



University of Computer Studies (Mandalay)

Department of Advanced Science and Technology

Ministry of Science and Technology

The Republic of the Union of Myanmar

JOURNAL OF INFORMATION TECHNOLOGY, RESEARCH AND INNOVATION

December, 2022

JiTRi 2022

JOURNAL OF INFORMATION TECHNOLOGY, RESEARCH AND INNOVATION

December, 2022

Organized by

University of Computer Studies

(Mandalay)

Volume-2, Issue-1

JOURNAL OF INFORMATION TECHNOLOGY, RESEARCH AND INNOVATION

JITRI, 2022

Vol-2, Issue-1

UNIVERSITY OF COMPUTER STUDIES (MANDALAY)

Journal of Information Technology, Research and Innovation (JITRI), 2022 Committee

University of Computer Studies (Mandalay)

Acknowledgements for Reviewers and Organizing members of JITRI

The Journal of Information Technology, Research and Innovation (JITRI) 2022 technical program committee would like to express the deepest appreciation to the external reviewers and organizing members for collaborating to publish this journal successfully. JITRI (2022) is published as the third time in order to undertake research in their respective fields of studies. Finally, We would like to thank all the authors who had submitted their research papers by making their great efforts in the Journal of Information Technology, Research and Innovation (JITRI), 2022, Vol-2, Issue-1.

Table of Contents

WEB SCRAPING FOR CAREER ANALYSIS BASED ON YOUTUBE DATA APIs USING WEB CONTENT MINING

Khin Than Nyunt [1] and Naw Thiri Wai Khin [2]

[1,2] Faculty of Information and Communication Technology,

University of Technology (Yatanarpon Cyber City)

[1]khinthannyunt.nyunt@gmail.com and [2]ntrwk87@gmail.com

ABSTRACT

Nowadays, APIs is frequently used to refer to "web application APIs", despite the fact that there are distinct APIs for numerous different software programs. Typically, a programmer will use HTTP to request some kind of data from an API, and the API will then provide the requested data in the form of XML or JSON. XML is still supported by the majority of APIs, but JSON is gradually overtaking it as the primary encoding standard. Notwithstanding the fact that millions of people do not consider using APIs to scrap webpages to be web scraping, both techniques provide similar outcomes. In this paper, in order to make the information more relevant, integrate data collected from a web scraper with data from publicly available APIs. This paper intends to scrap the videos content based on the data given YouTube APIs by using web content mining.

KEYWORDS

Web Application APIs, Web Mining, Web Content Mining, Web Data Scraping, YouTube Statistics

1. INTRODUCTION

Currently, everyone is discussing ways to make money with a YouTube Career [1]. Young people are focusing on YouTube in an effort to generate a respectable income from the creation of web videos. As a result, YouTube has given rise to a new generation of personalities known as YouTubers. It is reasonable to say that many celebrities have unavoidably joined Launchpad. It is also incredible how many individuals are drawn to the idea of creating a channel in order to become full-time YouTubers and earn a job from the platform [2].

By 2022, YouTube will have more than 51 million channels, a 36% increase from the current year. Millions of new subscribers are being added every day, and people from all over the world are starting YouTube channels and uploading videos of more than 100 million ethical action-takers each week. As a result, there are ever more video channels available online as YouTube, a platform for sharing videos, overtook Google to become the second-largest search engine in the world [4]. As YouTube channels expanded, so did careers on the platform. The channels for YouTube Career include a wide variety of content genres and is among the 10 most popular ones. In choosing such a business nowadays, in accordance with the changing times, success will only be realized it can accurately choose the most trending career by country. Therefore, this article proposes to use content mining as a dataset that will greatly assist in predicting which career is most suitable for which country by individuals who wish to make money from YouTube. The purpose of this study is to scrap YouTube video content by extracting pertinent information from the web resources.

2. WEB MINING

Web mining has grown in attention in recent years as a result of the Web's enormous growth in data sources and the rise of e-commerce among businesses [1]. Web mining is another essential component of the content pipeline for web portals. It is utilized in data validation, taxonomy construction, data integrity, opinion mining, correlation analysis, sentiment analysis, association analysis, regression analysis, classification, and clustering among other things. Web mining is a technique for extracting data from websites in order to automatically detect and extract relevant information from documents and services on the World Wide Web. Web mining is not the same as data mining. The unstructured nature of web data makes mining more difficult. A number of scientific communities, including database, information retrieval, artificial intelligence, psychology, and statistics have come together to study web mining. Find patterns in semi-structured data as well such as Internet (WWW) and can handle big data compare than traditional data mining [6]. Access rights are rarely required for web mining because the data is open and updated. Typically, web mining operation involves processing unstructured or semi-structured web-based data from websites. Based on the main data types used in the mining process, web mining operations can be divided into three groups: web content mining, web structural mining, and web usage mining, as illustrated in Figure 1.

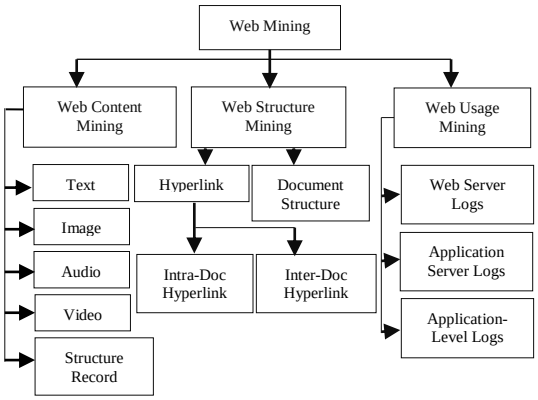


Figure 1. Types of Web Mining

3. WEB CONTENT MINING

Practically, the obtaining valuable information from the text of Web documents is known as Web content mining [2]. Content data refers to the assortment of data that a Web page was designed to display to viewers. It could include text, graphics, music, video, or structured records that could also include information and hyperlinks. The most thoroughly researched text mining application nowadays is for web content [1]. Topic discovery, association rule mining, web clustering, web classification documents and categories of web pages are problems that text mining solves. The technological advancement of other fields, such as information retrieval and natural language processing, is highly influenced by research activity on this area. Some of the data that makes up online content is hidden, some is dynamically generated in response to a query, and some is stored in a database. Typically, these data are not index. There are four techniques for web content mining. These are unstructured, structured, semi-structured, and multimedia [4]. The text portion of the web content data includes unstructured data, such as free text, semi-structured data such as HTML documents, and more structured data as a table or database / dataset. Undoubtedly, a lot of web content is data unstructured free text data [8]. The system flow diagram for my research is shown in Figure 2. I submitted this paper in an effort to learn and to develop dataset how to get data scrap and how to extract features for the dataset acquisition phase, which is the heart of the entire research. This paper is only the web data scraping portion of the research. The remaining research components will keep being written and submitted. The main contribution is based on TF-IDF method. In order to make the method better, we will enhance the lack of understanding of context because of sentence structure and the challenges memory and time found in the feature extraction section. Moreover, the dataset is self-constructed.

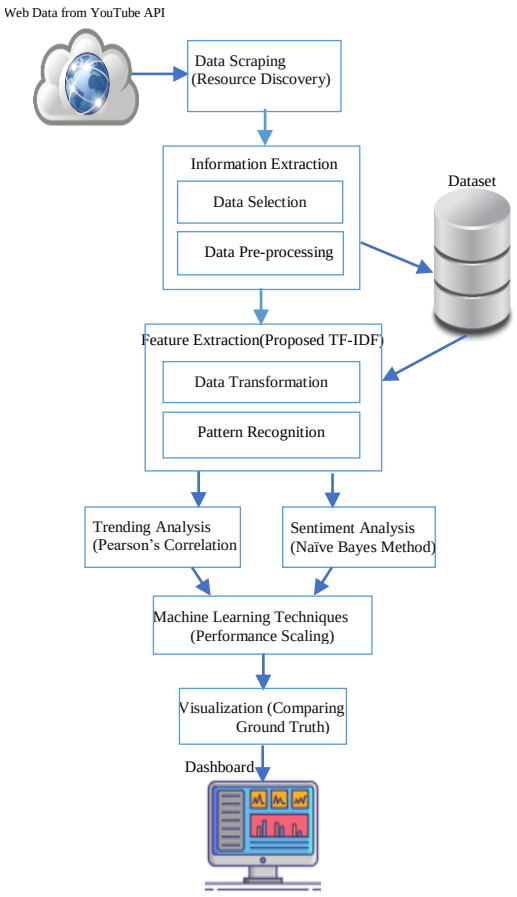


Figure 2. Proposed System Design

4. WEB SCRAPING TECHNIQUES

An effective method for obtaining vast amount of data from numerous websites is web scraping. This data can be kept in any kind of file or database. In web scraping, the HTML code of the web page is extracted [3]. The process of tracking and downloading information from websites and extracting semi-structured or unstructured data into a structured format is often referred to as web harvesting or web data extraction. To accomplish this, low-level implementations of the hypertext transfer protocol or the incorporation of certain web browsers resolve human exploration of the World Wide Web [6]. An orchestrator is a program that designs and carries out the requests to the browser during scraping. The elements to be searched must be clearly described, and the search's current status must be stated. Web scraping frameworks are typically a more comprehensive option. For instance, Python's quick and effective web scraping application framework, called "Scrapy" that offers all-around features for scraping and crawling by enabling the user to create a custom class called "Spider," conveniently parse and transform content, and get around website restrictions on scraping [7]. There are two stages to the web scraping process: In the first stage, known as extraction, data are retrieved from a site and preserved locally before being analysed to provide information in the second stage.

Data Scraping: Although social media data is accessible through APIs, major sources like Facebook and Google are making it more and more difficult for academics to obtain complete access to their "raw" data; very few social data sources offer affordable data offerings to academia and researchers [3]. This practice is known as scraping. Typically, to access to their data, news services like Thomson Reuters and Bloomberg often charge a fee.

Information Extraction: Internet web pages are systematically and automatically inspected by techniques like Spider, Crawler, Trackers. They help the network be tracked. The read the hypertext layout and click on all of the website's links. The most of the time, they are used to save copies of all viewed web pages so that search engines can index them and offer quick search functionality. It is a web platform that enables programmers and journalists to work together to create scrapers in order to extract and analyse publicly available data from the web.

Data Pre-processing: Due to its vastness and heterogeneous source origins, today's data has a higher possibility of being abnormal. The first and most important stage is data pre-processing, which is the act of preparing the raw data to transform into structured data and making it suitable. Additionally, it is necessary to clean the data and prepare it for a machine learning model, which improves accuracy and efficiency. Data preparation has become essential because high-quality data

produces stronger models and predictions. There are many steps involved in data pre-processing step using python as follow:

1. Getting the Dataset
2. Importing Libraries
3. Importing Datasets
4. Finding Missing Data
5. Encoding Categorical Data
6. Data Transformation
7. Splitting Dataset into train and test set
8. Feature Scaling
9. Data Visualization

Figure 3. Web Scraping

4.2. Scrapping YouTube Video Content using APIs Key

YouTube, an online video sharing platform, has become the world's second largest search engine, the number of video channels continues to grow. YouTube Careers started to rise as YouTube channels grew. YouTube Career's channel has many content categories and ranks in the top 10 trending of them [11,12]. Currently, as the number of YouTube users is also increasing, are the number of people who want to make a living from YouTube social media platform. However, if there is anything that makes it difficult for those people, YouTube's banning of the Dislike count from December, 14, 2021 has faced a lot of difficulty. Hence Career Analysis and in order to provide decision support for them, it becomes necessary to analyse YouTube data. To do that, it is very important to do data scrap. Only correct data can support the correct decision. If a wrong decision is made, a person's professional Career life will cost time, money, and effort and will not reach the goal of success. Therefore, data scraping is very important, so this is a paper that proposed how to do scraping. Data scraping design can be easily understood by looking at the Figure 4. The first step is to get the YouTube Website that we want to scrap. After that, the API key must be created on the Google Cloud Platform. If we get this key, need to enable YouTube Data API v3. After that, add the API key to the URL to be requested in the Python code and build. In order to be data enrichment, keywords must be generated. The next step is to add the file type to the server to output the obtained data in a format (CSV, JSON, etc.). In the last step, we get the streaming data / raw data we want. We need to convert raw data into useful features. Therefore, it converts raw data or dataset into vectors and each word has its own vector. It is necessary to develop the dataset based on what we want to analyse and implement.

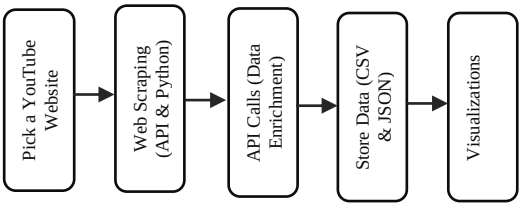


Figure 4. Web Scraping Process

4.2.1. Getting YouTube APIs Key

Firstly, we must have a YouTube API key in order to access YouTube video content. We really need a valid API key for the YouTube Data API. To obtain that key from the Google Cloud Platform, obtain the APIs key as shown in Figure 5 and set up credentials for the YouTube Data API v3 seen in Figure 6. The HTTP protocol offers four different methods for requesting data form a web server: GET, POST, PUT, and DELETE [1,4]. The following APIs key is used to access the web video content from YouTube social media.

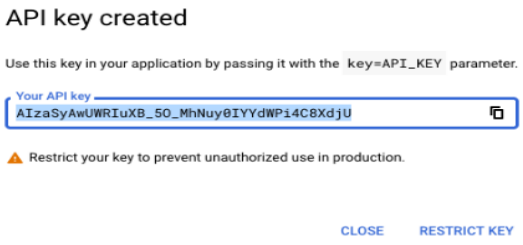


Figure 5. Creating YouTube APIs Key

Table 1. Result in CSV Format

video_id	title	publishedAt	channelId	channelTitle	categor	trending_date	tags	view_cour	likes	dislikes	comment_	thumbnail_lir	commen	ratings_c	description
0BYobDZut	I Proposed	2021-12-30T1	UCYfltqQ7l	Aaron Burriss	24	2022-01-01T0	aaron bur	1088048	100038	0	8345	https://i.ytim	FALSE	FALSE	Part 1: Asking her to be my Girl
I3zrQYycGf	Jackboy - L	2021-12-31T0	UC5utlnoY2	1804 Jackboy	10	2022-01-01T0	[None]	527256	32520	0	2329	https://i.ytim	FALSE	FALSE	Listen to the Single â€œLetter T
bvWRMAU	We Don't T	2021-12-29T0	UCgwv23FV	DisneyMusicVE	10	2022-01-01T0	Carolina|(	14686755	360904	0	23838	https://i.ytim	FALSE	FALSE	See Disneyâ€™s Encanto now s
2d-NnuM2	In-Depth c	2021-12-31T0	UC-2MJlKSq	Denver7 â€“ Th	25	2022-01-01T0	580 home	291799	2435	0	1207	https://i.ytim	FALSE	FALSE	The Denver7 team brings you i
JapWzIV1XI	X-MEN: All	2021-12-30T1	UCqTYHSnB	The Cosmonau	1	2022-01-01T0	[None]	560339	40822	0	3496	https://i.ytim	FALSE	FALSE	Please visit http://ayylienclothi
7VHfzUhRT	3 trials to l	2021-12-29T1	UCJswRv22	I am MoBo	23	2022-01-01T0	discord ba	6391327	783219	0	1890	https://i.ytim	FALSE	FALSE	Learning the secret technique v
xbV1csSM8	DUB RAN U	2021-12-30T1	UCDDgfKQE	PontiacMadeDI	26	2022-01-01T0	pontiac|p	633800	43996	0	3665	https://i.ytim	FALSE	FALSE	instagram: @PontiacMadeDDG
GmX7R9GX	All Animatr	2021-12-29T1	UChL4WU9'	SuperHorrorBr	20	2022-01-01T0	five nights	2303467	72319	0	3773	https://i.ytim	FALSE	FALSE	Five Nights at Freddy's Security
vGCCtFIzjw	i gave him	2021-12-29T1	UCgFvT6pU	Davie504	24	2022-01-01T0	davie504|	3712063	429715	0	3829	https://i.ytim	FALSE	FALSE	#shorts
XGuW5UKS	This VR gar	2021-12-28T17	UCQZi2YXSx	SportsNation	17	2022-01-01T0	espn|yout	6891055	472517	0	10902	https://i.ytim	FALSE	FALSE	This VR game went OFF THE RA
kV1jLbT27x	Eminem Be	2021-12-29T0	UCtab7TqLk	Frank Michael S	24	2022-01-01T0	shorts|yo	3364728	370626	0	3558	https://i.ytim	FALSE	FALSE	I respond on IG (say wassup):h
7yWcCCsY3	I Survived	2021-12-30T1	UCuSpeOwT	Painful	20	2022-01-01T0	Minecraft	655945	21084	0	1234	https://i.ytim	FALSE	FALSE	Celebrate Genshin Impact v2.3
GrhrPLUWJ	Building th	2021-12-29T2	UCOxoGn2	Braydon Price	24	2022-01-01T0	can am|d	664471	37837	0	2103	https://i.ytim	FALSE	FALSE	WE GOT THE OUTLANDER BACI
I8ato4vmb	My Final Vi	2021-12-30T1	UCzTKskwlc	NikkieTutorials	24	2022-01-01T0	my final vi	702208	33669	0	1207	https://i.ytim	FALSE	FALSE	This is my final videoâ€¦â€¦â€¦
X--HIW0jsV	The Prison	2021-12-30T17	UCWXCrltCI	ExplosmEnterta	23	2022-01-01T0	satire|sho	682319	52318	0	2700	https://i.ytim	FALSE	FALSE	Watch Part 2 of The Prison Bre
f6BKq432vE	I Spent $1,(	2021-12-29T1	UCWdBWdJ	KFC CLASH	20	2022-01-01T0	Clash|Roy	922696	48615	0	1927	https://i.ytim	FALSE	FALSE	SUBSCRIBE!FOLLOW MY STREA
nCqlR-HrnF	She Sells H	2021-12-29T1	UCYRg9w3A	Itsdanielmac	24	2022-01-01T0	[None]	2508344	77082	0	3868	https://i.ytim	FALSE	FALSE	I guess there is a market for an
EFZ3O4BwF	The SML ar	2021-12-30T0	UCqDZJlfBG	Chilly	22	2022-01-01T0	chilly|sml	307289	16857	0	5229	https://i.ytim	FALSE	FALSE	So happy to share probably the
pjunaHQ8h	Courtside v	2021-12-28T2	UCx18ywTc	Omar Raja - ESI	17	2022-01-01T0	Kevin Dur	2042510	135915	0	678	https://i.ytim	FALSE	FALSE	

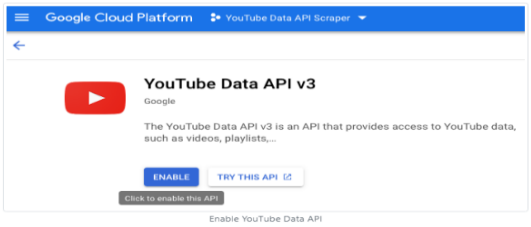


Figure 6. Creating YouTube Data APIs v3

4.2.2. Getting Bulk Data

After we have YouTube APIs key, the next step is to get bulk data from YouTube's search results – Data API Endpoint by the desired features such as category ID, YouTube API key, channel id, channel type (any, show), event type (completed, live, upcoming), language, coordinate's location radius, order (date, rating, relevance, title, videoCount, viewCount), published after, published before, query, region code, related to video ID, topic ID, video caption, and video category ID etc., as shown in Figure 7. We are unable to extract the desired data without the YouTube API Key in the URL. Because of this, we must first generate the API Key. Make a request by calling that key, adding it to the URL, and putting it in the Python code.

Figure 7. Scraping YouTube Video Content Using APIs Key

4.2.3. Extracted Data Format

The four most widely used format types for text markup are HTML, XML, JSON, and CSV [1,4]. Among them, The JSON and CSV format files are illustrated in Figure 8 is unstructured format and Table 1 transforms unstructured format into structured format respectively.


```
{
  "kind": "youtube#videoListResponse",
  "etag": "M6hyxJdrVuQgLdFGoPoLFql8mtM",
  "items": [
   {
     "kind": "youtube#video",
     "etag": "ZtHeEH2QDre4wGYV0ZxyO4nJeUw",
     "id": "gQRqe1nIZEM",
     "contentDetails": {
      "duration": "PT2M4S",
      "dimension": "2d",
      "definition": "hd",
      "caption": "false",
      "licensedContent": true,
      "contentRating": {},
      "projection": "rectangular"
     }
   },
```


Figure 8. Result in JSON File Format

1. **HTML:** Markup language used for web pages and other content that can be seen in web browser is called Hyper Text Markup Language.

2. **XML:** Extensible Markup Language or XML is used to categorize text data by applying tags to elements, such as and .

3. **JSON:** JavaScript Object Notation (JSON), a text-based open standard for human-readable data transport, is derived from JavaScript.

4. **CSV:** Each row starts on a new line, and each column's value is separated from the value of the adjacent column by a comma in a comma-separated values format.

5. SCRAPING YOUTUBE VIDEO CONTENT USING PYTHON

There are different ways to scrap websites such as online services, APIs or writing own code [9,10]. In this paper, demonstrate how to implement web scraping with Python language. A request is made to the URL we specified YouTube website when we run the web scraping code [5]. The server transmits the information in response to the request, enabling to see the HTML or XML or JSON page. After parsing the HTML or XML or JSON page, the code extracts the data [5,11]. To extract data using web scraping with Python, we need to the following basic steps:

1. **Find the URL that we want to scrap**: scrap YouTube website to extract the features: such as video_id, title, publishedAt, channelID, and channelTitle etc. The URL included the APIs key to extract the features is https://www.googleapis.com/youtube/v3/videos?part=contentDetails&chart=mostPopular&regionCode=IN&maxResults=50&**key=AIzaSyA5vrCzoRa5gcZjEkLdvGet1lfEZ3govKs** based on the country code we want to scrap. Notify, needs country codes for the countries we want to collect trending video contents from. These are two letter country abbreviations according to the ISO 3166-1.

2. **Inspecting the Page:** The data we want to scrap is nested, so define the inspected page to get, under which tag or page the data we want or which content we get.

3. **Find the data we want to extract:** define the features we extract such as: video_id, title, publishedAt, channelId, channelTitle, category_id, trending_date, tags, view_count, likes, dislikes, comment_count, thumbnail_link, comment_disable, rating_disable, and description for YouTube Career Analysis and Decision Support system.

4. **Write the code:** It must be possible to use Panda's libraries' functions of Python on information outlines for YouTube statistics data that include the frequency at which the channel was in the trending area. Using Python's matplotlib, the quantity of count for each channel is computed.

5. **Execute the code, then get the data:** After creating the python file to extract the YouTube video contents, by running it, the features we defined are columns, and we will get the YouTube video contents of the country we want to scrap.

6. **Store the data in the required format:** After extracting YouTube data, there are four ways to save it in systematically. Especially the most used is CSV and JSON file format.

The necessary libraries such as requests, sys, time, os, argparse for data scraping must be installed. We must write the code below if we want to scrap the video content as header feature.

```
# List of simple to collect features
snippet_features = ["title",
                    "publishedAt",
                    "channelId",
                    "channelTitle",
                    "categoryId"]

# Any characters to exclude, generally these are things that become problematic in CSV files
unsafe_characters = ['\n', '"']

# Used to identify columns, currently hardcoded order
header = ["video_id"] + snippet_features + ["trending_date", "tags", "view_count", "likes", "dislikes",
                                            "comment_count", "thumbnail_link", "comments_disabled",
                                            "ratings_disabled", "description"]
```

We might want to store the data in a format after executing the above steps. This format changes based on what we need. The extracted data will then be saved in CSV format by country code. The following lines will be added to my code to do this:

```
#write the extracted data in the csv file
for country_code in country_codes:
    country_data = [",".join(header)] + get_pages(country_code)
    with open(f"{output_dir}/{country_code}videos.csv", "w+", encoding='utf-8') as file:
        for row in country_data:
            file.write(f"{row}\n")
    file.close()
```

The workflow of the overall process of scraping using Python libraries is simply described in the steps that follow. These steps are thoroughly explained in the sections.

Step1. Import Requests and Pandas, Seaborn, Numpy, Matplotlib, Wordcloud Libraries.
Step2. Use page.content to get the raw HTML document of the webpage accessed and assign it to a variable 'content'.
Step3. Define prepare_feature (feature) method to get the string of original features scrapped.
Step4. Use api_request (page_token, country_code) method to request page tokenization and 2 letter country code to return JSON format of data and to build the URL.
Step5. Use get_videos (items) to get JSON format of data on movies items and return list of data on items.
Step6. Collect all individual list of data fields extracted into a Dictionary.
Step7. Use Python library Pandas to convert the dictionary into a data frame and export it into a.csv file.

6. DEVELOPMENT DATASET

In order to provide a lot of support for YouTube Trending Analysis and Sentiment Analysis, the web data is acquired from YouTube APIs to create a private dataset. The top 10 trending careers will be chosen first, followed by the top 10 YouTubers in each career, and finally, 100 YouTubers will be chosen. We will receive 1,000 channels when one of the 100 YouTubers selects the top 10 trending channels. This dataset contains 102192 rows and 16 columns from August, 01, 2020 to November, 18, 2022 of trending dates. Web Data is included for the India, US, Great Britain, Germany, Canada, France, Russia, Brazil, Mexico, South Korea, and Japan with up to 200 listed YouTube trending videos per day. There is a lot of video content data on each channel such as video category, title, channel title, channel ID, category ID, tags, links, view, like, dislike, comment, comment disabled, video error or removed, publishing date, trending date, rating, and description the amount of gigabytes GB. Among them, 9 features from the many features in the video content data were extracted, including view, like, dislike, comment, comment disabled, rating, and subscription. Web scraping, feature selection, and pre-processing steps were carried out. This dataset will be very helpful and usable for people who want to conduct research using the YouTube social media platform by utilizing the feature set as see fit.

This dataset is an excellent one that can provide evidence-based instruction that it can enable trending analysis, opinion analysis, and sentiment analysis by looking at the comments of the extracted video content data to build a YouTube career analysis that can help young people who are choosing a living from YouTube today make the proper decision. In Figure 9, for instance, we described how each feature is correlated with each other by using Pearson's correlation method for trending analysis with a heatmap using the video content data that was extracted from the dataset we created [12]. In Figure 10, using the dataset we created, we have exhibited the results of picking the 17 most trending careers right now in order to better decision support which careers are now trending and how those who are making a living using YouTube are developing over time in today's world.

In summary, according to the trending statistics displayed in Figure.11, a trending video has an average of 2,216,553.88 views. The median number of views for popular videos is 847,692.50, which indicates that half of them have fewer than that many views and the other half have more. We receive 277,791,741 views the most. The average number of views and likes for a trending video is 2216553.88 and 109467.09, respectively. The median comment count is 2299.00, while the average is 8624.58.

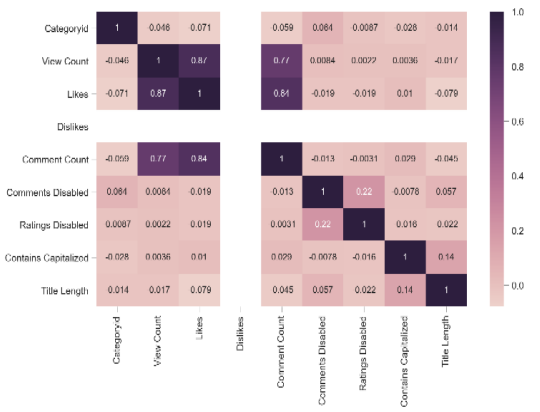


Figure 9. Correlation Analysis of Scraped Video Content Data with Heatmap

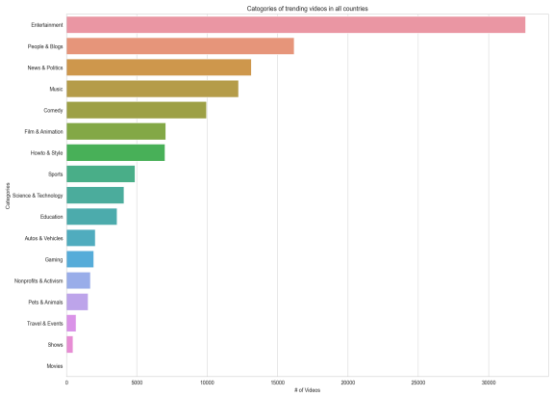


Figure 10. The Most Trending YouTube Categories in All Countries

	categoryId	view_count	likes	dislikes	comment_count
count	64398.00	64398.00	64398.00	64398.00	64398.00
mean	18.90	2216553.88	109467.09	0.00	8624.58
std	6.63	6996541.95	394047.12	0.00	75608.85
min	1.00	0.00	0.00	0.00	0.00
25%	17.00	431773.50	17249.25	0.00	1108.00
50%	20.00	847692.50	36951.50	0.00	2299.00
75%	24.00	1895633.00	88078.00	0.00	5137.75
max	29.00	277791741.00	12993894.00	0.00	3534337.00

Figure 11. The Statistical Analysis About the Numerical Column of Dataset

7. CONCLUSIONS

As already mentioned, a "explosion" of data services for scraping and sentiment analysis, career analysis, and social media analytics platform has resulted from the simple accessibility of APIs offered by YouTube social network sites. In order to commercialize their content, businesses are progressively restricting access to their data is arguably the largest cause for concern. In order to conduct experiments, researchers must have access to computer environments and, in particular, "huge" social media data. In any other case, computational social science might end up being the exclusive purview of large corporations, governmental organizations, and a select group of university researchers who have access to confidential information and who use it to write papers that cannot be questioned or replicated. For quantitative social science, it may be argued that cloud servers and computing environments in the public domain are required. Researchers get access to these resources through a cloud-based platform.

The dataset must have the correct data in order to get an accurate and worthwhile result. For ongoing research, this data is also a key component that must be continually developed. I have submitted a research article for people who are interested in learning how to do data scraping on YouTube social media platform, and this dataset is designed for trending analysis and sentiment analysis. When I finish my research, I will make it available to the public on my Github.

ACKNOWLEDGEMENTS

The Rector of the University of Technology (Yatanarpon Cyber City) who helped me accomplish this paper. Deputy Rector special thanks to the Dean of the Department of Information Science and everyone. Moreover, Special thanks to my parents and husband for giving me the necessary support.

REFERENCES

[1] Shyam Nandan Kumar, "World towards Advance Web Mining: A Review", American Journal of Systems and Software, 2015, Vol. 3, No. 2, 44-61.

[2] MUSKAN VERMA, RIYA GARG, RITIKA JAIN*, ANURADHA TALUJA, "Classification of YouTube data based on Opinion Mining", Journal of Remote Sensing & GIS, September 13, 2021.

[3] Ryan Mitchell, "Web Scraping with Python Collecting Data from the Modern Web" Printed in the United States of America. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

[4] Bogdan Batrinca • Philip C. Treleaven, "Social media analytics: a survey of techniques, tools and platforms", Springer Journal Publication, AI & Soc (2015) 30:89–116.

[5] Deepak Mokashi1, Darshan Rahude2, Diksha Fegade3, Vaishnavi Dhembre4, Mahesh Shinde5, Balaji Bodkhe6, "Career Prediction System using Big Data and Data Mining Techniques, A Review", International Research Journal of Engineering and Technology (IRJET), Mar, 2021.

[6] Mine Dogucu and Mine Çetinkaya-Rundel[b ,c ,d] , "Web Scraping in the Statistics and Data Science Curriculum: Challenges and Opportunities" Journal Of Statistics And Data Science Education 2021, Vol. 29, No. Sup1, S112-S122.

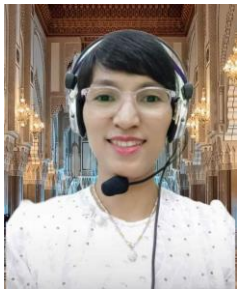
[7] Sowmiya Ka,1, Supriya Sb and R.Subhashinic, "Scraping and Analysing YouTube Trending Videos for BI", Smart Intelligent Computing and Communication Technology, 2021.

[8] https://medium.com/@wilirahmatm/extract-data-from-web-scraping-to-csv-and-json-files-using-python-705700cac050

[9] How to scrape data from web using python (crayondata.com)
[10] A Practical Introduction to Web Scraping in Python – Real Python
[11] Web Scraping With Python - Full Guide to Python Web Scraping (edureka.co)
[12] PEARSON'S CORRELATION COEFFICIENT - A BEGINNERS GUIDE (ANALYTICSVIDHYA.COM)

Authors

Biography



First A. Author I obtained Bachelor of Computer Sciences B.C.Sc in 2006, Bachelor of Computer Sciences (Hons:) B.C.Sc(Hons:) in 2007 and Master of Computer Sciences M.C.Sc in 2010. This paper is also a submission for the doctoral degree that I am currently attending. Received local and international journal papers obtained are 1. "Computer Assisted Learning System for Fundamental Concepts of UML", 2. "Automated Web Application Loaded Testing System with AppPerfect Load Test", 3. "Creating JavaScript Virtual Keyboard and Comparing to Physical Keyboard", 4. "Learning Assisted System Using Computer Assisted Instruction (CAI)", 5. "Scalable Mobile Continuous Integration Testing System", as the first author and 6. "Implementation of Site-to-Site IPsec VPN Tunnel between Routers", as the second author.

DETERMINING ROBBERY ACTIVITIES IN BANKS BASED ON GUN DETECTION USING DEEP LEARNING

Naing Zaw Aung[1], Hlaing Moe Than[2] and Kyaw Kyaw Lin[3]

[1,2,3] Department of Computer Technology, Higher Education Centre (Pyin Oo Lwin)

[1]naingzaw53@gmail.com, [2]hlaingmoe2008@gmail.com, [3]kklin1500@gmail.com

ABSTRACT

Robbery with a weapon in hand is increasing day by day, and it has grown to be one of the greatest issues facing the world nowadays. There are a variety of detecting devices on the current market, but they are not designed to automatically detect robbery activities based on gun detection and alarm the owner. To address this problem, deep learning-based gun detection for determining the robbery activities in banks' camera footage is used. The RetinaNet network will be used in our proposed system to detect guns, and an alert will be sent to either the police station or the bank owner. Using our own dataset, the method of using the NVIDIA Jetson Nano Developer Kit loaded with the RetinaNet network to realize the real-time detection of robbery activities in banks based on gun detection is proposed in this paper.

KEYWORDS

Anti-Robbery Device, Deep Learning, Jetson Nano Developer Kit, RetinaNet, TensorFlow

1. INTRODUCTION

Robberies are among the oldest and most common illegal activities, and they are becoming more common day after day. Robbery activity of valuable things such as phone, camera, money, and so on is some of the never-ending problem in the world [1]. The public has experienced loss and anxiety as a result of the increase in robberies. A robbery prevention system that is simple to use and mostly free of false alerts is required to prevent the rising rate of robberies throughout the world. So, in order to address all of these requirements, we are developing a system that can detect robbery in any store or surveillance area and alert the police station or the owner, giving them the opportunity to take the action against the robbery.

The development of an anti-robbery device using deep learning can detect robbery activities based on gun detection and alert the police station or any owner with an alert message. The system, which will have a simple user interface and be real-time, is useful to ensure the protection of both individuals and their valuable things.

In the modern, changing, and evolving environment, security against robbery is a major concern. More security is required for people's precious possessions. All banks, commercial shops, and surveillance areas have security cameras but need an anti-robbery system that can look after the safety of their property even when they are not there [2].

Our proposed system focuses specifically on the security of banks. The security of banks is important in Myanmar nowadays. In order to have a good security system, additional money must be spent on its installation. There are many different security devices available for detecting, identifying, and reducing crime or robbery. However, the banks can be supported by only a small number of intelligent devices. So, the main goal of our proposed system is to detect any robbery activity in banks using deep learning techniques based on gun detection and to alert the police station or bank owner with an alert message.

2. RELATED WORKS

Noor et al. [3] suggested mask detection using an infrared temperature sensor on the Jetson Nano Kit based on the occurrence of the Coronavirus disease-19 (COVID-19). Using a

webcam and the NVIDIA Jetson Development Kit series, this system determines if a mask may be worn. By attaching an IR array sensor device called the AMG8833, it is also possible to concurrently measure the surface temperature of a human body. A buzzer alarm will ring to alert you if the human body temperature exceeds the threshold value, which has been set at 37°C. Dixit et al. [4] have focused on a theft detection system using the Raspberry Pi 3B. They have proposed that this system would eliminate the need for continuous surveillance by human resources. This system continuously checks if someone is entering the store or not by sensor. The alert message is then sent to the owner with live images from various viewpoints captured by a rotating camera. Jaime et al. [5] proposed a technique for automatically detecting robberies based on actors' predicted poses. They have proposed that their algorithm correctly identified the various body segments in around 80% of the overall frames.

3. METHODOLOGY

The proposed system aims to improve the democratization of artificial intelligence using the NVIDIA Jetson Nano's high-performance AI edge device. The phases in this proposed method are as follows.

3.1. RetinaNet

Class imbalance that reduces the accuracy of the model is a common problem in deep learning, especially in classification problems. The RetinaNet solves this problem during training by using a focal loss function.

RetinaNet is one of the best one-stage object detection models that has proven to work well with dense and small-scale objects. It mainly comprises of three subnetworks: a residual network (ResNet), a feature pyramid network (FPN), and two fully convolutional networks (FCNs) [6]. The RetinaNet network architecture is shown in Figure 1.

3.1.1. ResNet

ResNet's main contribution is the concept of residual learning, which enables the original input data to be transmitted directly to the subsequent layer. A variety of network layers are compatible with ResNet. The most commonly used types of network layers are the 50-layer, 101-layer [6], and 152-layer varieties. The best training results are achieved with the 50-layer architecture. The features of the gun using ResNet are computed and then sent them to the next subnetwork.

3.1.2. FPN

FPN is a method for efficiently extracting the features of each dimension in an image using a standard CNN model. A one-dimensional picture is initially fed into ResNet. The FPN then takes the features from each layer, starting with the second layer of the convolutional network, and combines them to create the final feature output combination.

3.1.3. FCN

The class subnet in the FCN performs all the classification tasks. Whether a gun is included in the image or not might be determined by this subnet. The border regression task is performed by the box subnet in the FCN. Its task is to determine where the gun is located in the images and record the coordinates.

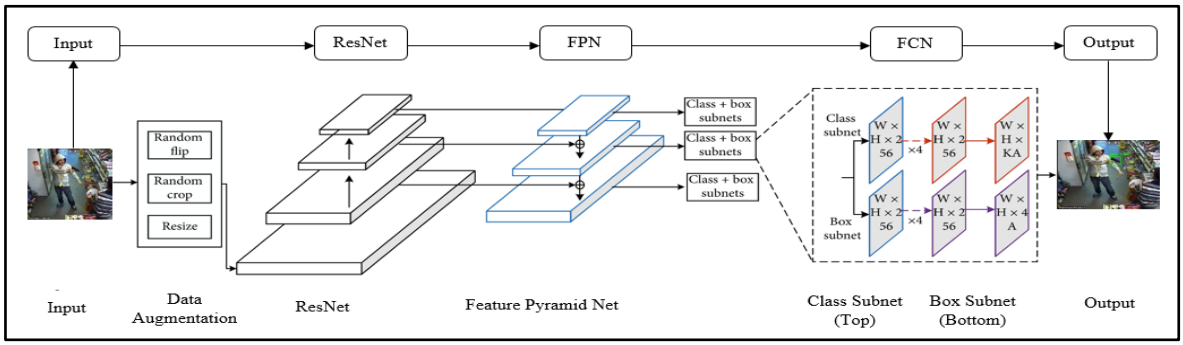


Figure 1. RetinaNet Network Architecture

3.2. Dataset

Data collection is the first step in constructing a gun detection model. While using object detection techniques, some of the major challenges such as viewpoint variation, deformation, lighting, a cluttered or textured background, and so on, are being faced. Therefore, appropriate data concerned with these challenges is collected particularly from the internet and CCTV footage of armed robberies. 3,500 pistol images and 3,500 rifle images are used to create our dataset. Using our own dataset, the LabelMe annotation tool in Anaconda3 2021 is used to make a ground truth label with bounding boxes and labels on the images. After that, data augmentation is used to boost the performance of deep networks. Preprocessing tasks like scaling are required. In many cases, models perform better if the size of images is reduced. The images used in this research were 416x416 pixels in size. Data must first be converted into an array in order to use the loop function. The ResNet-50 model will be used for preprocessing. During the evaluation stage, 20% of the images are used for testing, and 80% are fed into the model for training. All of the pistols and rifles are included in this data collection.

3.3. Building and testing the model

The default batch size is 3, which is used for the RetinaNet model. With a RestNet50 backbone, RetinaNet requires 139 MB of RAM for 3 batch-sizes. This model was trained on an x86 64-bit desktop computer with the NVIDIA GeForce RTX 2070 graphics card (Compute Capability = 7.5). when the dataset including 7000 images is trained, training time is about 120 minutes. The model is then uploaded on an NVIDIA Jetson Nano Developer Kit. Python version 3.6.2 is used to conduct several experiments. The mathematical formulas for evaluating the RetinaNet model are presented in the following equations (1)-(4) based on [7].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1\_Score = \frac{[Precision * Recall]}{[Precision + Recall]} \quad (4)$$

Symbols in equations (1)-(4) represent, respectively, false negative (FN), true positive (TP), false positive (FP), and true negative (TN) [7]. True positive values refer to images that have been labeled as true and have generated a true result as predicted by the model. In a similar manner, true negative images are ones that have been classified as true but produced an incorrect result because of prediction. False-positive images have been categorized as false yet produced false positives because of prediction. False-negative images are ones that are classified as false but appear to be true, producing false negatives. A measure called precision indicates how many predicted positive results there are. The F1-score evaluated test accuracy, whereas the recall score measured a classifier's ability to detect all positive values [7]. To achieve the precision and recall scores, equations (2) and (3) from the aforementioned equations have been used in this research.

3.4. Jetson Nano Developer Kit

Because of the new NVIDIA Jetson Nano Developer Kit, modern AI workloads can now be run at previously unheard-of scalability, power, and cost. Figure 2 shows how developers, educators, and makers may now create applications for object detection, classification tasks, speech recognition, and other activities using AI frameworks and models. This device includes everything from general-purpose input/output (GPIO) to a camera serial interface (CSI), and it is powered by a micro-universal serial bus (micro USB). This makes it easier to connect a variety of new detectors. It is very energy-efficient because it only consumes 5 watts of power [3]. The Jetson Nano Developer Kit has a Secure Digital (SD) card where Ubuntu 20.04 can be installed.

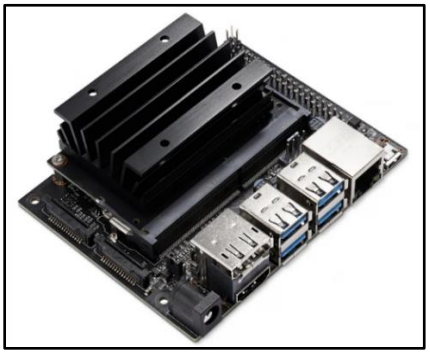


Figure 2. Jetson Nano Developer Kit

4. SYSTEM DESIGN

We propose a new method for addressing the problems with the current research because it will help to reduce the unnecessary utilization of hardware and reduce the cost of the project. Our proposed system would address classification error problems by reducing errors utilizing a variety of methods, including feature extraction, classification, and so on. The process flow for this system is as follows.

4.1. System Block Diagram

The system, which was structured by the system block diagram, is shown in Figure 3. The fundamental components of the system are a Jetson Nano Developer Kit and a CCTV camera. As soon as the system has been activated, the CCTV camera will start to capture video frames and send them to the Jetson Nano Developer Kit for processing using the TensorFlow library. The proposed system loaded on the NVIDIA Jetson Nano Developer Kit will then detect robbery activities based on gun detection and send an alert message to the police station or bank owner. An external power source is required by the Jetson Nano Developer Kit in this situation.

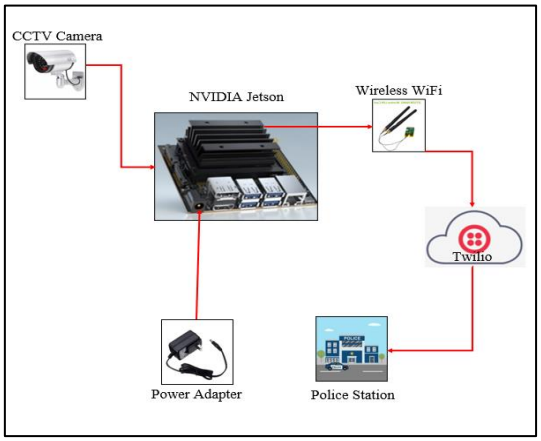


Figure 3. System Block Diagram

4.2. Flowchart

The complete process for the system is shown in the following flowchart, Figure 4. As an evaluation standard, a confidence score is computed. This confidence score shows the probability that the algorithm correctly detected the image. The confidence value is the image classification score, and the threshold value is defined at 0.5 by default in CNNs. This flowchart demonstrates that it captures a picture and sends it along with an alert message to the police station or bank owner.

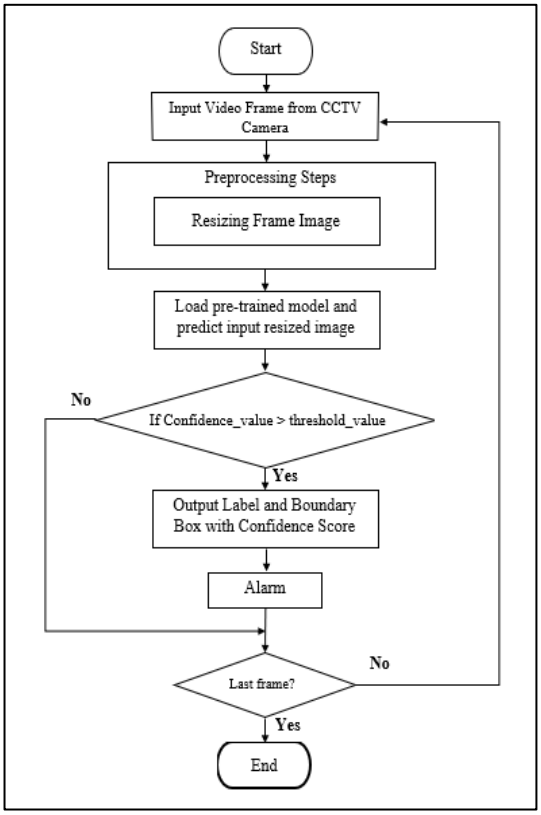


Figure 4. Flowchart of this Proposed System

4.3. Device

The Jetson Nano Developer Kit should be supplied with a 5 V to 12 V continuous power supply by the user before using it. The anti-robbery device will be switched on by the owner of the bank. The gun detection function will then be started using CCTV cameras, and video frames will be sent to the Jetson Nano Developer Kit.

4.4. Owner/ User

An alarm message is received by the police station or the owner of the bank if the system detects a gun. The alarm message will be checked by the police station or the owner of the bank, and they will take action against it.

5. EXPERIMENTAL RESULT

TensorFlow, Keras, and OpenCV are used to train a gun detection system. Using the gun

detection method, a model is trained and tested on a desktop computer whose hardware specifications are described in section 3.3, achieving 98% accuracy during training and testing. Experimentation on our own dataset for real-time detection, which is composed of 7000 images of rifles and pistols, has been trained using the RetinaNet model, and the best results were obtained through the model in terms of precision and recall. Table 1 shows the results for the precision and recall achieved by the RetinaNet model on our own dataset at a standard IoU threshold score of 0.5. The Jetson Nano Developer Kit is used to implement the model. A Jetson Nano Developer Kit and a CCTV camera are used to capture real-time video. Guns have been detected accurately even in low resolution and low light. The result obtained after training and testing models on our datasets is shown in Figure 5.

Table 1. The Results for the Precision and Recall achieved by the RetinaNet model on our own dataset.

Model	**IoU Threshold = 0.5**	
	Precision	Recall
RetinaNet	0.85	0.82

To evaluate an object detection model, the average precision (AP) from prediction scores is computed. AP summarizes a precision-recall curve as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight. The mean Average Precision, or mAP score, is computed by taking the mean AP over all classes or overall IoU thresholds.

$$\text{mAP} = \frac{1}{N}\sum_{k=1}^{N} AP_k \quad (5)$$

where,

AP is the average precision of class k

N is the number of classes

mAP is the mean average precision

The mAP considers both false positives (FP) and false negatives (FN), as well as the trade-off between precision and recall. Because of this property, mAP is a suitable measure for most detection applications. The mean Average Precision or mAP score of our proposed system is shown in Figure 6. A training dataset with 3500 images of pistols and 3500 images of rifles is used to train the model. To demonstrate the test precision for each of them, 70 epochs are used. We can notice that, after the 70th epoch, the value of mAP started to stabilize at approximately 0.85. When the proposed system, which achieves a mAP score of 0.85, is tested, the detection time is 22 ms for a frame (45 FPS).

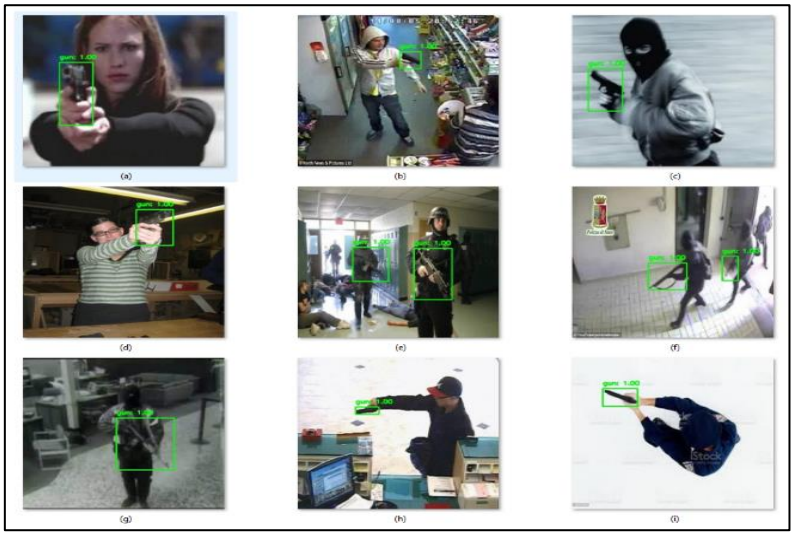


Figure 5. Detection Results of this Proposed System: (a) Front View Video Frame, (b) CCTV Video Frame, (c) Black and White Video Frame, (d) Front View Video Frame, (e) Vertical View Video Frame, (f) CCTV Video Frame, (g) Low Resolution Video Frame, (h) Video Frame from CCTV in Bank, (i) Image from Top View

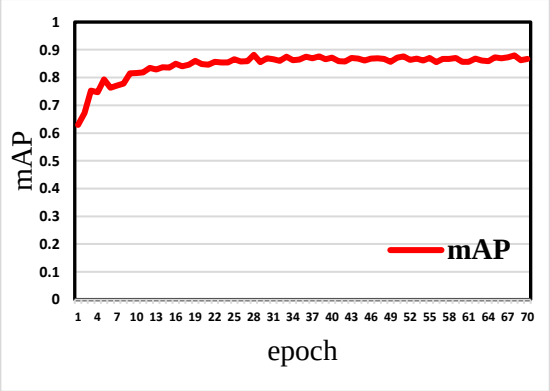


Figure 6. Mean Average Precision of Our Proposed System

6. CONCLUSIONS

The main purpose of the research is to create and develop an accurate gun detecting system. This is an anti-robbery device designed to address security problems and reduce or prevent robbery activities based on gun detection. Although many research have been done in the past to address such security problems, it remains challenging because of the numerous robbery activities that occur frequently. A transfer learning model called RetinaNet and the NVIDIA Jetson Nano Developer Kit are used to create the proposed gun detection system. As a result, the proposed system can be used in surveillance areas such as banks, supermarkets, office buildings and so on. The mean average precision of 0.85 is achieved by the proposed model when testing the gun detection and classification based on both the training data and the unknown data.

ACKNOWLEDGEMENTS

The author would like to express his gratitude to my dearest supervisor and co-supervisor for their guidance, motivation, and on-going support throughout this study. Throughout my work, their extensive knowledge and innovative thinking have been an invaluable support.

REFERENCES

[1] Ajay Kushwaha, Ankita Mishra, Komal Kamble, Rutuja Janbhare, and Prof. Amruta Pokhare, "Theft Detection using Machine Learning," *International Conference on Innovative and Advanced Technologies in Engineering*, 2018.

[2] Muhammad Tahir Bhatti, Muhammad Gufran Khan (Senior Member, IEEE), Massod Aslam and Muhammad Junaid Fiaz, "Weapon Detection in Real-Time CCTV Videos Using Deep Learning," *Digital Object Identifier* 10.1109/ACCESS.2021.3059170.

[3] Noor F. Abdul Hassan, Ali Abed, and Turki Younis Abdalla, "Surveillance system of mask detection with infrared temperature sensor on Jetson Nano Kit," *Bulletin of Electrical Engineering and Informatics*, Vol. 11, No. 2, April 2022, pp. 1047~1055.

[4] Dixit Suraj Vasant, Babar Apeksha Arun, and Meher Priya Shiivaji, "Raspberry-Pi Based Anti-Theft Security System with Image Feedback," *Journal of Information, Knowledge and Research in Electronics and Communication Engineering*, Volume – 04, Issue – 02.

[5] Jaime Dever, Niels da Vitoria Lobo, and Mubarak Shah, "Automatic Visual Recognition of Armed Robbery," *In Proceedings of the 16th International Conference on Pattern Recognition*, Montreal, QC, Canada, 11–15 August 2002; pp. 451–455.

[6] Meijun Yang, Xiaoyan Xiao, Zhi Liu, Longkun Sun, Wei Guo, Lizhen Cui, Dianmin Sun, Pengfei Zhang, and Guang Yang, "Deep RetinaNet for Dynamic Left Ventricle Detection in Multiview Echocardiography Classification," *Scientific Programming*, vol. 2020.

[7] Rutvik Kakadiya; Reuel Lemos; Sebin Mangalan, and Meghna Pillai, "AI Based Automatic Robbery/Theft Detection using Smart Surveillance in Banks," *In IEEE, Proceedings of the Third International Conference on Electronics Communication and Aerospace Technology* [ICECA 2019].

Authors

Biography

Naing Zaw Aung received the M.A.C.S degree in automatic control systems from the Moscow Institute of Electronic Technology (MIET), Russian Federation in 2015. He is now

attending the Ph.D. 18th Batch at the Higher Education Center (HEC), Pyin Oo Lwin.

Hlaing Moe Than received the M.I.C.Sc. degree from the Moscow Institute of Electronic Technology (MIET), Russian Federation in 2008. He continued his studies and got his Ph.D. (Computer Science) in 2014. He is now serving as an assistant lecturer at the Department of Computer Technology at the Higher Education Center (HEC), Pyin Oo Lwin.

Kyaw Kyaw Lin received the M.I.C.Sc. and Ph.D. (IT) from the Moscow Institute of Electronic Technology (MIET), Russian Federation in 2008 and 2017. He has served as an assistant lecturer at the Department of Computer Technology at the Higher Education Center (HEC), Pyin Oo Lwin.

IMPLEMENTATION OF SCIENTIFIC PAPER CLASSIFICATION AND RELEVANCY SCORE ANALYSIS BASED ON METADATA FEATURES

Aung Myint Myat[1] and Kyaw Min Thein[2] and Htin Kyaw[3]

[1,2] Department of Computer Science, Higher Education Center
[1]aungmyintmyat.ayeyarwaddy@gmail.com, [2]kyawlay82@gmail.com,
[3]mgyeinaye@gmail.com

ABSTRACT

The wide range of document classification methods currently exploits several data sources, including research paper text and metadata. In a variety of situations, metadata like the title, keyword, and abstract can be a superior option to the content. Researchers have evaluated the function of some of the metadata individually in the existing literature. Therefore, the metadata parameters can have a great impact on how scientific scope is categorized. This paper presents an evaluation of the role of metadata in achieving the goal of classifying research paper and closely related papers. In this evaluation, the metadata-based similarities are calculated by using the Bag-of-Words model and compared the cosine similarity and correlation coefficient. The results of this research presented that the combination of titles and keywords is superior in classifying research papers and closely related papers based on relevancy score. In this research, we implemented with the use of Excel VBA and CERMINE.

KEYWORDS

Bag-of-Words, Correlations Coefficient, Cosine Similarity, Document classification, Metadata, Research paper

1. INTRODUCTION

As the number of scientific papers increases, the need for academic classification system becomes more important. Classification scientific papers help researchers and find relevant papers. Scientific research paper is a very important communication channel in the scientific world. Document classification is becoming an increasingly vital role to manage and process a huge number of digital forms that are widespread exponentially. Research papers have expanded rapidly over the past several years. This vast amount delays the extraction of relevant papers. Researchers have been interested in this issue and are searching for relevant research paper in scientific area. To support the scientific community, collaboration among researchers within a research field is difficult. Todays, different approaches have been proposed to find relevant research papers. Therefore, we think to classify the area of research paper in the scientific field based on relevancy score by using correlation coefficient and cosine similarity. Categorization scientific literature can help in finding relevant research papers in several ways.

We collected 2000 research papers in the scientific fields that are currently performed most researchers from the domain of Computer Science. We extracted title and keyword from this research article by using CERMINE tool (a tool for automatic metadata extraction from the scientific literature). Metadata information is essential for finding related scientific literatures, providing search tools and creating citations networks [3]. The aim of this paper is to describe relevancy score evaluation on metadata features and to classify scientific research papers based on real academic datasets. The remainder of this paper is organized as follows. In section 2, the

related works are described. The proposed system in section 3. In section 4 experimental results and Conclusion is given in Section 5.

2. Related Work

Nowadays, the vast amount of research paper following over the Internet, huge collections of documents in digital library. Numerous similarity metrics, including squared Euclidean distance, cosine similarity, and relative entropy, have been applied to clustering [1]. In this paper, these measures are compared and analyzed in portion clustering for text document datasets. The research article measuring similarity methods are used with content-based approaches [6]. The majority of these methods rely on similarity techniques based in vector space, which make use of the vector space model of research articles. This similarity method is Cosine similarity [1]. It is one of the most commonly used content-based similarity method to find the relevant research papers.

Bayesian-based approach has been described to classify conference paper [7]. A feature selection algorithm was implemented to automatically extract keywords for each paper. The approach is based on keywords-based feature. Some measuring similarities are used relatedness scores between two research papers by using synonym sets of keywords that appear in the abstracts and titles of two papers [9].

Their analysis performed to determine the current scientific research area in a specific field of study [2]. They use traditional term-based approaches to identify the current relationships between the scientific domain. The proposed methodology uses the traditional topic and term frequency model to identify related articles [8]. This paper does not consider other relations.

Natural language processing (NLP) approaches can be employed in a variety of applications, including machine translations, natural language processing, multilingual and cross-language information retrieval (CLIR), speech recognition and Artificial Intelligence systems [5]. Text classification is one of the most interesting tasks in text mining, its purpose is to categorize text into a set of different classes and it has wide applications such as email filtering, sentiment analysis, search engines, and digital resource classification such as libraries, news, web pages. Moreover, text classification is a large and complex project.

3. The Proposed System

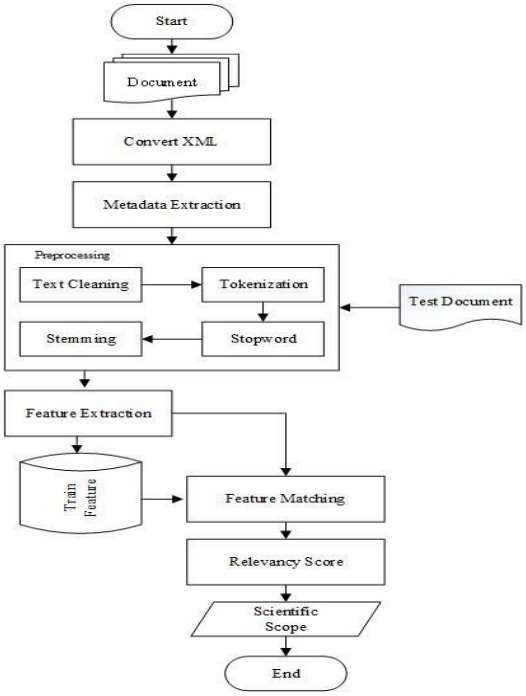


Figure 1. Proposed System Design of Scientific Paper Classification

The proposed system consists of two phases: training and testing. The data are collected from scientific literature published in the domain of Computer Science. This dataset contains metadata features including title, abstract, keywords, etc. Of all these metadata features, we extracted title and keywords from scientific paper and converted text by using CERMINE tool. We performed feature extraction and feature matching steps on the datasets. In first step, we performed preprocessing steps on the dataset with the use of Bag-of-Word. And then, we transformed data that converts words into numbers using the method of generating feature extraction. In second steps, we compared the measuring similarity such as cosine and correlation coefficient in feature matching. Next, we tested with new

document to classify research paper in scientific field and closely related papers based on relevancy score. In this evaluation, we presented that cosine similarity with the accuracy of 0.96 and correlation coefficient with the accuracy of 0.94. The workflow does not include any OCR phases, it analyzes only the PDF text stream. Figure 1 describes the proposed system design's flowchart.

3.1. Data Pre-processing

Data preprocessing step is the basic important step for Natural Language Processing before starting experiments. Text preprocessing consists of text cleaning, tokenization, stop word and stemming. We collected the data from scientific publication in PDF documents. Firstly, we transformed all input data to upper or lower case.

Tokenization is the splitting of sentences into the series of words. These token assist in the comprehension of the context or the development of NLP model. By evaluating the sequence of words, tokenization improves in interpreting the meaning of the text.

Stop word removal remove auxiliary verbs, conjunctions and articles in the sentences. It is crucial step in Natural Language Processing.

Reducing inflected words to their stem, base or root word form is known as stemming. Example: the word 'connection', 'connected', 'connects' will map to 'connect.

3.2. Feature Extraction

The bag-of-word model is widely used in Natural Language Processing (NLP). When using the bag-of-word model, a text is represented by a set of words call the "bag-of-words". Bag-of-words model is the method of pre-processing the text by formatting it as a number or vector, which finds the frequency of words in a given sentences. It is the extracting features method from the given text and use these features for model training. Bag-of-words is a method of converting words to numbers by creating vector embeddings from the tokens generated previously. For instance, supposed that there are two sentences: "It is a pencil" and "It is a pen for the baby". In these sentences, unnecessary words are filtered out, the set of words is equal to pencil and baby. In this case, vector (1,0) and vector (1,1) correspond to the first sentence and the second sentence respectively. That is, all sentences are represented by vectors, where each element of the vector represents the frequency of words [4]. We used the feature matching to define the measuring similarity between two documents.

3.3. Feature Matching

Before classification, similarity measures are determined. In this step training dataset is transformed into feature vectors and new features are appeared. We implemented the different measuring method in order to obtain the relevant feature from dataset. We used the approaches in feature matching are cosine similarity and correlation coefficient.

3.3.1. Cosine Similarity

When documents are represented as term vectors, the correlation between the vectors indicates how similar two documents are to one another. The cosine of the angle formed by two vectors, or the so-called cosine similarity. The cosine similarity between two research papers was calculated as follows:

$$Cosine\ Similarity = \frac{\sum_{i=1}^{n}(x_i - y_i)}{\sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}} \quad (1)$$

where,

$X = (x_1, x_2, x_3, \ldots, x_n)$ is the words vector of document A.

$Y = (y_1, y_2, y_3, \ldots, y_n)$ is the words vector of document B.

x_i and y_i are the frequency of i^{th} word in document A and document B.

3.3.2. Pearson's Correlation Coefficient

Pearson's correlation coefficient is widely used to verify whether there is linear relationship between research papers.

Pearson's correlation coefficient was calculated as follows:

$$r = \frac{n(\sum x_i y_i) - (\sum x_i)(\sum y_i)}{\sqrt{[n\sum x_i^2 - (\sum x_i)^2][n\sum y_i^2 - (\sum y_i)^2]}} \quad (2)$$

where,

$X = (x_1, x_2, x_3, \ldots, x_n)$ is the words vector of document A.

$Y = (y_1, y_2, y_3, \ldots, y_n)$ is the words vector of document B.

x_i and y_i are the frequency of i^{th} word in document A and document B.

The correlation coefficient is a value between -1 and +1. A correlation coefficient of +1 shows a positive correlation. A correlation coefficient of -1 shows a negative correlation.

3.4. Evaluations

The classification accuracy of the two different approaches were measured and evaluated with accuracy, precision, recall and f-measure and confusion matrix.

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

Precision is the fraction of the relevant documents that are relevant to the query.

$$Recall = \frac{TP + TN}{TP + FN} \quad (4)$$

The percentage of total relevant results correctly classified by the classifier is known as recall.

$$F1 - measure = \frac{2 * precision * recall}{precision + recall} \quad (5)$$

F-measures takes into account the mean of precision and recall. Confusion matrix is also developed for analyzing the classification results show the number of correct classification (true positive, true negative) and incorrect classification (false positive, false negative).

4. Experimental Results

We used the 2000 published scientific research paper that contains the domain field of Computer Science including metadata and their labels. In dataset preparation stage, we extracted metadata from scientific research papers by using python programming language and saved into CSV file. Then feature extraction stage, we extract the features from the dataset with bag-of-words model and classified research paper based-on relevancy score by using Excel VBA.

After the feature extraction step, feature matching is performed using two different measuring of cosine and correlation library. In this system, 80% of the dataset is used as training of classifiers and 20% is used as testing. The performed of each method is measured using precision, recall and f-measures.

The feature matching approaches upon the each of the two different measuring similarity are applied in this research paper classification systems. In accordance with results of individual and combined features, cosine similarity has the high accuracy in Bag-of-word based as presented in the following table:

Table1. Comparison results of two different measuring similarity

Method	Metric	Title	Keywords	Combined
Cosine	Precision	0.9367	0.9368	0.9725
	Recall	0.9356	0.9365	0.9778
	F1 Score	0.9336	0.9346	0.9777
Correlation	Precision	0.9322	0.9342	0.9578
	Recall	0.9337	0.9321	0.9575
	F1 Score	0.9357	0.9326	0.9578

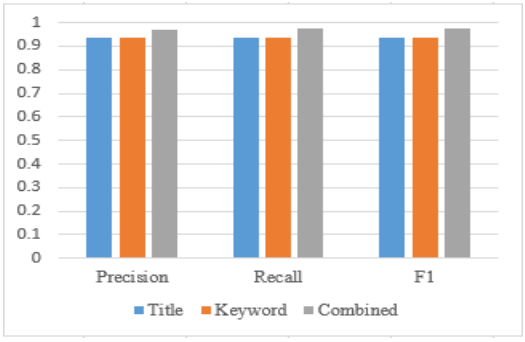


Figure 2. Individual and Combined Features results using Cosine Similarity

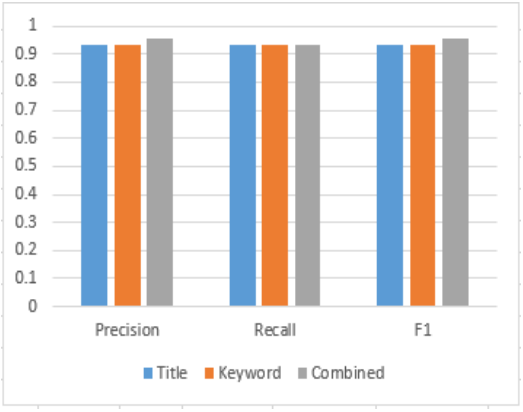


Figure 3. Individual and Combined Features results using Correlation Coefficient

Finally, we evaluated the relevancy score of metadata feature and classified the scientific research papers by using correlation coefficient and cosine similarity. The individual function of title and keywords are analyzed. We calculated the relevancy score of each title and keywords in metadata features that predicted the closely related with at least 5 papers. After that, counted the maximum frequency of paper index from them. Next, combination of title and keywords can be used to classify the scientific scope by calculating the relevancy score. The outcomes of relevancy score and classification scientific scope are shown in Fig. (4). It provides an approach to find relevant papers related to a particular paper. Moreover, it can also support relevancy score results for citing and cited papers.

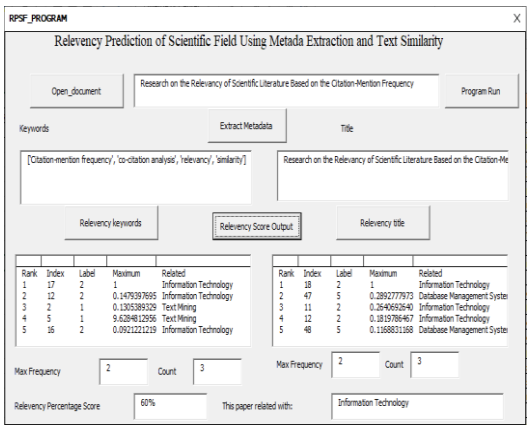


Figure 4. Classification of Research Papers Based-on Relevancy Score

5. Conclusion

Classification of scientific papers is one of the important research areas because that can be used in document clustering, research paper similarity, sentiment analysis and so on. In this paper, we presented the relevancy score of research papers and a classification that classifies the scientific research papers with the help of combination of metadata features such as titles and keywords. The combination of titles and keywords is superior in classifying research papers and closely related papers. In calculating relevance score, cosine similarity is found to be more accurate than correlation coefficient. The system is tested and evaluated using the research papers from scientific published in PDF documents real datasets. The proposed system benefits the scientific area and digital library to assist the researchers. In the future, possible combination of metadata features will be used to classify the documents by using different machine learning techniques.

Acknowledgements

The author thanks his supervisor Dr Kyaw Min Thein, Lecturer, Department of Computer Science, Higher Education Centre, Pyin Oo Lwin, who has brought him into a new territory of knowledge and Co-supervisors Dr Kyaw Myo Htun and Dr Htin Kyaw, Higher Education Centre, Pyin Oo Lwin, for their great support and invaluable advice during the progress of this research. And then, the author would like to express his sincere thanks to all compassionate individuals who helped directly and indirectly during this research.

References

[1] Anna Huang. "*Similarity Measures for Text Document Clustering*". In: New Zealand Computer Science Research Student Conference (NZCSRSC), Christchurch, April 2008.

[2] A. Fiallos, K. Jimenes, C. Vaca, and X. Ochoa, "*Scientific communities' detection and analysis in the bibliographic database: SCOPUS*", pp .118-124, 2017.

[3] D. Tkaczyk, P. Szostek, P. J. Dendek, M. Fedoryszak and L. Bolikowski, "*CERMINE - Automatic Extraction of Metadata and References from Scientific Literature*" 2014 11th IAPR International Workshop on Document Analysis Systems, 2014, pp. 217-221.

[4] G. Yu, W. Wang, C. Pak and T. Yu, "*Research on the Relevancy of Scientific Literature Based on the Citation-Mention Frequency*" in IEEE Access, vol. 7, pp. 181750-181757, 2019.

[5] Gobinda G. Chowdhury, "*Natural language processing*", Annual Revies of Information Science and Technology, pp (51-89), 2003.

[6] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. "*Research-paper recommender systems: a literature survey*". Int. J. Digit. Libr. 17, 4 (November 2016).

[7] Kok-Chin Khor and Choo-Yee Ting, "*A Bayesian Approach to Classify Conference Papers*", Multimedia University, 2006.

[8] Kim, SW., Gil, JM, "*Research paper classification systems based on TF-IDF and LDA schemes*", Hum. Cent. Comput. Inf. Sci. 9, 30 (2019).

[9] Qamar Mahmood, Muhammad Abdul Qadir, and Muhammad Tanvir Afzal, "*Finding relatedness between research papers using similarity and dissimilarity scores*". International Conference on Web-Age Information Management, 2014, pp. 707-710.

Authors

Biography

First A. Author Aung Myint Myat received

M.Sc degree in computer science from Higher Education Center (HEC), in 2018. He is currently working as Assistant Lecturer with Computer Science Department, Higher Education Center (HEC). His current research includes data science, text mining and sentiment analysis.

Second B. Dr Kyaw Min Thein received

M.I.C.Sc and Ph.D at Moscow Institute of Electronic Technology (M.I.E.T) in 2018. He is currently working as Lecturer with Computer Science Department.

Third C. Author Dr Htin Kyaw, a graduate of

B.C.Sc, was commissioned from the Defence Services Academy in 2005. He earned his M.I.C.Sc and PhD at the Moscow Institute of Electronic Technology (M.I.E.T) in 2015. Now, he is serving as an assistant lecturer at the Department of Computer Science in the Defence Services Academy (DSA)

MYANMAR VEHICLES' LICENSE PLATE DETECTION AND RECOGNITION SYSTEM USING OPTICAL CHARACTERS RECOGNITION IN IMAGE PROCESSING

Min Min Oo[1] and Myo Min Hein[2] and Zaw Ye Aung[3]

[1,2,3] Department of Computer Science and Technology, Higher Education Center
[1]minnnminnnoo@gmail.com, [2]myominhein48@gmail.com,
[3]alexanzawbmstu@gmail.com

ABSTRACT

A vehicle can be recognized by its license plate using the License Plate Detection and Recognition System, an image processing method. We propose using contour analysis to detect number plates from an image accurately and achieve high. This paper's research is mostly divided into four steps. First, the original image is preprocessed, and the color image is converted into a grayscale image. Second, the image is generated using the morphological operation of the image to display the image contour after having been binarized. The gradient edge features are extracted by the Canny operator. The license plate detection is performed for obtaining the region of the vehicle license plate and determined using the edge-based plate detection technique. Then, recognize and label the image's general contours. The performance of the proposed system has been tested on various images and provides better results.

KEYWORDS

Canny operator, image morphology, Transportation Information Management, and image contour analysis.

1. INTRODUCTION

A system for autonomously extracting vehicle license plate data from moving cars on a monitored road surface is known as a license plate detecting system. One crucial and extensively utilized element of contemporary intelligent transportation systems is license plate recognition. It examines the vehicle images or video sequences recorded by the camera to determine the particular license plate number of each car, concluding the recognition procedure. This research is based on digital image processing, pattern recognition, computer vision, and other technologies. When it comes to the issue of entering specific restricted places, security is a big concern that should be performed by the system. Vehicle detection, image acquisition, and license plate detection should all be a part of a single system for detecting license plates. Since the number of vehicles on the road daily, these technologies can provide faster and simpler responses in these situations.

When the vehicle detection part recognizes the entry of a vehicle, the image acquisition unit begins to capture the current video image. After processing the image and determining the location of the license plate, the license plate detecting unit creates the license plate number for output. Considering the actual situation, license plate images captured while going scroll has a variety of issues, such as light, weather, noise interference, license plate tilt, etc., which will result in poor picture quality and make it more difficult to recognize license plates. In order to minimize these interferences, it is essential to preprocess the image before license plate

localization. The following are the primary techniques for placing license plates that are often applied.

By detecting the vertical edges, the edge-based extraction technique successfully detects and extracts the character from an image. Finding all possible rectangles in the provided input image makes it simple to extract the license plate because the rectangular shape of the plate has a recognized aspect ratio property. Benefits include theft protection, access control management, and other functions that can also protect the safety of people and vehicles. ANPR systems are now widely used in a variety of public areas, such as parks and shopping malls. The ANPR systems are installed in more secure areas like military bases and governmental institutions like ministry offices. Moreover, in some protected areas like airports and military assets, this can also be utilized to monitor staff. Using the contour of license plate detection, an automated gate is proposed in this work. The other sections of the paper are structured as follows: Section 2 describes the issue, Section 3 describes the suggested solution, and Section 4 describes the results of the proposed approach.

2. RELATED WORK

Modern countries have seen a rapid increase in the number of vehicles in recent years. Transportation Information Management (TIM), includes toll management, smart parking, automatic gate security, and other criminal cases. Our nation's rapidly increasing vehicle population makes the traffic monitoring system especially important. The traffic monitoring system is supported by license plate recognition technology. But, this system has been unable to support detailed vehicle information, and the vehicle identification system takes on various challenging tasks. Deng Hongyao and Song Xiuli [3] presented a new method for Car License Plate Characters' Segmentation. The proposed approach is not only simple but also more effective than some of the existing methods reported earlier. The novelty lies in this case in its treatment which not only unites the projection and template match but also improves the techniques. Projecting the image in a horizontal direction can detect the approximate edge of a single license character in the vertical direction. Reza Fuad Rachmadi [9] showed how CNN is used to identify a vehicle's color. However, they showed that CNN can also learn classification based on color distribution. Traditionally, CNN is designed to understand the classification method based on shape information. This model effectively and precisely captures vehicle color.

A vehicle-type classification technique using deep learning technology was presented by Bensedik Hicham[10]. Different architectures of the CNN model are developed using parameters that have been learned from the training dataset. The automated traffic light control system is controlled by the proposed approach.

The studies mentioned above studied that the toll management system and traffic signal management system each detect a type of vehicle feature. In this paper, the proposed solution succeeded in getting over these constraints. The suggested system can be deployed at toll gates, bridges, airports, and other restricted places like military or governmental buildings for surveillance, automatic toll payment, etc. without constant human effort. Therefore, it is possible to efficiently reduce human effort, time, and cost.

3. THE PROPOSED SYSTEM

A license plate detection system based on contour analysis is analyzed in this paper. The system consists of two stages: license plate detection and character recognition. The license plate region is detected from the input image. Contour analysis which consists of contour extraction and vector representation is applied to the detected license plate. In this section, we discuss our proposed system. figure.1 shows the model of the license plate detection system used in this paper.

Vehicle color images are provided to the system as input. A global thresholding technique is used to convert RED GREEN BLUE-to-binary images in order to process license plate detection. By using the RGB-

to-binary conversion process, a color image (RGB) captured by a digital camera will be first converted to a grayscale image and then to binary images. Images that range from black at the lowest intensity to white at the highest are also referred to as black-and-white. The license plate recognition phase of the LPR system, and more importantly, this conversion, is the most important stage.

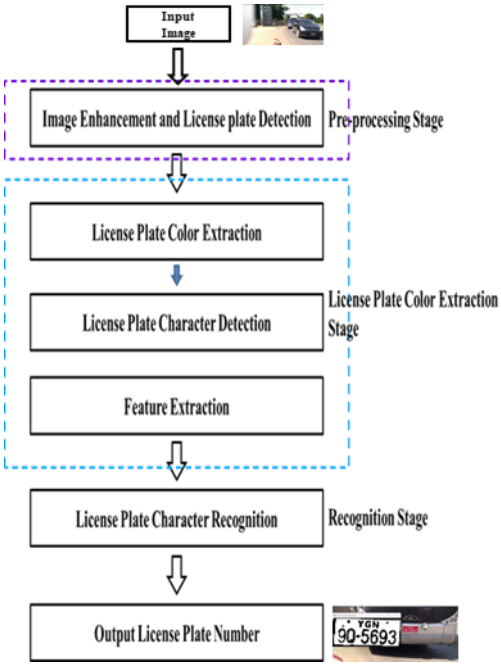


Figure 1. Flowchart of License Plate Detection and Recognition Algorithm

3.1 License Plate Detection

This research examined a contour-analyzing license plate detecting system. Character recognition and license plate detection are two of the system's four stages. From the input image, the license plate region is identified. The detected license plate is applied to contour analysis, which includes vector representation and contour extraction. Identifying the plate is a challenging task. Basically, the following reasons can cause the difficulty: The typical license plate makes up a small percent of the overall picture, and the formats, styles, and colors of license plates differ from country to country.

The video files of Myanmar vehicles' license plates which have been recorded at the toll gate by CCTV are collected. These video files are separated into each frame per second can be seen in Figure 2. Every frame has a resolution of 1920 x 1080 pixels.

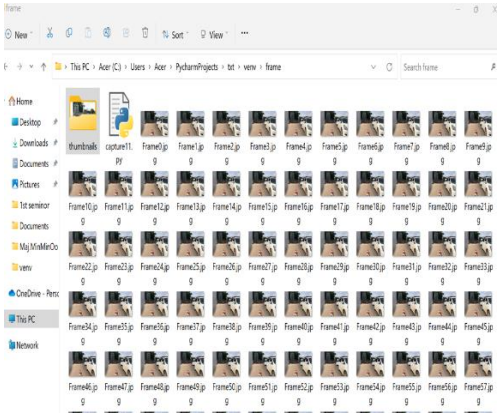


Figure 2. Collection of Dataset

For most of the period, the identification was carried out without prior knowledge of the location of the license plate in the image or the high probability of approaching some common issues that may affect the detection's efficiency, such as grayscale images, uneven or low illumination, looking to move vehicles, low image resolution, dirty plates, shadows or reflective surfaces, etc. There are many different ways, including region-based, edge-based, texture-based, etc. Our proposed system uses a contour-based method to recognize the plate from the license plate image.

Figure 3. Extracted License Plate

3.2 Extraction of the Region of Interest

Pre-processing of the real-time image capture includes grayscale conversion and median filtering noise removal. Based on their distinct structural features, the target

license plates are recognized using binarization and connected component labeling (Figure 1.). The headlights, upper, and lowers grills are recognized by the matching regions of interest (ROI) extraction. This area contains the differentiating data necessary to distinguish between different car makes and models. Figure 4. shows the ROI that was selected from a typical image.

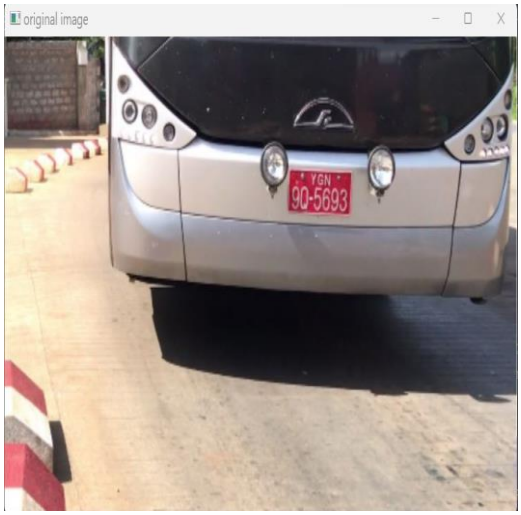


Figure 4. ROI selection for making Detection image

3.3 Bilateral Filter

A nonlinear filter known the bilateral filter [3] averaging space without smoothing the edges. It has been shown to be a successful image-denoising technique. It can also be used to decrease blocking effects. The selection of the filter parameters, which have a considerable impact on the results is a critical problem with the implementation of the bilateral filter. The follow is a formula for the bilateral filter:

$$BF[I]p = \frac{1}{Wp} \sum_{q\varepsilon s} G\sigma s(||p - q||)\ G\sigma r(|Ip - Iq)\ Iq \quad (1)$$

Where, $\frac{1}{Wp}$ is normalization factor, $G\sigma s(||p - q||)$ is space weight and $G\sigma r(|Ip - Iq)$ Iq is range weight. σs denotes the spatial extent of the kernel, i.e. the size of the neighborhood, and σr denotes the minimum amplitude of an edge. It helps to ensure that only pixels with intensity rates similar to the central pixel are considered for blurring while keeping sharp intensity changes. The edge gets sharper the smaller the value of r. The equation tends to have a Gaussian blur as r reaches infinity.

1. d: Each pixel neighborhood's diameter.
2. Sigma Color: The color space's value of. The nearer the colors are one another, the more they will begin to mix as the value is increased.
3. Sigma Space: The coordinate space value of. As long as the colors of a pixels are within the sigma Color range, the greater its value, the more pixels will mix together.

The following Figure is shown an original and smoothening image.

(a) (b)

Figure 5. Smoothening in image (a) original (b)smoothening

3.4 Binarization Algorithm

Image binarization is the process of converting a grayscale image into a binary image, which means reducing the information features of an image from 256 shades of gray only to two. Although thresholding may produce in images with more than two colors of gray, this is usually referred to as. It is a type of segmentation in which an image is divided into its individual objects. It is indeed normal procedure to perform this procedure while looking to extract an object from an image. It is not simple and primarily depends on

the content of the image, like many other image processing processes. Two different binarization methods are described here –

1. Binarization based on Global or Single Threshold: Generally, identify the single threshold that applies to the image sequence and binarize this. But this single threshold method generally eliminates or hides the local variation of the image, which may include some critical information or contents.
2. Binarization based on Region: Another method for binarization is also developed, in which the threshold is established based on the region. The image is actually divided up into various regions or windows.

Figure 6. Binarization in image

3.5 Edge Detection

In a digital image, edges are major local changes in intensity. A collection of connected pixels that forms a border between two existing regions is known as an edge. Three different edge types exist:

1. Horizontal edges
2. Vertical edges
3. Diagonal edges

A method for separating an image into regions of discontinuity is edge detection. It is a frequently used technique for processing images, like

1. pattern recognition
2. image morphology
3. feature extraction

Users can examine an image's features for a lift and drag in the grayscale by edge detection. This texture indicates the start of a new region in the image and the end of an existing one. It reduces the number of pixels in an image while keeping its structures. There really are two types of edge detection operators:

1. A gradient-based operator, such as the Sobel, Prewitt, or Robert operators, that calculates first-order derivations in a digital image
2. A Gaussian-based operator that produces second-order derivations inside a digital image, such as the Canny edge detector and the Gaussian Laplacian.

(a) (b)

Figure 7. Edge Detection in image (a) original image (b) edge image

A gaussian-based operator referred by Canny edge detector is applied to ways of getting. Noise doesn't always affect this operator. Without affecting or changing the feature, it extracts image features. The specialized algorithms in use by canny edge detectors are created from previous work on the Laplacian and Gaussian processes. It is a widely used technique for locating the best edges. Three components are applied to detect edges:

STEP I: Noise reduction through blurring by convolution Ing the input image I (I, j) with Gaussian filter G, noise in an

image is smoothed. The mathematical equation for the smooth final image is

$$F(i,j) = G * I(i,j) \quad (2)$$

Prewitt operators have become less complex for the operator to use than Sobel operators, and they're also more noise-sensitive.

STEP II: Searching for gradients means finding the edges where there is the maximum variation in grayscale intensity. An image gradient is used to determine the necessary areas. The gradient at each pixel of the smoothed image is determined using the Sobel operator. The following equations are the Sobel operators in *i* and *j* directions:

$$D_i = \begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix} \; and \; D_j = \begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{vmatrix} \quad (3)$$

These Sobel masks give gradients in the *i* and *j* directions after being convolved with a smoothed image.

$$G_i = D_i * F(i,j) \; And \; G_j = D_j * F(i,j) \quad (4)$$

Therefore, edge strength or magnitude of the gradient of a pixel is given by

$$G = \sqrt{G_1 + D_{ij}} \quad (5)$$

The direction of the gradient is given by

$$\theta = arctan \; G_i/G_j \quad (6)$$

G_i And G_j are the gradients in the *i* and *j* directions respectively.

STEP III: Non-maximum suppressions mean non-maximum suppression is being used, all local maxima in the gradient image are protected, and the edges remain thin by removing the others.

To compare the gradient magnitude of the pixels in the positive and negative gradient directions, first round the gradient direction 0° to nearest 45°. If the gradient is heading east, compare it to a gradient going east and west, represented by the letters E (i, j) and W (i, j), respectively. Retain the gradient's value and mark M (i, j) as the edge pixel if its edge strength is greater than that of E (i, j) and W (i, j), otherwise suppress or erase it.

STEP IV: Hysteresis thresholding: Local noise-induced maxima are still detectable in the output of non-maxima suppression. To avoid the problem of streaking, two thresholds 565 and 789 are used, or even one.

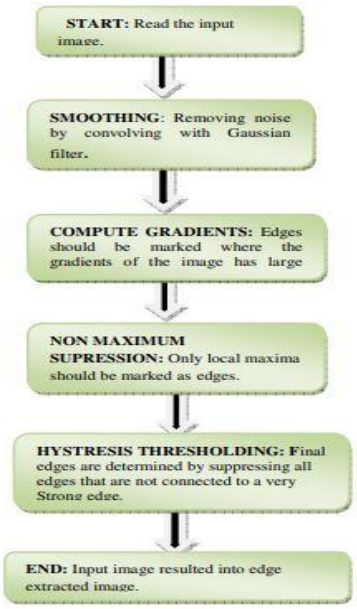


Figure 8. Flow chart of the Canny Edge Detection Algorithm

3.6 Optical Character Recognition

Optical character recognition (OCR) is the process of converting typed, handwritten, or written text images from documents, files, scenes (such the lettering on signs and buildings in a landscape photo), or subtitle text placed on an image into machine-encoded text. Another name for optical character recognition is text recognition (OCR). The data that an OCR system extracts from scanned documents, video images, and image-only PDFs is used for application areas.

OCR systems use suitable hardware and software to turn actual print texts into

machine-readable text. OCR software converts the document into a two-color or black-and-white representation. Dark areas are defined as characters that need to be recognized, while light parts are recognized as background in the scanned-in image or bitmap. After examining the dark regions, letters or numerals are found.

The system gets input data from vehicle color images. In order to conduct the license plate detection, RGB photos are converted to binary images using a global thresholding technique. A color image (RGB) captured by a digital camera will be converted to binary images using the RGB-to-binary transition process first. Black-and-white images are those with black as the lowest intensity and white as the highest. Generally, the conversion procedure is the most important time of the LPR system's license plate recognition phase. Figure 9 shows the general working of OCR.

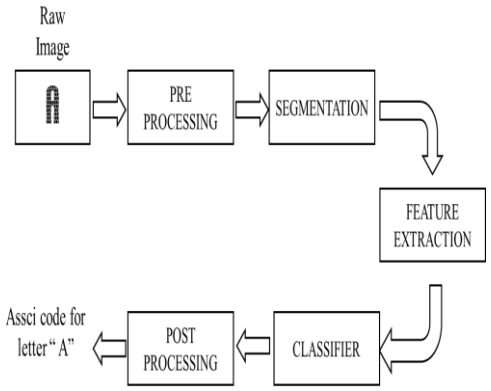


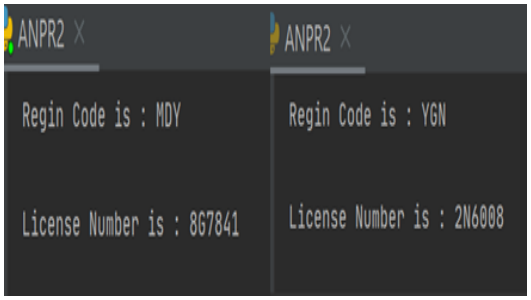
Figure 9. Flow Chart diagram of Optical Character Recognition (OCR)

4. Experimental Results

The method described in the article appears to be a contour-based system for number plate detection and recognition. Using reduced contours, the region of interest in this research, the number plate, is successfully detected. The contour approach shows how to separate the object's boundaries from the background of the image, emphasizing and rounding up the object's key features. Images produces all of the image data for an object, such as its area, features, angles between components, and contours.

The number plate is recognized in the input image and extracted based on the contours using the contours-based detection and optical character recognition methods. The figure represents how license plates were recognized using optical character recognition and contour-based license plate recognition on Myanmar vehicles. Python has been used to create this implementation, which included processes like detection and recognition. The data set we analyzed consists of more than 150 images that have been chosen at random. The Figure 11 show the regularity in which Myanmar vehicle license plates are detected and recognized.

(a)



(b)

(c)

(d)

Figure 10. Result of License Plate recognition using OCR (a) Original (b)Processing file (c) Test file (d) implementation of vehicle license plate

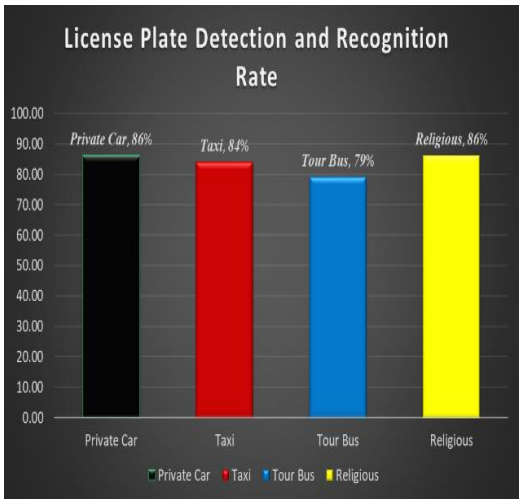


Figure 11. Detection and Recognition Rates.

5. Conclusion

This research paper's primary objective is to examine and observe an effective technique for recognizing Myanmar vehicles' license plates in different environmental conditions. Myanmar vehicles' license plate detection and recognition system, each vehicle's information are important for vehicle identification. In this paper, the vehicle license plate detection and recognition method using OCR are discussed in detail. Experiments and results show that the overall accuracy of the proposed methods reaches 86 %. Future work will focus on the comparison of our proposed methods with deep learning algorithms and YOLOV5 model recognition methods.

Acknowledgements

We would like to extend our appreciation to Dr. Kyaw Zin Win, the department's head of computer technology, Dr. Myo Min Hein and Dr Zaw Ye Aung for valuable pieces of advice, their close supervisor, kind guidance, support, and cooperation. Their energetic and enthusiastic supports are valuable encouragement during the period of study for this paper.

References

[1] JITENDRA MALIK February 23, 2001, “Contour and Texture Analysis for Image Segmentation”.

[2] Michael Randolph Maire (California Institute of Technology) 2003,” Contour Detection and Image Segmentation”.

[3] Deng Hongyao and Song Xiuli, “License Plate Characters Segmentation Using Projection and Template Matching”, International Conference on Information Technology and Computer Science, 2009.

[4] National Police Chief's Council. "Automatic Number Plate Recognition (ANPR)".

[5] Chirag Patel International Journal of Computer Applications (0975 – 8887) Volume 69– No.9, May 2013, “Automatic Number Plate Recognition System (ANPR): A Survey”.

[6] Md. Zainal Abedin 2017 IEEE Region 10 Humanitarian Technology Conference 23 Dec 2017, Dhaka, Bangladesh,” License Plate Recognition System Based On Contour Properties and Deep Learning Model”.

[7] Khin Pa Pa Aung, “Automatic License Plate Detection System for Myanmar Vehicle License Plates”, Japan Advance Institute of Science and Technology, 2018.

[8] M.M. Htay and A.K. Gopalakrishnan, “Localization and recognition of a Myanmar license plate based on partially cut character structure” 14th International Conference on ICT and Knowledge Engineering (ICT&KE), pp. 38 - 43, 2016.

[9] Reza Fuad Rachmadi, and Ketut Eddy Purnamay, "Vehicle Color Recognition using Convolutional Neural Network",15 Aug, 2018.

[10] Bensedik Hicham, Azough Ahmed, and Meknasssi Mohammed, "Vehicle Type Classification Using Convolutional Neural Network", IEEE 5th International Congress on Information Science and Technology (CiSt), Marrakech, Morocco, 21-27 Oct, 2018.

[11] Lubna 26 April 2021 "Automatic Number Plate Recognition: A Detailed Survey of Relevant Algorithms".

Authors

Biography

Min Min Oo received M.E (Computer) from the Bauman Moscow State Technology University (Kaluga) in 2011. He is currently working as Assistant Lecture with Computer Technology Department, Higher Education Center (HEC). His current research includes data science, text mining and sentiment analysis.

Myo Min Hein received M.I.C.Sc Russian State Technological University (MEPHI), (Russian Federation) in 2010 and Ph.D from Defence Services Academy Computer Science Department, in 2018. He is currently working as Assistant Lecture with Computer Science Department.

Zaw Ye Aung graduated with a B.C.Sc. and was commissioned by the University of Computer Studies, Yangon (UCSY) in 2002. His Ph.D., D.Sc. (Hons), and M.E. (IT) degrees are all obtained from Bauman Moscow State Technical University. He is working as a lecturer for the Higher Education Center's Department of Computer Science.

RESEARCH AND ANALYSIS OF DIESEL ENGINE FAULT CLASSIFICATION SYSTEM USING FEATURE FUSION AND MACHINE LEARNING TECHNIQUES

Nay Tun Aung[1], Htin Kyaw[2] and Aye Theingi Oo[3]

[1,2] Department of Computer Science, Higher Education Centre

[1]naytunaung.mm.53@gmail.com, [2]mgnyeinaye48@gmail.com

[3]ayetheingioo77@gmail.com

ABSTRACT

The diesel engine's acoustic signals can provide important information regarding the health of the engines. These signals can be used to identify engine defects. These signals, however, are complicated and made up of a defective component and additional background noise signals. Although Fault Diagnosis System (FDS) has been proposed by many researchers, the accuracy of fault diagnosis in diesel engines is still low due to a large number of components, demanding operation, and high vibration, which makes fault identification challenging. To address this problem, FDS needs to have the most discriminating features. This study examined the use of features, leading to the development of a novel feature extraction method for FDS. 30 Wavelet Transform features and 20 MFCC features are extracted and normalized. The concatenation-based feature fusion technique is applied to build a single feature vector. Utilizing the extracted feature vectors, the fault types are then classified using Machine Learning Techniques. The best-fit classifier is reported for the proposed feature vector. According to the experimental result, the proposed method achieved the best accuracy with the KNN classifier.

KEYWORDS

Diesel Engine, Fault Diagnosis, Feature Extraction, Machine Learning

1. INTRODUCTION

The process of determining the reasons and recommending corrective actions is known as the fault diagnosis system (FDS). Engine maintenance has become more crucial than ever due to the vehicle industry's rapid growth. Therefore, creating a precise FDS will be useful. An abrupt release of strain energy causes a stress wave known as acoustic emission (AE) waves to propagate through the materials. In various industries, AE has been utilized extensively for non-destructive testing. Additionally, AE has been used to identify and diagnose mechanical faults in gearboxes, engines, and bearings. The internal combustion engine's acoustic signals are made up of numerous parts that come from various sources. These sources include mechanical impacts, combustion, and a combination of both. To identify the requirements for acoustic signal analysis, the acoustic signal's components are crucial. The two components of a diagnosis system are feature extraction and classification. The classification accuracy increases as more significant features are extracted. As a result, the feature extraction process in FDS is crucial. Therefore, in this study, a new contribution to novel feature extraction in fault diagnosis for diesel engines is presented based on the acoustic signal. The rest of the paper is structured as follows. The adopted method is provided in section 3, while the fundamental theory and literature evaluation are covered in section 2. The experimental results and the proposed model are discussed in section 4 of the paper. Section 5 concludes to wrap everything up.

2. LITERATURE REVIEW

Depending on the nature of the signals and the quality of the gathered data, signal processing techniques could be appropriate. There are many different signal-processing techniques available in the literature. Acoustic signal processing now makes considerable use of deep features and hybrid features due to the

advancement of deep learning algorithms. Typically, it is challenging to extract the fault characteristics efficiently, which leads to poor diagnosis performance. [1] offers a bearing fault detection approach based on feature fusion and suggests a bearing fault feature extraction method to address the issue. To create the time-frequency characteristic matrix for the signal, the time-frequency feature of the bearing signal is first extracted using the Wavelet Packet Transform (WPT). Next, the Multi-Weight Singular Value Decomposition (MWSVD) is built using the singular value contribution rate and entropy weight. The features of the time-frequency feature matrix acquired by WPT are further extracted, and the fault-sensitive features are kept while the fault-insensitive features are subtracted from the time-frequency feature matrix. To diagnose bearing faults, the extracted feature matrix is finally fed into the Support Vector Machine (SVM) classifier. Maximum and minimum amplitudes, standard deviations, and means are calculated using time and frequency domain statistics in [2]. A novel extreme gradient boosting-based misfire fault diagnosis approach utilizing the high-accuracy time–frequency information of vibration signals is presented in [3]. [4] proposed a fault identification model for rotating machinery utilizing time series raw vibration data and k-means clustering. Using acoustic data recorded from the engine's cylinder heads, [5] proposed an automated model and hyperparameter selection approach for engine combustion fault classification. The acoustic waves were split up using WPT, and the statistical characteristics were then retrieved from the wavelet coefficients and fed into the three classification models. [6] presented the fault diagnosis method for motorcycle engines using sound analysis by the implementation of the MFCC algorithm.

3. RESEARCH METHODOLOGY

In this study, a feature extraction approach based on fusion is used to implement a diesel engine fault diagnosis system. Data collection and pre-processing, feature extraction, and engine state detection and classification make up the three steps of the proposed method.

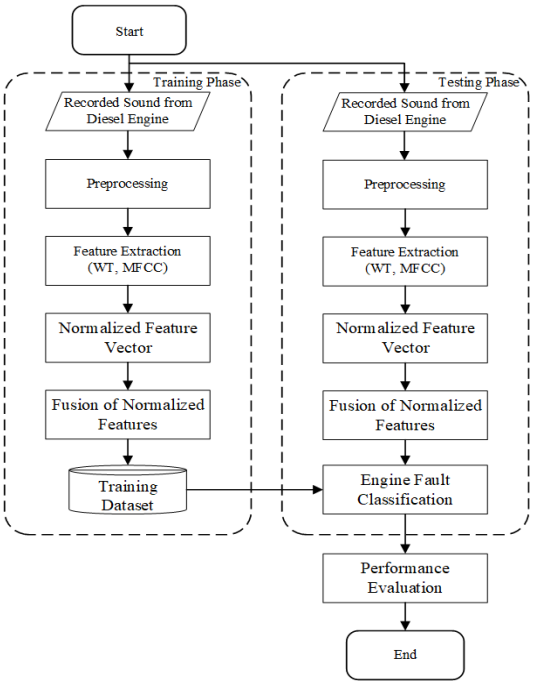


Figure 1. Proposed System Design

Firstly, data are collected from Sino Truck Engine under healthy and faulty conditions by using an external microphone with a frequency sampling rate of 44100 Hz and 16 bits resolution. It is then implemented to normalize the engine sound signal. Secondly, signal feature extraction for training and testing models is proposed. Finally, a classifier model for the detection of fault types using machine learning is implemented. Figure 1 describes the proposed system design's flowchart. The detailed procedures are described in subsequent sections.

3.1. Data Collection and Pre-processing

The dataset for this study is recorded from Sino Truck (WD 615.47 EURO-II) engine. The specification of the engine model is described in Table 1. The engine sound is recorded 1m away from the engine block using RP1 Re-max Voice Recorder. The data quality is the sampling rate of 44100 Hz, mono channel and 16 bits resolution. The duration of each audio clip is 2 seconds. Firstly, the sounds are recorded in healthy state. And then different fault types are created and recorded. The recorded acoustic data is stored in (.wav) format.

Table 1. Engine Specifications

Serial	WD 615.47 EURO-II
No. of Cylinder	6
Fuel Type	Diesel
Emission Standard	EURO-II
Rotating Speed	2200 rpm
Max. Horsepower	371 hp
Maximum Torque	1500 N.m
Noise	<= 98 dB

Applying a consistent amount of gain to an audio recording to reduce the amplitude to the desired level is known as audio normalization. Audio normalization is performed because the signal-to-noise ratio and relative dynamics are unchanged. Mathematically, the normalization equation is represented as,

$$x_{norm} = (x - x_{min})/(x_{max} - x_{min}) \quad (1)$$

Where x_{norm} is the normalized value of x.

3.2. Feature Extraction

Any ML algorithm's performance is based on the features used for training and testing. As a result, feature extraction is a crucial step in the machine-learning process. Wavelet Transform (WT) and Mel Frequency Cepstral Coefficients (MFCC) features are extracted during the feature extraction step, and their respective efficiency is examined. The result of each feature and the combining features are then reported. In the following subsection, specifics of the feature extraction techniques are covered in detail.

3.2.1 Wavelet Transform Decomposition

Multi-Resolution Analysis (MRA) is a framework used by WT. Two digital filters—a high-pass filter and a low-pass filter—are used by WT to deconstruct the signals. The scaling function ϕ(t) and wavelet function φ(t) are the names of the high-pass and low-pass filters, respectively. The original signal from both digital filters is passed to calculate the WT coefficients. The original signal is generated in detail by the high-pass filter, and approximately by the low-pass filter. Equations (2) and (3) define the wavelet function and scaling function that must be defined before the WT is applied to the signal.

$$\emptyset_{(j,n)}(t) = 2^n \sum_n c_{j,n} \emptyset(2^j t - n) \quad (2)$$

$$\varphi_{(j,n)}(t) = 2^n \sum_n d_{j,n} \varphi(2^j t - n) \quad (3)$$

Where, $\emptyset_{(j,n)}(t)$ and $\varphi_{(j,n)}(t)$ scale and translate the functions $\varphi(t)$ and $\emptyset(t)$ and c_j and d_j are the approximation and detail coefficients at scale j. The original signal is given $X_j(t) = (V_0, V_1, V_2, \dots, V_{N-1})$ with the length of N that $N = 2^j$ is the sampling number and j is an integer. The DWT mathematical formulation is as follows:

$$DWT\left(X_j(t)\right) = 2^{\frac{j+1}{2}}(\sum_{n=0} u_{j+1,n} \emptyset(2^{j+1}t - n) + \sum_{n=0} w_{j+1,n} \varphi(2^{j+1}t - n) \quad (4)$$

Where, j is the translation coefficient and $u_{j+1,n}$ and $w_{j+1,n}\varphi$ are the approximated and detailed version at scale $j + 1$ and are calculated as follows:

$$u_{j+1,n} = \sum_k c_{j,k} v_{j,k+2n}, \quad 0 \le k \le \frac{N}{2^j} - 1 \quad (5)$$

$$w_{j+1,n} = \sum_k d_{j,k} v_{j,k+2n}, \quad 0 \le k \le \frac{N}{2^j} - 1 \quad (6)$$

Where, $u_{j+1,n}$ and $w_{j+1,n}\varphi$is the approximated and detailed version at scale $j + 1$.

Figure 2 shows the decomposition and reconstruction of the WT. The letters A and D in Figure 2 stand for approximate and detailed versions, respectively. The detail and approximation coefficients of *X(t)* in Wavelet transform, which are high-pass and low-pass filters, respectively, are down-sampled to be half the length of *X(t)*. The approximate version of the previous decomposition level is sent through the low-pass filter to create the next decomposition level. This pattern keeps repeating up to a certain level of deconstruction.

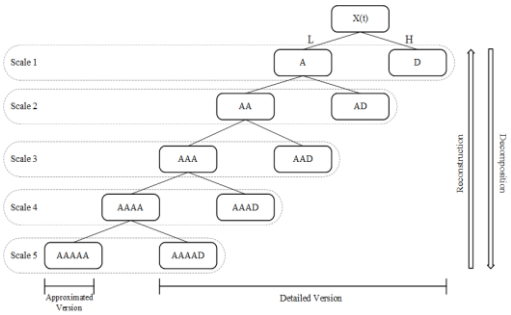


Figure 2. WT decomposition/reconstruction with five levels

The key benefit of the WT method is the use of resizable window widths that may be sharpened for higher frequencies and dilated for lower frequencies. In this study, signals are divided into five levels: one level, two levels, three

levels, four levels, and five levels. Features are extracted from the wavelet tree using a window size of 256 at 128 increments. Six features per window from each channel are selected to be extracted for a full tree at five levels. 30 features are obtained in a 5-level decomposition. The input signal is also used to extract MFCC coefficients. MFCC feature extraction method is extensively discussed in the next section.

3.2.2 Mel Frequency Cepstral Coefficient (MFCC)

MFCCs is the most widely used feature extraction method. Davis and Mermelstein proposed the MFFCs in the 1980s, and they have remained state-of-the-art since then. 20 MFCCs coefficients are extracted from the input signal. The theory of MFCCs feature extraction method is extensively discussed in the following.

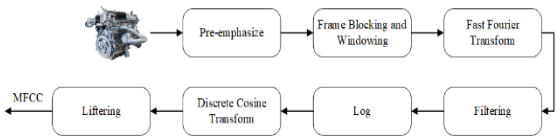


Figure 3. Block Diagram of MFCC Feature Extraction Method

The input signal must first go through a pre-emphasis filter to extend the high frequencies. The first-order filter in the equation below is used to apply this filter to a signal x.

$$y(t) = x(t) - \alpha\, x(t-1) \qquad (7)$$

Where, α is the filter coefficient.
After pre-emphasis, the signal is divided into low-time frames using framing. The signal is divided into brief time frames, and then each frame is given a different window function, such as the Hamming window.

$$w[n] = 0.54 - 0.46\cos(\frac{2\pi n}{N-1}) \qquad (8)$$

Where, $0 \le n \le N\text{-}1$, N is the window length.
Then the Fast Fourier Transform (FFT) is applied using the next formula:

$$P = \frac{|FFT(x_i)|^2}{N} \qquad (9)$$

Where, xi is the i^{th} frame of signal *x*.

The mapping from linear frequency to MFCC is then defined, and filter bank coefficients are determined using the following equation,

$$Mel(f) = 2595\log_{10}(1 + \frac{f}{700}) \qquad (10)$$

$$f = 700(10^{m/2595} - 1) \qquad (11)$$

Where, *f* is physical frequency and *Mel(f)* is the estimate of Mel from physical frequency. Finally, the MFCC was calculated using the discrete cosine transformation algorithm below.

$$MFCC_i = \sqrt{\frac{2}{N}}\ \sum_{j=1}^{N} m_j \cos(\frac{\pi i}{N}(j - 0.5)) \qquad (12)$$

Where, *N* is the number of bandpass filters, m_j is the log bandpass filter output amplitudes.

3.2.3 Feature Normalization

Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Max-Min normalization. Max-min normalization is one of the most common methods to normalize data. And it is an important part of machine learning for datasets. If the features are extracted from many datasets without a normalization method, one of the data may dominate the rest of the data. Therefore, the normalization method is used in this study. The mathematical form of normalization has been described in equation (1).

3.2.4. Feature Fusion

Fusion is a process of combining the specific extracted features which are stored in a dictionary to obtain a single feature file, which is very informative. To break the limitation of a single feature extraction algorithm, the study applied a fusion-based feature extraction method for diesel engine fault classification. The two normalized feature vectors are combined by concatenating.

$$Combined\ Feautre = [F1, F2] \qquad (13)$$

Where, $F1$ is the WT feature vector and $F2$ is MFCC feature vector.

3.3. Classification Algorithms

A subset of artificial intelligence is called machine learning. Algorithms for machine learning are used to classify or predict. An algorithm will provide an estimate of a pattern in the data based on certain input data, which can be labelled or unlabelled. There are three types of machine learning models: supervised, unsupervised, and reinforcement learning. This study looks at supervised machine learning techniques, which are frequently employed in condition monitoring to distinguish between sound and flawed signals. Six standard classifier algorithms were considered: decision tree, discriminant analysis, Naïve Bayes, Support Vector Machine (SVM), k-Nearest Neighbours (KNN), and ensemble classifier. These classifiers are explained briefly as follows.

3.3.1 Decision Tree

A decision tree carries out classification by creating a tree based on data attributes. To identify patterns in data and create decision-making rules, it employs hierarchical structures to calculate the correlation between independent and dependent variables. Minimum leaf size, maximum splits, and split criteria, which also include the Gini's diversity index, the twoing rule, or the maximum deviation reduction, are the hyperparameters for optimization in this case.

3.3.2 Naïve Bayes

Naïve Bayes is a type of supervised learning algorithm based on the Bayes theorem, which classifies new data using conditional probability to predict the outcome of an occurrence. The hyperparameters used for optimization here are the distribution type, the window width and the kernel type. The distributions used include Gaussian, multivariate multinomial, multinomial and kernel. The kernel smoothing window width is given by

$$f(x) = \frac{\sum_{j=1}^{n} k\{\frac{x-X_i}{h}\}}{nh} \qquad (14)$$

where *K(x)* is called the kernel function that is generally a smooth, symmetric function such as a Gaussian and $h > 0$ is called the smoothing bandwidth that controls the amount of smoothing. Basically, the kernel function smooths each data point X_i into a small density bumps and then sum all these small bumps together to obtain the final density estimate.

3.3.3 K-Nearest Neighbour

The KNN algorithm is a non-parametric approach to classification and is a supervised learning technique. It classifies new data by choosing the k closest instances. The KNN is optimized using three hyperparameters: the distance measure, the number of neighbours, and the distance weights. The distance between two instances $X = (x_1, x_2, \dots, x_n)$and $Y = (y_1, y_2, \dots, y_n)$can be calculated as:

$$Distance\ d(x,y) \sum_{i=1}^{N} \sqrt{(x_i - y_i)^2} \qquad (15)$$

Where, x_i is the extracted feature vector and y_i is the tested feature vector.

3.3.4 Support Vector Machine

Support Vector Machine (SVM), is one of best machine learning algorithms, which was proposed in 1990's and used mostly for pattern recognition. A supervised learning technique, SVM, performs classification by constructing a hyperplane with the greatest possible margin between two classes. The Box constraint, the kernel function, the kernel scale, and the polynomial order are the hyperparameters that are utilized to optimize the SVM. Box constraint manages the penalty for out-of-margin observations and aids in avoiding overfitting. Gaussian, linear, and polynomial kernel functions are used for optimization.

3.3.5 Discriminant analysis

Discriminant analysis is a powerful subspace method that additionally applies the Bayes theorem like Naive Bayes. Using probability densities, the Bayesian rule splits the data space into regions that are not related to each other and represent all the classes. Delta, Gamma, and the type of discrimination are the hyperparameters for optimization in this case. When estimating the covariance matrix, the amount of regularization to be used is called Delta, and the linear coefficient threshold is called Gamma. Linear and quadratic

discriminates are two instances of those employed.

3.3.6 Ensemble classifier

The ensemble classifier is a relatively new classification method that has become more and more popular in recent years. To increase the model's accuracy, several categorization approaches are used. The classification algorithm, the number of learning cycles, the learning rate, and other hyperparameters are being optimized in this case. Multiple techniques, including adaptive boosting, linear programming boosting, and random under-sampling boosting, are utilized to solve multiclassification issues. Bagging is the optimal ensemble approach for the suggested feature vector. An ensemble learning technique called bagging aims to have a varied collection of ensemble members by changing the training data. Following that, the ensemble members' predictions are combined using basic statistics like voting or average. The preparation of each dataset sample to train ensemble members is a crucial component of the technique. A distinct sample of the dataset is given to each model. A row is returned to the training dataset if it is chosen so that it may be chosen again using the same training dataset. For a certain training dataset, this means that a row of data may be chosen 0, 1, or multiple times. Instead of directly estimating from the dataset, a better overall estimate of the desired quantity can be obtained by creating numerous distinct bootstrap samples, estimating a statistical quantity, and then finding the mean of the estimates.

4. EXPERIMENTAL SETUP

The acoustic data are recorded in normal and three different fault conditions. And then, they are stored in (.wav) format. The total number of recorded audios is 300 and the duration of each audio clip is 2 seconds. MATLAB is used for the effective analysis of the proposed model.

4.1. Experimental Result

The main experimental process in this paper is divided into three parts. The first part is data collection and pre-processing. The second part focuses on the feature extraction of engine sound signals from healthy and faulty conditions. And the third part is mainly to establish a database of input sound signals which can be used as training data for different classifiers. Sound signatures of different fault types are shown in figure 3. Using an external microphone, the samples were collected. The total amount of the sample size is 300 and each has 2 seconds duration. To ensure the correctness of the study, the samples are collected under various circumstances. First, samples were gathered from functional, fault-free automobiles. These samples are crucial because they determine if the engine is defective or not. Then, three fault types were produced, and their noises were recorded. They serve as databases or reference sounds. There are several recordings made for each error. Additionally, the noises were captured both while the engine was idle and when it was rotating at 2000 Rounds Per Minute (rpm).

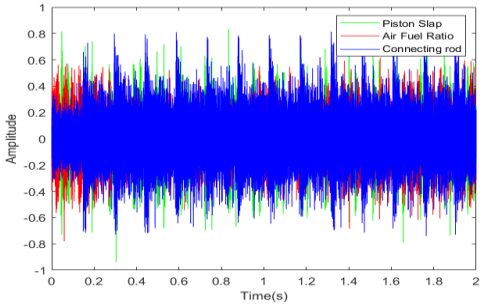


Figure 4. Sound Signature of the Diesel Engine

In feature extraction stage, the total dataset was divided into a training dataset and a test dataset in a ratio of 80:20. It means that 240 records are used as a training dataset and 60 records are used to test. Using maximum/minimum normalization, the input signal is normalized to become the desired level of amplitude because the signal-to-noise ratio and relative dynamics are unchanged. The adopted features extracted from the input acoustic data are the WT and the MFCCs. The number of MFCC features and the level of WT decomposition were updated to improve the performance of our proposed system. 30 WT coefficients and 20 MFCC features are selected for our proposed model. Following the time-frequency approach in feature extraction, the raw signal is windowed into some frames and the frequency rate (fs) of the acoustic signal decides how many frames can be obtained. Therefore, 688 frames are

obtained. For the WT, 6 coefficients of each of the frames are computed for each level. It means that there are 30 (6 x 5) coefficients in 5-level decomposition. For the MFCC, 20 coefficients are computed. Without any dimension reduction, the extracted feature vectors are normalized and combined into a single feature vector. These features are saved in (.mat) format. The total number of training data is (240 x 50). The extracted features are fed to the machine learning algorithms to classify the fault types of the engine. In testing stage, different machine learning algorithms are used to classify the fault type and compare their results. And then, the proposed classification algorithm is selected for our proposed feature vector. Table 2 shows the number of extracted features and the classification results obtained from different classification algorithms.

Table 2. Classification Accuracy Obtained from Different Models

Features	No. of Coefficient	DT	Discriminant analysis	NB	SVM	KNN	Ensemble classifier
WT	30	95.5%	96.7%	96.7%	93.3%	98.8%	85.4%
MFCC	20	80.0%	80.0%	83.3%	93.3%	96.7%	83.3%
WT + MFCC	**50**	**96.7%**	**100%**	**100%**	**96.7%**	**100%**	**86.7%**

The proposed model is obtained by running the data in each classifier using default parameters and finding the best-fit model for fault diagnosis. Although NB, KNN, and Discriminant Analysis have the same classification result, KNN is less training time. Therefore, KNN is selected for our proposed method. The best-fit classifier and default parameters used by the proposed model are listed in Table 2.

Table 2. Optimized Hyperparameters of the Proposed Model

Model	**Optimized Hyperparameters**		
	Neighbour	Distance	Distance Weight
KNN	3	Spearman	Equal

4.2. Performance Evaluation

The dataset for this study is recorded from Sino Truck (WD 615.47 EURO-II) engine. The engine sound is recorded 1m away from the engine block using RP1 Re-max Voice Recorder. The data quality is the sampling rate of 44100 Hz, mono channel and 16 bits resolution. The duration of each audio clip is 2 seconds. Firstly, the sounds are recorded in healthy state. And then different fault types are created and recorded. The recorded acoustic data is stored in (.wav) format.

For the experiments, a total of 300 audio data in which 240 samples are used to train the model and 60 samples under the healthy and faulty state are used to evaluate the performance of the proposed model. The accuracy comparison results are shown in Figure 5.

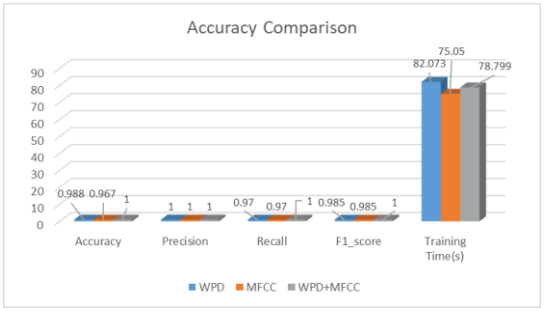


Figure 5. Accuracy Comparison of Proposed Model

WT and MFCC coefficients are then extracted and normalized from the input signal. The number of MFCC features and the level of WT decomposition were updated to improve the performance of our proposed system. 30 WT coefficients and 20 MFCC features are the optimised features for our proposed model. Therefore, each audio signal has size of (1 x 50). The total number of training data is (240 x 50). The extracted features are fed to the machine learning algorithms to classify the fault types of the engine. The accuracy of our proposed system obtained from different classifiers is described in Table 1. Obtained results were evaluated comparatively according

to accuracy, precision, recall, and f-score values calculated as given in Equations 16-19.

$$Accuracy = \frac{True\ Positive+True\ Negative}{P.+True\ N.+False\ P.+False\ N.} \quad (16)$$

$$Precision = \frac{True\ Positive}{True\ Positive+False\ Positive} \quad (17)$$

$$Recall = \frac{True\ Positive}{True\ Positive+False\ Negative} \quad (18)$$

$$F1-score = 2\ x\ \frac{Precision\ x\ Recall}{Precision+Recall} \quad (19)$$

5. CONCLUSIONS

In this study, the effectiveness of features and a novel extraction of the feature vector for FDS is investigated. Three faulty conditions, namely connecting rod bearing fault, piston slap fault, and air-fuel ratio variation, were used to train and test the applicability of the approach. The best accuracy and less computational time are achieved with the KNN classifier for the proposed method. The computational time for the proposed feature vector is 78 s. The aim of the study was not only to achieve the highest accuracy but also to reduce computational time. In doing so, the feature sets with reduced dimensions would be more suitable for future implementation in engine fault diagnosis applications. Currently, the author is collecting more datasets to test the accuracy changes when the data size and fault types are increased.

ACKNOWLEDGEMENTS

The author thanks his supervisor Dr Htin Kyaw, assistant lecturer, Department of Computer Science, Higher Education Centre, Pyin Oo Lwin, who has brought him into a new territory of knowledge and Co-supervisors Dr Aung Myo Thu and Dr Htike Aung Kyaw, Higher Education Centre, Pyin Oo Lwin, for their great support and invaluable advice during the progress of this research. And then, the author would like to express his sincere thanks to all compassionate individuals who helped directly and indirectly during this research.

REFERENCES

[1] Huibin Zhu, Zhangming He, Juhui Wei, Jiongqi Wang and Haiyin Zhou (2021) "*Bearing Fault Feature Extraction and Fault Diagnosis Method Based on Feature Fusion"*, Sensors 2021, 21, 2524.

[2] Mayuraj Ekatpure1, Rohan Sevlikar, Saurabh Kamble (2022), "*The Condition Monitoring of I.C. Engine using Acoustic Signal Analysis*", Department of Mechanical Engineering, Savitribai Phule Pune University, Maharashtra, India, Volume: 09.

[3] Tao J, Qin C, Li W, Liu C. Intelligent Fault Diagnosis of Diesel Engines via Extreme Gradient Boosting and High-Accuracy Time-Frequency Information of Vibration Signals. Sensors (Basel). 2019 Jul 25;19(15):3280. doi: 10.3390/s19153280. PMID: 31349707; PMCID: PMC6695824.

[4] QingFeng Wang, Jiahe Liu, Bingkun Wei, Wenwu Chen, ShuJian Xu (2020) *"Investigating the construction, training and verification methods of k-means clustering fault recognition model for rotating machinery"*, IEEE, 2020.3028146.

[5] Steve Koshy Mathew, Yu Zhang (2020) "*Acoustic-Based Engine Fault Diagnosis Using WPT, PCA and Bayesian Optimization",* Applied Sciences 10, no. 19: 6890.

[6] A. K. Kemalkar and V. K. Bairagi, "Engine fault diagnosis using sound analysis," 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), Pune, India, 2016, pp. 943-946, doi: 10.1109/ICACDOT.2016.7877726.

Authors

Biography

First A. Author Nay Tun Aung, a graduate of

B.C.Sc, was commissioned from the Defence Services Academy in 2010. He earned his M.E (Radio Electronic) at Moscow Power Engineering Institute (MPEI) in 2016. Now, he is a PhD candidate at the Higher Education Centre (HEC) in Pyin Oo Lwin.

Second B. Author Dr Htin Kyaw, a graduate of

B.C.Sc, was commissioned from the Defence Services Academy in 2005. He earned his M.I.C.Sc and PhD at the Moscow Institute of Electronic Technology (M.I.E.T) in 2015. Now, he is serving as an assistant lecturer at the Department of Computer Science in the Defence Services Academy (DSA).

Third C. Author Aye Theingi Oo, a graduate of

BE.EC, was commissioned from TU(Mawlamyine). in 2012. He earned his ME.EC in 2020. Now, she is a PhD candidate at UTYCC in Pyin Oo Lwin.

USAGE EXTENT OF MULTI-USERS USING WEB USAGE MINING

Tin Nilar Win[1], Nang Khine Zar Lwin[2]

[1]Faculty of Information Technology Supporting and Maintenance, University of Computer Studies (Thaton), [2]University of Technology, Yadanarbon Cyber City (UTYCC)

[1]tinnilarwin@ucstt.edu.mm,[2]nangkhinezar@gmail.com

ABSTRACT

Today, the web sites are increasing tremendously in its volume and these are much essential for the customers to choice the quality products. The increasing popularity of the web pages are becoming complex so that multi-users are tracking the navigational behaviors using Web usage mining. Web mining is the research area of data mining techniques on web data. Web Usage Mining (WUM) for personalized services, adaptive websites, customer profiling, are used in log data to extract user behaviors used in application such as creating attractive website. To get it, usage mining depends on the access log file of web. In this paper, we mainly focus to extract the relevant information to identify the web pages visited by users most. The system applies five phases such as data acquisition, data preparation, data cleaning, user identification, and session identification to extract the visited web sites of users in log data.

KEYWORDS

Web usage mining, access log file, visited web sites, relevant information

1. INTRODUCTION

The World Wide Web is a very rich source of information gathering, so, a lot of people are increasingly interested in analyzing web log files, which can provide more useful insight into website usage. Every web user spends most of their time on the internet and their behaviors are different from each other. Web servers store the traffic of web pages requested by users as the web access log files. To understand the users' behavior on the internet, web usage mining accesses characteristics from the web log files, extracting hidden pieces of knowledge of users; such as internet usage. Web usage mining is used to discover the user's private information from the web data, it is one of the techniques of data mining used to understand and better serve the need of the web-based application. Also, it is a web mining process that tracked and analyzed the browsing behavior of users on the web. Therefore, the websites' owners take to recognize the access patterns of their users. By collecting and analyzing the behaviors of user activities according to mine the web access log files, web sites owners can enhance the quality and performance of services to catch the attention of existing as well as new users [3].

Web Usage Mining mines the usage features of web pages, such as user's IP address, time and date of access, status code, number of transmitted data size, referrers' address, user agent, and POST or GET method. Reorganize websites effectively, links and navigation improvements, its aim to help with better recommendation and personalization. Therefore, Web log data represents the typical behavior of the users while browsing web pages that can be stored on the Web server.

The main purpose of this paper is to identify the usage extend of user's path or referrer, count, page, time stamp and to discover the usage extent of users which helps in better serving the websites. To identify the web pages visited by users most, the system comprises four different tasks: data acquisition, data preparation, data cleaning,

user identification, and session identification. This research paper provides an analysis of web usage methods, including the preprocessing stages used in web usage mining. This paper discusses web usage mining for users as follows. In the Section II, discuss about the related work. Section III describes the background knowledge of accessibility to web mining techniques. In Section IV, overview of the proposed system architecture is explained how to use the tasks to recognize the most visited users and implements the details of the results and concludes in Section V respectively.

2. RELATED WORKS

The rising popularity of electronic commerce makes data mining an indispensable technology for several applications, especially online business competitiveness. The World Wide Web provides abundant raw data in the form of Web access logs. However, without data mining techniques, it is difficult to make any sense out of such massive data. Web mining aims to discover useful information or knowledge from the web hyperlink structure, page content, and usage data. For the literature [1], in this paper is that Web mining is the application of data mining techniques used to extract the suspected behaviors of users, different types of web attacks, criminal pattern identification thru the web log files. A web log file contains noisy data thru preprocessing step; we eliminate irrelevant data so this step is very important in web mining. After preprocessing pattern discovery and analysis phase play an important role to identify criminal activity and predict the suspected user behavior. For the literature [2],in this paper emphasizes on the idea of better understanding of user's profile and site objectives, as well as the way users will browse web pages to better serve them with list of web pages which are relevant to him by comparing with user's historic pattern using different web usage mining techniques. In the Author [3], this paper gives a detailed discussion about these log files, their formats, their creation, access procedures, their uses, various algorithms used and the additional parameters that can be used in the log files which in turn gives way to an effective mining. It also provides the idea of creating an extended log file and learning the user behavior. In the literature [4], knowledge obtained from web usage mining does the task of finding the hidden important information about user behavior, and facilitate more effective browsing, to enhance web design, its page surfing pattern and other valuable information which is used for various purposes. In this paper, we provide detailed review of work done for different phases of web usage mining. For the Author [5], in this research paper gives analysis of web usage methods including preprocessing stages used in web mining.

3. KNOWLEDGE OF WEB MINING

Web Mining has defined the processes of data mining to automate fetching and evaluating information from web documents and services [2]. Therefore, data mining processes include documents, hyperlinks, and web access log files, used to extract the information from web data.

3.1 Web Mining Techniques

Web Mining is divided in the following three techniques:

- Web Content Mining
- Web Structure Mining
- Web Usage Mining

The content of web mining deals with the web data content to examine data collected by search engines. It is the text of web pages, the relevance of web contents to search is determined by scanning the images and graphs. In effect to use Web Content Mining, the content database is handled with a specific topic.

The structure of web mining is used to examine the information related to the structure of a website. It is linked by the information or direct links to establish the relationship between web pages. The main objective the structure of mining in web is the studying of information related to patterns of specific websites. If not, it is also

used to extract previously the relationships between web pages of the unknown users.

The usage of web mining deals with web information that is stored in a web server, to extract and analysis of user behavior, while they are interacting with websites. Web information provide paths to access web pages. This information is gathered automatically access web log files through the Web Server. These web log files mainly consist of textual logs that are collected web users access of web servers and might be represented in standard formats [4, 5].The usage of web mining is determined the user's usage patterns in web log files that are analyzed to give recommend the right products to the user. To provide an effective recommendation, user's click stream data (navigational data) provides information to identify the customers' preferences and interests which are the shopping patterns and behavior like details about the view, buying, and adding the products to their shopping carts.

3.2 Locations of Web Log Data

To discover the usage pattern of users, Web Usage Mining is taken the web access log data from three main server of locations:

(i) web servers
(ii) proxy servers
(iii) application servers

The server of web collect a lot of information in their web log files of the databases they use. These records usually contain the basic information in web logs are IP address of the remote host, date and time of the request, referrers, POST or GET method, etc. These information are usually represented in a standard format.

The Proxy servers accept the incoming requests from the clients and forward those requests to the destination servers. Therefore, if the web server receives the client's request through a proxy server. Entries to the log files will be the information of the proxy server not the original user. These web proxy servers maintain a separate log file for gathering the information of the user. It provides to improve the navigation speed through catching by giving internet service providers (IPS) to their customers' services [1].The application servers or web clients refer to extensions and helper applications that enhance the browsers to support special services from the sites. It collects the usage data on the application or client-side of web server using JavaScript and Java Applets, etc. In order to effectively manage a web server, the web clients' techniques avoid the problems of users' session identification and the problems caused by catching using clients' log files.

3.3 Features of Log Files

The main focus of this research paper is analyzed Nginx server access of web log using the usage mining of log file to discover the usage extent of the visited users on the websites. The basic features of the access log file of Nginx server as shown in Figure 1 are IP address, timestamp, request type, visiting path, status code, and user gent. IP address is assigned by the ISP to identify of the user who has visited the website. Timestamp means the time spent on each web page by the user while visiting the website. It is identified as a session. The request type is used the method for data transfer, such as GET or POST. To set the path of visiting, it taken by the user while visiting the website. This is done by using directly the unknown resources location or URL and clicking on a link or through a search engine. A status code is sent by the web server after processing the client request on the web server. The User agent is called the browser that sends the request from the user to the web server. It's just a string that describes the type and version of browser software being used.

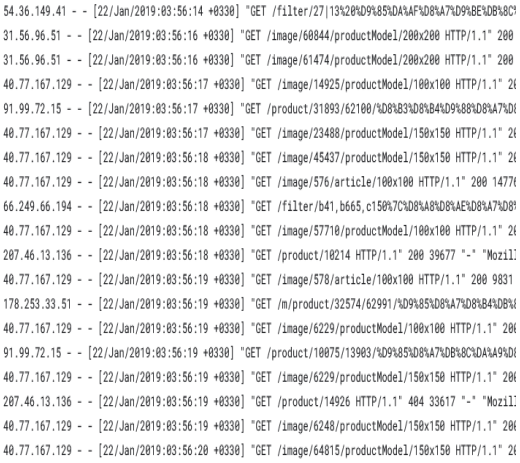


Figure1. Part of the access log file

4. PROPOSED SYSTEM ARCHITECTURE

The proposed system describes collecting and analyzing the entire clickstreams data in the log file for the usage extent of users or visitors who visit the websites and to discover the users or visitors which helps in better serving the websites. The path of the user's behavior gives an attention on Web usage mining to way the direction of web visitors' click- streams data based on web server access log file. This research paper use to implement the result of maximum potential addresses which is intelligent for visitors or users of behavior to show on the dashboard that is the entire web log dashboard shows where users are coming from and what scopes they usually look on their navigation behavior of the visitors. Log file get from the website of kaggle and downloaded and then detail as shown in data acquisition state. This data has about 4GB of data access log .In the data preparation state to collect the web log data and need to be cleaned of unrelated data. In this state has been used the many algorithms and many heuristic techniques to recommend and as shown details in data preparation. And then the data cleaning state to remove the unwanted data such as images, sound file and date/time section etc. and then more in data cleaning process. As part of the user identification process; a user may visit a website more than once. The server logs a number of times for each user. In the absence of authentication mechanisms, the most common way to identify among unique visitors is the use client-side cookies and show more detail in user identification process. Session Identification is the process of segmenting each user's activity history into sessions, each representing a single visit to the site. The purpose of the session identification heuristic is to reconstruct from clickstream data what a user has done during a visit to the site and to show more in session identification state. In the Figure2 shows the proposed system architecture to identify the usage extent of visitors who are coming from the website.

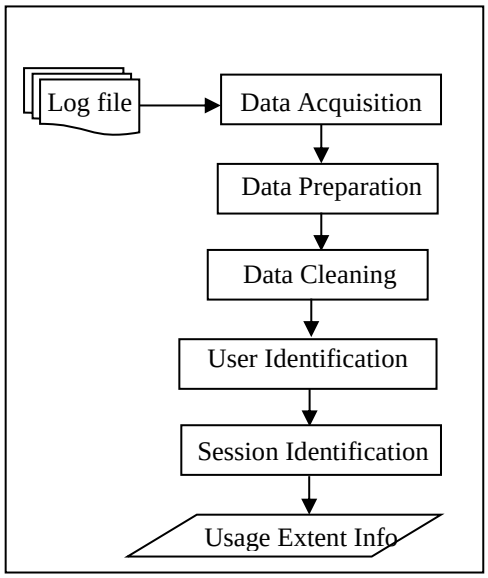


Figure2. Proposed System Architecture for Usage Extent Info of Visitors

(i) Data Acquisition

The data acquisition makes distinguish of the data format in log file. Therefore, the entire clickstreams data includes the attributes such as IP address, time stamp, request type (GET and POST), visiting path, status code, and user agent to analyze the webpages user most. The log file is downloaded from https:// www. kaggle. com/datasets/eliasdabbas/ web-server-access -logs. The data size is about 4G because it runs by row to work in one hundred thousand.

(ii) Data Preparation

In data preparation, it takes the data in log file as input to describe the most visited sites based on the behavior of visitors. For effective usage extent information, collected web log data need to be cleaned of irrelevant and inconsistent data. For this data processing operation, which includes various sub-operation such as the cleaning of data, the identification of the users and session. Many algorithms, heuristic techniques have been developed and recommended, it strong recommend for this that a reliable and integrated data source can be created and subsequently various data mining techniques can be effectively applied on them.

(iii) Data Cleaning

After that, the data cleaning makes the removal of unwanted data used to extract act the reliable data from the log file. The request types and the visiting paths are eliminated all irrelevant items such as file name suffixes (jpeg, GIF, JPEG, and JPG), CSS loading file, Bot, Mobile App, date string and drop time zone information for easier handling, sound files, references and it is not downloaded basis on of user's request and so it may be unnecessarily recorded in the log files due to spider navigation to improve the quality of analysis. The status codes in log data remove the values which are less than or greater than 200. The HTTP status code is also a concern for data cleaning [1]. Thus, data cleaning reduces the number of data that had recorded in log file so that the log file size is smaller than the original log file to perform the next processes quickly.

(iv) User Identification

To retrieve each user's access credentials, user identify provides requested service to users according to the IP addresses. We observe that the different users are distinguished on the user access behavior patterns. ID of the user will be assigned to different the address of IP. In this case, the same user of IP addresses are assigned on log data, the distinguished information and browser detail in these log data are analyzed among different web users. As part of the user identification process; a user may visit a website more than once. The server logs a number of times for each user.

(v) Session Identification

Session identification captures all references to the user's page in one record. It also defines the number of times a user visits a particular web page. The session identification is defined as sequence of web pages visited by a user. Only one session is created based on the new IP address to refer to a unique user. A session is defined as queries coming from the same IP address in less than a 30 minute timeframe. Thirty minutes is set to the time limit of maximum session. If a session has an average query rate of 1 second or less, it's probably not defined a human. Based on the query rate, check whether a human or crawler is likely to generate the session.

In each unique user, we retrieve the session in descending order based on the most recent timestamp of data. For each session, we fetch the different paths on categories of web site and analyze the usage extent of referrer and counts of visitor's behavior in Figure 3. Referrer means the source of website that is the visitor start to use this site and count means the number of visitors who visit on this website. In this figure show the count of the visitors who visited the web site among the website of www.goole.com is the most visited website and the count number of 6229. In the figure 4 to show the time stamp of visitors, referrer and the page of direction. In the figure 5 to implement the site navigation of visitors or users on product and then the next figure to implement the usage extend of visitors on product in Figure 6. In this picture, when extracting the visitors to the website, the number of visitors is extracted depending on the incoming IP addresses to search from the website of visitor's behavior.The system implements web usage mining to analyze the usage extent of web log file using Python language.

referrer	count
https://www.google.com/	6299
http://api.torob.com/	548
https://emalls.ir/Redirect.aspx?i=10201	209
android-app://com.google.android.googlequicksearchbox	160
http://www.google.com/	70
https://www.google.com	45
android-app://com.google.android.googlequicksearchbox/https/www.google.	33
android-app://org.telegram.messenger	29
https://emalls.ir/%D8%AE%D8%B1%DB%8C%D8%AF_%D9%81%D8%B1%D9	24
https://www.ask.ir/Shop/Product/679162	19
https://www.ask.ir/Shop/Product/484979	18
https://www.mobile.ir/directory/buyonline.aspx?id=276338	17
https://www.bing.com/	15
android-app://com.google.android.gm	15
https://www.ask.ir/Shop/Product/679238	14
http://www.khanesazan.com/materialsaprices/wall/29-gache-pish-sakhteh.l	13

Figure3. Implement the usage extent of Referrer and Counts of Visitors Behaviour

ts	referrer	page
2019-01-22T13:36:36	https://www.google.com/	/
2019-01-22T13:37:06	https://www.zanbil.ir/	/browse/accessories/%D8%B3%D8%A7%D8%B9%I
2019-01-22T13:37:15	https://www.zanbil.ir/	/browse/digital-supplies/%D9%84%D9%88%D8%A3
2019-01-22T13:37:16	https://www.zanbil.ir/	/browse/art-music/%D9%87%D9%86%D8%B1-%D
2019-01-22T13:37:26	https://www.zanbil.ir/browse/accessories/%D8%B	/browse/smartwatch/%D8%B3%D8%A7%D8%B9%I
2019-01-22T13:37:29	https://www.zanbil.ir/browse/accessories/%D8%B	/browse/pens-and-pencil/%D9%82%D9%84%D9%
2019-01-22T13:37:33	https://www.zanbil.ir/browse/accessories/%D8%B	/browse/jewelry/%D8%B2%D8%BC%D9%88%D8%I
2019-01-22T13:38:37	https://www.zanbil.ir/browse/pens-and-pencil/%	/browse/luxury-pens-set/%D8%B3%D8%AA-%D9%
2019-01-22T13:39:17	https://www.zanbil.ir/customer/profile	/order/track/9711020447
2019-01-22T13:39:42	https://www.zanbil.ir/browse/art-music/%D9%87%	/browse/musical-instruments/%D9%84%D9%88%D
2019-01-22T13:39:50	https://www.zanbil.ir/browse/jewelry/%D8%B2%D	/browse/men-jewelry/%D8%B2%D8%BC%D9%88%
2019-01-22T13:39:52	https://www.zanbil.ir/browse/jewelry/%D8%B2%D	/browse/women-jewelry/%D8%B2%D8%BC%D9%8

Figure4. Implement the usage extent of TS, Referrer and Page's Navigation of Visitor Behaviour

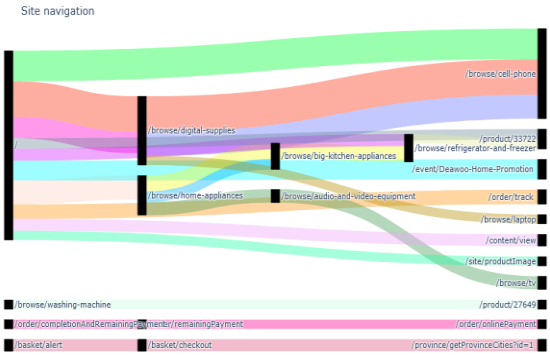


Figure5. Implement the site navigation of visitors on products

Path	visitors
/	143
/browse/digital-supplies	52
/browse/home-appliances	30
/browse/cell-phone	68
/browse/big-kitchen-appliances	20
/browse/audio-and-video-equipment	10
/browse/refrigerator-and-freezer	21

Figure6. Implement the usage extent of visitors on products

5. CONCLUSION

As the web based information, it is becoming difficult for web log file to extract the relevant information. The main focus of this paper is applies five phases such as data acquisition, data preparation, data cleaning, user identification, and session identification to extract and count of the visited web sites for visitors. In this paper, the web usage mining is used to discover the usage extent of multiusers and the popular web sites according to web log data are described on descending order of visitors to implement the recommended products. It is used to implement the usage extent of web page, time stamp, referrer and count of the popular web sites and to show the implementation of visitor's behavior and the number of users who visit the website which helps to better serving the web based application. In the future work, this preprocessing state will be done and the resulting data will be implemented as a model using methods.

References

[1] A. Pratap Singh, R. C. Jain, Dr., "A Survey on Different Phases of Web Usage Mining for Anomaly User Behavior Investigation", International Journal of Engineering and Computer Science, Vol. 3, Issue 3, pp. 70-75, 2014.

[2] D. Racha, "Web Usage Mining for extracting Users' Navigational Behavior", International Journal of Engineering and Computer Science, Vol. 3, Issue 5, pp. 5989-5995, 2014.

[3] L. K. Joshila Grace, V. Maheswari, D. Nagamalai, "Analysis of Web Logs and Web User in Web Mining", International Journal of Network Security & Its Applications (IJNSA), Vol. 3, No. 1, pp. 99-110, 2011.

[4] Neeru, R. Nandal, "A Systematic Review on Data Processing and Pattern Discovery of Web Usage Mining", International Journal of Advanced Research in Computer Science, Vol. 9, No. 2, pp. 305-308, 2018.

[5] T. Tabassum, S. Saxena, "A Survey on Web Usage Mining Techniques", International Journal of Engineering Research & Technology, Vol. 2, Issue 10, pp. 3824-3829, 2013.

COVID-19 MEDICAL CHATBOT FOR MYANMAR CITIZEN

Kyaw Myo Tun[1], Htin Kyaw[2], Aung Myo Thu[3], Zar Nay Lin[4]

[1,2,3,4] Department of Computer Science, Higher Education Centre (Pyin Oo Lwin)
[1]kyawmyotun53.ps@gmail.com, [2]mgnyeinaye@gmail.com,
[3]aungmyothu22@gmail.com, [4]naylin730@gmail.com

ABSTRACT

In today's world, the COVID-19 disease has become a global pandemic. At this time, the flow of high number of cases may become a problem because there are not enough doctors and nurses in the hospitals to handle them. As for patients, it is becoming difficult to consult information related to covid because the number of doctors/nurses in the hospital is not sufficient. Especially for our Myanmar citizens, there is a need for consultation in Myanmar Language. This paper proposes a solution using a supervised learning algorithm to build a Myanmar medical COVID-19 chatbot using Myanmar Language. Machine learning and text similarity models will help that the chatbot understand the questions asked by the patient and assist the chatbot respond. The chatbot maintains information about COVID-19 informs in the database to categorize the sentence keywords, create a question, and generate an answer. In this system, we use the combination of two feature extraction methods (BOW and Tf-idf) to get more accurate answers and to train and predict classification. Finally, we implemented program with Python GUI to simulate a medical chatbot system, providing automatic answers to frequently asked questions of patient for COVID-19 information.

KEYWORDS

Chatbot, Combination of Two Text Similarity Methods (BOW and TF-IDF), COVID-19, Machine Learning, Natural Language Processing.

1. INTRODUCTION

Chatbot is an intelligence system creating with a computer program. Chatbot has two types; text-base and voice-base chatbots. Nowadays, chatbots are being used a lot of fields such as business, education, transporting, marketing, and medicines etc. In the current world situation, COVID-19 epidemic is being come a global pandemic. Due to the outbreak of COVID-19, it is becoming a socio-economic impact. It is needed to keep the spread of virus in the upcoming days. In this time, lockdown staying at home was declared by the government. During lockdown, it is needed to obtain the consultation with the doctor for COVID-19. To solving these problems, it is being get the idea which is to create a medical chatbot using intelligence system that can give COVID-19 information as discussed with the doctor.

This research focus on Myanmar Citizens who are English illiterate especially in remote areas. So, this system is needed to understand Myanmar Language. Myanmar (Burmese) is a Sino-Tibetan language belonging to the Burmese Lolo group. The dataset is taken from World Human Organization (WHO) of magazines, journals and web-site etc. In preprocessing stage, the dataset is split into a training stage and a testing stage with word segmentation and stopword removal. Myanmar word segmentation is the most essential task in natural language processing. Stopword removal is removed unnecessary word getting from segmentation. For feature extraction, it is used techniques such as term frequency-inverse document frequency (TF-IDF) and bag-of-words (BOW). Proposed method is a feature technique by combining TF-IDF and BOW. Finally, three machine learning models, Cosine similarity, Correlation coefficient and K-nearest neighbors (KNN) are trained to test their performance on the

test data. And then, it is compared the results from three machine learning models with BOW, TF-IDF and proposed method. In this study, contributions are using the standard feature techniques TF_IDF and BOW, performance analysis of three machine learning models, Cosine similarity, Correlation coefficient, and KNN for the classification of COVID-19 data. Propose method is based on BOW and TF_IDF and evaluate its performance. Project contribution is built on Excel VBA.

This paper is composed of six sections. First section (1) is Introduction and the second section (2) is Related Work. The Methodology is in section (3) and Proposed System is in section (4). Implementation and results are displayed in section (5) and concludes the paper in section (6) respectively.

2. RELATED WORK

When patients are exposed to nCOV-19, Gopi Battineni [2] recommends the design of a sophisticated artificial intelligence (AI) chatbot for diagnostic evaluation and fast measures. Furthermore, by presenting a virtual assistant, the infection severity can be measured and registered doctors can be contacted if symptoms grow critical.
ChatBot is frequently used to check on one's health at any time. It is the same as going to a doctor and getting a prescription. Tamizharasi B., [9] presents about medical chatbot using the machine learning algorithm to predict the accuracy of the disease.

Naing Naing Khin, Khin Mar Soe [7] focused on developing Myanmar chatbot based on sequence-to-sequence model. This paper was built RNN, sequence to sequence learning, Attention Mechanism, Natural Language Processing (NLP) and question answering system. In this paper, the data collection coming from the admission team of University of Computer Studies, Yangon, Myanmar and from its Facebook page messenger was segmented with the Myanmar words segmenter of UCSY NLP Lab. This paper was shown a hybrid method which was Recurrent Neural Network based Encoder Decoder model via Attention mechanism. This paper was proposed the system getting a support for university education sector.

H. H. Y. Oo, M. T. Sue, Y. K. Thu et. al. [4] presented the rule-based Myanmar language Chatbot for travel and tourism domain. This paper was shown how a rule-based chatbot can be created artificially by using Regular Expression, Damerau-Levenshtein distance and Web scraping. Evaluate making two versions were the first version measuring String similarity with the whole dataset without separating categories and the next one which was with separating the dataset into several categories. The main purpose of this paper's proposed system was to develop a chatbot prototype of travel and tourism in Myanmar.

3. METHODOLOGY

3.1. Natural Language Processing

Natural Language Processing (NLP) which is a subfield of linguistics, computer science, and artificial intelligence is the talent of a computer program to understand human language as it is spoken and written, referred to as natural language [1]. It integrates statistical, machine learning, and deep learning models with computational linguistics and rule-based modelling of human language. With the use of these technologies, computers have become able to process human language in the form of text or voice data and fully "understand" what is being said or written, including the speaker's or writer's intentions and sentiment.

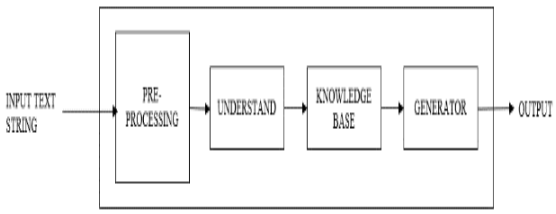


Figure 1. The process of Natural Language Processing(NLP)

NLP has been around for more than 50 years and has linguistics roots.

3.2. The Characteristic of Myanmar Language

The Myanmar script is an abugida in the Brahmic family used for writing Myanmar [3]. The characters are round in shape, because the traditional palm leaves used for writing on with a stylus would have been ripped by straight lines. It is written from left to right and no need to put white spaces between words. But the modern writing style contains spaces after each clause in order to enhance readability. Historically it was adopted from the Mon script, which further derived from Indian Brahmi script flourished in the Indian subcontinent between 5th Century BC and 3rd Century AD. Myanmar script has 33 consonants characters and 12 vowels characters.

3.3. Syllabification

Syllabification divide a word into its individual vowel sounds. The process of syllabification is the division of words into smaller parts which are known as syllables. A syllable is a voice or pronunciation unit that can be identified. Each voice is referred to as a syllable, which is also known as 'Wanna' in Myanmar. The fragmentation of each syllable unit is referred to as syllabification. The text of each syllable in a word, phrase, or sentence is independently split by syllabification [8]. Syllabification is also known as syllable break, syllable segmentation, and other terms.

The syllable break ‘wanna’ is commonly divided in two ways. Phonetical and orthographical breaks are the two types of breaks. With a table, here's an example of a phonetic and orthographic break:

Table 1. Syllabification

Myanmar Text	Phonetic Break	Orthographical Break
မင်္ဂလာပါ	မင်~ဂ~လာ~ပါ	မင်္~လာ~ပါ
လိမ္မော်သီး	လိမ်~မော်~သီး	လိမ္မော်~သီး

3.4. N-gram

A group of N words is referred to as an N-gram. N-grams in a document are recurrent groups of letters, numbers, or other tokens. The term "N-gram" refers to a group of N words. Using the N-gram model, the most likely term in a sentence is determined [6]. N-grams are applied in this system to extract the pertinent keywords from the database, compress the text, or decrease the amount of data in the document.

Table 2. N-gram

Uni-gram	Bi-gram	Tri-gram	N-gram (4)
ကို	ကိုဗစ်	ကိုဗစ်၁၉	ကိုဗစ်၁၉ရောဂါ
ဗစ်	ဗစ်၁၉	ဗစ်၁၉ရောဂါ	ဗစ်၁၉ရောဂါကို
၁၉	၁၉ရောဂါ	၁၉ရောဂါကို	၁၉ရောဂါကိုဆို
ရောဂါ	ရောဂါကို	ရောဂါကိုဆို	ရောဂါကိုဆိုတာ
ဂါ	ဂါကိုဆို	ဂါကိုဆိုတာ	ဂါကိုဆိုတာဘာ
ကို	ကိုဆိုတာ	ကိုဆိုတာဘာ	ကိုဆိုတာဘာလဲ
ဆို	ဆိုတာဘာ	ဆိုတာဘာလဲ	
တာ	တာဘာလဲ		
ဘာ	ဘာလဲ		
လဲ			

3.5. Feature Extraction

A huge set of raw data is divided into smaller groups for processing using the dimensionality reduction technique known as feature extraction. These big data sets require a lot of computer resources to process due to the enormous number of variables. In order to handle less data while precisely and completely defining the original data set, feature extraction refers to techniques for choosing and/or combining variables into features. It is possible to extract features manually or automatically. There are some methods of feature extraction;

(a) Word2Vec
(b) Bag-of-word (BOW)
(c) Term frequency-inverse document frequency (TF-IDF)

3.5.1. Bag-of-word

Bag of words model is the technique of pre-processing the text by converting it into a number/vector format, which finds the

frequency of a word in a given sentence. It is the method of extracting features from the given text and use these features for model training. Bag of Word is a process of converting words into numbers by generating vector embedding's from the tokens generated above. This is given as an input to the machine learning model for understanding the written text. One hot encodes the output or targets is process of converting words into numbers by generating vector embedding's from the tokens generated above.

3.5.2. Term Frequency-Inverse Document Frequency

Term frequency-inverse document frequency (TF-IDF) is a method for determining which words in a corpus are likely to be useful based on their document frequency. TF-IDF is a method to measure the main of a word's (or term's) connection to a document. Frequency is used to check the number of times a sentence occurs.

$$tf = tf_i \quad (3.1)$$

Term frequency (tf) is calculated by counting the number of conditions of the term ti in the document d_i with tf is the term frequency, and tf_i the number of conditions of term ti in the document d_i.

$$tf = logN/df \quad (3.2)$$

$idfi$ is inverse document frequency, N is the number of documents retrieved by the system, and $idfi$ is the number of documents in the collection where term ti appears in it.

$$W_i = tf_i * logN/df \quad (3.3)$$

With Wi is the document weight, N is the number of documents retrieved by the system, tf_i is the number of occurrences of the term ti in the document d_i, and df_i is the number of documents in the collection where term t_i appears in it. Document weight (W_i) is calculated for obtaining a weight resulting from the multiplication or a combination of term frequency (tf_i) and Inverse Document Frequency (df_i) [5].

2.7. Text Similarity

Similarity is the distance between two vectors where the vector dimensions represent the features of two objects. Finding the text's degree of proximity is really the main goal. In simple terms, similarity is the measure of how different or alike two data objects are. If the distance is small, the objects are said to have a high degree of similarity and vice versa. Generally, it is measured in the range 0 to 1. This score in the range of [0, 1] is called the similarity score.

Text similarity measurement is the basis of natural language processing tasks, which play an important role in information retrieval, automatic question answering, machine translation, dialogue systems, and document matching.

2.7.1. Cosine Similarity

Cosine similarity is a measure of how similar vectors in an inner product space. The cosine value of 0 is one and less than one for any other angle; the cosine's lowest value is -1. The cosine of a vector's angle indicates whether or not two vectors are pointing in the same direction. In text/document matching, cosine similarity is commonly utilized. When two documents A and B are compared, the cosine similarity of their vector representations is computed, and the cosine of the angle between vectors is measured. The resultant similarity ranges from -1, which indicates complete opposite, to 1, which indicates complete similarity, with 0 suggesting independence and in-between values indicating moderate similarity or dissimilarity. The cosine similarity formula is presented by equation below:

$$Similarity = \cos(\theta) = \frac{A.B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (3.4)$$

Where:

A_i= components vector of A

B_i = components vector of B

2.7.2. Correlation Coefficient

A correlation coefficient is a statistical method that measures two-variable relationship for how strong relationship between two variables. A correlation is a single number that determines the degree of interaction between two variables. The testing of correlation coefficient is best used for variables that express a linear relation with each other. The correlation coefficient is capable of taking any value from -1 to 1. A computed number less than -1 or greater than 1 means that there was an error in the measurement of correlation coefficient. The correlation has three kinds that are positive correlation, negative correlation and no correlation. In positive correlation, the variables tend to change in the same direction. In negative correlation, the variables tend to move in opposite directions (one variable is increase and another variable is decrease) and the variables do not have a connection with one another in no correlation.

$$r = \frac{N\sum xy - \sum x \sum y}{\sqrt{(N\sum x^2 - (\sum x)^2)(N\sum y^2 - (\sum y)^2)}} \quad (3.5)$$

Where:

N is the number of pairs of scores,

$\sum xy$ = the sum of the products of paired scores,

$\sum x$ = the sum of x scores,

$\sum y$ = the sum of y scores,

$\sum x^2$ = the sum of squared x scores,

$\sum y^2$ = the sum of squared y scores.

2.7.3. K-Nearest Neighbours (KNN)

The KNN algorithm is a supervised Machine Learning technique that may be used to solve both classification and prediction regression issues. However, in industry, it is mostly utilized to solve predictive classification difficulties. A class member-ship is the result of K-NN classification. An item is classified by a majority vote of its neighbors, who assign it to the most common class of its k nearest neighbors (k is a positive integer, typically small). The result of the K-NN regression is the value of the object's property. This value is the average of the k closest neighbors' values.

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2} \quad (3.6)$$

4. PROPOSED SYSTEM

In this proposed system design, it consists of two phases; training phase and testing phase. Each phase has pre-processing step in which word segmentation and stopword removing. Word segmentation is to segment the collected data and stopword removing is to remove unnecessary word in sentence. Final word kept in the dictionary as training corpus dataset. And then, these words extracted with feature extraction to convert words into numbers. This model is trained on Cosine similarity method. Finally, the obtained training and testing data will be tested to get the output answer. The proposed system design for the COVID-19 medical chatbot application is shown in Fig. 2.

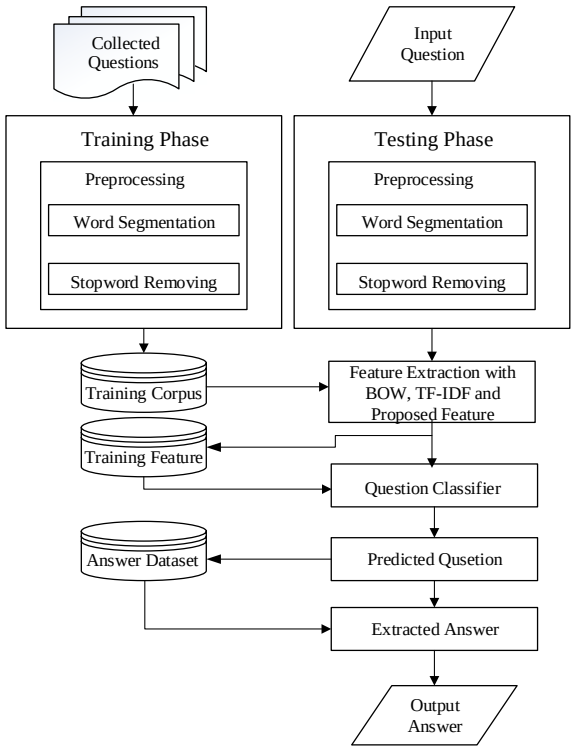


Figure 2. Proposed system Design of Covid-19 question and answering medical chatbot system

4.1. Data Collection

In Natural Language Processing, data collection is the most important component and saves time process for developing a system, especially for low-resource languages. Most NLP models require large amounts of training data and linguistic processing. The more training data is kept, the more correct information is obtained. Since Myanmar is a low-resource language, creating training data presents a problem of data scarcity. We obtained a corpus of 1000 question-answer pairs to train the conversation model (with a total of roughly 5000 phrases). The data is based on medical COVID-19 data, as well as information from its Facebook page messenger, website, COVID-19 center, healthcare news, and medical experts.

4.2. Data Pre-processing

Data pre-processing is the fundamental task of Natural Language Processing. In per-processing state consist of two states; word segmentation and Stopword removing. Firstly, the collected data are needed to segment with syllabification method that is a method of word segmentation. In Myanmar Language, it is not be used white space for word boundaries. So, Myanmar word segmentation is being become the main task. In this system, a dictionary was created for word segmentation. N-gram base method is used to get more accurate segmented word. When the exact syntax or spelling of the target object is unclear, N-gram search is used to find it. The task of Stopword removing is to remove unimportant word in the sentence. It is the vital step in the text pre-processing of Natural Language Processing (NLP). Stopwords in English include words like "is," "are," "in," "for," and "that." As can be seen, those terms have little or no meaning outside from grammatical structure. In Myanmar Language, stopwords include words like "နဲ့, ဤ, ထို, ဟို, ယခု, ၎င်းတို့, သည့်, က, မ , ၌, ၍", etc. These words are removed. Normalization is the process of converting a text into a canonical form, as well as mapping similar words: for example, such as "ကိုဗစ်-၁၉ရောဂါဆိုတာကဘာလဲ", "ကိုဗစ်-၁၉ရောဂါဆိုတာ ဘာလဲ", "ကိုဗစ်-၁၉ရောဂါဆိုသည့် မ ာ အဘယ္ယနည္း", "What is COVID-19?" in English to "ကိုဗစ်-၁၉ရောဂါကဘာလဲ" in Myanmar Language Stopwords are removed from the corpus, which reduces its size and improves the efficiency and accuracy of NLP tasks.

In this proposed system, training phase and testing phase involve. Before training the conversation model, the data is pre-processed. After that, text similarity method is used to train the model.

4.3. Vector Embedding's

Bag of Word is a process of converting words into numbers by generating vector embedding's from the tokens generated above. Term frequency-inverse document frequency (TF-IDF) is a technique for finding which words in a corpus are most likely to be helpful based on their document frequency. One of the best techniques for searching and labelling documents is the TF-IDF representation. However, there is no compelling argument to favour TF-IDF over other experience-based problem-solving methods. Therefore, in this paper, a better result should be found by combining two techniques, BOW and TF-IDF. A higher combination of the two methods indicates a stronger connection between the word and the document in which it appears.

4.4. Question Text Classification

This system enables computers to converse with people by utilizing machine learning and natural language processing (NLP). We created the COVID-19 disease question classification strategy in the closed data domain using the cosine similarity supervised learning technique, and we developed a Q&A chatbot application using this machine learning model in this study. By allowing patients to publicly ask

inquiries about their health via text queries, the suggested approach improves how well hospitals and medical institutions serve their patients. We must increase the number of tagged questions per Intent in the training dataset in order to maximize the performance of the aforementioned model. Future work on the Semi-Supervised Learning model will focus on enabling the bot to develop its own learning capabilities depending on newly submitted questions.

5. EXPERIMENTAL RESULT

Different chatbots with ruled-based, machine learning, and deep neural network architectures are based on various languages. Our contribution is the design and implementation of a Myanmar COVID-19 medical chatbot system for questions and answers using machine learning. We installed the Cosine similarity, Correlation Coefficient, and KNN algorithms on Python using the scikit-learn package to assess the effectiveness of the suggested approach in developing a Myanmar COVID-19 chatbot system.

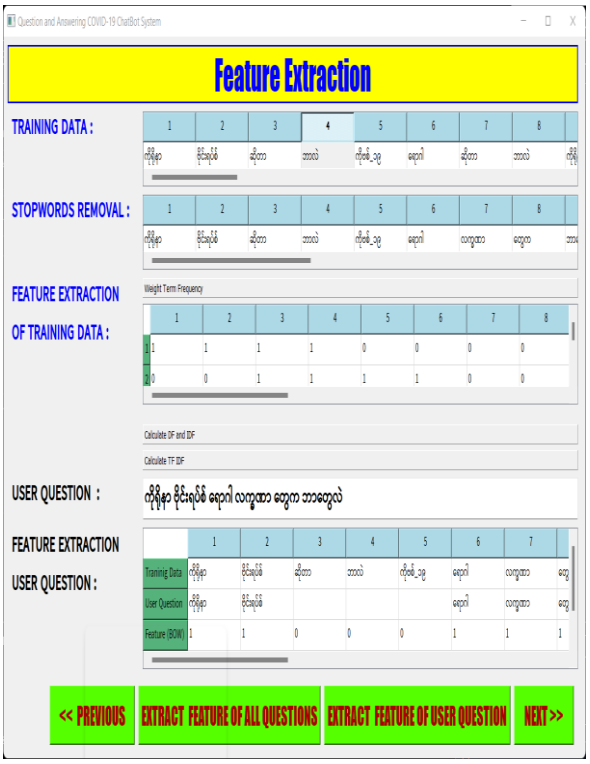


Figure 3. Feature Extraction using TF-IDF

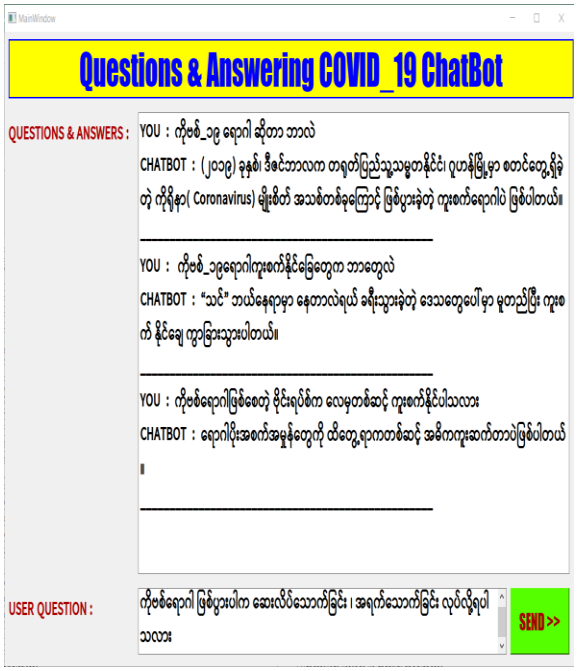


Figure 4. Experimental result of the proposed system

To evaluate the model for the similarity classification problem, we use the following formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

After comparing the three methods with Term Frequency-Inverse Document Frequency (TF-IDF) given in the below table, the accuracy of the Cosine Similarity is greater than Correlation Coefficient and KNN methods which is around 92%.

Algorithm Comparison (TF-IDF)				
Sr.No	Algorithm	User Questions	Accurate Answers	Accuracy
1.	Cosine Similarity	250	230	0.92
2.	Correlation Coefficient	250	213	0.852
3.	KNN	250	185	0.74

And then, when comparing the three methods comparing with Bag-of-Word (BOW) as shown in the below table, the accuracy of the Cosine Similarity is greater than Correlation Coefficient and KNN methods which is near about 90%.

Algorithm Comparison (BOW)				
Sr. No	Algorithm	User Questions	Accurate Answers	Accuracy
1.	Cosine Similarity	250	226	0.904
2.	Correlation Coefficient	250	207	0.83
3.	KNN	250	179	0.72

Finally, when comparing the three methods with our propose method, the combination of two methods (BOW and TF-IDF), as shown in the table below, the accuracy of Cosine Similarity is higher than that of Correlation Coefficient and KNN methods, which is near about 94%.

Algorithm Comparison (Combination of BOW and TF-IDF)				
Sr. No	Algorithm	User Questions	Accurate Answers	Accuracy
1.	Cosine Similarity	250	235	0.94
2.	Correlation Coefficient	250	224	0.896
3.	KNN	250	210	0.84

Therefore, in three feature extraction methods, our propose method is more accuracy other than BOW and TF-IDF methods.

6. Conclusions

By using natural language processing (NLP) and machine learning, this system allows computer to converse between human to computer. In this article, we presented the COVID-19 disease question text classifier method, based on the Cosine Similarity, Correlation Coefficient, and KNN model to build a Q&A chatbot application. According to compare result, the combination of two methods (BOW and TF-IDF) is better than TF-IDF and BOW. By looking at the final result, Cosine similarity is more accurate than Correlation Coefficient and KNN methods. The proposed method helps medical institutes and hospitals support patients by allowing them to freely raise medical-related questions via text queries. To improve the efficiency of the above model, we need to add the number of labelled questions per Intent in the training dataset. In the future, we will continue to study the subtraction of two methods (BOW and TF-IDF) with the Deep Learning model, so that the bot learns on its own based on the new questions entered by the user during the operation of the chatbot system.

Acknowledgements

I would like to express my sincere thanks to Dr. Zar Nay Lin, Dr. Aung Myo Thu, Dr. Htin Kyaw and my honourable teachers, Department of Computer Science, Higher Education Center, Pyin Oo Lwin, who have brought me into a new territory of knowledge and for providing the technical guidelines and suggestions writing this research. Lastly, I'm very grateful to all compassionate individuals who helped directly and indirectly during this research.

References

[1] Gobinda G. Chowdhury, "Natural language processing," Annual Review of Information Science and Technology, Volume 37, Issue 1, Pages 51-89, 2003.

[2] Gopi Battineni, AI Chatbot Design during an Epidemic like the Novel Coronavirus, 9 May 2020, MDPI.

[3] Htay, H. H, Murthy, K. N, "Myanmar Word Segmentation Using Syllable Level Longest Matching," proceeding of the 6th Workshop on Asia Language Resources, pp. 41-48, 2008.

[4] Htet Htet Yadanar Oo, May Thae Sue, Ye Kyaw Thu, Hin Aye Thant, Nang Aye Aye Htwe, "Rule-based Myanmar Language Chatbot for Travel and Tourism Domain", ICSE, December, 2019.

[5] Kavitha B. R., Dr. Chethana R. Murthy, "Chatbot for healthcare system using Artificial Intelligence", IJARIIT, Volume 5, Issue 3, 2019.

[6] M. F. Lynch, P. Willett and F. Ekmekcioguly, "Stemming and N-gram Matching for Term Conflation in Turkish Texts", Information Research News, Volume1, No.1, pp.2-6, 1996.

[7] Naing Naing Khin, Khin Mar Soe "Question Answering based University Chatbot using Sequence to Sequence Model", Conference of the Oriental COCOSDA, Nov 2020.

[8] Pa, P. W, Ye, K. T, Finch, A and Sumita, E, "Word Boundary Identification for Myanmar Text Using Conditional Random Fields", International Conference on Genetic …, September 2015.

[9] Tamizharasi B., Building a Medical Chatbot using Support Vector Machine Learning Algorithm, NCSET, 2020.

Authors

Biography First

A. Author

Kyaw Myo Tun, a graduate of B.C.Sc, was commissioned from the Defence Services Academy in 2010. He earned his MC.Sc at Higher Education Center (HEC), D.S.A, in 2017. Now, he is serving as an assistant lecturer at No. (6) Advance Training Military School.

Biography second

B. Author

Htin Kyaw, a graduate of B.C.Sc, was commissioned from the Defence Services Academy in 2005. He earned his M.I.C.Sc and Ph.D. at Moscow Institute of Electronic Technology (M.I.E.T) in 2015. Now, he is serving as an assistant lecturer at the Department of Computer Science in Defence Services Academy (DSA).

Biography third

C. Author

Aung Myo Thu, was commissioned and obtain his bachelor degree specializing in Computer Science from Defence Services Academy in 2004. He earned his Master of Engineering from Moscow Aviation Institute (MAI), (Russian Federation) in 2010. He also obtained his Ph.D. (Degree Doctor of Philosophy) from Moscow Aviation Institute (MAI) (Russian Federation) in 2014. Now he is serving as an associated professor of the Department of Computer Science.

Biography fourth

D. Author

Zar Nay Lin, was commissioned and obtain his bachelor degree specializing in Computer Science from Defence Services Academy in 2001. He earned his Master of Information and Computer Science from National Research University of Electronic Technology (MIET), (Russian Federation) in 2005. He also obtained his Ph.D. (Degree Doctor of Philosophy) from National Research University of Electronic Technology (MIET) (Russian Federation) in 2009. Now he is serving as an associated professor of the Department of Computer Science.

DEEP LEARNING MODELS FOR MYANMAR ROAD SIGN RECOGNITION

Tin Zar Htun[1] and Aung Nway Oo[2]

[1,2]Faculty of Computer Science, University of Information Technology, Yangon, Myanmar

[1]tinzarhtun@uit.edu.mm, [2]aungnwayoo@uit.edu.mm

ABSTRACT

For Intelligent Vehicles system, automatic road sign recognition and identification is extremely important. Due to drivers' irresponsibility, a significant number of traffic fatalities and collisions are reported each and every year. The system is commonly designed for self - driving vehicles and drivers in order to alert them to traffic conditions and minimize the rate of accidents across the nation. Deep learning, a rapidly growing area of artificial intelligence, succeeds at a variety of tasks, including speech authentication, facial identification, and natural language processing (NLP). This study evaluates the performance of deep convolutional neural networks like AlexNet and GoogLeNet. In the experiments, these pre-trained CNN models are trained and tested using the Myanmar Road Sign dataset. An evaluation of the two pre-trained models for identifying road signs found that the GoogLeNet model outperforms the AlexNet model. The best recognition accuracy comes from GoogLeNet, which is 98.32% and AlexNet achieves recognition accuracy with a 97.95% rate.

KEYWORDS

Artificial Intelligence, Intelligent Vehicles System, Deep Convolutional Neural Network (DCNN), AlexNet, GoogLeNet

1. INTRODUCTION

People around the world are hurt in car accidents each year, most of which can be avoided by paying attention while driving [1]. One of the most important ways to significantly decrease the probability of these incidents is the use of driverless cars. The capacity to achieve nearly complete awareness of the surrounding operating area is one of the fundamental requirements for the development of autonomous systems. The primary perceptual issue in fulfilling these criteria is the identification of road signs. Traditional machine learning methods cannot handle all perception in autonomous driving, which makes it difficult to identify road signs.

Because AI significantly enhances the performance of conventional machine learning approaches, it has become essential to the development of autonomous cars. Convolutional neural networks (CNN) are the foundation of Deep Neural Networks (DNN), which dominate AI and are crucial to the development of driverless cars. Input, hidden, and output layers are a few of the numerous layers present in CNN topologies. Using a Myanmar Road Sign dataset, various innovative pretrained CNN models are utilized to provide improved accuracy. There is currently no recognition system in Myanmar that has been compared using pretrained CNN models.

So, these are the contributions that this paper produced.

- A larger collection of road sign categories than in other studies is created in Myanmar, where there are 25 classes that are distinctly different from one another.
- On our created dataset, proposed pretrained CNN models' classification accuracy is evaluated.

In this study, we propose a deep pre-trained CNN-based road sign detection system. For the proposed application, deep pre-trained models AlexNet and GoogLeNet have been utilized. This is how we arrange our paper: In Section 2, you'll find a summary of the most recent studies on approaches for identifying and detecting road signs. The proposed road sign-based pre-trained models are then described in section 3. The results of the experimental performance analysis are presented in Section 4. In part 5, the paper's conclusions and future studies are discussed.

2. Related Work

The associated works relating to road sign recognition systems are highlighted in this section. Numerous studies have been conducted on the recognition of road signs. Due to the inconsistent types and designs of road signs in different nations, there is still no ideal answer in this area. A system for real-time road sign recognition was proposed by the authors [2]. Images of Beijing taken in various weather situations were used to construct a new dataset. To detect color and shape, the authors employed color segmentation and the Hough transform; to identify the road signs, they used a template matching method. The authors claimed that the road sign identification system had an accuracy of 80% and that the method based on template matching is effective for recognizing road signs.

However, the recognition accuracy is not good enough. Deep learning methods can outperform feature-based computer vision methods due to technological advancements. The authors used pre-trained CNN models to study on the Malaysia traffic sign recognition system [3]. According to experimental outcomes, the accuracy of most pretrained models is 90%, while the DenseNet169 model has a 98.33% accuracy rate. On the German Traffic Sign Recognition Benchmark dataset (GTSRB), the researchers evaluated VGG16, VGG19, AlexNet, and ResNet50 in order to increase accuracy. The researchers concluded that the AlexNet model outperforms all other implemented models among these pretrained models in terms of accuracy [4]. In 2014, the GoogleLeNet architecture, including the use of the Inception Block, has largely been successful in resolving the problems that large networks faced. According to the authors, the F-RCNN Inception v2 model used a fully convolutional network with transfer learning to produce accurate, dependable, and quick results even in challenging real-world road circumstances, with a recognition accuracy of 96% [5]. So, we proposed AlexNet and GoogLeNet for a road sign detection and identification system based on prior research.

3. Myanmar Road Sign Recognition System

Road sign recognition often involves the use of preprocessing, feature extraction, and classification. This section outlines the system architecture we've developed to use AlexNet and GoogLeNet to recognize road signs. A pretrained CNN model (AlexNet and GoogLeNet) is mainly used for feature extraction from images and then using these extracted features by a fully connected layer of CNNs to identify road signs.

3.1. Preprocessing

Preprocessing is an important step in the process of recognizing road signs. This initial step's goal is to get the image ready for more processing. An RGB-colored image serves as the input image. Images are resized as their respectively network architecture's input image size. The training data set will then be expanded using the augmentation method. For data augmentation, rotations, translations, and shearing were applied to training images. The number of training images increases as a result. Rotation creates a randomized rotation transformation that rotates the image by an angle selected. Translation creates a translation transformation that shifts the input image horizontally and vertically by a distance selected. Shearing creates a horizontal shear transformation with the shear angle selected. Figure 1 shows the sample output after augmentation.

Figure 1: Sample Image after Augmentation

3.2. Network Architecture

The four transfer learning approach types are relational knowledge transfer, feature representation transfer, parameter transfer, and instance transfer [6]. The ideal approach for this study is feature representation transfer learning. A technique called feature representation transfer learning employs features to represent the image information. So, we could utilize pre-trained models like AlexNet and GoogLeNet to extract the essential features from a training image dataset, and then we could use those features to represent the images.

GoogLeNet: Developed by Google researchers, the Inception Network is a Deep Convolutional Neural Network with 22 layers [7]. GoogLeNet is a variation of this network. The GoogLeNet architecture addressed computer vision issues such image categorization and object recognition in the

ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). Most of the issues that huge networks had were resolved by the GoogleLeNet architecture, primarily through the use of the Inception Block (See in Figure 2). The Inception block is a neural network design that uses dimensional reduction to lower the computational cost of training a large network while leveraging feature identification at various sizes through convolutions with various filters. Nine inception blocks are included in the total of 22 layers that make up the GoogLeNet architecture (27 layers when pooling layers are included). The standard GoogLeNet architecture is displayed in Table 1.

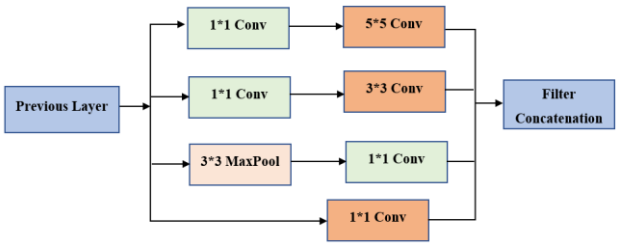


Figure 2: Inception Block

Table 1. GoogLeNet Architecture

Type	Patch size/Stride	Output size
Conv	7*7/2	112*112*64
Max pool	3*3/2	56*56*64
Conv	3*3/1	56*56*192
Max pool	3*3/2	28*28*192
Inception (3a)		28*28*256
Inception (3b)		28*28*480
Max pool	3*3/2	14*14*480
Inception (4a)		14*14*512
Inception (4b)		14*14*512
Inception (4c)		14*14*512
Inception (4d)		14*4*528
Inception (4e)		14*14*832
Max pool	3*3/2	7*7*832
Inception (5a)		7*7*832
Inception (5b)		7*7*1024
Avg pool	7*7/1	1*1*1024
Dropout (40%)		1*1*1024
Linear		1*1*1000
Softmax		1*1*1000

The GoogleLeNet architecture's input layer receives an image with a resolution of 224 x 224. The first convolutional layer in Table 1 uses a filter (patch) size of 7x7, which is different from other patch sizes used in the network. This layer's primary objective is to immediately reduce the input image while preserving spatial detail by employing large filter sizes. More feature maps are created, but the input image's size (height and width) is reduced by a factor of four at the second convolution layer and by a factor of eight before it reaches the first inception block. The second convolution layer, which has a depth of two, leverages the 1x1 conv block, which results in dimensionality reduction.

The GoogLeNet design is comprised of nine inception blocks, as shown in Figure 3. Two max-pooling layers connect a few inception blocks. The input height and width are decreased to 1x1 by the average pooling layer, which calculates the mean of all the feature maps created by the previous inception block. There is a dropout layer (40%) used before the linear layer. The dropout layer is a regularization technique used during training to prevent the network from being overfit. The linear layer has 1000 hidden units, which are equivalent to the 1000 classes identified in the ImageNet dataset. The final layer, called Softmax, uses the Softmax activation function to ascertain the probability distribution of a set of integers contained within an input vector.

In this study, the final layer of the GoogLeNet model is removed in order to acquire the feature extraction module, which creates a set of features from each input image. The feature vector has a length of 1024. In order to recognize Myanmar road signs, the resulting feature vectors are passed into a new fully connected layer.

Max pool
Inception (3a)
Inception (3b)
Max pool
Inception (4a)
Inception (4b)
Inception (4c)
Inception (4d)
Inception (4e)
Max pool
Inception (5a)
Inception (5b)

Figure 3: Partition of GoogLeNet

AlexNet: Alex Krizhevsky et al. produced the first pre-trained prototype in 2012 to demonstrate substantial outcomes in the computer vision domain. A huge, deep convolutional neural network was trained to classify the 1.2 million high-resolution images in the ImageNet into the 1000 different categories [8]. Compared to the previous

state-of-the-art, its top-1 and top-5 error rates are much lower at 37.5% and 17.0%, respectively. It achieved cutting-edge recognition accuracy when compared to every other machine learning and computer vision approach currently in use. In the original network, the There are four blocks that make up AlexNet. A single Convolutional Layer (CL) with the ReLU function, followed by a pooling layer with unique parameters for each block, makes up both the first and second blocks. Block 3 is composed of three CLs, followed by ReLU functions and additional parameters, and a pooling layer (PL). Following these blocks is a flatten layer, followed by an FC block comprised of three FC layers.

In this system, the final two fully connected layers of the original networks were discarded. To the original network, another layer was added—a fully connected layer with Softmax activation. The stochastic gradient descent optimizer (SGD) is employed with a learning rate of e-4, and the batch normalization is set to 32. The input image is 227x227 in size. Figure 4 (left hand side) shows the original architecture of AlexNet and Figure 4 (right hand side) displays the detailed architecture of AlexNet used in this system.

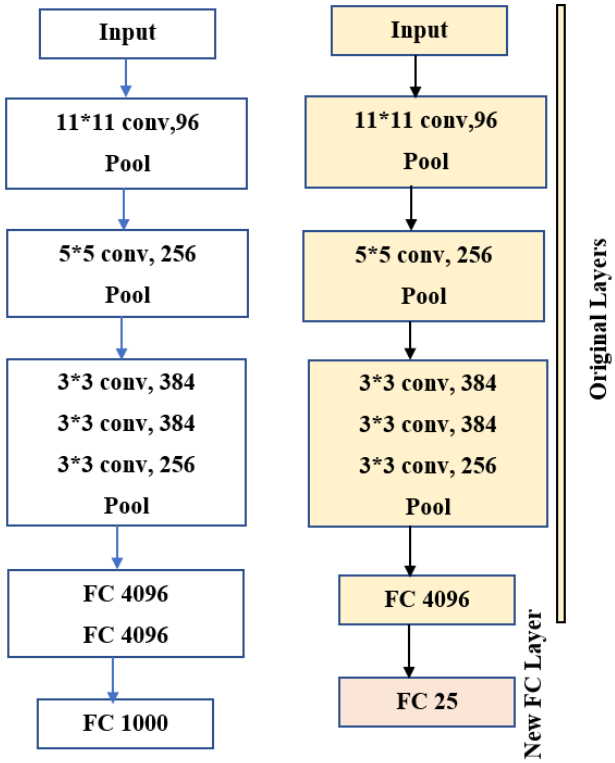


Figure 4. AlexNet Architecture

3.3. Proposed System Architecture

As depicted in Figure 5, this paper proposes using a Deep pre-trained model, AlexNet and GoogLeNet rather than a straightforward CNN model to recognize road signs. To evaluate the system, Myanmar Road Sign Dataset has been used. For training, 80% of the dataset is used, and for testing, 20%. The proposed system's design is depicted in Figure 5.

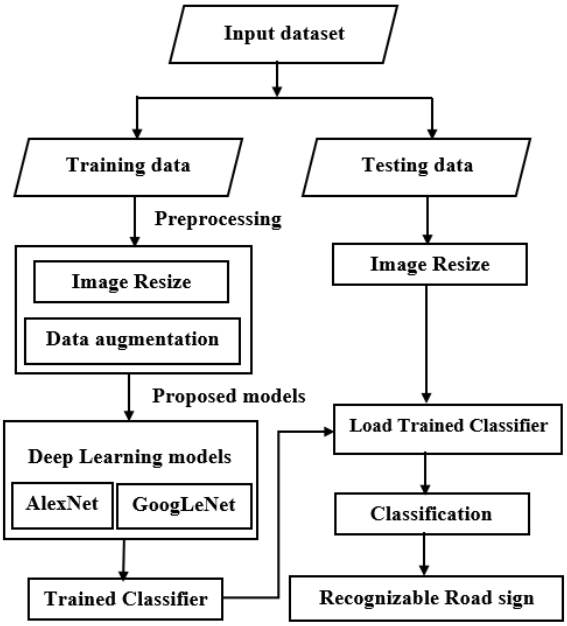


Figure 5. Proposed system design

4. Experiments and Results

The Myanmar Road Sign Dataset was used in this section to analyze how well the proposed system performed. The proposed methods are implemented using the MATLAB R2022b programming language. An Intel ® i5-8250U processor running at 1.6GHz, 8GB of RAM, a 256GB SSD, and an NVIDIA GeForce MX130 GPU use to implement the system. The proposed system's execution time is measured using MATLAB's timeit function.

4.1. Datasets

Each class in the Myanmar Road Sign Dataset, which consists of 25 classes, has between 11 and 160 photos. Red, blue, green, and yellow are the colors used on road signs in Myanmar. They commonly have circular, rectangular, diamond, or triangular patterns as well. The background color of the signs used for this study are red, yellow, and blue, and they have circular, rectangular, and diamond-shaped shapes. There are a total of 536

images for testing and 2142 images for training before augmentation. The training data would comprise 6426 images after augmentation. Images were captured under a range of lighting settings and viewpoints. The photos in the collection have pixel sizes ranging from 32 by 32 to 225 by 225. The sample images from the Myanmar Road Sign Dataset are shown in Figure 6.

Figure 6. Example of road sign dataset

4.2. Evaluation Metrics

The proposed methods are analyzed using the confusion matrix. Performance, recall, accuracy, and precision of the proposed approaches were evaluated. The experiment's results are computed using the following formulas. Tables 2 and 3 present the evaluation results.

Accuracy: the percentage of all predictions that were accurate.

$$\text{accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

Precision: the percentage of positive cases that were appropriately identified

$$\text{precision} = \frac{TP}{TP+FP} \quad (2)$$

Recall: the percentage of positive cases that are really and accurately identified

$$\text{recall} = \frac{TP}{TP+FN} \quad (3)$$

Where TP and TN, respectively, stand for the number of True Positives and True Negatives. Then, FP and FN are abbreviations for the number of false positives and false negatives, respectively.

4.3. Results

The results of the proposed methods are presented in Tables 2 and 3. Results from experiments show that GoogLeNet achieves the highest accuracy (See Table 2). Table 2 describes the recall, and accuracy for each classifier and Table 3 describes the performance and precision for each classifier. The execution time of AlexNet and GoogLeNet are 23m 2s and 45m 43s. While GoogLeNet takes a long time to execute as it contains more deeper layers than AlexNet, AlexNet can process in a short period of time (See Table 3).

Table 2. Recall and Accuracy comparison

Model	Recall	Accuracy
AlexNet	91.27	97.95
GoogLeNet	95.60	98.32

Table 3. Performance and Precision comparison

Model	Performance	Precision
AlexNet	23m 2s	90.19
GoogLeNet	45m 43s	93.89

Figure 7. Prediction results of proposed system

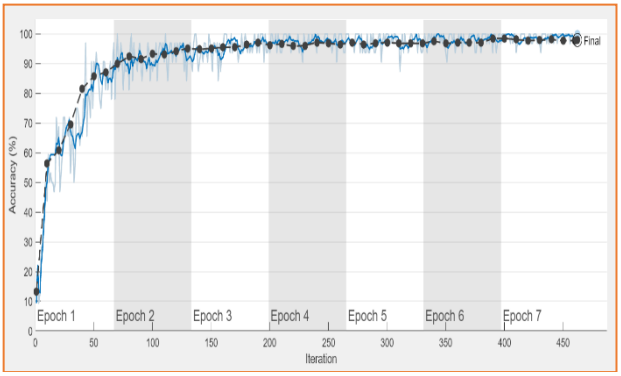


Figure 8. Accuracy of AlexNet. The number of epochs is represented in the row, and the column is the accuracy score.

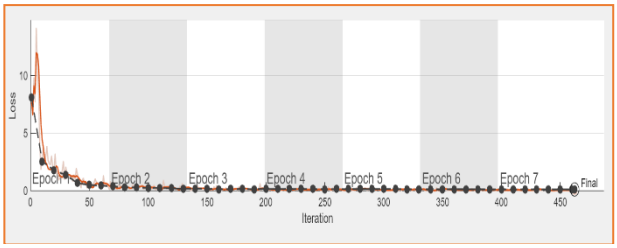


Figure 9. Loss of AlexNet. The number of epochs is represented in the row, and the column is the loss score.

The sample output of the testing data is shown in Figure 7. The accuracy and loss of the AlexNet on the Myanmar Road Sign training dataset are depicted in Figures 8 and 9, respectively. On the Myanmar Road Sign training dataset, Figures 10 and 11 depict the accuracy and loss of GoogLeNet, respectively. Figure 8 and 10 illustrate how the accuracy curve quickly converged to a maximum value of 100% and Figure 9 and 11 show how the loss function reaches a minimum value near 0 for both classifiers although the two classifiers are slightly different in terms of overall performance accuracy and execution time.

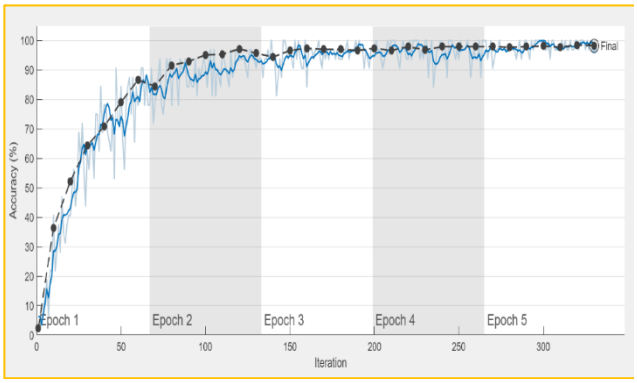


Figure 10. Accuracy of GoogLeNet. The number of epochs is represented in the row, and the column is the accuracy score.

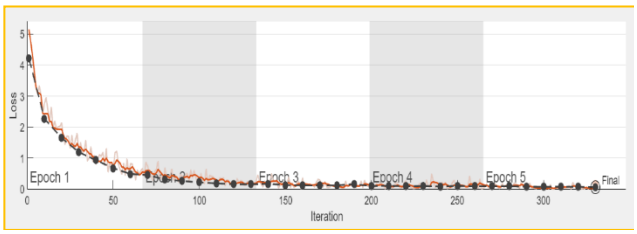


Figure 11. Loss of GoogLeNet. The number of epochs is represented in the row, and the column is the loss score.

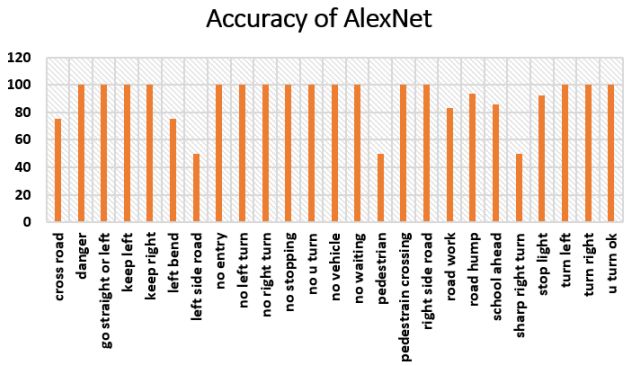


Figure 12. Classification Accuracy of AlexNet

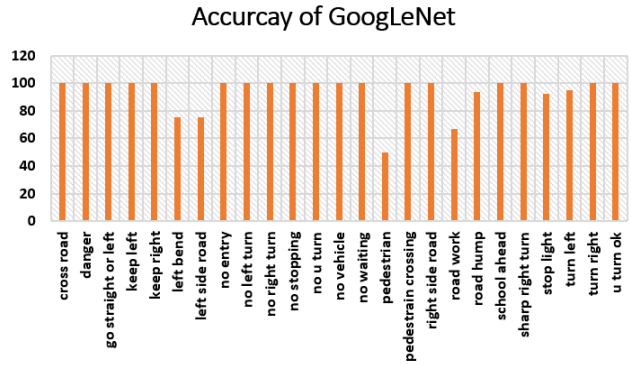


Figure 13. Classification Accuracy of GoogLeNet

The accuracy score of AlexNet for each class is displayed in Figure 12. Figure 13 displays the GoogLeNet accuracy result for each class. The comparison results between AlexNet and GoogLeNet are shown in Figure 14.

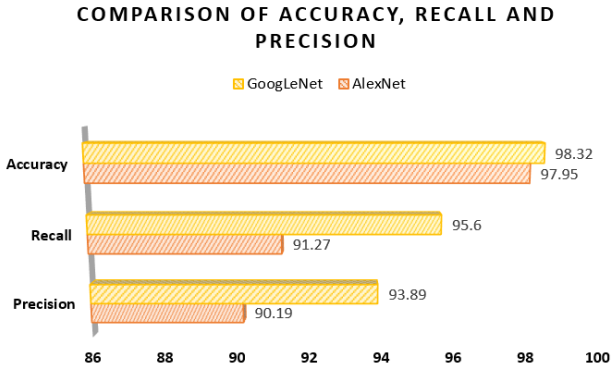


Figure 14. Classification results of AlexNet and GoogLeNet

5. CONCLUSION

For Myanmar, it is crucial to have detection and recognition system for road signs. The number of people who suffer from accidents on the roadways every year is significantly impacted by drivers who disregard road signs. In order to create recognition of road signs system, this study employs pre-trained AlexNet and GoogLeNet CNN models. According

to this research observations, GoogLeNet behaved well for the detection and recognition of road signs, achieving a higher recognition rate when compared to AlexNet model results on the Myanmar Road Sign Dataset. The created dataset will be enhanced in later studies to include more significant road signs and to make the system effective at night and in low - light situations.

6. REFERENCES

[1] Ratheesh Ravindran, Michael J. Santora, Mariam Faied, and Mohammad Fanaei (2019) "Traffic Sign Identification Using Deep Learning", CSCI, pp.1-3.

[2] Jiayuan Yu, Huiling Liu and Huayan Zhang (2019) "Research on Detection and Recognition Algorithm of Road Traffic Signs", IEEE, pp. 2-6.

[3] Dickson Neoh Tze How, Khairul Salleh Mohamed Sahari, Yew Cheong Hou and Omar Gumaan Saleh Basubeit (2019) "Recognizing Malaysia Traffic Signs with Pre-Trained Deep Convolutional Neural Networks", 4th International Conference on CRC, pp.1-5.

[4] Soulef Bouaafia, Seifeddine Messaound, Amna Maraoui, Ahmed Chiheb Ammari, Lazhar Khriji and Mohsen Machhout (2021) "Deep Pre-trained Models for Computer Vision Applications: Traffic sign recognition", 18th International Multi-Conference on SSD, pp. 1-6.

[5] Marco Magdy William, Pavly Salah Zaki, Bolis Karam Soliman, Kerolos Gamal Alexsan, Maher Mansour, Magdy EI-Moursy and Kerolos Khalil (2019) "Traffic Signs Detection and Recognition System using Deep Learning", ICICIS

[6] Sinno Jialin Pan and Qiang Yang Fellow, and IEEE (2009) "A Survey on Transfer Learning", IEEE.

[7] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich (2014) "Going deeper with convolutions ".

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton (2012) "ImageNet Classification with Deep Convolutional Neural Networks".

NATURALNESS ANALYSIS OF MYANMAR TTS USING MEAN OPINION SCORE

Htet Wai Linn[1], Thet Htoo Aung[2], Kyaw Myo Htun[3]

[1]Department of Computer Technology, [2]Department of Computer Technology,
[3]Department of Computer Technology, Higher Education Centre (Pyin Oo Lwin)
[1]htetwailinn53@gmail.com, [2]thethtooaung@gmail.com,
[3]kyawmyohtun.2013@gmail.com

ABSTRACT

One of the most popular technologies based on human language processing at the moment is text-to-speech synthesis (TTS). The naturalness of speech, however, is one of the important outstanding challenges with synthesized speech. This article describes the use of a speech quality assessment technique to rate the naturalness of end-to-end RNN-based Myanmar TTS. In this paper, there are four main parts: collecting text and recorded voice, preprocessing for voice and labelling text, applying RNN model, and evaluating the output speeches. Firstly, we create a speech corpus of 5112 sentences containing Myanmar daily conversation, and syllable segmenter and text normalizer are used for data preprocessing of text, and features extraction of voice are utilized on the recorded audio using the Short Time Fourier Transform (STFT) technique. In the third step, RNN model that synthesizes speech directly from the sequence of text characters are applied. Finally, the naturalness of synthesized speech is analyzed in the training process of steps - 50000, 100000, 150000, and 200000. The optimal model is obtained at the training step-200000, that achieves a mean opinion score (MOS) of 3.9.

KEYWORDS

DSP, RNN, GRU, MOS, Tacotron, Text-to-Speech

1. INTRODUCTION

Modern text-to-speech (TTS) synthesis systems, using deep neural networks trained on large quantities of data, are able to generate natural speech. Generating natural speech from text remains a challenging task despite decades of investigation [1].

Some of the techniques even produced state-of-the-art voice synthesis performances. There are a number of various approaches and technologies that can simultaneously produce speech that is high-quality, very understandable, and very natural-sounding in the recent development of text-to-speech systems. In contrast to other languages, there are still certain challenges in understanding and translating the Myanmar language. Furthermore, there isn't a publicly accessible speech corpus for Myanmar. Our main objective is to construct a speech corpus, create speech that sounds like human speech, and evaluate the naturalness of synthesized speech using MOS.

2. RELATED WORKS

A. M. Hlaing et al. [2] have suggested a word representation for the Myanmar text-to-speech (TTS) system that is based on neural networks. In order to increase the naturalness of Myanmar's text-to-speech system, researchers looked into three acoustic modeling techniques: Deep Neural Network (DNN), Long Short-Term Memory Recurrent Neural Network (LSTM-RNN), and a hybrid of DNN and LSTM-RNN (Hybrid-LSTMRNN). According to their experimental findings, Myanmar TTS systems based on DNN, LSTM-RNN, and Hybrid-LSTM-RNN might enhance the subjective evaluation of the naturalness of the synthesized speeches.

A Deep Neural Network (DNN)-based Myanmar speech synthesis was proposed by Aye Mya Hlaing [3]. They presented research on the alignment of input language features and acoustic information at the state and phone levels for training DNN. Additionally, they contrasted speech synthesis based on HMM

and DNN on a question set for the Myanmar language. In their investigations, it was found that DNN-based state-level speech synthesis produced superior results to HMM-based speech synthesis.

Y. Win et al. [4] proposed Tacotron model for Myanmar text-to-speech system, which is an end-to-end generative Text-to-Speech model that takes a character sequence as input and outputs the corresponding spectrogram. According to their experiments, the results showed that they could obtain the output speech with good quality of intelligibility and naturalness.

3. AIM OF PAPER

The aim of the research paper is to synthesize the Myanmar naturalness speech from input text by using the end-to-end RNN network and evaluate the naturalness of synthesized speech using Mean Opinion Score.

4. MEAN OPINION SCORE

How natural the synthesized speech sounds is one of the key performance measures used to assess Text-To-Speech (TTS) systems. Tests of auditory hearing are conducted to evaluate the naturalness.

The most popular method for evaluating speech synthesis quality is the mean opinion score (MOS) hearing test. The subjective nature of these listening tests is inherent. Participants are asked to judge the naturalness and accuracy of samples of synthetic speech. Due to the subjective nature of the exam, there are significant variations among individuals and among various synthesized texts.

Participants in these examinations score naturalness using a five-point absolute category scale. The naturalness MOS (mean opinion score), which is calculated as the average across all examinees. There is now no dependable instrumental model available, despite the fact that this examination is time-consuming.

A variety of systems are used to generate utterances for a MOS test. A listener is asked to judge the stimulus' quality and naturalness on a scale ranging from 1 to 5, with the ratings denoted as Bad, Poor, Fair, Good, and Excellent [5]. It is possible to determine the intelligibility and naturalness test for the MOS score by using Equation 1.

$$MOS = \frac{\sum_{j=1}^{S}\left(\frac{\sum_{i=1}^{N} R_{ij}}{N}\right)}{S} \quad (1)$$

Where, S refers to the total number of speech outputs, N is the total number of evaluators, R is the total number of test results analyzed by evaluators and j is the speech index.

5. PROPOSED SYSTEM ARCHITECTURE

The steps involved in converting text to speech are as follows: encoding the input text, feeding it into a model that can produce Mel Spectrograms, and then feeding those Mel Spectrograms into a model that can turn them into synthesized audio. In this section, we go over the specifics of our proposed system as follows and figure 1 shows the block diagram of our proposed system.

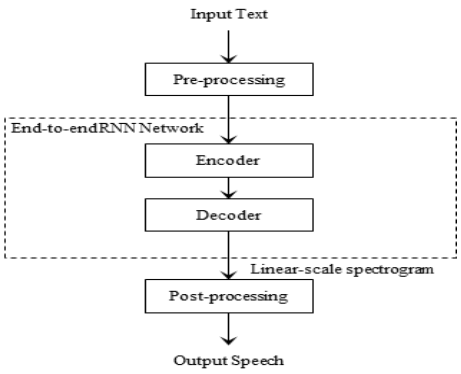


Figure 1. Block Diagram of Proposed System

5.1.Pre-processing for Audio and Labelling Text

The text corpus was provided by manual creation and it contains 5112 lines of Myanmar texts that are either daily conversation sentences or phrases. Before creating the speech corpus, Myanmar text is needed to segment and normalize for data pre-processing. In Myanmar text, there are no clear rules for using word segmentation because Myanmar words are not usually separated by white space. In this paper, we segment the

Myanmar text into a sequence of characters by using a syllable segmenter.

Audacity recorder is used to create the speech corpus. The recorded audio is saved as a Wav file at a sampling rate of 22050 Hz. The length of the audio is between 2 seconds to 8 seconds. In this way, the speech corpus is created and the final speech contains 5112 <text, audio> pairs for male native speaker.

In our proposed system, we implemented feature extraction of a voice using Short Time Fourier Transform technique for Myanmar Text-to-Speech synthesis system. The process of voice features extraction system is pre-emphasis, framing and windowing, short-time flourier transform and Mel-frequency filter bank.

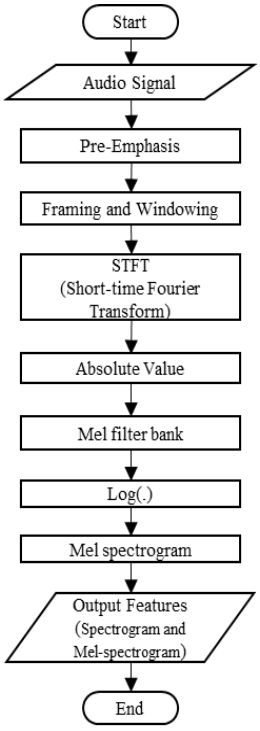


Figure 2. Flow Chart of Voice Features Extraction System

5.1.1. Pre-emphasis

The term "pre-emphasis" refers to the signal flowing through a filter that emphasizes higher frequencies. This will improve the energy of signal at higher frequencies. Pre-emphasis is required because high frequency of speech signal have small amplitude with respect to low frequency components.

5.1.2. Framing and Windowing

An audio signal can be divided into a number of frames in this step, with certain sections of each frame overlapping. As a result, each frame can be analyzed and synthesized separately without losing information. So, each frame can be represented by a single vector. A continuous audio signal is divided into N samples in a frame using this method. The frame's width is usually around 50 milliseconds, with a 12.5 millisecond overlap. The window function is performed to smooth the signal for the computation of the FFT. The output of windowing is

$$Y_m = X_m W_{n(m)}, 0 \leq m \leq N_{m-1} \quad (2)$$

Where, N_m is the quantity of samples in every frame and X_m is input speech signal and $W_{n(m)}$ is the hamming window.

Techniques of windowing functions are Hamming, Hanning, Blackman, Gauss, rectangular, and triangular. Among them, the Hanning window is usually used in speech signal spectral analysis because of its resulting frequency resolution is better than other windowing methods. For the Hanning window, the equation is

$$W_{n(m)} = 0.54 - 0.46 \cos\left(\frac{2\pi}{(N_{m-1})}\right), 0 \leq m \leq N_{m-1} \quad (3)$$

Where, N_m is the quantity of samples in every frame.

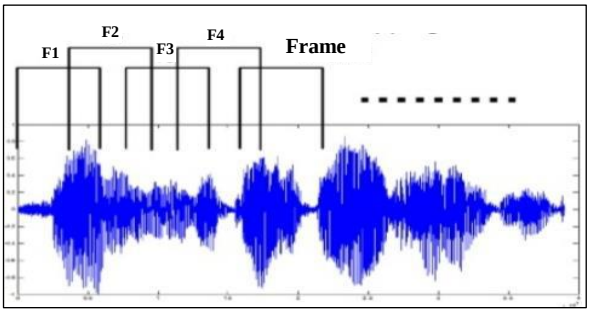


Figure 3. Sample of Framing for Speech Signal

5.1.3. Short-Time Fourier Transform

Short-Time Fourier transform which convert each frame of N samples frame time domain into frequency domain. The input for the Fast Fourier Transform is the windowed signal and its output is the discrete frequency bands. The Discrete Fourier Transform formula is

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi kn/N} \quad (4)$$

Where, N is the number of samples, n is the current sample, k is the current frequency, x_n is the sine value at sample n and X_k is the DFT that includes information of both amplitude and phase.

5.1.4. Mel Filter Bank

Mel filter bank processing is an important step. All frequency bands are not equally sensitive to human hearing. Higher frequencies, roughly greater than 1000Hz, are less sensitive, indicating that human perception of frequency is nonlinear. According to the spectrum, the Mel scale is applied to the filter banks and the output is the sum of filtered spectral components. The logarithm of the square magnitude of the Mel filter bank's output is used to calculate log energy. It also reduces the dynamic range of data.

$$Mel(f) = 2595 * log10(1 + \frac{f}{700}) \qquad (5)$$

This feature extraction system for speech digit .wav file which is implemented by recording Myanmar daily conversation sentences.

In figure 4 describes the spectrogram and Mel-spectrogram of input speech signal using STFT and Mel Filter Bank at each frame. After that the values of spectrogram and Mel-spectrogram are needed to change the normalization values for the training phase in speech synthesis system. In the figure 4, the normalization of spectrogram and Mel-spectrogram are also shown.

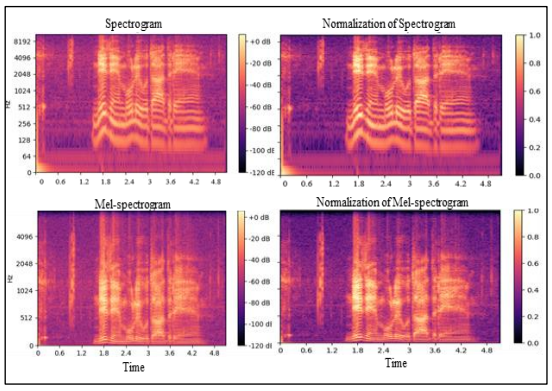


Figure 4. Visualization of Speech Features

After creating the speech corpus, text data is well normalized and then audio files are converted into both spectrograms and Mel-spectrograms. Labelling the audio features and text data to train in proposed RNN model are shown in figure 5.

Figure 5. Labelling the Audio Features and Text

5.2.End-to-End RNN Network

The end-to-end generative model [6], which consists of several neural networks, is used. The Encoder and Decoder networks make up Tacotron's two primary network architectures. Character embedding is the encoder's primary focus. Every word's vector may be produced in character embedding, even if it is an out-of-vocabulary (OOV) word because every word can be spelt using these letters, whereas word embedding handles observed words better. We handle Myanmar texts by using character embeddings of size 256. The encoder is used to convert the input character sequence into a vector.

For each character embedding, a set of non-linear transformations are used. The encoder CBHG module, which includes a bidirectional GRU, 4-layer highway net, max-pooling, and 1-D convolutions bank, comes after the pre-net. The CBHG module is effective in separating robust sequential representation from a character sequence. The attention module's final encoder representation is created by a CBHG module using the pre-net outputs as a starting point.

The decoder itself is made up of a number of parts, including a Pre-net, an RNN wrapped with attention, a stack of GRUs with residual connections, and a CBHG-based post-processing network, as was previously mentioned. The entire goal of the RNN decoder's components is to enable the ability to learn the alignment between the speech signal frames and the text. The targets can be significantly compressed by learning the alignment for decoder targets, in our instance 80-band Mel-scale spectrograms, as long as they offer enough prosody and intelligibility

information for waveform synthesis. This is due to the fact that these decoder targets, or anticipated Mel-scale spectrograms, will be passed into a post-processing net where waveform generation takes place.

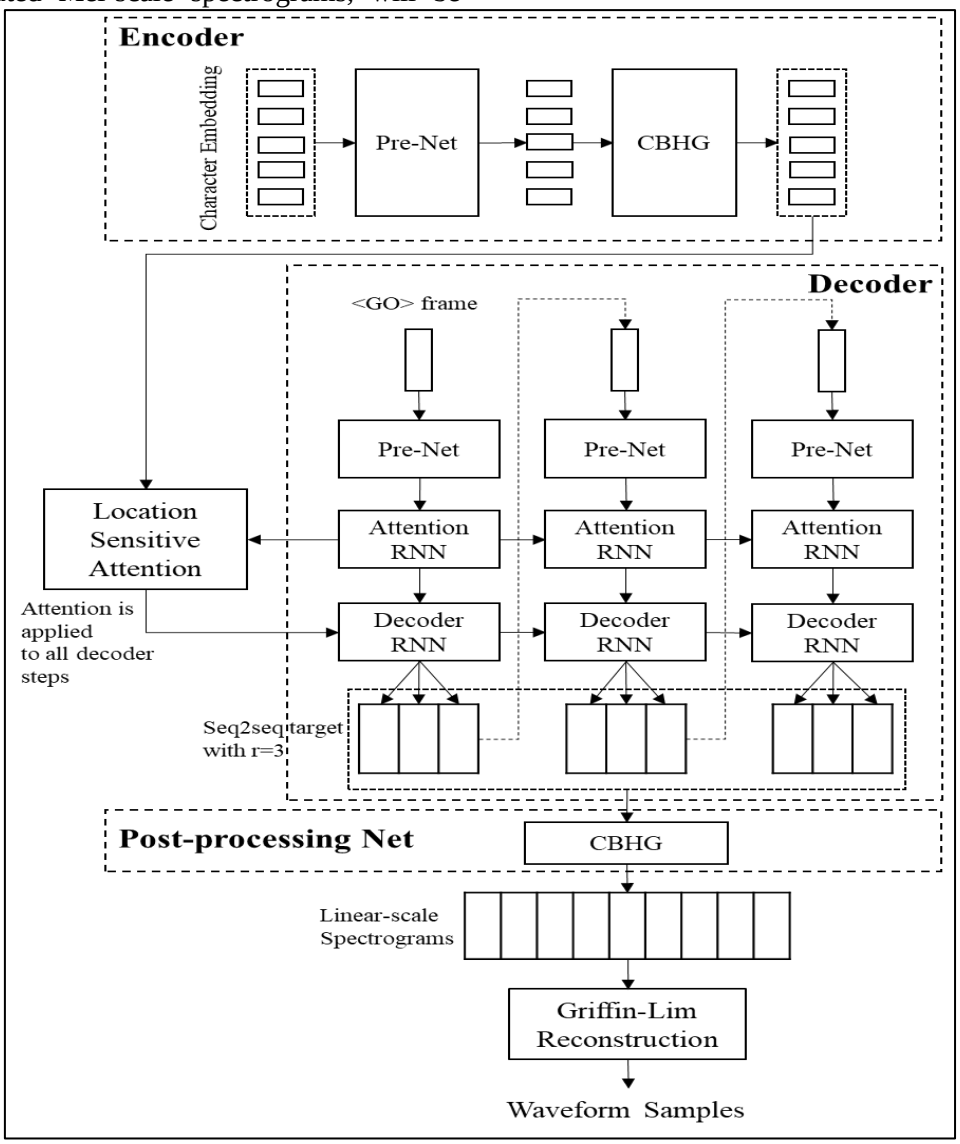


Figure 6. RNN (Tacotron) Network Architecture for Myanmar TTS Synthesizer

The speech audio, which is actually a waveform that represents the speech, is synthesized as the last step by the post-processing network. Predicted Mel-scale spectrograms from the preceding RNN decoder's outputs are passed into another CBHG module. The CBHG module, which is used in both encoder and decoder as previously mentioned, has a somewhat different composition in the decoder than it does in the encoder. The post-CBHG module may fix the prediction error for each frame since the final layer is a bidirectional GRU, which has both forward and backward information. The linear spectrograms are the targets that can be produced with the aid of the post-CBHG module.

Griffin-Lim algorithm is used as the synthesizer to create the waveform from the expected linear spectrogram. Nowadays, many vocoders are available, such as Griffin–Lim algorithm, World, WaveNet and so on. Since the Griffin-Lim algorithm can effectively estimate the phase information from linear spectrograms to reconstruct speech waveforms, we use it as the vocoder in our model in this paper.

Griffin–Lim is an algorithm to reconstruct a signal from amplitude spectrograms without knowing the phase information. Suppose $x(n)$ and $X_w(mS,\omega)$ to be the speech signal and its short-time Fourier transform, respectively. Meanwhile, let $Y_w(mS,\omega)$ be the amplitude spectrogram for the signal that needs to be constructed. Notice that the above-mentioned both linear spectrograms and mel spectrograms are representations of the amplitude spectrograms. The objective function of Griffin–Lim algorithm is to minimize the distance between $x(n)$ and $Y_w(mS,\omega)$. Specially, it can be expressed as

$$D[x(n),Y_w] = \sum_{m=-\infty}^{+\infty} \frac{1}{2\pi} \int_{-\pi}^{\pi} |X_w(mS,\omega) - Y_w(mS,\omega)|^2 d\omega \quad (6)$$

where, S is a positive integer which indicates the sampling period of $X_w(n,\omega)$ in the variable n, w is the window function and ω is the frequency. One iterative method can be used to minimize the objective function $D[x(n),Y_w]$, and the iterative process is shown in figure 7. At the beginning, a speech signal $x^0(n)$ needs to be initialized randomly. In each iteration, the signal $x^i(n)$ obtained in the previous iteration is first transformed into $X_w^i(mS,\omega)$ by short-time Fourier transform. Then, the phase of $X_w^i(mS,\omega)$ is kept unchanged and the amplitude of $X_w^i(mS,\omega)$ is replaced by $|Y_w(mS,\omega)|$ to get $\widehat{X_w^i}(mS,\omega)$. Finally, the reconstructed speech signal $x^{i+1}(n)$ at this iteration is obtained by the inverse short-time Fourier transform of $\widehat{X_w^i}(mS,\omega)$. It is worthwhile to mention that the Griffin–Lim algorithm is very efficient. The missing phase information in the amplitude spectrogram $Y_w(mS,\omega)$ can be recovered within 50 iterations. As a result, we can obtain the reconstructed speech signal $x^{50}(n)$. The magnitude and phase spectra of speech are encoded with this vocoder, and the predicted phase spectrum is given a more realistic sound during reconstruction.

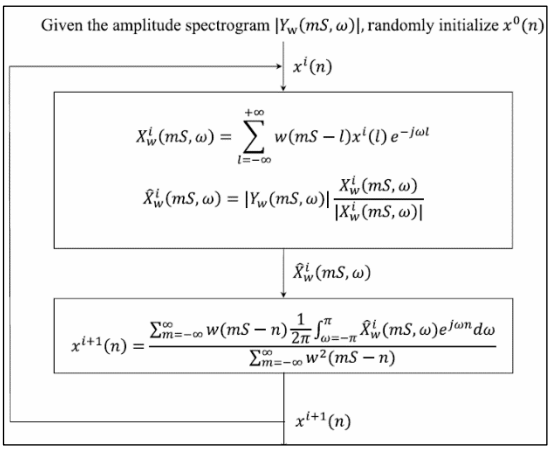


Figure 7. Flow Chart of Griffin-Lim Algorithm

6. TRAINING AND EVALUATION ON PROPOSED RNN MODEL

We train a neural network-based text-to-speech system on Myanmar text corpus that contain about 5 hours and 30 minutes of speech data containing Myanmar daily conversations. The length of the audio is between 2 seconds to 8 seconds. The maximum text length is 146 and the maximum audio frame is 640. All the utterances in speech corpus are used as training data.

Although we utilize similar network design and hyper-parameters, the reduction factor r is different. The preliminary experiment determines the reduction factor r to be 3. The model size, training time, and inference time can all be decreased because three frames will be predicted at once. To eliminate distortions, increase the estimated magnitude spectrograms' power by 1.2 before putting them into the Griffin-Lim algorithm [6]. However, the power of magnitude in our implementation is set to 1.5. In our version, the number of iterations is set to 60, which is still rather quick despite many implementations claiming that Griffin-Lim converges between 30 and 50 iterations. Additionally, we employ the Noam scheme learning rate decay and the Adam optimizer.

The proposed model tries to predict a random input from the current batch into an output utterance every 1000 steps. In this step, a speech wav file and an alignment graph are

generated and placed in the output location. To get a smooth, linear curve, we train 200,000 times. The precision of the matching between the input characters and the output waveform frames is indicated by the alignment graph. During training, it's crucial to keep an eye on alignment plots; if they don't appear linear, training will probably need to be redone. A few alignment plots obtained using our suggested model at intervals of around 10,000 steps. The alignment graph of 10000 steps is shown in figure 8.

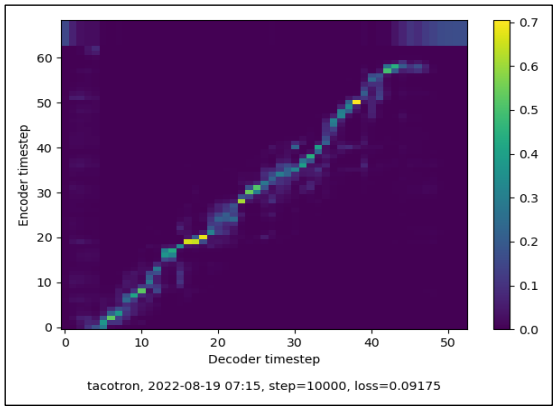


Figure 8. Alignment Graph at 10000 Steps

A large variety of subjective and objective testing methods for TTS system are available nowadays. Among of them, this paper uses subjective evaluation using Mean Opinion Score (MOS) method to evaluate the experimental results. By measuring how much of the voice output can be understood clearly and how much of it sounds natural and similar to human speech, we were able to perform the evaluation.

Using the MOS from five native speakers, we assessed the output of the synthesized speech (1-bad, 2-poor, 3-fair, 4-good, and 5-excellent). For the evaluation experiment, two different types of speech were prepared. The first is the original recorded speech, while the second is the output of our speech synthesizer.

The 10 speech outputs for the naturalness test are selected as the following;

“ မင်း မိုးလေဝသ သတင်း ကြည့်မိသေးလား ”

“ ဒီနေ့တော့ မိုးမရွာလောက်ဘူး ထင်တာပဲ”

“ ရန်ကုန် မှာလည်း ပုသိမ် ဟာလဝါ ဝယ်စားလို့ ရပါပြီ”

“ ကျိုက်ထို က နေ ကျိုက်ထီးရိုး ဘုရား ကို ဘယ်လောက် ဝေးလဲ”

“ မိုးမရွာ လောက်ဘူး ထင်တာပဲ ဒီနေ့တော့”

“ ထင်တာပဲ ဒီနေ့တော့ မိုးမရွာ လောက်ဘူး”

“ မိုးလေဝသ သတင်း ထဲမှာ ဒီနေ့ နေသာမယ်လို့ ကြေညာ ထားတယ်”

“ ရန်ကုန် မှာ ဒီနေ့ မိုးမရွာ သလို ပုသိမ် မှာလည်း နေသာ ပါတယ်”

“ ရန်ကုန်နဲ့ ကျိုက်ထို က ဘယ်လောက် ဝေးလဲ”

“ ကျိုက်ထီးရိုး ကို လာပြီး ပုသိမ် ဟာလဝါ ဝယ်စားချင် လို့တော့ မရဘူးလေ”

In the testing data, the first 4 lines are included directly in the training dataset and the 5th and 6th lines are shifted for phrases. The last 4 lines are tested by including unknown data. The differences of testing data are also shown in table 1.

Table 1. Testing Data of Daily Conversation

Total data	**Original data**	**Shifted phrases data**	**Included unknown data**
10	4	2	4

Then, the five raters had to record everything they heard, even if they didn't understand what they were hearing. When we determined MOS for each individual in training process of steps - 50000, 100000, 150000, and 200000 respectively, the male speaker's MOS for the naturalness test is described at the below section.

The linear alignment graphs of the obtained models we tested in the training process are shown in figure 9.

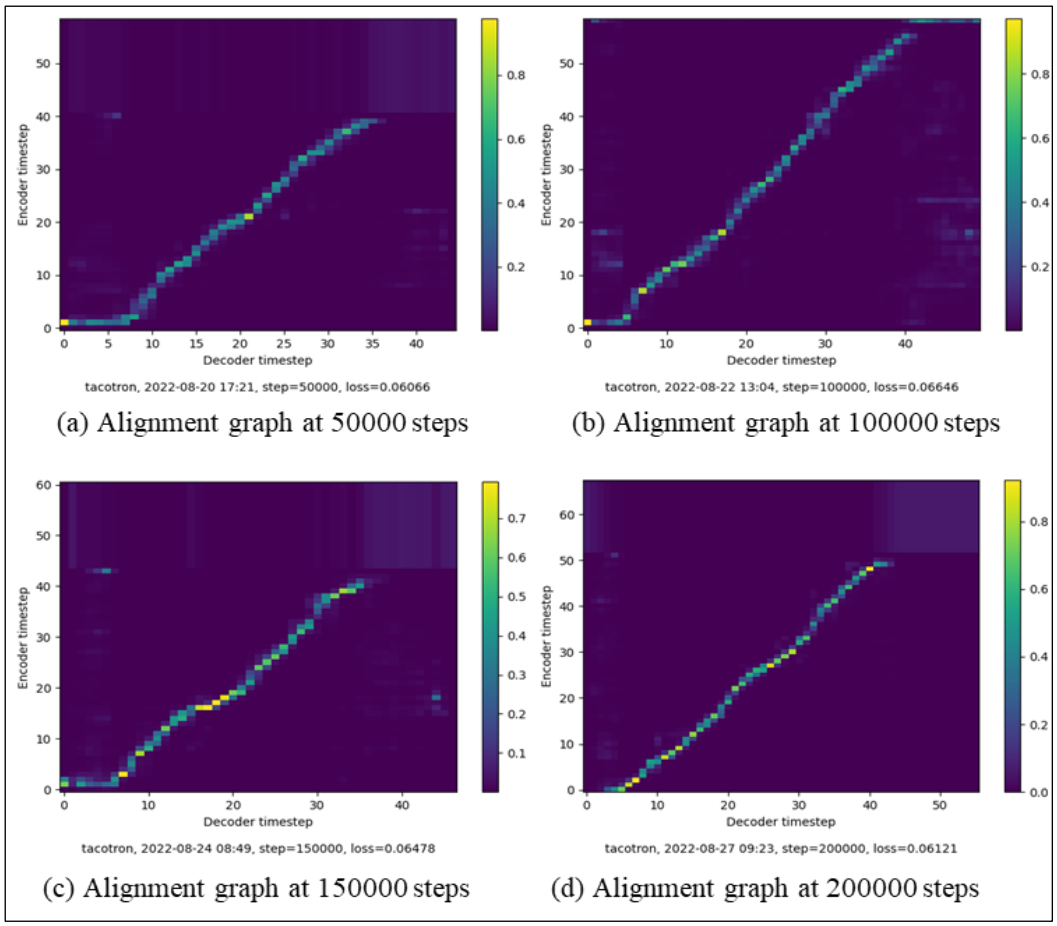


Figure 9. Linear Transformation of Alignment Graphs at the Evaluation Steps

In the training process, as the number of training steps increases, the value of the loss decreases and the alignment graph is more linear. The evaluation results of the obtained model after training steps- 50000, 100000, 150000 and 200000 are shown in figure 10.

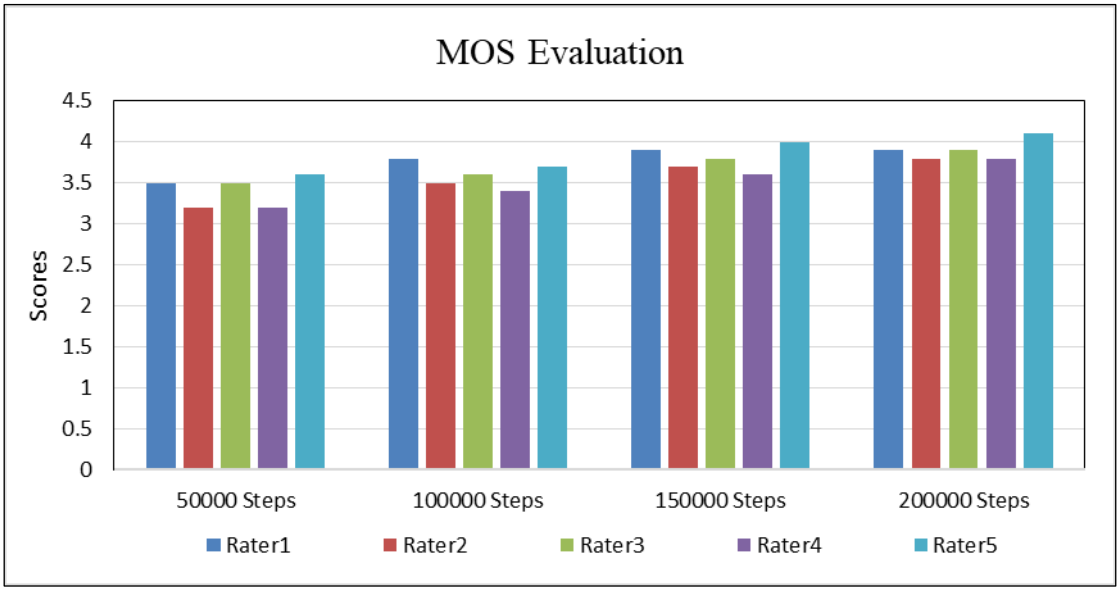


Figure 10. Evaluation Results of the Five Native Raters

The results of the MOS rating for the naturalness test are depicted in table 2.

Table 2. Mean Opinion Score (MOS) for Naturalness Evaluation

Trained Steps	Original Audio	Synthesized Speech
50000	4.7	3.4
100000	4.7	3.6
150000	4.7	3.8
200000	4.7	3.9

According to scores from other tests using end-to-end speech synthesizers created for other languages [6], the average MOS score for the human voices is 4.7, while the average score for proposed method synthesizer is 3.9 at the end of training. Additionally, the MOS scale produced by our proposed approach is competitive with the results of comparable MOS evaluations in Myanmar text-to-speech systems based on HMM [7] and DNN [3].

7. CONCLUSIONS

In this study, we implement an RNN-based Myanmar text-to-speech system that utilizes character sequences as input and generates spectrograms that correspond to those sequences. The original recorded audio and the trained model's output speech are compared with the subjective evaluation to see the difference in naturalness using mean opinion score (MOS). The experimental results showed that we could synthesize speech with a reasonable level of technical accuracy and naturalness. In the future plan, we'll implement multi-speaker (male and female voice) text-to-speech systems with more data set for additional field of usage areas.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to Dr Thet Htoo Aung, my dearest supervisor, Department of Computer Technology, and Dr Kyaw Myo Htun, my co-supervisor, who have, as always, given us their support throughout writing of this paper.

REFERENCES

[1] P. Taylor (2009), "*Text-to-Speech Synthesis,"* Cambridge University Press, New York, NY, USA, First edition.

[2] A. M. Hlaing, and W. P. Pa (2020), "*Word Representations for Neural Network Based Myanmar Text-To-Speech System*," International Journal of Intelligent Engineering and Systems, vol. 13, no. 2, pp. 239–249, doi: 10.22266/IJIES 2020.0430.23.

[3] A. M. Hlaing, W. P. Pa, and Y. K. Thu (2018), "*DNN-based Myanmar speech synthesis*," The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages, pp. 29-31.

[4] Y. Win, H. P. Lwin, and T. Masada (2020), "*Myanmar Text-To-Speech System Based on Tacotron*," Carleton University, IEEE 2020.

[5] A. Rosenberg and B. Ramabhadran (2017), "*Bias and Statistical Significance in Evaluating Speech Synthesis with Mean Opinion Scores*," INTERSPEECH 2017, August 20–24, Stockholm, Sweden.

[6] Y. Wang, RJ. Skerry-Ryan, D, Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z, Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, R. A. Saurous (2017), "*Tacotron: Towards End-To-End Speech Synthesis,*" arXiv:1703.10135v2 [cs. CL], 2017.

[7] Y. K. Thu, W. P. Pa, J. Ni, Y. Shiga, A. Finch, C. Hori, H. Kawai, and E. Sumita (2015), "*HMM Based Myanmar Text-to-Speech System,*" in Proc. INTERSPEECH 2015, pp. 2237-2241.

Authors

Biography

Htet Wai Linn received the M.A.C.S degree in automatic control systems from the Moscow Institute of Electronic Technology (MIET), Russian Federation in 2015. He is now attending the Ph.D. 17th Batch at the Higher Education Center (HEC), Pyin Oo Lwin.

Thet Htoo Aung received the M.I.C.Sc. degree from the Moscow Institute of Electronic Technology (MIET), Russian Federation in 2009. He continued his studies and got his Ph.D. (Computer Technology) in 2016. He is

now serving as an assistant lecturer at the Department of Computer Technology at the Higher Education Center (HEC), Pyin Oo Lwin.

Kyaw Myo Htun received the M.I.C.Sc. and Ph.D. (IT) from the Moscow Institute of Electronic Technology (MIET), Russian Federation in 2008 and 2013. He has served as a lecturer at the Department of Computer Technology at the Higher Education Center (HEC), Pyin Oo Lwin.

A NEW UNSUPERVISED TEXT SUMMARIZATION SYSTEM USING NATURAL LANGUAGE PROCESSING TECHNIQUES

A Mar Myo Thu[1], Wut Yi Oo[2]

University of Technology (Yatanarpon Cyber City)

[1]amarmyothu2015@gmail.com, [2]wutyioo@gmail.com

ABSTRACT

Nowadays, numerous texts are spread out the whole over the world. Reading enormous texts may be a tedious case. So, text summarization is necessary and important in the area of research fields. Two categories of automatic text summarizations are supervised and unsupervised methods. Supervised text summarization needs already trained data to summarize. In my paper, a new unsupervised text summarization model is proposed. This system helps people efficiently reading from articles in an abstract format.

Keywords:

Supervised-learning, Unsupervised-learning, summarization

1. INTRODUCTION

In recent years, text summarization is rapid attention in research fields because of the huge amounts of text documents, users are difficult to read useful information from them. Due to the huge amount of enormous information, users cannot read many relevant text documents. Then, the available online text documents are increased exponentially and the user makes research and text summarization applications are very important and may be attractive research fields.

The techniques of text summarization can be summarized into abstraction methods and extraction methods, multi-documents or single-documents supervised or unsupervised methods. Input texts and output texts are the same in extraction methods. But input sentences and output sentences are not equivalent in the abstraction method. The system must know the semantics of original text documents.

Extractive summarization can be classified into two groups: supervised text summarization and unsupervised text summarization. In supervised methods for summarization, the task of high-ranking high-rank sentences is represented as a binary classification problem, extracting all sentences in the input into summary and non-summary sentences. Unsupervised learning-based text summarization does not require any training data, and thus can be applied to any text data without requiring any human effort. The two popular unsupervised learning methods commonly used in the context of text data are clustering and topic modeling [1_2].

Many of text summarization systems used extractive approaches and supervised methods. This paper focuses on unsupervised text summarization method-based extractive document summarization. My system can summarize any text files and any web pages. The result will show my model is an effective and useful area and can help users with the abstractive format.

2. RELATED WORKS

The most automatic summarization is divided into extraction-based and abstractive summarization. The extractive model extracts

several sentences and words from the original article by choosing the highest-rank sentences. (Mihalcea and Tarau, 2004[3] Xuet al., 2019 Text Rank algorithm which is based on sentence rank importance, and the extraction method based on a supervised model (Liu, 2019) [4] proposed the extraction method based on supervised model. The abstracts obtained by the extraction model output may be obtained the most relevant to the article, but because the extracted sentences are spread out in different parts of the whole article, the coherence of the abstracts is a problem to be challenged. Seq2seq structure is used in abstraction models. And, BART (Lewis et al., 2019), proposed the pretraining model that is used to achieve a better generation effect [5].

Single-Document Myanmar Textis Summarization using Latent Semantic Analysis (LSA) is proposed by Soe Soe Lwin(2018) which summarize Myanmar news from Myanmar official [6].

3. THEORICAL BACKGROUND

There are 6 algorithms which belong to 3 categories are showed as the following.

- Term Frequency
- Latent Semantic Indexing
- Non Negative Matrix factorization
- Sum Basic
- Page Rank
- Embedding Page Rank

To explain about **Term Frequency**, frequencies of terms including a sentence are added to calculate term score. Sentence length is used to calculate regularization. After top score is selected from the sentence, the frequency of all the words including the selected sentence is reduced to reduce more variance in the selected sentences. **Matrix factorization** is done with Singular Value Decomposition in Latent Semantic Indexing. In **Text Rank**, cosine similarity is used to fine similarity between sentences using sentence term matrix. The graph is created to find similarity between sentences. Similarity matrix is based on created graph. Then this matrix is put to Page Rank Algorithm base on the graph. **Embedding Text Rank** is similar, similarity matrix is calculated using sentence embedding vector .Sentence embedding vector is calculated by averaging containing of all the words in the sentence.

The internet is a very complex network. The relation between pages can be represented as a graph. The importance of any vertex is the in-degree (out-degree); which is the number of inbound (outbound) links to this vertex. The inbound of a given page, is an indicator of a page's importance or quality. PageRank algorithm uses this idea to rank the pages that appear in the search results. PageRank does not consider all the inbound links from the pages equal, the link will take an extra importance depending on the importance of the page that comes from it.

The PageRank algorithm of a Web page u, denoted by PR (u), where d is the damping factor with 0.85 and N: is the total number of nodes. Algorithm 1 shows how the PageRank algorithm works; it starts by initializing the rank of each node by 1/N, where N is the number of nodes in the graph. Then the algorithm iterates and calculates the new ranks of the nodes according to Formula .After calculating the new ranks for all nodes, the ranks are updated. This process is repeated depending on iterations count. The iteration here is used to update the rank for each sentence to get the best and most stable rank, since the order of the sentence is directly associated with the order of the associated sentences, which changes every time the algorithm is applied.

```
   Input: Weighted Graph G.
   Output: Scored Graph.
1  Configure N = Number of Nodes in G.
2  Current_Rank ← Double[N]
3  Temp_Rank ← Double[N]
4    Foreach n = 1 to N
5    Current_Rank[n] = 1/N
6      For i = 1: Number_of_Iterations
7    Foreach nd: G.Nodes
8      Temp_Rank[nd.index] = Calc_Page_Rank(nd)
9    Current_Rank = Temp_Rank
```

Figure 1: Page Rank Algorithm

This paper is used page rank algorithm based new sentence similarity equation to get more relevant summarize texts.

4. PROPOSED UNSUPERVISED TEXT SUMMARIZATION SYSTEM

In proposed system, the following steps are implemented.

- Input text document or web page link
- Preprocessing
- Calculate Term Frequency and Inverse document frequency for each sentences.
- Calculate sentence similarity with new proposed method
- Building similarity graph
- Sort the sentences based on the calculated importance.
- Evaluating Performance of the system
- Display final summary of user needs.

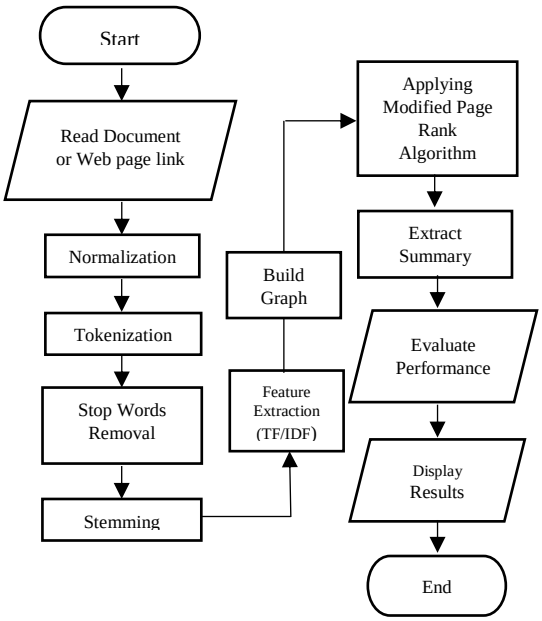


Figure 2: Flow Chart of the proposed system

In preprocessing step, for text file, removing stop words which are unnecessary for summarizing texts, and term frequency and inverse document frequency, and new weight values are calculated using new sentence similarity method. In processing steps, cut off sentences below limited threshold value. Threshold value is chosen by manually. Because human can read about 7 lines of texts without stress. Finally, user can view smart summarize texts.

To explain about new similarity measure weight, the following equation is used.

The new similarity measure is proposed by my system. So, T = {t_1, t_2… t_n } represent all the distinct terms that occurred in the collected documents D, where n is the number of terms including in D. In the vector space model, each sentence s_i is represented using these terms as a vector in m –dimensional space, W_i = {w_{i1}, …, w_{in}} where each component reflects the weight of a relevant term. The popular one of weighting scheme is the Term Frequency–Inverse Document Frequency (TF-IDF) weighting scheme.

In this paper, instead of using a simple Term Frequency inverse sentence frequency tf-isf scheme, my system uses the new symmetric weight measure:

$$w_{ij} = \frac{tf_{ij}}{tf_{ij} + 0.65 + 0.5 * \frac{l_j}{avgl}} * sf_j$$

Where inverse sentence frequency sf is obtained by dividing the total number of sentences by the number of sentences containing the term, and then taking the logarithm of that quotient:

$$sf_j = \log(\frac{n}{n_j})$$

where n is the total number of sentences in the document collection D; n_j is the number of sentences in which the term s_j occurred; tf_{ij} is the number of occurrences of term tj in sentence S_i, l_i is the sentence length S_i and avgl is the average length of the sentence. This formula normalizes the length of sentences rather than the simple tf-idf method.

Firstly, the user can input not only any text files but also web page text with getting any http link. Then, term frequency of all texts and the new symmetric weight method is used to extract the frequency of each term.

Then, the highest rank sentences are chosen and constructed as summarized sentences. Combine all summarized sentences and cut off 7 sentences with threshold value 7. Finally, final summary will get in a few minutes.

In the proposed system, the user can enter text files and automatically summarize within a single minute. The results show how much relevant and effective. Figure 1 shows input document and output summarize texts.

Input Document:

We are currently experiencing another gold rush in AI. Billions are being invested in AI startups across every imaginable industry and business function. Google, Amazon, Microsoft and IBM are in a heavyweight fight investing over $20 billion in AI in 2016. Corporates are scrambling to ensure they realize the productivity benefits of AI ahead of their competitors while looking over their shoulders at the startups. China is putting its considerable weight behind AI and the European Union is talking about a $22 billion AI investment as it fears losing ground to China and the US.

AI is everywhere. From the 3.5 billion daily searches on Google to the new Apple iPhone X that uses facial recognition to Amazon Alexa that cutely answers our questions. Media headlines tout the stories of how AI is helping doctors diagnose diseases, banks better assess customer loan risks, farmers predict crop yields, marketers target and retain customers, and manufacturers improve quality control. And there are think tanks dedicated to studying the physical, cyber and political risks of AI.

So who will make the money in AI?

Output Summary

Google, Amazon, Microsoft and IBM are in a heavyweight fight investing over $20 billion in AI in 2016. And there are think tanks dedicated to studying the physical, cyber and political risks of AI. From the 3.5 billion daily searches on Google to the new Apple iPhone X that uses facial recognition to Amazon Alexa that cutely answers our questions. Corporates are scrambling to ensure they realize the productivity benefits of AI ahead of their competitors while looking over their shoulders at the startups. Billions are being invested in AI startups across every imaginable industry and business function.

Figure 3: Proposed Input and Output Texts

5. EVALUATION OF THE PROPOSED SYSTEM

The system's compression ratio can be seen as dynamically low in the following figure.

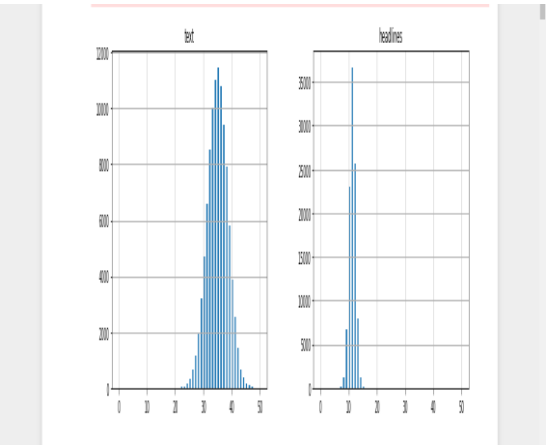


Figure 4: Text and Headline compression ratio

Precision and recall can be seen in the following table: Precision is 84 % and Recall is 58%. So, my system can be seen in the good nature of summarization.

Table 1: Precision and Recall

System-human choice of total sentences in Summary	196
The system chooses total sentences in Summary	335
Humans choose total sentences in Summary	231
Precision	84%
Recall	58%

With simple testing with original page rank algorithm, Precision is only 76% and Recall 50%. So, proposed method improves slightly compared with original method.

6. CONCLUSION

Nowadays, text Summarization is a hot research area and current summarization methods which produced a new unsupervised text summary are mainly focused on natural language processing methods. Proposed system can summarize any text and any web page. Supervised methods must train with big data to summarize. And then, proposed unsupervised text summarization system is not necessary to overall training. So, time complexity and space complexity can be reduced. So, proposed system can produce a promising good summary for every learners.

REFERENCES

1. Alyguliyev R.M. *"The two-stage unsupervised approach to multi-document summarization,* Automatic Control and Computer Sciences*"*, 2009, vol.43, no.5, pp.276–284.

2. Aliguliyev R.M. *"Multidocument summarization through clustering and ranking of sentences"*, Problems of Information Technology*"*, 2010, no.1, pp.26–37.

3. Paul Tarau. And Rada Mihalcea. *"Textrank: Bringing order into the text"*, 2004 conference on empirical methods in natural language processing, 2004, pages 404–411.

4. Yang Liu. *"Fine-tune bert for extractive summarization"*, 2019, preprint arXiv: 1903.10318.

5. Lewis, M., L. Bart, "*Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*", 2019, arXiv preprint arXiv: 1910.13461.

6. Lwin, S.S, Nwet, K.T, "*Single-Document Myanmar Text Summarization using Latent Semantic Analysis(LSA)"*, 2018 , 16th international conference on Computer Appplications (ICCA2018).

A Mar Myo Thu received the bachelor of computer science from University of Computer Studies, Dawei in 2006 and Master from the University of Computer Studies, Dawei in 2010. She was working as tutor in Computer University, Dawei from 2006 to since 2010. Then, she promote as assistant lecturer in University of Technology (Yatanarpon Cyber City) at 2013. Then, she was transferred to University of Computer Studies (Pyay) from 2017 to 2021. Now she is working at University of Technology (Yatanarpon Cyber City) form 2021 to present. Her recent interest on research area includes natural language processing and machine learning.

OFFENSIVE SPEECH DETECTION USING LOGISTIC REGRESSION

Ei Phyoe Hein[1] and Ei Ei Thu[2]

[1,2] Department of Computer Science, University of Computer Studies, Taunggyi
[1]eiphyoehein@ucstgi.edu.mm,[2]eieithu@ucspinlon.edu.mm

ABSTRACT

Nowadays, Social Media platforms becomes the dissemination of offensive speech. As online content continues to grow, the offensive speech spread rapidly. Therefore, many countries have developed law to prevent online offensive speech. Offensive speech identification is to detect the inflict tweets, abusive comments, execrate words on social media platforms. In order to detect hurtful tweets, the efficient offensive speech detection technique is used. In order to detect offensive speech, Offensive Speech Dataset is used in this system that contains 23353 tweets collected from Twitter social media website. The experiments are conducted with 80% for training data and 20% for testing data. This system provides a simple machine learning model using logistic regression method. An efficient logistic regression model is build using tweets feature which obtain from word vectorization process. The proposed system intends to solve the problem of offensive speech detection. Results show that the proposed logistic regression approach gains high accuracy in offensive speech detection.

KEYWORDS

Machine learning, logistic regression, offensive detection

1. INTRODUCTION

Nowadays, the impact of social media is significantly increased on human daily activity. Accordingly, social media become the essential set of human communication. Social media are the primary media for committing offensive speeches. For this reasons, social media parts and comments can control the human emotion and behaviours. Therefore, social media are the primary media for committing offensive speeches. Most country set rules and laws to protect minority people from suffering abusive words and speech. Correctly detecting offensive word is still difficult task. It is challenging problem to identifying the word whether it is abusive or not.

Machine learning plays vital role in identification and detection domain. In machine learning, there are many methods to detect offensive speech. Among them, logistic regression obtains the higher accuracy in detection than other methods. The detection method described in this paper is based on the logistic regression model. An efficient Logistic Regression model is proposed to predict the offensive speech. This system intends to predict tweets from a twitter dataset [6] whether it is offensive or not.

2. RELATED WORK

In recent works, many of the research have been conducted to detect offensive speech. Some of the related works are presented in this section.

The authors proposed the Logistic Regression method for detection offensive and hate speech [1]. Their approach detects hate speech and cyber bullying using XGBoost, Logistic Regression and Decision Tree machine learning algorithms. According to their result, the Logistic Regression and Decision Tree model outperform than other machine learning model.

The researchers [2] proposed hate speech detection approach for the Amharic language. They collect of 6120 instances of

Amharic posts from Facebook, and they labeled the speech as “hate” and “not hate” using word2vec and term frequency-inverse document frequency (TF-IDF) feature extraction. In their system, they used Naïve Bayes (NB) and random forest (RF) classifier to detect the features of “hate” and “not hate” speech. According to their result, NB classifier is outperformed than RF classifier.

The researchers [3] conducted the identification of offensive tweets English language. They build machine learning model using Scikit and logistic regression.

3. MACHINE LEARNING

The main goal of machine learning is to transform input data into significant outputs and to learn useful representations of the input data. The machine ‘learns’ and uses its algorithms through supervised and unsupervised learning. Supervised learning means to train the machine to translate the input data into a desired output value. Unsupervised learning means discovering new patterns in the data without any prior information and train [4]. Machine Learning techniques have been applied to many real-life industrial problems with success. One of the commonly used techniques is logistic regression. Logistic regression technique belongs to the family of supervised learning algorithms.

The application of creating artificial intelligences (AI) is supervised learning. In this case, input data has been labelled for a particular output and a computer algorithm is trained on that input data. Detections of the underlying patterns and relationships between the input data and the output labels are trained by the model until it can detect.

3.1. Logistic Regression

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable [5]. The most common logistic regression models a binary outcome such as true or false, yes or no, and so on.

The origin of logistic regression comes from the *logistic function*, also known as *sigmoid function*. Logistic Regression uses this function to predict the probability of a given point to belong to a class. The function maps any real value into a value between 0 and 1.

In Logistic Regression, the input values (x) are combined linearly using coefficient values (β), called weights to predict an output value (y). Figure 1 depicts the function of logistic regression.

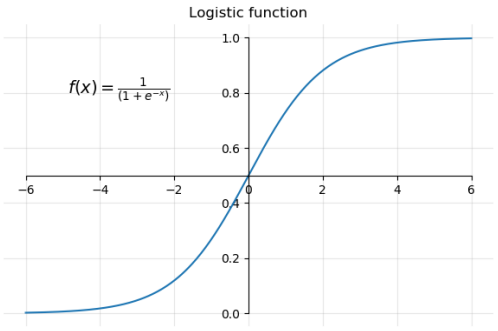


Figure 1. Logistic Function

3.2. Naive Bayes

The Naïve Bayes algorithm is a set of supervised learning which is based on the Bayes theorem that is used to solve classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Naïve Bayes has three type of models that are Gaussian, multinomial and Bernoulli. The formulation of Naïve Bayes classifiers is depicted below the equation 1 where ‘p(x)’ is the probability of feature given values for ‘y’ features.

$$P(x|y) = \prod_{i=1}^{p} p(xi|y) \qquad (1)$$

3.2. TF-IDF

In order to measure the frequency of word in document, TF-IDF method is used. TF

means term frequency and it can obtain by dividing count of one word by told number of words. IDF means inverse document frequency and it can obtain by taking log value in dividing number of specific terms including documents and total number of documents. Equation 1 describes the weight of word.

$$tfidf(d,t) = tf(t) \times idf(d,t) \qquad (2)$$

In this paper, TF-IDF is used to transform tweeted words into numeric word vector.

4. System Design And Architecture

The aim of this system is to detect offensive speeches. In order to detect offensive speech, the proposed system build efficient machine learning model using logistic regression method. There are three main parts in the proposed system architecture. Figure 2 represents the overview architecture of the proposed system.

I. Data Pre-processing
II. Feature Selection
III. Model Building

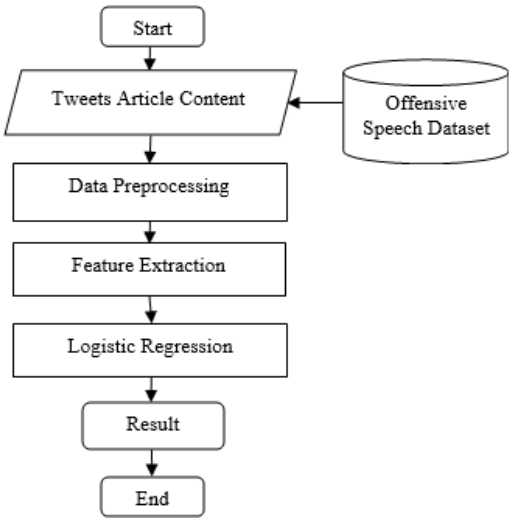


Figure 2. System Architecture

4.1. Data Pre-processing and Dataset

In order to pre-process the tweeted data, the proposed system tokenize, stem and remove the username, punctuations, special characters, emoji, hash symbol and hashtag using NLTK, PANDAS, NUMPY, Scikit-learn, Porter Stemmer, Word-net Lemmatizer, Stopword remover under the NLTK library. After removing unnecessary characters, the system changes all texts into lower case.

Natural Language Library (NLTK) is mainly used for interaction between computer and human language. In this system, NLTK use for tokenization, porter stemmer, lemmatizations, lower case conversion, stop words removal, remove twitter handle (e.g.; @user), remove url from tweet (e.g.; http://www.twitter.com), remove punctuations, numbers (e.g.; 1, 2, 3…) and remove special characters. PANDAS is used for handling data in this system. It can handle missing data, cleaning up the data and it supports multiple file formats. NUMPY is use to provides efficient multi-dimensional array objects and various operations to work with these array objects. There are six features in offensive dataset [6]. These are id, count, offensive, neither, class and tweet data.

4.2. Feature Selection

In feature selection stage, the system extracts word from tweets data using Count Vectorizer function. The goal of using TF-IDF is to reduce the effect of less informative tokens that appear very frequently in the data corpus. The following Table 1 show the original tweets and Table 2 show result of the preprocessing step.

Document = [“I call him my bitch @MaKennaM_”, “I dont have hoes 😴”, “Is that an albino Mexican?”]

Table 1. Original Tweet

	Original Tweets
D-1 : "I call him my bitch" @MaKennaM_ D-2 :"I dont have hoes" 😴” D-3 : "Is that an albino Mexican?"	

Table 2. Result of the Pre-processing

Pre-processing Result
D-1 : call bitch
D-2 : hoes
D-3 : albino Mexican

Table 3 show the TF-IDF calculation result for three documents.

4.3. Model Building

In regression model building process, the weighted results from feature selection process is used to train model.

In order to train model, the coefficient for independent variables is calculated first. Table 4 shows the calculation results of coefficient for independent variables (x). By using these coefficients, the predictor variable (y) is calculated. Equation (3) is the formula for dependent variable (y_i) and equation 4 is the formula of logistic regression probability.

$$y_i = \beta_0+\beta_1 x_1+ \beta_2 x_2+..+ \beta_p x_i \qquad (3)$$

where, for i=n observations:
y_i = dependent variable
x_i = independent variables
β_0 = y-intercept
β_p = coefficients for each independent variable

Formula of Logistic Regression Probability:

$$P(y=0) = e^{(y_i)}/1+e^{(y_i)} \qquad (4)$$

For Offensive;

$$P(y)=e^{-0.799+(-0.189589(0.23856))+(-0.380224(0)+(0.1913535(0))}/1+e^{-0.799+(-0.189589(0.23856))+(-0.380224(0)+(0.1913535(0))}$$

P(y)=0.30064

Probability:

For D-1 : P(y) = 0.30064 < 0.5
For D-2 : P(y) = 0.27281 < 0.5
For D-3 : P(y) = 0.67991 > 0.5

< 0.5 is Offensive tweet
> 0.5 is Neither tweet

According to the result, D-1 and D-2 are offensive tweets and D-3 is neither tweet. Table 5 show the final result of the proposed logistic regression method.

Table 5. Final Result

	Input	Label
D-1	"I call him my bitch" @MaKennaM_	offensive
D-2	:"I dont have hoes" 😴	offensive
D-3	"Is that an albino Mexican?"	neither

4.4. Experimental Results

Confusion Matrix: It is the representation of the information about the actual and predicted classifications by the classification algorithm. It gives the insight into the accuracy of each of the classes that are present in the target feature meaning that the system gets the number of test instances for each class that are correctly as well incorrectly predicted by proposed model.

From this comparison four categories are identified: True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN). True positives refer to instances of offensive speech text that were correctly identified as offensive speech whereas True Negatives are instances of not offensive speech text correctly predicted as not offensive speech. False positives are the instances of not offensive speech text incorrectly classified as offensive speech whereas False Negatives are instances of offensive speech text incorrectly determined to be not offensive speech. These four categories can be illustrated in a confusion matrix as in Table 6.

TF-IDF Calculation for Offensive Speech Detection										
Word	**Count**			**Term Frequency (TF)**			**Inverse Document Frequency (IDF)**	**TF*IDF**		
	D-1	**D-2**	**D-3**	**D-1**	**D-2**	**D-3**		**D-1**	**D-2**	**D-3**
call	1	0	0	0.5	0	0	0.47712	0.23856	0.00000	0.00000
bitch	1	0	0	0.5	0	0	0.47712	0.23856	0.00000	0.00000
hoes	0	1	0	0	1	0	0.47712	0.00000	0.47712	0.00000
albino	0	0	1	0	0	0.5	0.47712	0.00000	0.00000	0.23856
mexican	0	0	1	0	0	0.5	0.47712	0.00000	0.00000	0.23856
							Average	0.23856	0.47712	0.23856

Table 3. TF-IDF Calculation Result

Table 4. Coefficient of Logistic Regression

Coefficient of Logistic Regression Calculation for Offensive Speech Detection							
label	**D-1**	**D-2**	**D-3**	**Logit**	**EXP(Logit)**	**Probability**	**Log Likelihood**
0	0.2386	0.0000	0.0000	-0.8442	0.4299	0.3006	-0.3576
0	0.0000	0.4771	0.0000	-0.9804	0.3752	0.2728	-0.3186
1	0.0000	0.0000	0.2386	-0.7533	0.4708	0.3201	-1.1392
							-1.8153
b0	-0.799		0.799				
b1	-0.1896	P(y=0)	0.1896	P(y=1)			
b2	-0.3802		0.3802				
b3	0.1913		-0.1913				

Table 6. Confusion Matrix

		Predicted Label	
		Offensive	**Not Offensive**
Actual Label	**Offensive**	TP	FN
	Not Offensive	FP	TN

Accuracy measures the percentage of inputs in the test set that the model correctly labelled either as hate speech or non-hate speech. It is calculated as in Equation 5.

$$\text{Accuracy} = \frac{True\ Positives + True\ Negatives}{N} \quad (5)$$

Precision is the ratio of correctly classified documents to the total number of documents classified under a particular category. Precision is a measure of false positives calculated as in Equation 6.

$$\text{Precision} = \frac{True\ Positives}{True\ Positives + FalsePositives} \quad (6)$$

Recall is defined as the number of correctly classified documents among all documents belonging to that category. Recall is a measure of false negatives calculated as in Equation 7.

$$\text{Recall} = \frac{True\ Positives}{True\ Positives + FalseNegative} \quad (7)$$

F1-Score is mean of precision and recall calculated as in Equation 8.

$$\text{F-1Score} = 2 \times \frac{Precision * Recall}{Precision + Recall} \quad (8)$$

In order to measure the accuracy of the system, an accuracy score metric is used. This metric compares the predicted result with actual results. After testing the model, this system achieves 95% accuracy. The confusion matrix for proposed regression model is shown in Figure 3.

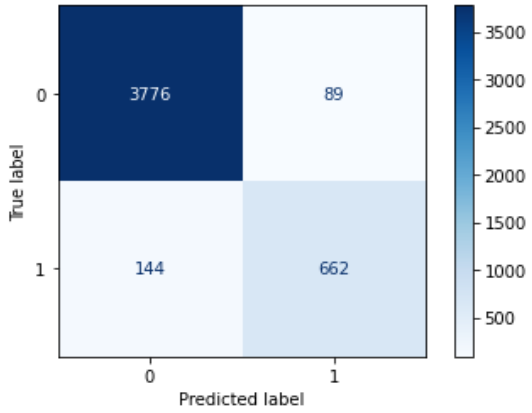


Figure 3. Confusion Matrix

4.5. Comparative Study

The comparative study results obtained using the Logistic Regression and Naïve Bayes classifier are displayed below Table 7. This system also provided the resulted confusion matrix. The score of accuracy of the model are the following:

Table 7. Comparison Table

	Accuracy	**Precision**	**Recall**	**F1- Score**
LR	0.95	0.98	0.97	0.97
NB	0.91	0.91	0.99	0.95

This is comparative study shows that the prediction accuracy of Logistic regression is higher than another prediction algorithm in the offensive speech detection of tweets data on dataset. Logistic Regression is the most accurate algorithm having classified the sample with 95%. Figure 4 describes the accuracy function for two comparison models with chart.

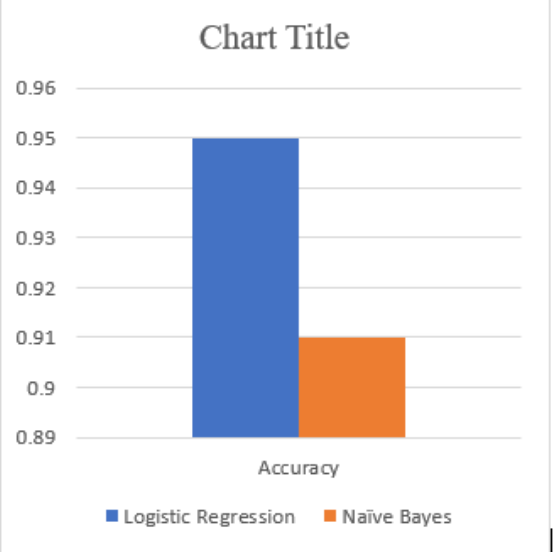


Figure 4. Comparison Chart

5. CONCLUSION

A simple machine learning model based on logistic regression method is presented. The problem of offensive speech detection is solved by using proposed model. Experimental results show that the proposed method can efficiently detect the offensive speech in Twitter. The offensive speech detection for other social media platform (Facebook, Instagram, Twitter, etc.) will conduct in future.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my Pro-Rector, Dr. Myint Myint Khaing for giving me a golden opportunity to do this thesis.

I would like to express my sincere thanks to Pro-Rector Dr. San San Tint for giving me the chance to write journal, journal of Information Technology, Research and Innovation (JITRI-2022), VOLUME-2, ISSUE-1 at University of Computer Studies, Mandalay.

REFERENCES

[1] ORIOLA, O. and KOTZÉ, E. "*Evaluating Machine Learning Techniques for Detecting Offensive and Hate Speech in South African Tweets"*.IEEEAccess.2020.

[2] Mossie, Z.; Wang, J.H. "*Social network hate speech detection for Amharic language"*. In Proceedings of the 6th International Conference on Computer Science and Information Technology, Copenhagen, Denmark, pp. 41–55, 28–29 April, 2018.

[3] T. Pedersen, "*Duluth at Semeval-2020 task 12: Offensive tweet identification in English with logistic regression*," Proceedings of the Fourteenth Workshop on Semantic Evaluation, Barcelona, December, 2020.

[4] https:// chatbotsmagazine.com/machine-learning-neural-networks-and -algorithms-5c0711eb8f9a

[5]https://www.sciencedirect.com/topics/computer-science/logisticregression#:~:text=Logistic%20regression%20is%20a%20process,%2Fno%2C%20and%20so%20on.

[6]https://www.kaggle.com/datasets/mrmorj/hate-speech-and-offensive-language-dataset

Authors

Biography

Ei Phyoe Hein she received the B.C.Sc (Hons) degree from University of Computer Studies (Meiktila) in 2020. Now she is a Tutor of Faculty of Information Science in University of Computer Studies (Pinlon). Her main interests in Database Management System and Deep Learning.

Ei Ei Thu she received the Ph.D (IT) from University of Computer Studies (Mandalay) in 2018. Now she is a Lecturer of Faculty of Computer Science in University of Computer Studies (Pinlon). Her main interests in Software Engineering and Machine Learning.

MOVING OBJECT TRACKING USING KANADE-LUCAS-TOMASI (KLT) ALGORITHM

Myo Min Paing[1], Dr. Myo Min Hein[2] and Dr. Bawin Aye[3]

[1,2,3]Department of Computer Science, Higher Education Centre (Pyin Oo Lwin)

[1]myominpaing351@gmail.com, [2]myominhein48@gmail.com, [3]bawinaye@ieee.org

Abstract

There are many computer vision technologies that we are using in many aspects of our everyday life. Object tracking technology plays a vital role in computer vision field. Object tracking technologies are used in many applications, such as robotics, traffic monitoring, vehicle tracking and more. It can also be applied to advanced military technologies such as guided missile, anti-tank missile, air-to-surface missile, surface-to-air missile and switchblade, etc. Object tracking algorithm finds the position of the interested object in every frame of the video sequence. In this paper, moving object tracking system using KLT algorithm was proposed. The KLT algorithm extracts corner features of the object and tracks the object by calculating the displacements of the extracted features. By using this algorithm, the moving object was able to be tracked with the accuracy of 90%.

Keywords

Corner Feature, Displacement, Extract Feature, KLT, Object Tracking

1. INTRODUCTION

Object tracking plays a vital role in computer vision field. Object tracking is finding the position of the interested object in every frame of the video. It is also used in deep learning to track a particular object or multiple objects by training the algorithms with some datasets.

Generally, there are three main types of Object tracking methods: point-based, kernel-based and silhouette-based tracking [1]. Point-based tracking is tracking object by means of feature points extracted from the Region of Interest (ROI), using feature extraction algorithms. The general method of kernel-based tracking involves calculating the non-stationary object, which is represented by the embryonic object region, from one frame to the next [1]. Silhouette tracking is done by using an object model represented by color hologram, object edges or contour. Among these methods, point-based tracking method was used in the system presented in this paper. Minimum eigenvalue algorithm, also known as Kanade-Lucas-Tomasi (KLT) algorithm, is used here for extracting the feature points of the interested object and for tracking it by means of the extracted features.

2. AIM

The aim of the research paper is to track moving object by using KLT algorithm.

3. RELATED WORK

In recent years, object tracking technologies have become crucial in many areas. Many researchers have researched for the improvement of object tracking algorithms, and many developers have developed object tracking systems to track single or multiple objects by using various methods.

Mr. Pramod R. Gunjal, Miss. Bhagyashri R. Gunjal, Mr. Haribhau A. Shinde, Mr. Swapnil M. Vanam, Mr. Sachin S. Aher [2] proposed a method for moving object detection and tracking using Kalman filter.

In this system, the target object in each frame are identified by means of color recognition. So, it has drawbacks in illumination changing condition. And then the target object is tracked by the help of Kalman Filter which can only be used in a linear or stable system with noise distributed normally (Gaussian) [3].

Li Tan, Xu Dong, Yuxi Ma and Chongchong Yu [4] proposed a multi-target tracking algorithm based on YOLO. It uses You Only Look Once (YOLO) algorithm to detect the object, Long Short Term Memory (LSTM) to track the detected object and MOT-16 data sets and MSR data set to train and experiment the method. Although accuracy is high, it can only detect and track the pre-trained objects under its trained condition.

M. Sushma Sri [5] presented that KLT algorithm is applied to track the detected object. It uses the Viola-Jones algorithm to detect the human face, and KLT algorithm to extract features from the face and track the face frame to frame by means of the extracted features. In this paper, it is only attempted to track the human face, the static object.

4. METHODOLOGY

In this paper, moving object tracking system using KLT algorithm was proposed.

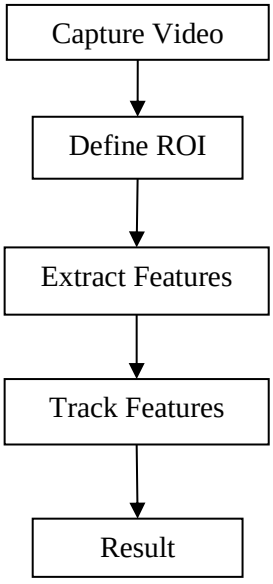


Figure 1. Flowchart of Proposed System

In the flowchart of proposed system, it will capture video first. And then the Region of Interest (ROI) must be defined by clicking the upper left and the lower right corners of the object. After that it will extract features within ROI, and track the extracted features frame to frame.

4.1 Extracting Features

The first step of proposed system is to extract features within ROI manually defined by clicking the upper left and the lower right corners of the object. To do so, the gradient of the image is first calculated. An image gradient is the changes of the image intensity or color in directions. In this paper, sobel gradient operator is used for calculating image gradient. It is a derivative mask and used to detect vertical and horizontal edges of an image. The operator computes the image gradient, convolving the original image with two 3×3 kernels.

The gradient image in x and y directions is got after the original grayscale image have been convolved with mask. It can be methmatically represented as:

$$g(x,y) = f(x,y) \otimes h(x,y) \qquad (1)$$

Where *f(x,y)* is the original grayscale image, *h(x,y)* the mask or filter and *g(x,y)* the gradient image, I_x and I_y, in x and y direction. A grayscale image is a pattern of intensities where the intensity values range from 0 to 255.

-1	0	1
-2	0	1
-1	0	1

-1	-2	-1
0	0	0
1	2	1

Figure 2. Sobel Operators for Vertical and Horizontal Edges

After doing convolution process, corner features are extracted winthin the Region of Interest (ROI) by using minimum eigenvalue algorithm, Kanade-Lucas-Tomasi (KLT) algorithm. Corner is the intersection of two edges, the sudden changes of discontinuities in an image.

Minimum eigenvalue method calculates the eigenvalues of the sum of the squared difference matrix, M.

The sum of the squared difference matrix, M, is defined as follows:

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \quad (2)$$

$$A = (I_x)^2 \otimes \mathrm{w} \quad (3)$$

$$B = (I_y)^2 \otimes \mathrm{w} \quad (4)$$

$$C = I_x I_y \otimes \mathrm{w} \quad (5)$$

Where I_x and I_y are the gradients of the input image, I, in the x and y direction. This $\otimes$ symbol denotes a convolution operation.

A filter coefficient vector is defined using the Coefficients for separable Smoothing filter parameter. To generate a matrix of filter coefficients, w, the block multiplies this vector of coefficients by its transpose.

The minimum eigenvalue of the sum of the squared difference matrix which corresponds to the corner metric matrix is calculated by the block.

4.2 Tracking Features

The KLT algorithm tracks an object in two steps. The first step is locating the trackable features in the initial frame of the video. The second step is tracking the features frame to frame by calculating its displacements. The fundamentals of tracking can be explained by looking at two images in an image sequence. If the first image is captured at time t and the second image at time $t + \delta t$, the incremental time δt will depend on the frame rate (capture rate) of the video camera and should be as small as possible. A higher frame rate allows for better tracking. An image can be represented as function of variables x and y. The variable t is added as the time the image was captured. Thus, any section of an image can be defined by an intensity function $I(x, y, t)$. If a window is defined in an image taken at time $t + \delta t$ as $I(x + \delta x, y + \delta y, t + \delta t)$.

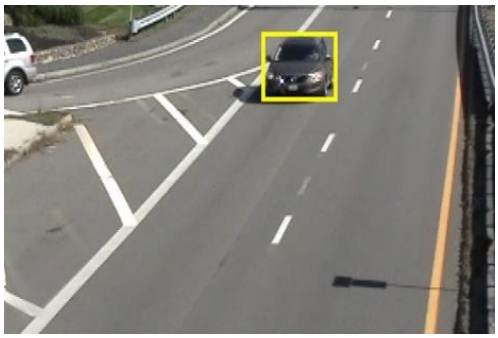
The basic assumption of the KLT tracking algorithm is

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \quad (6)$$

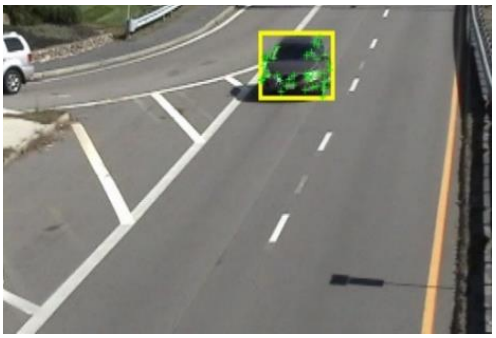
From (6), it is clear that every point in the second window can be obtained by shifting every point in the first window by an amount $(\delta x, \delta y)$. The main goal of traking is to calculate this amount $(\delta x, \delta y)$ which can be denoted as the displacement $d = (\delta x, \delta y)$.

5. EXPERIMENTAL RESULT

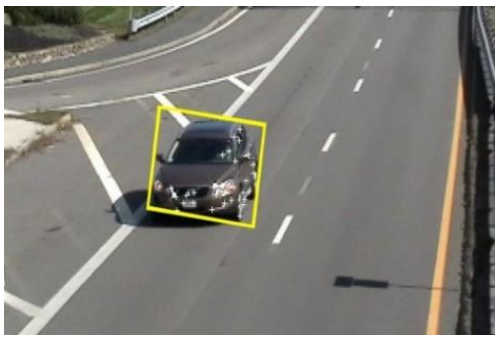
In this section, the effectiveness of proposed method is identified by doing experiments on 20 videos of moving objects and three of them are shown in the following figures. The system was developed in MATLAB R2021a platform.



(a) Defining ROI



(b) Extracting Feature



(c) Tracking

Figure 3. Tracking Moving Vehicle

Figure 3 is the illustration of tracking moving vehicle. First, ROI is manually defined as shown in figure 3(a). Second, the system extracts corner features within ROI as shown in figure 3(b). Then it tracks the extracted features frame to frame as shown in figure 3(c).

(a) Defining ROI

(b) Extracting Feature

(c) Tracking

Figure 4. Tracking a Human in the Train Station

Figure 4 is the illustration of tracking a human in the train station. First, ROI is manually defined as shown in figure 4(a). Second, the system extracts corner features within ROI as shown in figure 4(b). Then it tracks the extracted features frame to frame as shown in figure 4(c).

(a) Defining ROI

(b)) Extracting Feature

(c) Tracking

Figure 5. Tracking Military Vehicle from the Aerial View

Figure 5 is the illustration of tracking military vehicle from the aerial view. First, ROI is manually defined as shown in figure 5(a). Second, the system extracts corner features within ROI as shown in figure 5(b). Then it tracks the extracted features frame to frame as shown in figure 5(c).

Table 1. Accuracy Table

No.	Number of Frames	Tracked Frame	Tracked Index
1.	100	100	1
2.	85	85	1
3.	140	140	1
4.	213	213	1
5.	100	100	1
6.	200	130	0
7.	110	110	1
8.	80	80	1
9.	132	132	1
10.	130	130	1
11.	84	84	1
12.	200	200	1
13.	200	200	1
14.	80	80	1
15.	100	100	1
16.	200	200	1
17.	150	150	1
18.	100	100	1
19.	120	120	1
20.	100	30	0
Total	**2824**	**2754**	**18 (90 %)**

The accuracy of the system is shown in table 1. The system successfully tracked 20 videos with the accuracy of 90%.

6. CONCLUSIONS

In this paper, moving object tracking system using KLT algorithm was proposed. First, the system captures the image or video. And then, the Region of Interest must be manually defined in this system. If the Region of Interest is defined by the system using deep learning algorithms, there will be limitations about which objects the system will detect and track. In the system proposed in this paper, there is no limitation about objects that the system will track. After that, it extracts corner features using minimum eigenvalue algorithm and tracks the object frame to frame by calculating the displacements of the extracted feature points. It can track the stationary or non-stationary object with moving or non-moving camera. It successfully tracked 20 testing videos with the accuracy of 90%. Using this system, many security and surveillance systems can be developed. It can also be used in advanced military technology like tracking of the enemy's military vehicles or buildings from the aerial view.

ACKNOWLEDGMENT

We would like to express our sincere thanks to Dr. Soe Win Myint, Head of Department of Computer Science, and Dr. Zar Nay Linn who have, as always, given us their support throughout writing of this paper.

REFERENCES

[1] S. R. Balaji and Dr. S. Karthikeyan, A SERVEY ON MOVING OBJECT TRACKING USING IMAGE PROCESSING, Sathyabama University, Chennai, India, 11th International Conference on Intelligent Systems and Control (ISCO) 2017.

[2] Mr. Pramod R. Gunjal, Miss. Bhagyashri R. Gunjal, Mr. Haribhau A. Shinde, Mr. Swapnil M. Vanam and Mr. Sachin S. Aher, MOVING OBJECT TRACKING USING KALMAN FILTER, Amrutvahini College of Engineeirng, Sangamner, India, International Conference on Advances in Communication and Computing Technology (ICACCT) 2018.

[3] Barga Deori and Dalton Meitei Thounaojam, A SERVEY ON MOVING OBJECT TRACKING IN VIDEO, National Institute of Technology, Silchar, India, International Journal on Information Theory (IJIT) 2014.

[4] Li Tan, Xu Dong, Yuxi Ma and Chongchong, A MULTIPLE OBJECT TRACKING ALGORITHM BASED ON YOLO DETECTION, Beijing Technology and Business University, HaiDian District, Beijing, China, 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI) 2018.

[5] M. Sushma Sri, Object Detection and Tracking using KLT Algorithm, Osmania University College of Engineering, Hyderabad, India, IJEDR 5th International Conference on Computer and Communications 2019.

Authors

Biography

Myo Min Paing received M.C.Sc. degree from Higher Education Center (HEC), in 2018. Now he is a Ph.D. candidate of Computer Science Department, Higher Education Center (HEC).

Dr. Myo Min Hein earned his Master's Degree from Russian State Technological University (MEPHI), (Russian Federation) in 2010. He obtained his Ph.D. (Computer Science) from HEC in 2018. Now he is an assistant lecturer of Department of Computer Science in HEC.

Dr. Bawin Aye earned his master degree from National Research University of Electronic Technology (MIET), (Russian Federation) in 2006. He also obtained his Ph.D. degree from National Research University of Electronic Technology (MIET), (Russian Federation) in 2010. Now he is a lecturer of Department of Computer Science in HEC. He is also a member of the IEEE Computer Society.

FACE PHOTO TO SKETCH CONVERSION SYSTEM BASED ON GENERATIVE ADVERSARIAL NETWORK

Hayma Thida Kyaw[2], Dr.Myo Min Hein[2], Dr.Ye Wint Mg Mg[3]

[1,2,3]Department of Computer Science, Higher Education Centre (Pyin Oo Lwin)
[1]haymathidakyaw@gmail.com, [2]myominhein48@gmail.com, [3]linwai46@gmail.com

ABSTRACT

Visual media has long been a primary means of human communication. While there are several methods for transforming a face sketch into a photo, research on photo to sketch transformation is limited. Learning the mappings between these two various visual styles or domains is required for this assignment.Generative Adversarial Network (GAN) models like Pix2Pix, CycleGAN, and BicycleGAN, to name a few, can be used in a variety of ways to do this. In this article, we express a novel face photo to sketch conversion system by using Pix2PixGAN model. We use the advantages of Pix2PixGAN, as well as a feature-level loss, to ensure the high quality of the fake face sketch from photo. Experimental results are presented compare with traditional image processing technique.

KEYWORDS

Convloutional Neural Network, Face Recognition, GANs, Photo to Photo Matching, Sketch Identification

1. INTRODUCTION

The analysis of visual imagery frequently makes use of a deep neural network called a convolutional neural network (CNN or ConvNet). The convolutional neural network performs similar operations to a conventional feed-forward neural network, with the exception that the operations in its layers are spatially structured and the connections between layers are sparse (and well planned). Convolution, pooling, and ReLU are the three kinds of layers that are frequently found in a convolutional neural network. Each layer in the convolutional network is a 3-dimensional grid structure, which has a height, width, and depth [1].

Convolutional networks were influenced by natural processes in that their neuronal connectivity pattern mirrors the structure of an animal's visual cortex. CNNs need much less pre-processing than other image classification methods, in comparison. In those other, the network recognizes filters that were manually programmed in traditional algorithms, the network picks up the filters that were manually programmed in conventional algorithms. The freedom here is from human effort and prior knowledge. Some of the applications include recommender systems, medical image analysis, image classification and natural language processing, image and video recognition [2].

In 2014, Ian Goodfellow created the machine learning system known as the generative adversarial network (GAN). In the context of the image estimating paradigm, CNNs and GANs can be viewed as deep learning techniques in general. Convolutional neural networks (CNNs) were demonstrated to be effective in image-to-image intensity projections in addition to recognizing, classifying, and retrieving visual patterns. Generative adversarial networks (GANs) employ CNNs as generators, and estimated images are categorized as true or false using a separate discriminator network.

GANs are an ingenious method for learning a generative model which frames the issue

like a supervised learning problem with two sub-models—the generator model, which we train to create new instances, and the discriminator model, which tries to categorize real or fake.

In this article, we provide a novel method for generating a sketch from a photo using a generative adversarial network. The Pix2Pix model, CGAN, or a conditional GAN, in which the creation of the result output image is reliant on an input, was employed in our suggested system. An input image give to the discriminator and a target output, and it must decide whether the output is a realistic conversion of the input images. By using deep learning techniques to reduce image noise, we can produce the correct output image.

2. AIM

The aim of the system is to generate the output sketch from the photo of the suspected person by using Pix2Pix GANs model.

3. RELATED WORK

In order to train the Conditional GAN on input-output image pairings, the authors [1] employ the generator's U-Net architecture, which consists of an encoder and decoder network as well as a specially designed discriminator that is put forth in the study. The generator includes two parts: a decoder layer and an encoder layer that use down-sampling to construct lower-dimensional representations of the given sketches, or X. Between the decoder's layer it and the encoder's layers 8-i, the generator has skip connections. Here, they take a drawing as input and attempt to create an image for it while comparing how well it resembles the desired image. Accordingly, they created their loss function.

The authors [2] suggested that the paper includes both image production and image recognition components. U-Net networks with skip connection generators are used in current architectures. The sketch's warped structures are kept in the final image since they have a skip link in the generator part. During testing, they also manually entered the sketch's name into the class, which we did not do.

Integrating GAN with CNN for Face Sketch Synthesis is presented by the authors [3]. In this article, they provide a thorough deep learning based model for face recognition. In this scenario, a U-Net generator creates an input photo by learning a nonlinear mapping relationship between a face shot and a sketch image from a training dataset. For the network, the generated sketch is used as a coarse estimation image with noise and distortion. In comparison to previous procedures, their suggested method produces more delicate identical synthesized outputs.

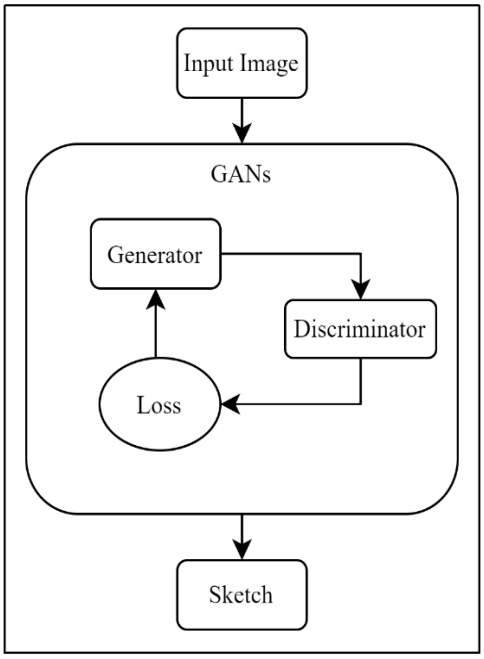
We proposed an adversarial model for generating high-quality face photos from sketches in the paper [4].

We have introduced a feature-level loss to ensure higher similarities between the synthesized photos and the ground-truth photos, in addition to the commonly used discriminators for evaluating the authenticity of the generators. It is important to note that during network training, performance may vary between epochs. Given the limited data, they plan to investigate different pre-training methods in the future to improve the convergence of the sketch-to-photo transformation model. Furthermore, they want to investigate the possibility of generating photos from more difficult forensic sketches, which can differ significantly from photos [4].

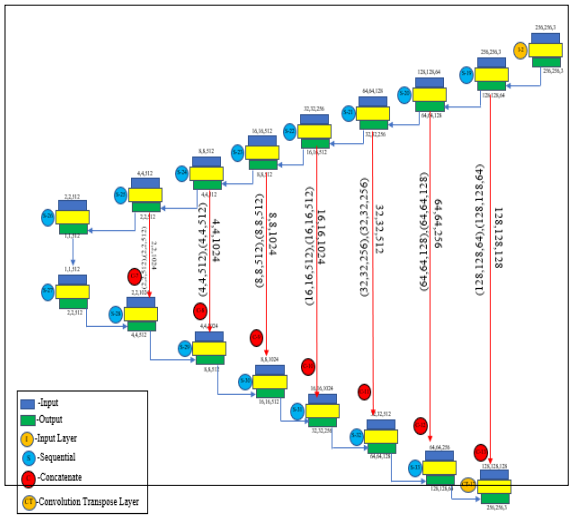
4. METHODOLOGY

The algorithm recognition of hand-draw sketches is known as sketch recognition. It comprises of a variety of distinct algorithms, some created for use in particular fields and others for broad recognition. In this article, we proposed a new algorithm to convert the photo of a

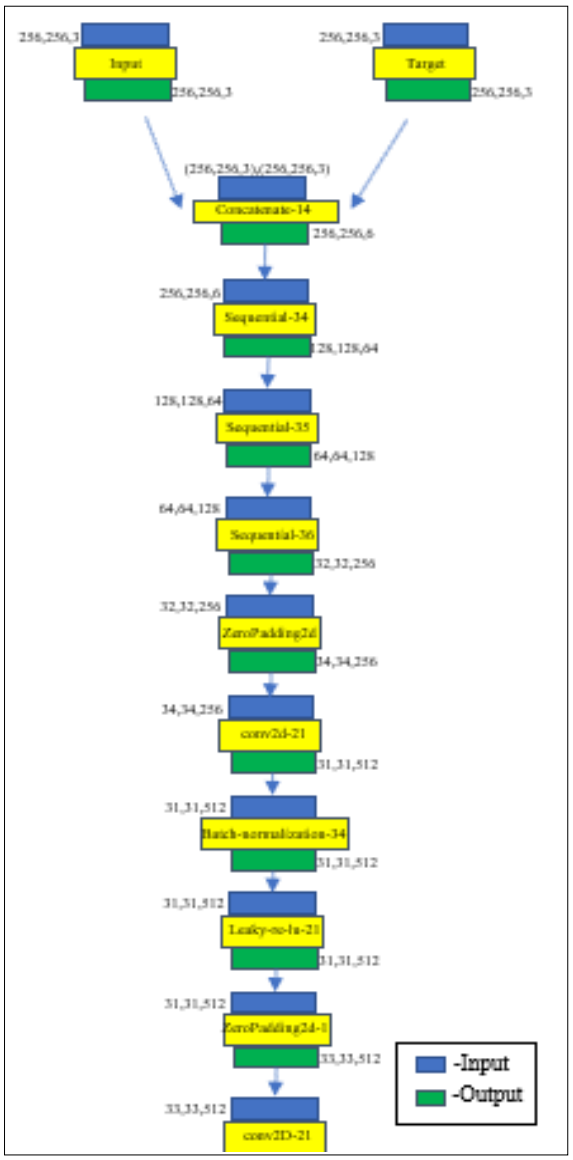
suspect image to a sketch image using Generative Adversarial Network. The photo of the suspected person is given as input to Pix2PixGAN model. After that, Pix2PixGAN model generated sketch. The proposed system's block illustration is as shown in figure1.



(a) Block Diagram of Prpopsed System



(b) Generator of Proposed System



(c) Discriminator of Proposed System

Figure 1. Proposed System

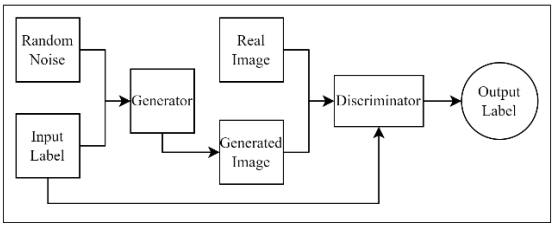


Figure 2. Flow chart of Pix2PixGAN

As in Pix2PixGAN mode, there are two networks—a generator model and a discriminator model—that function together. In the generator model, the input

image's random noise and the ground truth image as input label are passing through to the U-Net generator. Then, this generator generated the fake (generated) image. After that, the fake (generated) image, the real input image and the ground truth image (input label) are passing through to the Discriminator model. And then, this Discriminator detects the input image which is real or fake. Finally, this Discriminator model predicts the output label. The Discriminator model is binary classification model.

4.1 Computer Vision

Computer vision is a branch of artificial intelligence (AI) that allows systems and computers to support and extract useful information from digital photos, movies, and other graphical inputs, act on that information, and make recommendations. In the preceding four decades, hundreds of smart and creative minds have worked on this unexpectedly difficult issue, but we are still far from being able to create a general-purpose "seeing machine." Technically, machines retrieve, process, and interpret visual data using specialized software algorithms [2].

Computer vision (CV) encompasses a variety of tasks, including object identification, image classification, and more. Image classification: In 2012, AlexNet, which effectively increases the depth of LeNet and employs various strategies like ReLU and Dropout, won ILSVRC with five convolutional layers and three fully connected layers. Object detection: Two-stage and one-stage approaches are the two categories into which we may divide the existing deep learning-based detectors. R-CNN, Fast R-CNN, Faster R-CNN, CoupleNet, Light-Head R-CNN, and other two-stage approaches, which first produce candidate regions using CNN or conventional methods, and then categorize them [2].

4.2 Machine Learning

Machine Learning is a field of artificial intelligence that uses algorithms and statistical models to enable computer systems to learn and improve from experience, without being explicitly programmed. It involves training a model on a dataset and using it to make predictions or decisions on new data [2].There are various approaches based on the type and volume of data depending on the nature of the business problem being addressed. It can be classified into three types: supervised learning, unsupervised learning, and reinforcement learning [2].

4.3 Deep Learning

A neural network which has three or more layers is essentially what deep learning, a class of machine learning, is all about. These neural networks make an effort to mimic how the human brain functions, however they fall well short of being able to replicate it, so that it can "learn" from vast amounts of data. Even though a single-layer neural network can approximate, additional hidden layers will be able to aid in optimization and accuracy [3]. Figure 3 illustrates the connections between artificial intelligence, deep learning and machine learning.

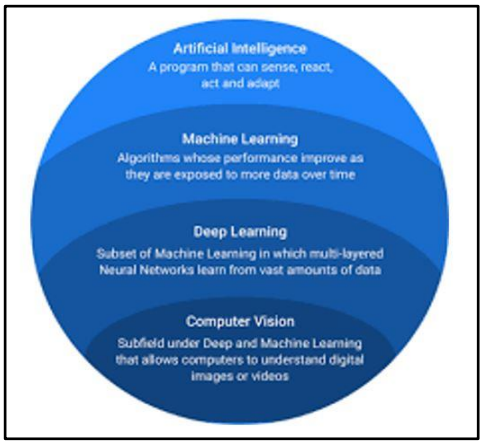


Figure 3. Relationships between AI, ML and DL

For many computer vision applications, including semantic segmentation, instance segmentation, anomaly detection, object recognition, and others, deep learning is a significant method. Deep learning is capable of performing a number of tasks that humans are not capable of, such as the

board game Go and the categorization of objects. Picture segmentation, in particular, has received a lot of attention and has a variety of practical applications, including driverless vehicles, medical image diagnosis, aerial image segmentation, and more [3].

4.4 Pix2PixGAN Model

A Generative Adversarial Network (GAN) model called Pix2Pix was created for all-purpose image-to-image translation. In their article titled "Image-to-Image Translation with Conditional Adversarial Networks," which was expressed at CVPR in 2017 [8], Phillip Isola et al. introduced the method. The GAN architecture is made up of two models: a generator model that generates new, feasible photos and a discriminator model that determines whether the images are real (from the dataset) (generated). While the former does not update it, the discriminator model does. As a result, the two models are developed concurrently in an adversarial process in which the generator attempts to fool the discriminator while the discriminator attempts to detect the fake images [4].

The Pix2Pix model is a conditional GAN, or CGAN, that generates an output image in response to an input image. The discriminator model is given an input photo and a target output and must decide if the target is a realistic conversion of the source image.

The generator is trained using adversarial loss, which motivates it to produce believable images in the target domain. Additionally, the generator is updated using the loss, L1, between the created image and the desired output image. This additional loss motivates the generator model to produce accurate translations of the original image.

Several image-to-image translation tasks have been evaluated, including converting the maps to the satellite images, black-and-white photographs to color photographs, and product sketches to product photographs.

4.5 U-Net Generator

The generator is an encoder-decoder network, as we know (first a series of down sampling levels, followed by a bottleneck layer, and then a series of upsampling layers). The E-D network used the "U-Net" architecture with skip connections. The spatial resolution of the layers being connected already matches each other, therefore no scaling, projections, etc. are needed for the U-Net skip connections. We employ the U-Net generator in the system we've proposed [4].

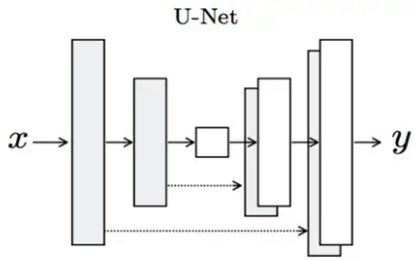


Figure 4. Architecture of U-Net

5. EXPERIMENTAL RESULT

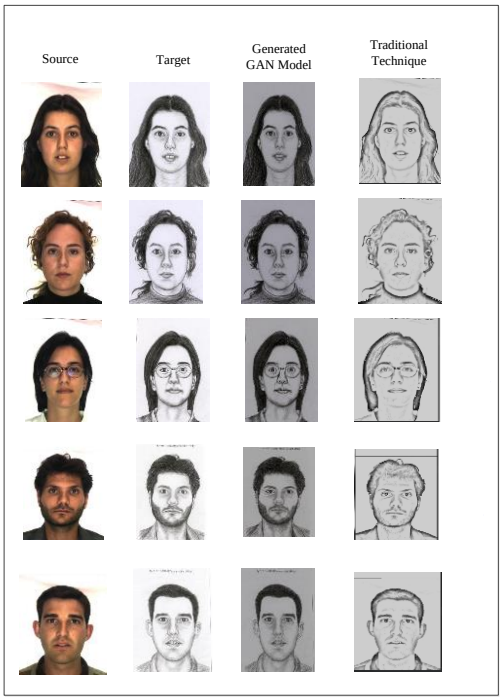


Figure 5. Experimental result of Photo to Sketch

There are 188 sketch images in the dataset. 151 images are used for training and 37 images are used for validation data. After training the model, 88 images are used to generate skectch images. The experimental result is based on the 88 testing dataset.

6. EVALUATION

The correctness of a model can be evaluated in different ways. There are several loss functions in machine learning algorithms. Loss functions are based on the problem statement to which machine learning is being applied. In our proposed system, MAE, MSE, RMSE, Correlation Coefficient, and PSNR functions are used to calculate the loss of the system.

The Mean Absolute Error, or MAE, is one of several metrics for summarizing and evaluating the quality of a machine learning model. The difference of two continuous variables are referred to statistically as Mean Absolute Error (MAE). The formula is as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \quad (6.1)$$

The mean squared error is the average squared difference between the data set's original and anticipated values (MSE). It computes the variance of the residuals.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2 \quad (6.2)$$

Root Mean Squared Error is the square root of Mean Squared Error. It computes the residuals' standard deviation.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2} \quad (6.3)$$

A correlation coefficient is a number ranging from -1 to 1 that indicates the strength and the direction of a relationship between variables. The formulas produce a value between -1 and 1, with 1 indicating a strong positive relationship, -1 indicating a strong negative relationship, and zero indicating no relationship at all. It can be calculated as follow:

$$CorCoff = \frac{\sum[x(i)-mean(x)][y(i)-mean(y)]}{\sqrt{\sum[x(i)-mean(x)]^2 \sum[y(i)-mean(y)]^2}} \quad (6.4)$$

Peak Signal to Noise Ratio is known as PSNR. This ratio is frequently used to compare the original and compressed images' quality. The quality of compressed or rebuilt image improves with increasing PSNR. PSNR is determined as:

$$PSNR = 10\log_{10}\frac{R^2}{MSE} \quad (6.5)$$

Where n stands for the number of observations, y_i denotes the actual value, x_i denotes the predicted value, mean(x) denotes the mean of all x values, mean(y) denotes the mean of all y values, and R=255.Utilizing the aforementioned formula, we compare the testing outcomes for image processing technique called Haar-Cascade detetction method, Gaussian smoothing and Bitwise NOT operation and Pix2PixGAN model as follow:

The comparison chart of testing using image processing technique and GANs are as shown in following figure.

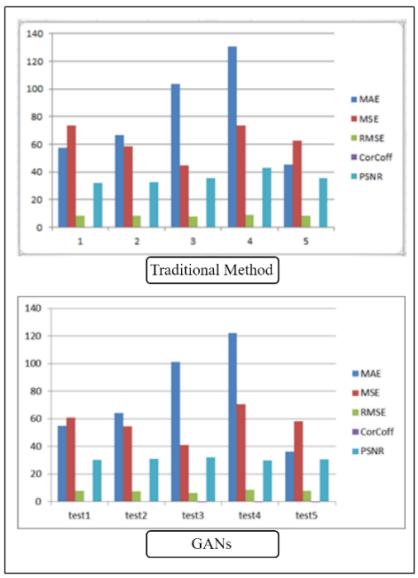


Figure 6. Comparison Results of Traditional Technique and Pix2PixGAN

Table1. Evaluation Result for Pix2PixGAN Model and Traditional Image Processing Technique

Method / Image	MAE	MSE	RMSE	CorCoff	PSNR
1	54.68	60.92	7.8051	0.3826	30.2832
2	64.12	54.44	7.3783	0.0121	30.7716
3	101.32	41	6.4031	-0.2358	32.0029
4	122.24	70.64	8.4048	-0.478	29.6402
5	36.08	58.4	7.6419	-0.0989	30.4667

Traditional Method

Image / Method	Test1	Test2	Test3	Test4	Test5
MAE	57.64	66.87	103.43	130.5	45.52
MSE	73.67	58.55	45	73.64	62.4
RMSE	8.324	8.367	7.583	9.2134	8.5186
CorCoff	0.4286	0.2101	-0.3285	-0.746	-0.789
PSNR	32.2823	32.7716	35.67	43.2376	35.5648

GANs

7. CONCLUSION

In this article, we proposed face photo of the suspect image to the sketch image conversion system by using Pix2PixGAN model. We plan to make face photo to sketch conversion system for criminal case. The Pix2Pix GAN, or Generative Adversarial Network, is a learning model for deep convolutional neural networks for image-to-image translation tasks. The Pix2Pix GAN has been being trained to accomplish the image-to-image translation tasks, such as converting the maps to the satellite photographs, the black-and-white photographs to the color photographs, and the product sketches to the product photographs. The needed line with face in image processing technique is more than in our proposed system. So, proposed model is better than traditional face photo to sketch conversion system. According to the result, our proposed system is a well face photo to sketch conversion algorithm for face sketch recognition.

ACKNOWLEDGMENT

We are thankful to express my sincere thanks to Dr. Soe Win Myint, Head of Department of Computer Science, and my honorable teachers in Higher Education Centre, for giving this opportunity to read this paper and helping me throughout this research paper.

REFERENCES

[1] Lisa Fan, Jason Krone, Sam Woolf, Sketch to Image Translation using GANs, Project Report at Stanford University, 2016.

[2] Nischal Tonthanahal1, Sourab B R, Dr Sharon Christa, Forensic Sketch-to-Face Image Transformation Using CycleGAN, Dept. of Information Science and Engineering, Dayananda Sagar College of Engineering, Bengaluru.

[3] Shikang Yu, Hu Han, Shiguang Shan, Antitza Dantcheva, Xilin Chen, Improving Face Sketch Recognition via Adversarial Sketch-Photo Transformation, FG 2019 - 14th IEEE International Conference on Automatic Face and Gesture Recognition, Lille, France ,May 2019.

[4] Vinay,Aditya Lokesh, Vinayaka R Kamath,K N B Murtht and S Natarajan, Sketch to image using CNN and GAN, CPRIM PES University Bengalura, India, 2019 IEEE.

Authors

Biography

Hayma Thida Kyaw received Master of Computer Science (M.C.Sc.) degree from Higher Education Center (HEC), in 2019. Now she is a Ph.D. candidate of Computer Science Department, Higher Education Center (HEC).

Dr. Myo Min Hein earned his Master's Degree from Russian State Technological University (MEPHI), (Russian Federation) in 2010. He obtained his Ph.D. (Computer Science) from HEC in 2018. Now he is an assistant lecturer of Department of Computer Science in HEC.

Dr. Ye Wint Mg Mg earned his Master's Degree from Moscow Engineering Physics Institute (MEPHI), Moscow in 2009. He obtained his Ph.D. (Computer Science) from HEC in 2015. Now he is an assistant lecturer of Department of Computer Science in HEC.

MULTICLASS SKIN LESION CLASSIFICATION USING DEEP LEARNING

Kyi Pyar Zaw[1], Atar Mon[2], Theint Theint Soe[3], May Zin Tun[4]

[1,2] Department of Electronic Engineering, University of Technology (Yatanarpon Cyber City)

[1]kyipyarzaw89@gmail.com, [2]atarmon@gmail.com, [3]theint.soe@gmail.com, [4]mayzintun954@gmail.com

ABSTRACT

Skin cancer is one of the most lethal of all cancers. It also occurs while the tissue is revealed to light from the sun, mainly due to the rapid development of skin cells. Both image processing and deep learning are used in the technique for successful treatment of skin cancer. This Proposed System is to visualize a reliable assistant in indicating skin lesion areas of patients to medical doctors. The system will be developed for the skin lesion classification using transfer learning method with VGG16 architecture and will provide the most accurate results for skin lesion to medical doctors. ISIC dataset is used to train the data with the VGG16. The system implementation is performed for balanced and unbalanced nature of ISIC dataset. This system achieves accuracy of 83% in unbalanced dataset and 91% in balanced dataset.

KEYWORDS

Skin Cancer, VGG16, Unbalanced, Balanced

1. INTRODUCTION

Nowadays, computer-aided diagnosis (CAD) systems have become a necessity to diagnose and evaluate medical images. The computer-aided diagnosis systems have aided the principal element in mammogram to the breast cancer observation in the United States (US). Moreover, it is essential in medical fields in imaging and diagnosis radiology. The prediction ability of diseases in early stages is directly proportional with the efficient determination and treatment of kinds of cancers. There were over 14 million events by over 8.2 million deaths as the worldwide cancer reporting at 2012. This statistical analysis provides the cancer as the main occurrence in death in humans. Skin cancer is a common kind of cancer and this mainly happens at skin which had been exposed in sunlight even though cancer presents at every portion of the body. The computer-aided diagnosis systems have the ability of the utilization of the images at skin lesion no assessment of every relevant and adaptable information for the observation of initial diagnosis. Melanoma is the most common type of skin cancer occurred at humans and this type of cancer happens pigmented ticks on moles at upper surface of the skin. The occurrence of melanoma is the abnormal happening in the melanin-providing cells that provide coloration at the skin.

If this disease is not diagnosed at the earlier stages, melanoma develops and continues in spreading among the outer surface layer of the skin before the penetration of deep surface layers at where it coordinates with the lymph vessels and blood. If the skin detection is performed at the earlier steps, there will have the higher probability for treatment in skin lesion. Otherwise, there will have the lower probability for treatment in skin lesion. This is expensive for the skin cancer detection at the earlier steps. As the lesions at skin is same with each other, it has the difficulty for analysis to determine the skin lesion is benign or malignant. The Stanford University developed the deep learning convolutional neural networks for the outperforming at the dermatologists in the classification of the keratinocyte carcinoma among the benign seborrheic keratosis and benign nevi vs. malignant melanomas. Transfer Learning provides a state-of-the-art prediction outcome in computer vision fields.

2. RELATED WORKS

The authors presented the analysis in the comparison of various algorithms for feature selection and prediction of melanoma [1]. This diagnosis systems were developed with the data collection at real events and the model was constructed by the set of rules in the aid of domain experts. This approach was expensive as the fast image processing and more detailed knowledge for the optimal set of rules development. However, they did not utilize deep learning approaches for this system.

In this paper [2], the authors proposed one-class deep neural network learning and compared with other classifiers performance according to the techniques of Isolation Forest, Gaussian Mixtures, and OC-SVM. According to the evaluation results, the optimum technique for this dataset is Isolation Forest. Isolation Forest is based on the features of anomalies are very distinct from the normal samples. This idea is for the development of an ensemble in isolation trees at where anomalies possess short average path lengths on those trees.

The authors developed the automatic skin lesion analysis, potentially melanoma determination and semantic segmentation [3]. This system applied deep learning methods for the prediction among public available dermoscopic images of skin lesion. The first implementation of deep convolutional neural network architecture was performed among raw images for classification. Moreover, the deduction of edge histogram, multiscale color local binary patterns, and 166-D color histogram distribution through the images was performed and conducted at random forest classifier. Then, the final prediction outcome was produced by the average of the values of the two classifiers. The accuracy results by 80.3% with precision score by 0.81, and Area Under the Curve (AUC) score by 0.69 were achieved. The convolutional-deconvolutional architecture was implemented for the segmentation and the dice coefficient by 73.5% was provided by this segmentation model.

The skin lesion, melanoma is occurring more and more prevalent. Moreover, the chances for the patient can be potentially enhanced by the detection at initial stages. But, the effective and precise determination of this cancer is becoming the challenges according to the difficulty in the categorization among non-melanoma and melanoma skin lesion cancer. Various techniques have been developed for the classification of skin lesion cancers utilizing deep learning. In this paper [4], the authors presented the enhancement in skin lesion determination and classification applying convolutional neural network. The principal goal was to apply only lesion textural information as the input of convolutional neural network instead of the whole images. The textural lesion was achieved with the projection of the segmented object and texture components produced by the application of the multi-scale decomposition.

3. THEORICAL BACKGROUND

3.1. Skin Lesion

Skin Lesion is the irregular development of skin cells which usually occurs on skin outer layer exposed with the sunlight. However, this skin lesion cancer may develop at skin layers with no exposure with the sunlight. Three kinds of skin lesion cancers are squamous, melanoma, and basal. The most popular and common kinds of skin lesion cancers are squamous and basal. They initially occur at the outermost layer of skin and this layer relates with the exposure of the sunlight. The determination of the kind of skin lesion cancer is performed according to the places at skin lesion occurs. The person happens the basal type skin lesion when the skin cancer develops at basal skin cells. Melanoma initiates if this skin cells happens the skin color to occur the cancerous.

Basal cell carcinoma: Basal cell is the most occurrence kind in skin cancer and it usually happens at the person who possesses the fairy skin [5]. This type of skin cancer occurs at the inner portion of the basal cell layer, the epidermis. This skin cancer looks as the flesh-colored round development, a pinkish patch of skin, or pearl-like bump. It grows at the exposure portions of the sunlight such as the head, neck, and face. It continually develops with slowly and this has rare cases to basal cell cancer for distribution in other portions in the body. However, there is no treatments, this cancer cell may develop to nearest skin surfaces and penetrates into the other tissues or

the bone below the skin. When there is no complete removal, basal cell may return at the same portion at the skin. The person who possesses basal skin lesion has the chance for achieving new skin cancer cells at another portions. As the deep development of this cancer cells, the initial diagnosis and treatment to basal cell carcinoma is very essential. Figure 1 presents basal cell carcinoma.

Figure 1: Basal Cell Carcinoma

Squamous cell carcinoma: Squamous cell is the second most occurrence kind in skin cancer and the person who possesses light skin that are most likely to grow squamous cell [6]. These cells are flattened and occurs at the outer portion of the epidermis that are shed like new skin cancer cells type. This kind of skin lesion cancer grows at the person who possesses the dark-colored skin and this may appear as red-colored bump, scaly sore or patch which returns and heals. This kind of skin lesion cancer grows at the exposure body portions of the sunlight as the ears, neck, face, backs of the hands and lips. This becomes at the skin in the genital portion. They may deeply develop at the skin happening disfigurement and damage. Squamous cell can be prevented through developing deeply and distributing to other portions of the body by the earlier treatment and diagnosis. Figure 2 presents squamous cell carcinoma.

Figure 2: Squamous Cell Carcinoma

Melanoma: Melanoma is the most serious and dangerous skin cancer cell as it possesses the tendency for distribution. This type of skin cancer cells grows through melanocytes, the pigment-producing cells occurred at the epidermis. They make brown pigment-producing cells known as melanin that produces the skin brown or tan color. These cells act like the natural sunscreen of the body preventing the deep surfaces of the skin through some dangerous effects of the sun [7]. This type of skin cancer develops at these cells, this may appear at the trunk or face of the infected men and this may grow in the lower legs of infected women. Figure 3 presents melanoma.

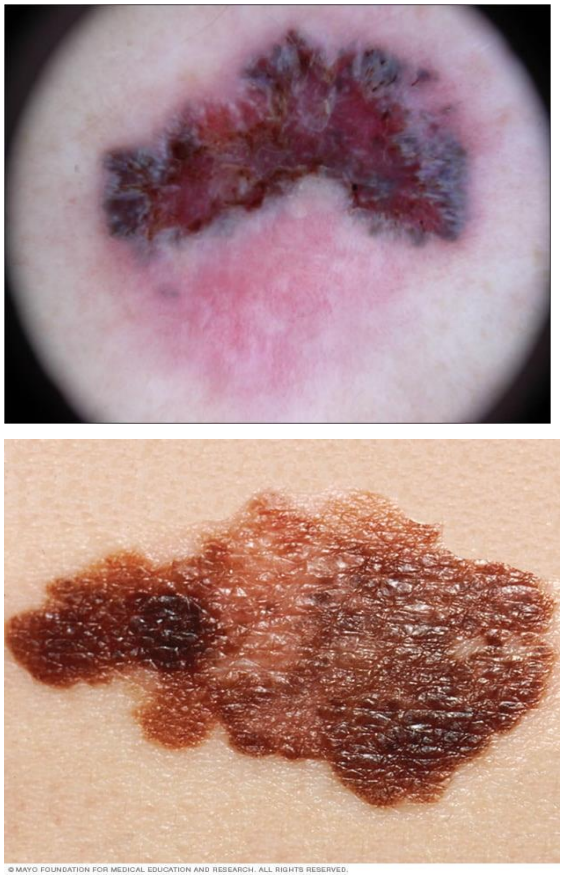


Figure 3: Melanoma

3.2. Random Sampling

Random sampling is one kind of probability sampling in that every instance possesses the equality in probability for the selection. A random selected instance is an unbiased description for the total population. If the instance does not represent the population, the variation becomes a sampling error. Random sampling is a method for choosing each participant or a subset of the population for providing the statistical inferences from them and estimation the characteristics of the whole population. It can be utilized as a data reduction method as it allows a large data set to be represented by a much smaller random data instance (or subset).

3.3. Oversampling

Over-sampling is a balancing approach for asymmetric datasets with maintaining all of the data in the majority class and increasing the size of the minority class by adding the instances to it. In another words, the duplication of samples to the minority class in the training dataset and this sampling can occur overfitting to some models as learning algorithms focus on replication of minority instances. Simple random sample with replacement (SRSWR) of size is similar to simple random sample without replacement (SRSWOR), except that each time an instance is chosen from dataset, it is recorded and then replaced, i.e., after an instance is drawn, the back placement is performed at dataset so that the choosing may be done again. This sampling is appropriate in such conditions as there is no enough information. One class is the majority, or abundant, and the other class is the minority, or rare. The number of rare instances is increased in this sampling. Figure 4 shows oversampling.

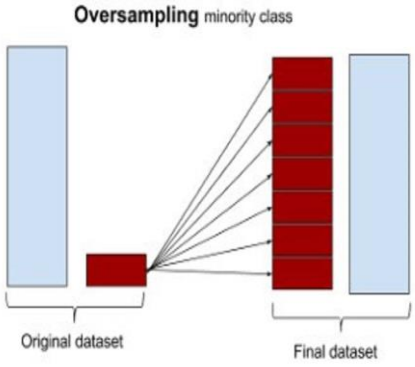


Figure 4: Oversampling

3.4 VGG 16 Architecture

VGG16 is a convolution neural network architecture that was applied for achieving the winner at ILSVR(Imagenet) competition at 2014. This is an update and excellent vision model architecture. The most unique element is that instead of owning various hyper-parameter targeting on owning convolution layers of 3x3 filter by a stride 1 and usually applied same padding and maxpool layer with 2x2 filter by stride 2. This architecture obeys the arrangement of max pool and convolution layers consistently among the whole architecture. At the end, it owns two fully connected layers with softmax function for the outcome. The 16 in VGG16 architecture means that it owns 16 layers with weights. This network is a large network with owning

about 138 million (approximation) parameters. The VGG16 architecture is shown in figure 5.

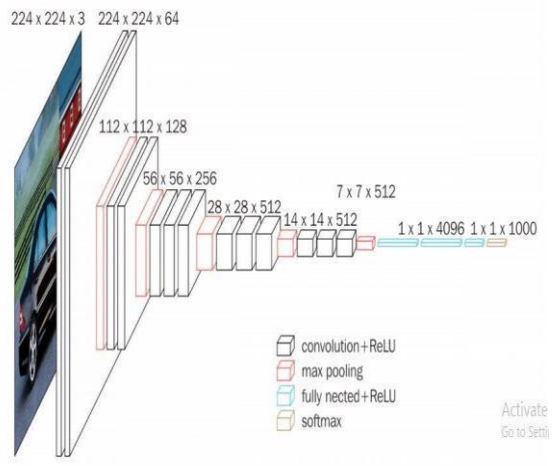


Figure 5: VGG16 Architecture

3.5 Transfer Learning

Transfer learning is the machine learning approach and at which a pre-trained model is reused as the beginning point for a model on an incoming job [8]. For the improvement in generalization with other, what has been observed at a job is exploited by using transfer learning. The weight at a network observed task "X" to a new task "Y" is transferred. The initial and middle layers are utilized and the latter layers are retrained at transfer learning. The labelling data of the job that was early trained is leveraged. Transfer learning provides the best outcomes and the computing effectiveness utilizing small dataset. The knowledge obtained through the previous pretrained model such as features and weights is utilized for continuing the job. Moreover, this transfer learning approaches provide the best and faster performance results than traditional machine learning approaches.

4. PROPOSED SYSTEM

The aim of the research is to develop reliable visualization assistance for medical doctor in skin lesion. This work will be focused on deep learning techniques, and will therefore consider and analyses different type of models for classification. The experiments will be examined by means of various indicators, for example completeness, appropriateness and quality, and will illustrate the Precision, Recall, F1_measure, confusion matrix and accuracy of the used deep learning algorithm. The proposed system design for skin lesion classification system can be described as follow in figure 6.

The workflow of proposed system design is composed of two main parts: training and testing. In all stages, firstly images are pre-processed to improve the quality of operation. This pre-processing step generally includes cleaning, and scaling, in order to achieve more accurate deep learning model. After the pre-processing process, training model is built using Transfer Learning. VGG16 with the Adam optimizer is applied in the system.

The well-established convolutional neural networks architectures are used to extract optimized features from the images, called the VGG16 model. All the convolution and separable convolution layers are followed by batch normalization. To extract features from this pretrained VGG16 model, fine-tuning of this model is conducted using ISIC skin lesion images to obtain higher quality features from the images. The new fully connected layer is added by removing the old fully connected layer from this architecture. This system develops the skin lesion system using VGG16 model on unbalanced dataset and balanced dataset. In this system, the skin lesion dataset at Kaggle, ISIC dataset [9], is employed. This dataset contains 2357 images of malignant and benign oncological diseases, which were become from the International Skin Imaging Collaboration (ISIC). All images were sorted based on classification taken with ISIC, and all subsets were categorized into the same number of images, by the exception of melanomas and moles, these images are slightly dominant.

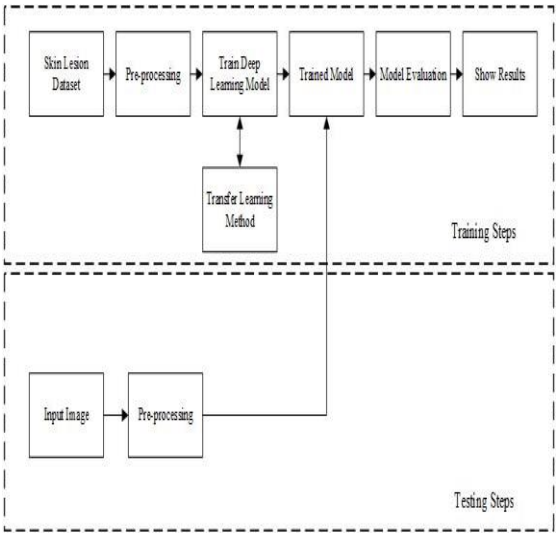


Figure 6: Proposed System Design

5. PERFORMANCE EVALUATION

In this system, the skin lesion dataset at Kaggle, ISIC dataset is used. From these dataset, basal cell carcinoma, squamous cell carcinoma, and melanoma images are extracted for classification. The system is developed with Python Language.

The system is firstly tested with unbalanced nature of original dataset. Then the system is tested with balanced nature by applying oversampling. The key metrics of performance measures (Accuracy, Recall, F1_measure, and Precision) are evaluated for the proposed system analysis. The Accuracy is computed using equation 1, the Precision is calculated with equation 2, the computation of Recall is done with equation 3, and the F1_measure is computed according to equation 4.

$$\text{Accuracy} = \frac{\textit{Exactly predicted sample}}{\textit{Total number of samples}} \times 100 \quad (1)$$

$$\text{Precision} = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Positive}} \quad (2)$$

$$\text{Recall} = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Negative}} \quad (3)$$

$$\text{F1\_measure} = \frac{2 * \textit{Precision} * \textit{Recall}}{\textit{Precision} + \textit{Recall}} \quad (4)$$

Figure 7, and figure 8 show the comparative results for the performance of VGG16 model on unbalanced, and balanced datasets. Table 1 shows the comparative results for the performance of proposed model for Basal Cell Carcinoma on unbalanced, balanced by oversampling.

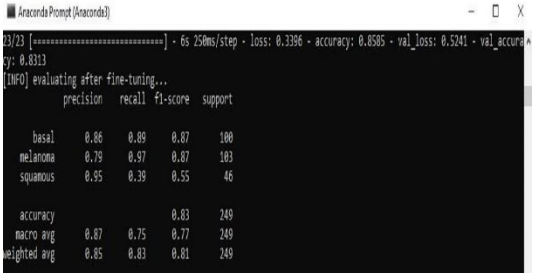


Figure 7: Performance Results for VGG16 in Unbalanced Dataset

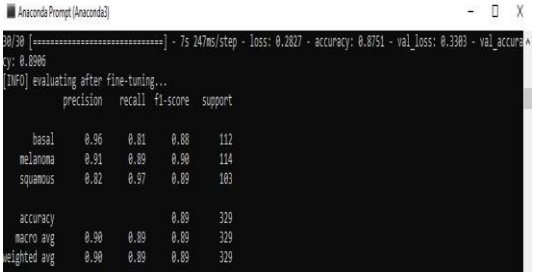


Figure 8: Performance Results for VGG16 in Balanced Dataset

Table 1. Performance Results for Basal Cell Carcinoma

Dataset	Basal Cell Carcinoma		
	Precision	Recall	F1-measure
Unbalanced	0.86	0.89	0.87
Balanced by Oversampling	0.96	0.81	0.88

Table 2 shows the comparative results for the performance of proposed model for Squamous Cell Carcinoma on unbalanced, balanced by oversampling.

Table 2. Performance Results for Squamous Cell Carcinoma

Dataset	Squamous Cell Carcinoma		
	Precision	Recall	F1-measure
Unbalanced	0.95	0.39	0.55
Balanced by Oversampling	0.82	0.97	0.89

Table 3 shows the comparative results for the performance of proposed model for Melanoma on unbalanced, balanced by oversampling. Table 4 shows the comparative of accuracy of proposed model on unbalanced, balanced by oversampling.

Table 3. Performance Results for Melanoma

Dataset	Melanoma		
	Precision	Recall	F1-measure
Unbalanced	0.79	0.97	0.87
Balanced by Oversampling	0.91	0.89	0.90

Table 4. Performance Results for Unbalanced and Balanced Dataset

Dataset	Accuracy (%)
Unbalanced	83
Balanced by Oversampling	91

According to the evaluation results, the improvement of accuracy of VGG16 is achieved by applying random sampling technique: oversampling.

6. CONCLUSIONS

As the skin is the body's central part, it is reasonable to assume that skin cancer is the most frequent disease in humans. Skin cancer indications can now be quickly and easily diagnosed using computer-based techniques. Multiple noninvasive methods have been proposed for assessing skin cancer signs. In the system, a better performance model for skin lesion classification is proposed using deep learning method VGG16 that makes the system faster and easier. A deep learning model to train the model with ISIC data set, VGG16, is implemented. In the unbalanced dataset of Basal Cell Carcinoma, the system achieves the value of Precision 0.86, Recall 0.89, and F1_measure 0.87 and achieves the value of Precision 0.96, Recall 0.81, and F1_measure 0.88 in the balanced dataset of Basal Cell Carcinoma. Also, the value of Precision 0.95, Recall 0.39, and F1_measure 0.55 are achieved in the unbalanced dataset of Squamous Cell Carcinoma and the value of Precision 0.82, Recall 0.97, and F1_measure 0.89 are achieved in the balanced dataset of Squamous Cell Carcinoma. In the unbalanced dataset of melanoma, the value of Precision 0.79, Recall 0.97, and F1_measure 0.87 are achieved and the value of Precision 0.91, Recall 0.89, and F1_measure 0.90 are achieved in the balanced dataset of melanoma. This proposed system achieves accuracy of 83% in unbalanced dataset and 91% in balanced dataset. Therefore, the accuracy of the system is improved by utilizing random sampling technique: oversampling.

REFERENCES

[1] M. A. Rasel, H. U. Obaidella, S. A. Kareem, "Convolutional Neural Network-Based Skin Lesion Classification with Variable Nonlinear Activation Functions," *IEEE Access (Volume: 10)*, IEEE, 05 August 2022, pp. 83398 – 83414.

[2] T. Alkarakatly, S. Eidhah, M. A. Sarawani, A. A. Sobhi and M. Bilal, "Skin Lesions Identification Using Deep Convolutional Neural Network", *2019 International Conference on Advances in the Emerging Computing Technologies (AECT)*, IEEE, Al Madinah Al Munawwarah, Saudi Arabia, 10 February, 2020, pp. 209-213.

[3] A. C. Salian, S. Vaze, and P. Singh, "Skin Lesion Classification using Deep Learning Architectures", *2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA)*, IEEE, Mumbai, India, 03-04 April, 2020, pp. 168-173.

[4] W. Gouda, N. U. Sama, and G. A. Waakid, "Detection of Skin Cancer Based on Skin Lesion Images Using Deep Learning," *Volume 10*, MDPI, 24 June 2022.

[5] https://www.aad.org/public/diseases/skin-cancer/types/common

[6] https://www.mayoclinic.org/diseases-conditions/skin-cancer/symptoms-causes/

[7] https://www.osmosis.org/answers/skin-lesions#

[8] J. Brownlee, "A Gentle Introduction to Transfer Learning for Deep Learning", September 16, 2019.

[9] https://www.kaggle.com/datasets/nodoubttome/skin-cancer9-classesisic

INTEGRATION OF K-MEANS CLUSTERING AND C4.5 DECISION TREE ALGORITHMS FOR PERFORMANCE ENHANCEMENT OF CLASSIFICATION

Moe Phyu Phyu Khaing[1] and Myint Myint Zaw[2]

[1,2] University of Computer Studies, (Taunggyi)
[1]moephyuphyukhaing23@gmail.com, [2]myintmyintzaw@ucstgi.edu.mm

ABSTRACT

Data mining is a field of study that discovers useful information from massive amount of data and modifies these data into categorized knowledge. Classification is one of the data mining methods and it can extract the model that provides useful information from large databases. To achieve desired objectives, the categorization model that is created is utilized to analyse trends or patterns and make judgments. Data mining hybrid algorithms are the natural integration of many pre-existing methodologies to strengthen performance and supply better results. K-means clustering is a clustering technique for breaking down big datasets into smaller, more manageable chunks. C4.5 is a decision tree classifier that may be used to make a judgment based on a valid sample of data. In this proposed method, the data with equal characteristics are clustered by k-means clustering as a first step. The cluster results of k-means clustering are classified by C4.5 decision tree algorithm as a second step. The experimental findings reveal better accuracy when the suggested approach is tested on these datasets.

KEYWORDS

Data Mining, K-means Clustering, C4.5 Decision Tree Classifier

1. INTRODUCTION

Nowadays, data is growing in time and it is hard to extract useful information from these data. Data mining algorithms can handle the massive data which is difficult for human to operate manually and therefore it can also save time and finally it can produce accurate results.

Classification becomes an important technology in this information age. The main function of classification is to predict future trends and behaviours based on the historical or previous data. Classification use classification algorithms to categorize data objects into a pre-set set of classes. The disadvantage of decision tree classification is that if the dataset is big, a single tree might become complicated and over fit. To handle this problem, clustering algorithm is used to partition the dataset based on equal characteristics of data objects.

Clustering is the division of a set of data items into meaningful subsets known as clusters. For its efficacy, k-means clustering is the most common and well-known method. Smaller subgroups are obtained by using k-means clustering. Building C4.5 decision tree and generating decision rules from these smaller subgroups can reduce the problem of information overload. Furthermore, decision rules generated from these subgroups are accurate and the performance of classification is enhanced in the experiments.

Analysis paralysis, which occurs when decision makers are overwhelmed with knowledge as a result of a large input dataset, is one of the downsides of the decision tree technique. Decision trees' decision-making capacity is slowed by the

time required to analyze information. Duplicate sub-trees on distinct routes are also a possibility. Clustering tackles these difficulties by isolating the input dataset across the attribute with the best information gain ratio before producing the actual decision tree [1].

2. RELATED WORK

Hybrid algorithms are created by taking some of the best characteristics of current technology and mixing them in interesting ways. This combination makes use of the hybrid algorithm's core technology, resulting in consistent and accurate outcomes. Creative algorithm choices create efficient combinations, reducing memory footprint and computation time. Many academics have been researching on combining multiple mining techniques. This section covers various hybrid techniques that combine decision tree algorithms and k-means clustering [1].

KanchanJha and Divakar Singh were able to harvest the data by combining clustering and classification methods. They came to the conclusion that a hybrid clustering and classification algorithm might be the best algorithm for the property tax system, and that it could help close the revenue gap and close the property tax gap [9].

Heena Sharma and Navdeep Kaur Kaler explore clustering and decision tree methods by combining hybrid algorithms such as k-means, Self-Organizing Maps (SOM), and Hierarchical Agglomerative Clustering (HAC) algorithms from clustering and chi-square automatic interactions) to extract data detection (CHAID) using) and the C4.5 algorithm. Clustering methods are used to form clusters of similar groups to facilitate feature or attribute extraction, and decision tree methods are used to make optimal decisions to extract valuable information. After effectively combining the two methods, they compared the effectiveness of HAC and SOM clustering data mining algorithms with traditional algorithms using C4.5 and CHAID decision tree algorithms on the dataset. When compared to previous algorithms, the hybridized approach achieves the highest classification accuracy while also being more resilient and generalizable. In comparison to other algorithms, the Kohonen SOM and HAC algorithms give the greatest classification accuracy, as well as more resilience and generalization capabilities [14].

D.Lavanya and Dr.K.Usha Rani proposed a hybrid approach to the breast cancer dataset including CART decision tree classifier, clustering and feature selection. In the absence of feature selection, the accuracy of the hybrid technique was compared with CART with feature selection, clustering, and classification. According to this study, a combination of data mining tasks outperformed a single data mining activity. According to this study, a combination of data mining tasks outperformed a single data mining activity [7].

Driyani Rajeshinigo presented a work in which she showed how integrating C4.5 with k-means clustering, which is used to handle continuous data, improves its accuracy. Because of the continuous data, decision trees suffer from limitations, and their accuracy suffers as a result. To convert continuous values of characteristics into categorical values, K-means clustering is utilized. K-means clustering returns the classified value for that particular attribute to the C4.5 algorithm once the values have been categorized. C4.5 then builds the tree using these categorical values for that property. The accuracy of the classifier is considerably improved by transforming continuous variables to categorical values using k-means clustering during tree construction. The accuracy improves from 73 percent to 92 percent when C4.5 is combined with k-means clustering [20].

When compared to the results of Decision tree given in the literature, the suggested cascaded K means clustering and Decision tree classifier achieves one of the best classification accuracies [4].

3. THEORY BACKGROUND

Data mining is a common technique for extracting previously unknown and possibly relevant facts from large databases. Data mining may also be defined as the process of using many layers of analysis to uncover correlations in massive relational databases. It's a powerful tool with a lot of promise for supporting a company or organization in growing sales and profit by exploiting client information.

Data mining can provide us with information that searches and reports cannot. Database applications can only project the information that is available in the database, and data mining often pulls information that isn't directly presented in the database, with restricted operating capabilities. As a result, the most accurate definition of data mining is the procedure for extracting information from a database [14].

3.1. Clustering

Clustering is the process of organizing a set of data items into groups or clusters, with objects in each cluster sharing a high degree of similarity but differing greatly from those in other clusters. Attribute values that describe the objects are utilized to determine dissimilarities and similarities, and distance measures are typically used. Clustering as a data mining technique has origins in a variety of fields, including biology, security, business intelligence, and Web search [18].

In the literature, there are several clustering methods. Because these categories may overlap, clustering algorithms are difficult to categorize precisely, resulting in a method that incorporates information from many categories. Nonetheless, providing a well-organized understanding of clustering techniques is useful.

3.1.1. k-means Clustering

The user specifies the k value, and k-means clusters or organizes the given records into k clusters. K-means is a clustering technique that is based on distance. Euclidean, Manhattan, Minowski, and other distance measures can be utilized. The user chooses the initial centroids for the k clusters, or k records are chosen at random. The distance metric is used to calculate each record's distance from each of the cluster centroids. The record is assigned to the cluster that is the most closely related to it (minimum distance). After the records have been assigned, the cluster centroids are recalculated. Each record is examined once again to see if the cluster assignment is valid by calculating its distance from each cluster and allocating it to the cluster closest to it. These repetitions continue until the record assignments to the clusters remain unchanged. Based on record assignments, the clusters holding the records differ as do the centroids [5].

3.1.2. k-means Learning Algorithm

Algorithm: k-means. The k-means approaches for partitioning, in which the mean value of the objects in each cluster represents the cluster's center [18].

Input: k: the number of clusters,

D: a data set comprising n items.

Output: A collection of k clusters.

Method:

1. Choose k items at random from D as the first cluster centers;
2. repeat
3. (Re) assign each item to the cluster with the highest mean value.
4. calculate the mean value of the items for each cluster to update the cluster means;
5. until no change occurs;

3.2. Classification

Classification is one of the most used Data Mining strategies for analyzing a dataset. It's used to extract models from a dataset that accurately characterizes significant data

classes. It takes two steps to classify something.

Step -1 A classification algorithm is used to develop the model, which is then used to the training data set.

Step -2 To measure the model's training performance and accuracy, the extracted model is tested against a preset test dataset. So classification is the process of assigning a class label to a dataset with an unknown class label [8].

To classify the data, this system employs C4.5 decision tree classification. Ross Quinlan created the C4.5 decision tree algorithm. It generates Decision Trees (DT) that can be used to categorize the dataset. Because C4.5 can deal with both continuous and discrete features, it extends the ID3 algorithm. C4.5 can additionally handle missing values and post-construction tree trimming.

3.2.1. Decision Tree Induction

Decision tree induction refers to the process of learning decision trees from class-labelled training tuples. A decision tree is a tree structure that looks like a flowchart, with each internal node (non leaf node) representing an attribute test, each branch representing the test's result, and each leaf node (or terminal node) storing a class label. The root node is the highest node in a tree. Internal nodes are shown as rectangles, whereas leaf nodes are shown as ovals. Some decision tree algorithms can only generate binary trees (trees with two internal nodes), but others can generate both binary and nonbinary trees.

The decision tree is used to evaluate the attribute values of an unknown class labelled tuple, X. A path is traced from the root to a leaf node that contains the tuple's class prediction. Classification rules can be easily converted from decision trees from the root to a leaf node that carries the tuple's class prediction. Decision tree induction approaches have been used for categorization in many domains, including health, manufacturing and production, financial analysis, astronomy, and molecular biology. Decision trees serve as the foundation for several commercial rule induction systems [18].

3.2.2. Decision Tree Learning Algorithm

Algorithm: Produce a decision tree. Create a decision tree using the data partition training tuples, D.

Input: The data partition is made up of a set of training tuples and their associated class labels, while the attribute list is made up of a set of candidate attributes;

A method for establishing the "best" splitting criterion for breaking data tuples into specified classes is the attribute selection method.

Output: A decision tree [18].

Method:

1. make a node N;
2. If all of the tuples in D belong to the same class, C, then
3. return N as a leaf node with the class C indicated;
4. If attribute list is null,
5. return N as a leaf node tagged with the majority class in D; // majority voting
6. Use the Attribute selection method (D, attribute list) to get the "best" splitting criterion;
7. assign the splitting criterion to node N;
8. if splitting attribute is discrete and multiway splits are permitted, then // not limited to binary trees
9. attribute list — splitting attribute; // delete splitting

attribute

10. for each j of splitting criterion outcomes // divide the tuples and create sub trees for each partition
11. Let Dj be the set of data tuples in D that meet result j; // a partition
12. if Dj is empty then
13. Connect node N with a leaf labeled with the majority class in D.;
14. Otherwise, attach node N to the node provided by Generate decision tree (Dj, attribute list).;

endfor

return N;

3.2.3. Attribute Selection Measure

The attribute selection measure scores each attribute that characterizes the training tuples under consideration. As the dividing attribute for the provided tuples, the attribute with the greatest measure score is picked. If the splitting attribute is continuous-valued or we are limited to binary trees, we must define a split point or a splitting subset as part of the splitting criterion. The partition D tree node is labelled with the splitting criterion, branches are created for each criterion outcome, and tuples are partitioned accordingly.

C4.5 constructs a decision tree using the Information Gain Ratio as an attribute selection measure. The information predicted to be required to classify a tuple in dataset D is given by,

$$Info\ (D) = -\sum_{i=1}^{m} pi\ log_2\ pi \qquad (1)$$

p_i = probability that a tuple in D belongs to class C_i

m = distinct classes C_i

D = dataset

The expected information required to classify a tuple from D based on a partitioning by A,

$$Info_A\ (D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} * Info(D_j) \qquad (2)$$

A = sub attribute from attribute

v = number of subsets

D_j = the number of tuples that belongs to the subset

$$Gain\ (A) = Info(D) - Info_A\ (D) \qquad (3)$$

$$SplitInfo\ (D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} * log_2(\frac{D_j|}{|D|}) \qquad (4)$$

$$Gain\ Ratio = {Gain(A)}/{SplitInfo(D)} \qquad (5)$$

4. Hybrid Algorithm Of The System

Algorithm: A stepwise description of the algorithm

1. For all attribute x€ X
2. Begin
3. //computation of parameters
4. $Info\ (D) = -\sum_{i=1}^{m} pi\ log_2\ pi$
5. $Info_A\ (D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} * Info(D_j)$
6. $Gain\ (A) = Info\ (D) - Info_A\ (D)$
7. $SplitInfo\ (D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} * log_2(\frac{|D_j|}{|D|})$
8. $Gain\ Ratio = {Gain(A)}/{SplitInfo(D)}$
9. end
10. Select attribute with highest gain value xbest for Clustering
11. Chose centroids for clusters
12. Apply k-means (S)
13. Cluster set C = {c1, c2,…, ck }
14. for each cluster ci € C
15. begin
16. return decision tree
17. end

The program is fed a tabular dataset S. For each attribute, compute the information gain values in steps 1 through 8. The attribute with the highest gain ratio is chosen for clustering the dataset in step 9.

Step 10 chooses centroids based on dataset size and range, then cluster S using the k-means algorithm. After step 10, the clusters c1, c2..., ck are obtained. The results are presented in the form of individual decision trees for each cluster, as well as their associated accuracy statistics [1].

5. FLOW OF THE SYSTEM

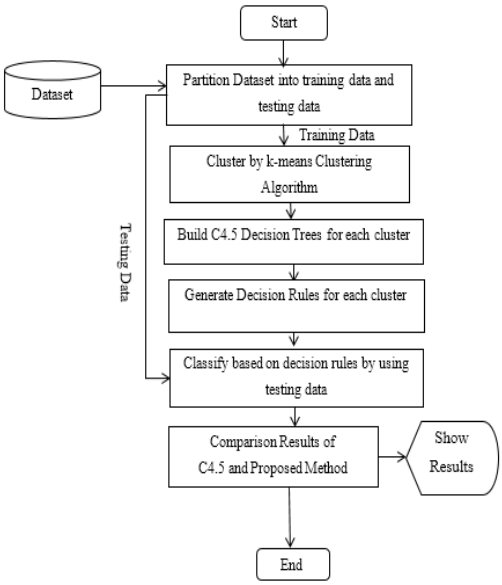


Figure 1. System Flow Diagram

6. PERFORMANCE ANALYSIS

To analyze performance, this system uses accuracy, precision, recall and Cohen's kappa coefficient. Confusion matrix for analyzing accuracy, precision, recall and Cohen's kappa coefficient are shown in Table 1.

Accuracy is the number of items that are correctly identified as either truly positive or truly negative out of the number of items. The number of things that are accurately recognized as positive class out of the total items identified as positive class is known as precision (Positive Predictive Value). The percentage of objects accurately detected as positive class out of the total number of real positives is known as recall (also known as sensitivity or True Positive Rate). Cohen's kappa is another metric used to assess the correctness of a decision tree model. Its value spans from 0 to 1 and is proportional to the accuracy of the decision tree model.

$$Accuracy = \frac{TA}{S} \quad (6)$$

$$Precision = \frac{TP\ (Class\ Name)}{Column\ Total} \quad (7)$$

$$Recall = \frac{TP\ (Class\ Name)}{Row\ Total} \quad (8)$$

$$Cohen's\ Kappa = \frac{TA-TE}{S-TE} \quad (9)$$

Where, TA = Total number of Agreements (a+ e+ i)

TP = The number of True Positives (a, e, i)

Column Total = D, E, F (Predicted as Class Name)

Row Total = A, B, C (Actual Class Name)

S = Total number of test records

TE = Total Expected Frequencies of Agreements

6.1. Dataset

This system uses Banknote dataset and Red Wines dataset from UCI Machine Learning Repository. The attributes in these datasets are types of numeric.

Data from Banknote dataset were extracted from images that were taken for the evaluation of an authentication procedure for banknotes. This dataset has 1372 records and the classes are 0 (real) and 1 (fake). Each record is defined by the following attributes:

1. variance of Wavelet Transformed image (V1)
2. skewness of Wavelet Transformed image (V2)
3. curtosis of Wavelet Transformed image (V3)
4. entropy of image (V4)
5. Class (0 and 1)

Red Wines dataset has 1599 records divided into 6 classes (wine quality). Each record is defined by the following attributes:

1. fixed acidity
2. volatile acidity
3. citric acid
4. residual sugar
5. chlorides
6. free sulfur dioxide
7. total sulfur dioxide
8. density
9. pH
10. sulphates
11. alcohol
12. quality (3 to 8)

6.2. Results and Analysis

Banknote dataset and Red Wines dataset are partitioned into training data 80% and testing data 20%. In proposed method, training data is partitioned into two clusters (k=2) using k-means clustering across the attribute with highest information gain ratio.

Decision rules obtained from individual clusters are as follows:

Banknote Dataset Cluster - 1

1. If V1 ≤ -0.3157 And V2 ≤ -5.5932 Then Class – 2

2. Else If V1 ≤ -0.3157 And V2 > -5.5932 And V3 ≤ 3.5999 Then Class – 2

3. Else If V1 ≤ -0.3157 And V2 > -5.5932 And V3 > 3.5999 Then Class – 2

4. Else If V1 > -0.3157 And V4 ≤ 0.226 Then Class – 1

5. Else If V1 > -0.3157 And V4 > 0.226 And V3 ≤ 3.0582 Then Class – 1

6. Else If V1 > -0.3157 And V4 > 0.226 And V3 > 3.0582 Then Class – 1

Banknote Dataset Cluster – 2

1.If V2 ≤ 5.3879 And V1 ≤ 0.3237 And V3 ≤ -1.2428 Then Class – 2

2. Else If V2 ≤ 5.3879 And V1 ≤ 0.3237 And V3 > -1.2428 Then Class – 2

3. Else If V2 ≤ 5.3879 And V1 > 0.3237 And V3 ≤ -0.8452 Then Class – 2

4. Else If V2 ≤ 5.3879 And V1 > 0.3237 And V3 > -0.8452 Then Class – 1

5. Else If V2 > 5.3879 And V1 ≤ 1.0774 And V4 ≤ -4.0086 Then Class – 1

6. Else If V2 > 5.3879 And V1 ≤ 1.0774 And V4 > -4.0086 Then Class – 1

7. Else If V2 > 5.3879 And V1 > 1.0774 Then Class – 1

There are a few weaknesses in decision tree.

1. If the attribute list is empty, the majority of classes in the dataset are returned as a leaf node or class label.

2. If a class has few instances, the information for this class is insufficient to predict.

3. When some classes dominate, decision tree learners produce biased trees.

According to decision rules 2 and 6 in Banknote dataset cluster – 1, there are no more attributes on which to separate the data, the node predicts the most common class.

Banknote dataset comprises of only two classes, and the number of occurrences for these two classes is not significantly different. These two classes' decision criteria are sufficient to classify data samples. This dataset's performance is good. (Table 2)

Red Wines Dataset Cluster – 1

1. If volatile acidity ≤ 0.5414 And alcohol ≤ 12.3987 And pH ≤ 3.2915 Then Class – 7

2. Else If volatile acidity ≤ 0.5414 And alcohol ≤ 12.3987 And pH > 3.2915 Then Class – 6

3. Else If volatile acidity ≤ 0.5414 And alcohol > 12.3987 Then Class – 7

4. Else If volatile acidity > 0.5414 And alcohol ≤ 12.1342 Then Class – 6

5. Else If volatile acidity > 0.5414 And alcohol > 12.1342 And pH ≤ 3.4993 Then Class – 5

6. Else If volatile acidity > 0.5414 And alcohol > 12.1342 And pH > 3.4993 Then Class –7

Cluster-1 has 43 decision rules. The system can predict for four classes (5, 6, 7 and 8) but not for the other two (3 and 4). Because class distribution is imbalanced and information for the other three classes (3, 4 and 8) is insufficient, decision rules are for classes 5, 6 and 7.

Red Wines Dataset Cluster – 2

1. If volatile acidity ≤ 0.6456 And alcohol ≤ 9.8065 Then Class – 5

2. Else If volatile acidity ≤ 0.6456 And alcohol > 9.8065 Then Class – 6

3. Else If volatile acidity > 0.6456 And alcohol ≤ 10.0735 Then Class – 5

4. Else If volatile acidity > 0.6456 And alcohol > 10.0735 And pH ≤ 3.3406 Then Class – 6

5. Else If volatile acidity > 0.6456 And alcohol > 10.0735 And pH > 3.3406 Then Class – 5……

Cluster-2 has 162 decision rules, and the system can predict five classes with the exception of class 8. Because class distribution is unbalanced and information for the other three classes (3, 4, and 8) is insufficient, decision rules are for classes 5, 6, and 7.

Red Wines dataset is imbalanced. There are clusters in imbalanced datasets that include minority classes and clusters that do not contain data objects from all classes. When information for minority classes is insufficient to classify data items, majority classes are biased for the decision tree leaves. When these conditions occur, the classification performance for imbalanced datasets decreases. (Table 3)

Classification accuracy depends on partitioning dataset into training and testing percentage, training and testing data that are partitioned randomly and initial centroids in k-means clustering. This system can operate on numeric attributes dataset only because clustering is made based on distance measures of records. The more the number of cluster (k), the more declines the performance of classification.

Table 2 and 3 show analysis results of C4.5 and proposed method. Accuracy, precision, recall and Cohen's kappa coefficient are improved by using clustering at first in cluster-2.

In table 3, the accuracy of the Red Wines dataset cluster-1 is low because there is less training records and the class distribution is highly different when compared to cluster-2.

According to the table 2, it is suitable to conclude that this system is best for binary classification.

7. CONCLUSIONS

Classification is the process of extracting the model that assigns useful knowledge from vast amount of data. K-means clustering and C4.5 decision tree classification are used to classify data samples. In this system, the dataset is partitioned into training data (80%) and testing data (20%). The cluster results of k-

means clustering are classified by C4.5 decision tree algorithm. C4.5 decision trees and decision rules are produced for each cluster. Then the system calculates accuracy, precision, recall, and Cohen's kappa coefficient using the testing data. Finally, the system shows the comparison results of C4.5 decision tree algorithm and the proposed hybrid algorithm. Decision rules obtained from individual clusters are accurate and the performance of classification is improved.

ACKNOWLEDGEMENTS

Thanks, are given to all the people who helped me directly or indirectly for the successful completion of this journal.

Table 1. Confusion Matrix for Performance Analysis

	Predicted Class					
Actual Class	Class	Class Name 1	Class Name 2	Class Name 3	Row Total	Recall
	Class Name 1	**a**	b	c	A	a/A
	Class Name 2	d	**e**	f	B	e/B
	Class Name 3	g	h	**i**	C	i/C
	Column Total	D	E	F	S	
	Precision	a/D	e/E	i/F		

Table 2. Experimental Results of Banknote Dataset

Method	Dataset	Accuracy	Precision	Recall	Cohen's kappa
C4.5	Banknote	0.88	0.89	0.89	0.77
Proposed Method (k-means+C4.5)	Banknote Cluster-1	0.85	0.86	0.86	0.71
	Banknote Cluster-2	0.92	0.92	0.92	0.84

Table 3. Experimental Results of Red Wines Dataset

Method	Dataset	Accuracy	Precision	Recall	Cohen's kappa
C4.5	Red Wines	0.52	0.17	0.22	0.2
Proposed Method (k-means+C4.5)	Red Wines Cluster-1	0.32	0.1	0.2	0.03
	Red Wines Cluster-2	0.55	0.18	0.23	0.26

REFERENCES

[1] A. Ahlawat and B. Suri, "Improving classification in data mining using hybrid algorithm," in *2016 1st India International Conference on Information Processing (IICIP)*, Delhi, India, Aug. 2016, pp. 1–4, doi: 10.1109/IICIP.2016.7975380

[2] A. Cherfi, K. Nouira, and A. Ferchichi, "Very fast C4.5 decision tree algorithm," *Applied Artificial Intelligence: An International Journal*, vol. 32, no. 2, pp. 119–137, 2018, doi: 10.1080/08839514.2018.1447479.

[3] A. George Eapen, "Application of data mining in medical applications," pp. 1-117 University of Waterloo, Waterloo, Ontario, Canada, 2004.

[4] A. Gowda Karegowda, P. V, M. A. Jayaram, and A. S. Manjunath, "Rule based classification for diabetic patients using cascaded k-means and decision tree C4.5," *International Journal of Computer Applications (0975 – 8887)*, vol. 45, no.12, pp. 1–6 May 2012.

[5] A. P K and Dr. R. Shettar, "Hybrid decision tree using k-means for classifying continuous data," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 3, no. Issue 10, pp. 1-8, Oct. 2015, doi: 10.15680/IJIRCCE.2015. 0310068.

[6] A. Singh, A. Yadav, and A. Rana, "K-means with three different Distance Metrics," *International Journal of Computer Applications IJCA*, vol. 67, no. 10, pp. 13–17, Apr. 2013, doi: 10.5120/11430-6785.

[7] D. Lavanya and Dr. K. Usha Rani, "A hybrid approach to improve classification with cascading of data mining tasks," *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, vol. 2, no. 1, Jan. 2013.

[8] D. Rajeshinigo and J. P. A. Jebamalar, "Accuracy improvement of C4.5 using k-means clustering," *International Journal of Science and Research (IJSR)*, vol. 6, no. 6, pp. 1–4, Jun. 2017.

[9] D. Singh, "Hybrid Algorithm of Clustering and Classification in data Mining: A Survey," *Advances in Computer Science and Information Technology*, vol. 1, no. 2, pp. 1-4, 2014.

[10] H. Sharma and N. K. Kaler, "Data mining with improved and efficient mechanism in clustering analysis and decision tree as a hybrid approach," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 2, no. 5, pp. 1-4, Apr. 2013.

[11] J. Han, M. Kamber, and J. Pei, *Data Mining*, Third Edition, 2011.

[12] J. Irani, N. Pise, and M. Phatak, "Clustering Techniques and the Similarity Measures used in Clustering: A Survey," *International Journal of Computer Applications IJCA*, vol. 134, no. 7, pp. 9–14, Jan. 2016, doi: 10.5120/ijca2016907841.

[13] R. Sudrajat, I. I, and K. D, "Analysis of data mining classification by comparison of C4.5 and ID3 algorithms," presented at the IOP Conference Series: *Materials Science and Engineering*, 2017, pp. 1–9. doi: 10.1088/1757-899X/166/1/012031.

[14] S. Agarwal, "Data Mining: data mining concepts and techniques," in *2013 International Conference on Machine Intelligence and Research Advancement*, Katra, India, Dec. 2013, pp. 203–207, doi: 10.1109/ICMIRA.2013.45

[15] S. llkin Serengil, "A step by step C4.5 decision tree example" http://sefiks.com/2018/05/13/a-step-by-stepc4.5-decision-tree-example.

[16] S. Neelamegam and Dr. E. Ramaraj, "Classification algorithm in Data mining: An overview," *International Journal of P2P Network Trends and Technology (IJPTT)*, vol. 3, no. 5, pp. 1–5, Oct. 2013.

[17] K. Ganesan, "How to compute precision and recall for a multi-class classification problem." https://kavita-ganesan.com/how-to-compute-precision-and-recall-for-a-multi-class-classification-problem/#.X_E3DFVKjIU.

[18] M. Gupta, "Calculating Precision & Recall for Multi-Class Classification." *Medium, an American online publishing platformhttps://medium.com/data-science-in-your-pocket/calculating precision-recall-for-multi-class-classification* 9055931ee229.

[19] "Cohen's Kappa: Index of Inter-rater Reliability." *The University of Nebraska–Lincoln, a public land-grant research university in Lincoln, Nebraska* [Online]. Available: https://psych.unl.edu/psycrs/handcomp/hckappa.PDF.

[20] "Data Structures - Algorithms Basics." https://www.tutorialspoint.com/index.htm.

Authors

Biography

Moe Phyu Phyu Khaing recieved the B.C.Sc in Computer Science from University of Computer Studies, (Monywa) in 2018 and M.C.Sc from University of Computer Studies, Mandalay (UCSM) in 2022. She works as a tutor at University of Computer Studies, (Taunggyi). Her research interest includes Data Mining and Machine Learning.

Myint Myint Zaw is a teacher. She is an assistance lecturer at University of Computer Studies, (Taunggyi). She achieved D.C.Sc in 2002 from University of Computer Studies, (Kyaing Tong), B.A English in 2000 and M.A English in 2020 from Taunggyi University.

PANCREAS CANCER DIAGNOSIS SYSTEM USING NB AND ID3 CLASSIFIERS

Lin Lin Htun[1], Yin Su Hlaing[2] and Yin Nyein Aye[3]

[1,2,3] Faculty of Information Science, University of Computer Studies (Taunggyi)

[1]linlinhtun@ucstgi.edu.mm, [2]yinsuhlaing@ucstgi.edu.mm, [3]yinnyeinaye@ucstgi.edu.mm

ABSTRACT

These days, individual's way of life changes as per the cutting-edge patterns and the food styles change and become various diseases. Pancreas releases enzymes that aid digestion and produces hormones that help manage of the blood sugar. Pancreatic cancer is seldom detected at its early stages when it's most curable. This is because it often doesn't cause symptoms until after it has spread to other organs. The pancreas disease recognizable proof is a significant and the right computerization would be entirely attractive. Each person can't be similarly dexterous as specialists. Besides, all specialists can't be similarly gifted in each sub strength. In this circumstance, a robotized clinical analysis framework improves the clinical consideration and it can likewise lessen costs. This framework is proposed the clinical finding framework by arranging pancreas malignant growth. For arrangement, this framework utilizes the ID3 decision tree and Naïve Bayesian (NB) classifiers. Besides, this framework analyses the exhibition of these two classifiers. The framework causes the client to rapidly know the pancreas malignant growth status and to decrease the registration costs.

KEYWORDS

Pancreas Cancer, Classifications, ID3, NB Classifier

1. INTRODUCTION

The pancreas is the organ of the stomach related framework and situated behind the stomach in the upper mid-region. It makes compounds to process proteins and fats in the digestive organs, and produces the hormones insulin. Pancreatic malignant growth is illness made harm the DNA in a cell in your pancreas. A solitary malignant growth cell develops and isolates quickly, turning into a tumor. It is difficult to know the pancreatic disease from the earliest starting point, so it tends to be known as the quiet executioner. It can the spread the inward organs without any problem. So, in this paper, the side effects of pancreatic malignancy are introduced as informational indexes to distinguish the illness. The dangers of pancreatic malignancy are likewise portrayed and informational indexes are taking a gander at the overview from 500 patients.

Along these lines, this framework proposes the clinical analysis framework that can check the patient who endures the pancreas disease or not. For checking, this framework utilizes the order technique. There are numerous grouping techniques that are Bayesian classifier, choice tree, k-nearest neighbor (KNN) classifier, uphold vector machine, etc. Among them, the proposed framework utilizes ID3 choice tree and Naive Bayesian classifiers for pancreas malignancy arrangement. These two classifiers have performed well on wide scope of uses including clinical diagnostics, text portrayal, data recovery and spam

separating. Along these lines, the proposed framework gives the right and exact pancreas malignant growth sickness result for the patient.

2. Related Work

Machine learning is a paradigm that may refer to learning from past experience to improve future performance. The sole focus of this field is automatic learning methods. Learning refers to modification or improvement of algorithm based on past "experiences" automatically without any external assistance from human. [5]

K. J. Danjuma, proposed the identification and performance evaluation of machine learning classification algorithms for lung cancer patients [3]. Thoracic surgery datasets from the UCI machine learning repository were utilized to train and evaluate models using multilayer perceptron, decision tree, and Naive Bayesian techniques. Multilayer perceptron performed best, with a classification accuracy of 82.3%, according to the comparative analysis. Naive Bayes performed the poorest, with a classification accuracy of only 74.4%, while decision tree came in second.

A decision tree is a tree-based technique in which any path beginning from the root. It is the hierarchical exemplification of knowledge relationships that contain nodes and connections. The details of this method, such as using algorithms/approaches, datasets, and the findings achieved are summarized. In addition, this study highlighted the most commonly used approaches and the highest accuracy methods achieved.[2]

Naive Bayes and decision tree data mining classifiers were presented by K. Krishnaveni and E. Radhamani [2] in 2016 to diagnose and assess attention deficit hyperactivity disorder (ADHD). The Naive Bayes and decision tree classifiers are used to divide the ADHD-prone samples into moderate (ADHDmod) and high (ADHDhigh) classes, and their classification performance is assessed using the WEKA tool's accuracy metrics, ROC curve, and confusion matrix. According to the experimental findings, the decision tree classifier provides 100% accuracy on the ADHD dataset while taking 94.3% less time than Naive Bayes. In this paper, datasets are sample viewed Urinary biomarkers for pancreatic cancer in Kaggle website. And then added the common symptoms in pancreas cancer attributes and insert data in the datasets for this paper. [7]

Hepatitis was identified in 2019 by S. U. Emon, T. I. Trishna, and R. R. Ema [3] based on symptoms, the mode of transmission, and pertinent testing. WEKA software uses Naive Bayes, KStar, and J48 classifiers to determine the outcome. For text categorization with high dimensional training datasets, this system is utilized. The accuracy of the result for 10-fold cross-validation in Naive Bayes is 93.8%. By employing 10-fold cross-validations, the accuracy of the J48 classifier is 98.6% while that of the KStar classifier is 97.2%.

3. Classification Analysis

Classification is a form of information investigation that concentrates models depicting significant information classes. Such models, called classifiers, foresee categorical (discrete, unordered) class names.

Numerous arrangement strategies have been proposed by analysts in AI, are memory inhabitant, normally accepting a little information size. Ongoing information mining research has based on such work, creating adaptable characterization and measures of circle occupant information. Arrangement has various applications, including misrepresentation location, target promoting, execution expectation, assembling, and clinical conclusion.

Classification examination is utilized to plan informational collections into predefined gatherings and classes. As the classes are resolved before the analysing the datasets, it is viewed as the regulated learning. The learning of classifier is 'administered' in that it is advised to which class each preparation tuple has a place. It

grouping, in which the class name of each preparation tuple isn't known, and the number or set of classes to be learned may not be known ahead of time.

3.1. Naive Bayesian (NB) Classifier

Naive Bayesian (NB) classifier shows the connection between the autonomous factors and the objective variable [6]. The preparing steps of NB classifier are as per the following:

1. Every information test is addressed by an n-dimensional element vector, X = (x1, x2,... xn), which depicts n-estimations based on the example from n-credits, A1, A2,....., An.
2. Assume there are m classes, labelled C1, C2,..., Cm. If and only if P(Ci X) > P(Cj X) for 1 j m, j I, the Nave Bayesian classifier assigns an obscure sample X to the class Ci given an obscure information test X.
3. The most extreme posteriori theory is applied to the class Ci where P (CiX) is amplified. According to Bayes' theorem, P(CiX) = P(XCi) P(Ci)/P(X)
4. Since P(X) is constant across all classes, only P(XCi) P(Ci) needs to be increased.4. Given informational indexes with numerous properties, the Naive supposition of class contingent freedom is made. Hence, P(X\Ci) =∏nk=1 P(xk\Ci). The likelihood P(x1\Ci), P (x2\Ci)… P(xn\Ci) can be assessed from the information tests.
5. In request to group an obscure example X, P (X\Ci) P (Ci) is assessed for each class Ci,. Test X is then allocated to the class Ci if and just if P(X\Ci) P(Ci) > P(X\Cj) P(Cj) At the end of the day, it is allocated to the class Ci for which P (X\Ci) P(Ci) is the most extreme [1].

3.2. ID3 Decision Tree Classifier

Decision tree is a stream outline like tree structure, where every inside centre point demonstrates a test on a quality, each branch addresses a consequence of the test, and leaf centres address classes or class disseminations [1].

3.2.1. ID3 Decision Tree Algorithm

ID3 decision tree calculation is as per the following:

Algorithm: Generate_decision_tree.
Input: Training Samples
Output: A Decision Tree
1. create a hub N;
2. if examples are the entirety of a similar class, C at that point
3. return N as a leaf hub named with class C;
4. if trait list is vacant at that point
5. return N as a leaf hub named with the most well-known class in examples;
6. select test-trait, the characteristic among property list with the most elevated data gain;
7. label hub N with test-trait;
8. for each known worth ai of test-characteristic
9. grow a branch from hub N for the condition test-attribute=ai;
10. let si be the arrangement of tests in examples for which test-attribute=ai;
11. if si is vacant at that point
12. attach a leaf named with the most widely recognized class in examples;
13. else connect the hub returned by Generate _decision _tree [1];

4. ARCHITECTURE OF THE SYSTEM

From the outset of the framework, the client must information the obscure example about pancreas information. At that point, the client can pick the ID3 decision tree classifier or Naive Bayesian (NB) classifier.

As indicated by the NB classifier, this framework figures the likelihood about every pancreas trait. At that point, this framework likewise figures the likelihood about obscure example over each class. In the wake of figuring every likelihood, this framework picks the class that has most extreme likelihood.

In the ID3 decision tree classifier measure, this framework must figure the data

increase about every pancreas characteristic. In view of data gain result, this framework produces the decision tree and rules. As indicated by the choice guidelines, this framework creates the class result for the obscure example.

At that point, this framework figures the precision of every classifier and analyses the classification of every classifier. At long last, this framework shows the class result about the obscure example. Proposed framework configuration is appeared in Figure 1.

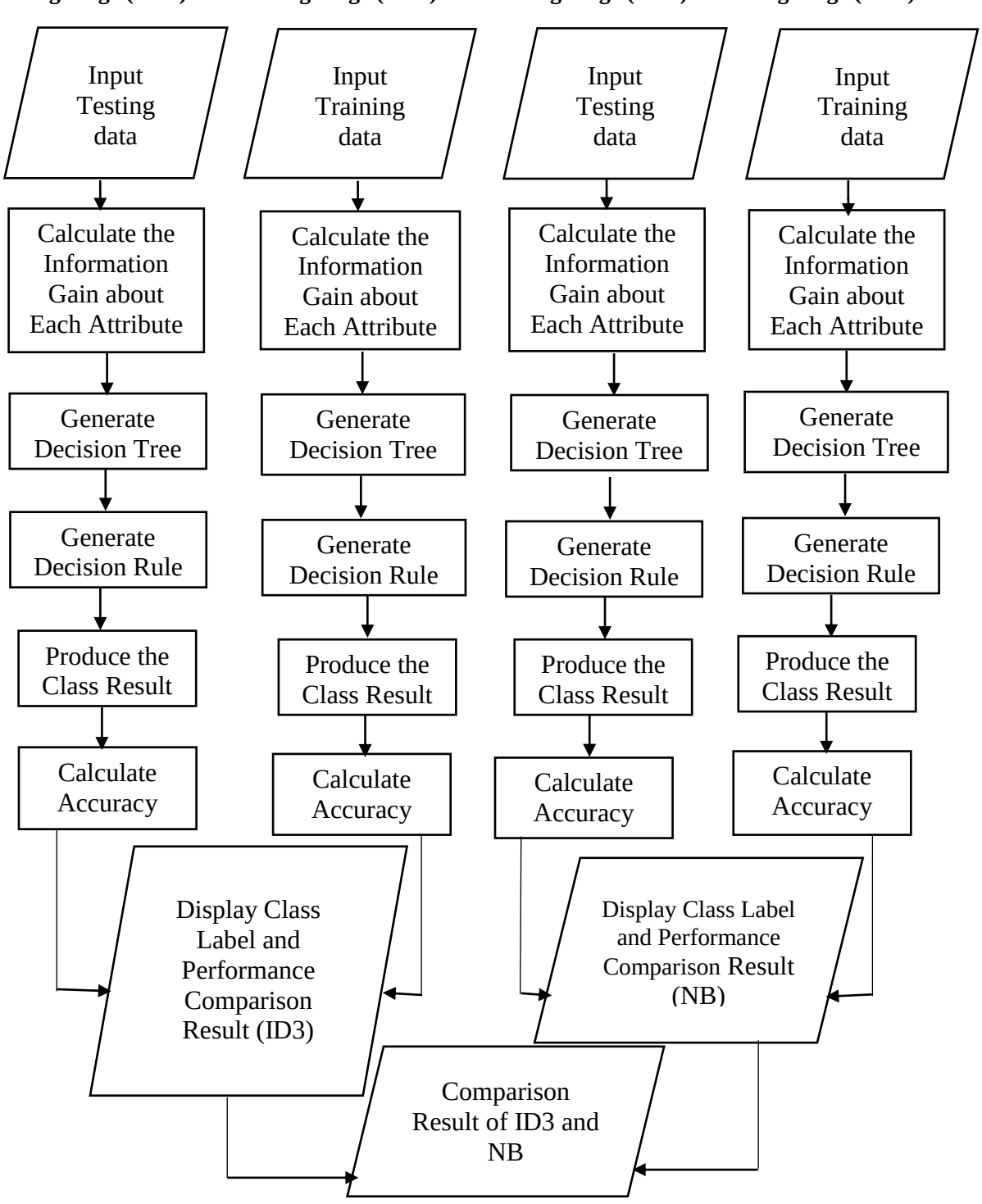


Figure 1. Proposed System Design for Comparison Result of ID3 and NB

This system introduced pancreas cancer classification by using ID3 and NB classifiers. Firstly, by using ID3 classifier is shown in Figure 1. There are two stages, these stages are training stage (80%) and testing stage (20%). In the training stage, calculate the Information Gain about Each Attribute (ID3), generate Decision Tree, generate Decision Rule, produce the Class Result of Positive or Negative for pancreas cancer, calculate Accuracy and then display the Class Label and Performance Comparison Result. In the testing stage, calculate the Information Gain about Each Attribute (ID3), generate Decision Tree, generate Decision Rule, produce the Class Result of Positive or Negative for pancreas cancer, calculate Accuracy and then display the Class Label and Performance Comparison Result as shown in Figure 1. The next classifier of NB is shown in Figure 1. There are two stages, these stages are training stage (80%) and testing stage (20%). In the training stage, calculate the Information Gain about Each Attribute (NB), generate Decision Tree, generate Decision Rule, produce the Class Result of Positive or Negative for pancreas cancer, calculate Accuracy and then display the Class Label and Performance Comparison Result. In the testing stage, calculate the Information Gain about Each Attribute (NB), generate Decision Tree, generate Decision Rule, produce the Class Result of Positive or Negative for pancreas cancer, calculate Accuracy and then display the Class Label and Performance Comparison Result as shown in Figure 1.

4.1 Abstract Pancreas Cancer Information

In this system, eleven symptom of pancreas cancer and its class are used for classification. Pancreas cancer attribute and its values are shown in Table 1.

Table 1. Pancreas Cancer Attribute

Attribute Name	Data Type	Attribute Value
Sex (A1)	Nominal	Male (1), Female (2)
Unexpected Weight Loss (UWL) (A2)	Nominal	Moderate (1), Significant (2)
Abdomen Pain (AP) (A3)	Nominal	Yes (1), No (0)
Dark Urine (DU) (A4)	Nominal	Yes (1), No (0)
Loss of Appetite (LA) (A5)	Nominal	Moderate (1), Significant (2)
Liver Enlargement (LE) (A6)	Nominal	Yes (1), No (0)
Jaundice (A7)	Nominal	Yes (1), No (0)
Itchy Skin (IS) (A8)	Nominal	Small (1), Big (2)
Vomiting (A9)	Nominal	Yes (1), No (0)
Weakness (A10)	Nominal	Yes (1), No (0)
Constipation (A11)	Nominal	Yes (1), No (0)
Class	Nominal	Positive (P), Negative (N)

4.2. Explanation of the System

Pancreas cancer diagnosis system is tested by using ten patient records. Sample data is shown in Table 2.

Table 2. Sample Pancreas Cancer Training Data

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	Class
1	2	1	1	2	0	1	1	1	1	1	P
1	1	1	0	2	0	0	2	0	1	0	P
2	1	0	0	2	0	0	1	1	0	0	N
1	2	1	0	1	0	0	2	0	0	1	N

1	2	0	1	1	0	1	2	0	0	1	P
2	1	0	1	2	0	1	1	1	1	0	N
2	2	1	1	2	1	1	1	1	1	1	P
2	1	1	0	1	1	0	1	1	1	0	N
1	2	1	1	1	0	1	2	0	0	0	P
1	1	0	1	2	0	1	2	1	1	0	P

To know the pancreas cancer result, the user (patient) first inputs the unknown case (sample) that includes Sex "male", Unexpected weight loss "moderate", Abdomen pain "no", Dark urine "yes", Loss of appetite "significant", Liver enlargement "yes", Jaundice "no", Itchy skin "small", Vomiting "yes", Weakness "yes" and Constipation "no". By using training ten records from Table 2, this system checks the pancreas cancer result. For checking, this system uses the ID3 and NB classifiers.

4.2.1. Naive Bayesian Classification Process

According to the Naive Bayesian (NB) classifier, this system calculates the probability and chooses the maximum probability result for the unknown sample. The probability results about each symptom (attribute) are shown in Table 3.

Table 3. Probability Results about Each Attribute

Attribute (Symptom) Name	**Probability Result**	
	Class: Positive (P)	**Class: Negative (N)**
Sex: Male	0.83	0.25
Unexpected Weight Loss (UWL): Moderate	0.33	0.75
Abdomen Pain (AP): No	0.33	0.5
Dark Urine (DU): Yes	0.83	0.25
Loss of Appetite (LA): Significant	0.67	0.5
Liver Enlargement (LE): Yes	0.17	0.25
Jaundice: No	0.17	0.75
Itchy Skin (IS): Small	0.33	0.75
Vomiting: Yes	0.5	0.75
Weakness: Yes	0.67	0.5
Constipation: No	0.5	0.75

After calculating each attribute probabilities, this system obtains the "positive" probability that is "0.0008" and the "negative" probability that is "0.0005". So, this system produces the "**Pancreas Cancer is Positive**" result to the patient according to the inputted unknown sample.

4.2.2. ID3 Decision Tree Classification Process

In the ID3 decision tree grouping measure, this framework figures the data gain for every indication (property). This framework picks the trait as the hub that has the most elevated data gain results among different qualities. At that point, this framework plays out every cycle by utilizing data gain results. In the wake of acquiring each class, this framework delivers the decision tree. Decision tree is appeared in Figure 2.

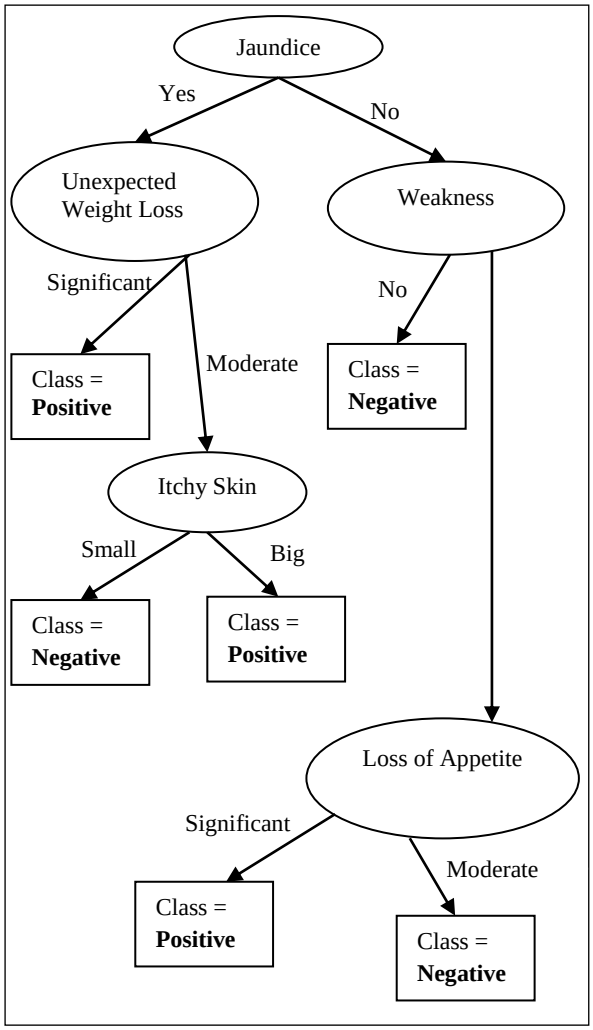


Figure 2. Decision Tree

According to the decision tree, this system produces the decision rules. Decision rules are as follows:

1. **IF** "Jaundice = Yes" **AND** "Unexpected Weight Loss = Significant" **THEN** "Class = Positive".
2. **IF** "Jaundice = Yes" **AND** "Unexpected Weight Loss = Moderate" AND "Itchy Skin = Small" **THEN** "Class = Negative".
3. **IF** "Jaundice = Yes" **AND** "Unexpected Weight Loss = Moderate" AND "Itchy Skin = Big" **THEN** "Class = Positive".
4. **IF** "Jaundice = No" **AND** "Weakness = No" **THEN** "Class = Negative".
5. **IF** "Jaundice = No" **AND** "Weakness = Yes" **AND** "Loss of Appetite = Significant" **THEN** "Class = Positive".
6. **IF** "Jaundice = No" **AND** "Weakness = Yes" **AND** "Loss of Appetite = Moderate" **THEN** "Class = Negative".

Finally, this system produces the "**Pancreas Cancer is Positive**" result to the patient according to decision rule 5.

5. EXPERIMENTAL RESULT

This system uses the holdout method to measure the performance of the system. Hold out method splits up the dataset into a 'train' and 'test' set. Hold-out method uses the 80% of data for training and the remaining 20% of the data for testing. In 500 patient records, 400 patient records are using for training data and 100 patient records are using for testing data. The precision method is as follows

$$Precision(i,j) = n_{ij}/n_i \quad (1)$$

where, the i and j represents all the classified data and the correct classified data. The n_j and n_i represents the number of i and j.
To show the performance of the system, this system is using 500 patients record about pancreas cancer. Moreover, this system compares the performance between Naive Bayesian and ID3 decision tree classifiers. Table 4 shows the experimental result of the system.

$$Entropy(S) = \sum_{i=1}^{c} - P_i \log_2 p_i \quad (2)$$

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values} \frac{|S_v|}{|S|} Entropy(S_v) \quad (3)$$

Where Values A is all possible values attribute A and S_v is the subset of S for which attribute A has value v.

Table 4. Experimental Result of the System

Number of Testing Pancreas Cancer Record	Accuracy (%)	
	Naive Bayesian Classifier	ID3 Decision Tree Classifier
100	85%	88%

200	87%	89%
300	88%	90%
400	89%	91%
500	89%	93%

Performance comparison result is shown in Figure 3. According to the comparison result, the ID3 decision tree classifier is more accurate than the Naïve Bayesian classifier.

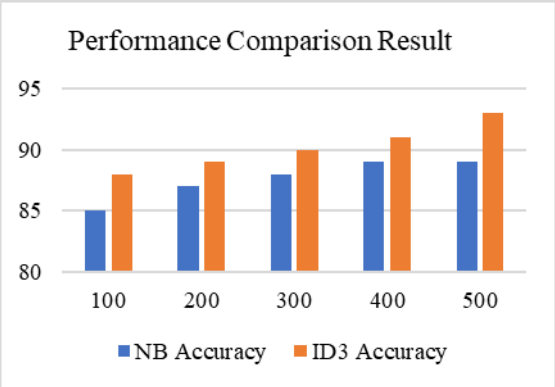


Figure 3. Performance Comparison Result

6. CONCLUSIONS

Characterization framework for pancreas malignancy is executed by utilizing Naive Bayesian (NB) and ID3 decision tree classifier. The framework can order the sickness result rapidly and effectively. It helps patients for testing their pancreas malignant growth status. At that point it likewise underpins the lesser specialists when they group the infection issue. Also, it can set aside time and cash since patients don't have to proceed to counsel the experts. By contrasting the NB and ID3 classifier, this framework brings up the ID3 classifier is more precise than the NB classifier.

ACKNOWLEDGEMENTS

I would like to thank all people who directly and indirectly contributed towards the success of this paper for the support encouragement, useful ideas, valuable guidance and help in preparing this paper.

REFERENCES

[1] H. Jiawei and K. Micheline, "*Data Mining Concepts and Techniques*", Simon Fraser University, United States of America, 2001.

[2] Gizem, Aksahya & Ayese, Ozcan (2009) Communications & *Networks*, Network Books, ABC Publishers.

[3] S. U. Emon, T. I. Trishna and R. R. Ema, *"Detection of Hepatitis Viruses Based on J48, KStar and Naive Bayes Classifier*", IEEE, 2019.

[4] Kurdistan Region, " *Classification Based on Decision Tree Algorithm for Machine Learning*", Journal of Applied Science and Technology Trends, vol. 02, no. 01, pp.20 - 28, 2021.

[5] K. Krishnaveni and E. Radhamani, *"Diagnosis and Evaluation of ADHD using Naive Bayes and J48 Classifiers"*, IEEE, 2016.

[6] M.S. Basarslan and I. D. Argun, "*Classification of A Bank Data Set on Various Data Mining Platforms*", IEEE, 2018.

[7] www.kaggle.com "Urinary Biomarkers for Pancreatic Cancer"

Authors

Biography

First A. Lin Lin Htun received the B.Sc (Physics), D.C.Sc and M.I.Sc degrees in Computer Science from University of Computer Studies, Mandalay (UCSM) in 2002, 2003 and 2006 respectively. She is now working as an Associate Professor in University of Computer Studies (Taunggyi) and her research interest includes Data Mining, Database Management System and Software Engineering.

Second B. Yin Su Hlaing received the B.C.Sc, B.C.Sc (Hons) and M.C.Sc degrees in Computer Science from University of Computer Studies, Mandalay (UCSM), in 2005, 2006 and 2009

respectively. She is now working as a Lecturer in University of Computer Studies (Taunggyi) and her research interest includes Data Mining, Database Management System and Software Engineering.

Third C. Yin Nyein Aye received the B.C.Sc, B.C.Sc (Hons), M.C.Sc and Ph.D (IT) degrees in Computer Science from University of Computer Studies, Yangon (UCSY), in 2005, 2006, 2009 and 2014 respectively. She is now working as an Associate Professor in University of Computer Studies (Taunggyi) and her research interest includes Data Mining, Database Management System and Software Engineering.

NAME ENTITY RECOGNIZATION FROM FOOD INFORMATION TO SUPPORT HEALTH CARE SYSTEM

April Su

Faculty of Computer Science, University of Computer Studies (Mandalay)
nwayapril97@gmail.com

ABSTRACT

Nowadays, the knowledge of extracted food data and their relationship to other biomedical entities is important for improving public health. However, this information is presented as separate text data. This means that the data has no predefined data format. Therefore, working with textual data is a challenge because of its variability. An automatic information extraction method is necessary to do it to utilize. Name Entity Recognition (NER) can be used for automated extraction of food entities from unstructured text. In the paper, the food entities are extracted by using rule-based NER method. The rule-based NER method extracts a sequence of words from the textual food data that match with a predefined pattern. Here NLTK POS tagger and USAS tagger is used to tag the words token associated with their semantic tag and part-of-speech tag. And then, the rules are generated with the helps of POS tagging and USAS semantic tagging. The generating rules are used to extract the food entities. Performance evaluation is measure by accuracy of food entities extraction.

KEYWORDS

Food Information Extraction, NER methods, Rules-based NER methods, POS tagger, USAS semantic tagger

1. INTRODUCTION

In different food information systems, only a few amounts of food information is highly relevant to the system [4]. In public health care system, the extracting food entities and the relations between these entities and other biomedical entities (like genes, drugs, diseases, etc.) are important for the system. In this paper, the extracting food entities can be utilized at public health care system. But, extracting the food entities from textual data is a challenging task because most of the food information are presented as heterogeneous data forms. Entities Extraction is an entity to shape structured data from unstructured data [7]. Unstructured data ensures that raw data such as social media such as Twitter, Facebook and Instagram are included. Food entities extraction is separating named entities from unstructured data. Entity Recognition is the most important technique for extracting, obtaining information, answering questions, and machine translation.

Information Extraction (IE) is the most important task of automatically extracting information from unstructured data. Especially, IE is deal with natural language processing (NLP) that is the processing of human language text by means of NLP. Basically, IE involves the process of structuring the text, extracting patterns in the structured data and finally does data analysis and interpretation. A concept of IE is to provide a structured representation of extracted information obtained from analyzed text. Name Entity Recognition (NER) is a well-known task of IE. NER enables an IE system to recognize and classify information units in an unstructured text. There are difference types of NER methods exist: *terminology-driven, rule-based, methods based on active learning (AL), and methods based on deep neural networks (DNNs)* [1]. Another type of named entity recognition methods for unstructured textual data are the hybrid approaches.

The paper mainly focuses on extraction of food products using rule-based NER

method. The rule-based NER method is based on computational linguistics and rules that combine statistical information. In the paper contains four main steps. Firstly, preprocessing step is performed in the unstructured text. In second step, the paper use Natural Language ToolKit (NLTK) Part-Of-Speech (POS) tagger and UCREL Semantic Analysis System (USAS) semantic tagger in order to get the morphological information from a textual data. In the next step, the specific rules are identified with the helps of POS tagger and USAS semantic tagger with the purposed of extracting the phrases in the text that are associated with food entities. The final step extracts the food entities by using the predefined rules. The resulted food entities are critical to support public health care system. The POS tagger gives information about roles of a word or a token in a sentence. Your POS labels, word tokens related to lemmas and semantic tags are supported by the USAS semantic tagger. Semantic tags describe semantic fields that combine senses of words that belong to the same contextual concept at some general level. Not just synonyms and antonyms, each group consists of hypernyms and hyponyms. To evaluate, the paper used a complexity matrix used to visualize the performance of NER methods. In the matrix, each row represents the features of the predicted class and each column represents the features of the actual class. [1].

2. RELATED WORK

IE is and effective way of populate the contents of a relational database. Once the information is encoded formally, it can be applied all the capabilities provided by database systems, statistical analysis packages, and other forms of decision support systems to address the problems to solve. Here, different methods are presented to be used for food related information extraction. The UCREL Semantic Analysis System (USAS) [6] is a system for automated semantic analysis of texts. It consists of 21 high level categories, and particular interested category is “food and farming”. However, one significant drawback is that USAS only works on a word level. FoodIE [3], contains rules engine. These rules are based on computational linguistics and terms that describe each food concept. Unlike USAS, FoodIE considers blocks of words when extracting and annotating food concepts. DrNER [8] is a rule-based NER method, the goal of this method is to identify information of evidence-based dietary recommendations. In the scope of this NER system, the nutritional information as well as food concepts are also participated as information.

NCBO Annotator [2] is a web framework that extracts and annotates food concepts from user-provided free-form text. An annotator's concepts and performance domains depend on the ontology chosen to be used in the background. The r-FG corpus is used to process Japanese recipe text with the help of ML approach. The r-FG corpus is comprised of Japanese food recipes only. A hybrid approach for NER [9] is trying to use POS tagger and syntactical structure of the document in combination with one of the semi-supervised learning algorithms to identify and categorize named entity. The POS and syntactical structure of the input data are applied to determine which is entity boundary and to construct extraction pattern. The extraction pattern then feed to one of semi-supervised algorithm which is Self-Training algorithm. One of the innovative hybrid approaches for NER system is developed and this hybrid NER system composed of rule-based deep learning and clustering-based approaches from unstructured text data [1]. The hybrid approach facilitates the extraction of generic entities out of natural language texts of domains that lack generic named entities labeled domain data sets. This approach takes the advantages of both deep learning and clustering approaches.

Electrical equipment malfunction text entity identification [10] is an object extraction system that uses a hybrid natural language processing algorithm. First, the system developed a dictionary-based method to identify proper nouns related to the field of electrical energy. Then, Time entities are

extracted using the Language Technology Platform (LTP) tool. With the help of BERT-CRF algorithm, it extracts strongly expandable objects including line names and manufacturer names. The system can perform statistical analysis based on organizational extraction to gain knowledge about preventive maintenance of failures from historical maintenance records, allowing maintenance to be more focused on equipment and components most likely to fail to prevent power pre-latent system failure. In entity recognition by NLP and Machine Learning [7], bi-directional long short terms memory is used to extract entities with conditional random fields. Bi-LSTM-CRF-CNN is a hybrid approach in which CNN layer is aimed at extracting sub-word information, CRF works well for POS tagging.

3. SYSTEM ARCHITECTURE

Figure 1 gives an architectural overview of the system. The system performs the food entities extraction in three main steps: Data cleaning, POS tagging, USAS semantic tagging and food entities extraction by using the defined rules. The system takes an unstructured data as input and produces the food entities. The performance is evaluated by using confusion matrix.

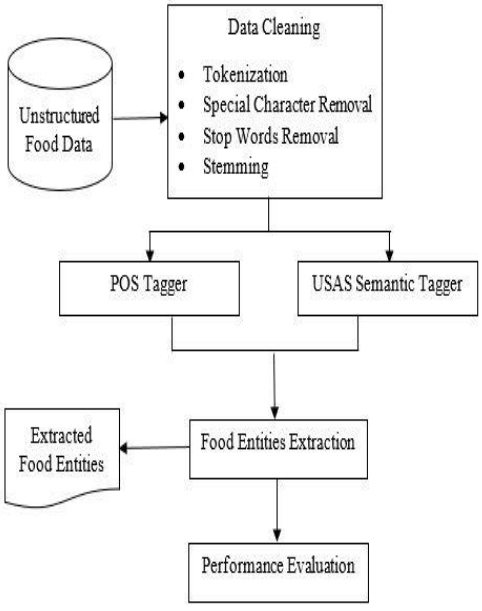


Figure 1. Food Entities Extraction System

3.1 Data

The input textual data was taken from http://cs.ijs.si/repository/FoodIE/FoodIE_datasets.zip and in this dataset contained 1,000 recipes data. These recipes obtained from Allrecipes which is the largest food-focused social network. The recipes were selected from five recipe categories: Appetizers and snacks, Breakfast and Lunch, Dessert, Dinner, and Drinks.

3.2 Data Cleaning

The input data obtained from various sources such as web sites, social media platforms and, other internet sources are unstructured data format. So, data preprocessing is needed to get structured data. The resulted data is used to extract and classify the named entities. NLTK libraries is used for NER. The following steps are done to clean data:

- Tokenization: Tokenization is a process of breaking up text into sentences and words.
- Special Characters Removal: The special characters such as punctuation, emoticons and non-alphabetic characters are removed from the text.
- Stopword Removal: The commonly used words such as am, is, are, can, you, he, etc. are removed.
- Stemming: Stemming is reducing related words to a common stem.

3.3 POS Tagging

POS tagging is a simply labeling process that gives an appropriate part-of-speech to a word or a token in a sentence. The part-of-speech show how a word is used in a sentence. There are eight main parts-of-speech: Nouns, pronouns, adjectives, verbs, adverbs, prepositions, conjunctions and interjections. POS tagging is a supervised approach to learning which uses features such as the previous words, next words, first letter capitalization, etc. NLTK has a POS tags feature and operates after the tokenization process.

The POS tagging is a crucial because the paper needs to search the food entities from the input sentences based on their tag sets. Mostly, the food entities may be noun (NN) or noun phrases (NNP). Therefore, this paper mainly focuses on NN or NNP pos tag. As an example, "*Add the steak and stir-fly for 2 minutes or until evenly browned.*" for this sentence, NLTK POS tagger will give the following result.

"*Add/IN the/DT steak/NN and/CC stir/NN fly/NN for/IN 2/CD minutes/NNS or/CC until/IN evenly/RB browned./NN*"

3.4 USAS Semantic Tagging

The USAS tagger is used for automatic semantic analysis of text. In this tagger, the revised tagset is arranged in a hierarchy with 21 major discourse fields expanding into 232 fine-grained semantic field tags. The following table shows the 21 labels at the top level of the hierarchy.

Tabel 1. Top level Semantic tagset

A	General & Abstract Terms
B	The Body & the Individual
C	Arts & Crafts
E	Emotional Actions, States & Processes
F	Food & Farming
G	Government & the Public Domain
H	Architecture, Building, Houses & the Home
I	Money & Commerce
K	Entertainment, Sports & Games
L	Life & Living Things
M	Movement, Location, Travel & Transport
N	Numbers & Measurement
O	Substances, Materials, Objects & Equipment
P	Education
Q	Linguistic Actions, States & Processes
S	Social Actions, States & Processes
T	Time
W	The World & Our Environment
X	Psychological Actions, States & Processes
Y	Science & Technology
Z	Names & Grammatical Words

Your POS labels, word tokens related to lemmas and semantic tags are supported by the USAS semantic tagger. Semantic tags describe semantic fields that combine senses of words that belong to the same contextual concept at some general level. Not just synonyms and antonyms, each group consists of hypernyms and hyponyms. To evaluate, the paper used a complexity matrix used to visualize the performance of NER methods. In the matrix, each row represents the features of the predicted class and each column represents the features of the actual class.

On paper to define input text phrases associated with food entities, the USAS semantic tagger is used to define these phrases. The semantic tagger firstly, tag the word token with its appropriate semantic tag. Then, the specific rules are defined by using this tag in order to determine the food tokens in the input text. For example, the sentence "Add the steak and stir-fly for 2 minutes or until evenly browned." is processed by USCS tagger. The resulted is presented by below table.

Table 2. Semantic tags obtained from USAS tagger for one sentence

ID	Token	Semantic Tag
1	Add	N5+ / A2.1
2	the	Z5
3	steak	F1
4	and	Z5
5	stir	A1.1.1
6	fly	L2
7	for	Z5
8	2	N1
9	minutes	T1.3
10	or	Z5
11	until	Z5
12	evenly	N6+
13	browned	O4.3
14	.	PUNC

3. Food Entities Extraction

The POS tag obtained from POS tagging and the semantic tag derived from USAS semantic tagging are used to create the food entities extraction rules. Every word with the POS tag and semantic tag are iterated through the predefined rules. If a word or token meet with a rule, the word is extracted

as food entity. In this paper, mentioned below rules are used to extract the food entities.

Tabel 3. Rules for Food Entities Extraction

ID	Rule
R1	$S_t = F_1 \vee F_{1+} \vee F_{1-} \wedge P_t = NN \vee NNP$
R2	$S_t = F_2 \vee F_{2+} \vee F_{2-} \wedge P_t = NN \vee NNP$
R3	$S_t = F_3 \vee F_{3+} \vee F_{3-} \wedge P_t = NN \vee NNP$
R4	$S_t = F_4 \vee F_{4-} \wedge P_t = NN \vee NNP$
R5	$S_t = O_{1.1} \vee O_{1.2} \wedge P_t = NN \vee NNP$

Where, S_t = Semantic Tag
P_t = POS Tag
F_1 = Food
F_{1+}= Abundance of food
F_{1-} = Lack of food
F_2 = Drinks and alcohol
F_{2+} = Excessive drinking
F_{2-} = Not drinking
F_3 = Non medical drugs
F_{3+} = Drugs abuse
F_{3-} = No use of drugs
F_4 = Farming & Horticulture
F_{4-} = Uncultivated
$O_{1.1}$ = Solid
$O_{1.2}$ = Liquid

4. PERFORMANCE EVALUATION

The performance of the system is evaluated by comparing the outputs of the food entities extraction with the annotations found in the ground truth dataset. The confusion matrix is used to measure the performance of the system and 1000 recipes from each recipe category are used as the evaluation test set. In a confusion matrix, four metrics can be defined: true positive (TP), true negatives (TN), false positive (FP), false negative (FN). For this system, only three metrics are selected to measure the performance.TN is not selected, because the system only interested in food entities, if a word or token is not identified as a food entity, it can be assumed that it is not a really food entity, and it is a TN.

These evaluation metrics are defined as:

- TP - This type of match occurs when all tokens from the system exactly match the same food item in the ground truth dataset.
- FN – This type of match occurs unless there is a certain comment that should be identified as an actual food. It occurs when the system does not extract a food item correctly.
- FP - This is a matching type from FN. This is what happens when an extractive food organization goes wrong. This occurs when something that is not a food is classified as one.

$$\text{Precision (P)} = \frac{TP}{TP+FP}$$

$$\text{Recall (R)} = \frac{TP}{TP+FN}$$

$$\text{F-measure (F1)} = \frac{2 * Precision * Recall}{(Precision + Recall)}$$

The following performance evaluation metrics for each recipe category show that the FN of dinner category more less than the FN of breakfast/lunch category. However, the FP rate of breakfast/lunch category provided is much greater than the FP rate of drink category. Furthermore, the system obtained 96% for precision (P), 94% for recall (R) and 96% for F1-score on the 1000 recipes of all recipes categories. Looking at the results, the system gives very promising and consistent results.

Tabel 4. Prediction for 1000 recipes

TP	FP	FN
11461	258	684

Tabel 5.Evaluation metrics for 1000 recipes

Precision (P)	Recall (R)	F1-score (F1)
0.9593	0.9437	0.9605

Tabel 6. Prediction for each recipe category

Category	TP	FP	FN
Appetizer/snacks	2147	27	162
Breakfast/lunch	2443	33	82
Desserts	2612	87	127
Dinner	3176	47	223
Drinks	1083	64	90

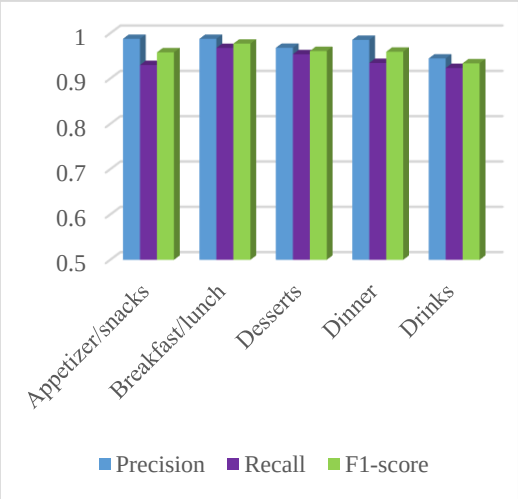


Figure 2. Performance Evaluation for each recipe category

5. CONCLUSIONS

This paper presents a rule-based NER approch for food information entities extraction where the rules are mostly concerned with the computational linguistics and semantic information that describe the food entities. Therefore, in this system, NLTK POS tagger and USAS semantic tagger are used to obtain more precious information about the input data. The main goal of the system is to provide a structrued representation of extracted food entities obtained from analyzed texual data. This resulted structured representation is critical for improving public health care. In addition, the goal of this system is to link food drawn from other fields, such as food recommendation systems, biotechnology, consumer and social sciences.The performance of the system show that the rule-based NER method for food entities extraction gives very consistent and accurate results.

In future work, the system planning to upgrade the food entities extraction NER method to support the extraction of insights from data relevant to all areas of data science.

REFERENCES

[1] *Anu Thomas, S. Sangeetha, "An innovative hybrid approach for extracting named entities from unstructured text data". In the proceeding of the Computational Intelligence (2019), pages 1-28.*

[2] *C. Jonquet, N. Shah, C. Youn, C. Callendar, M. Storey, and M. Musen, "NCBO Annotar: Semantic annotation of biomedical data", in International Semantic Web Conderence, Poster and Demo session, vol. 110 (2009).*

[3] *G. Popovski, B. K. Seljak, and T. Eftimov, "FoodIE: A rule-based named-entity recognition method for food information extraction", in Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods, (ICPRAM 2019), pp. 915-922.*

[4] *G. Popovski, B. Seljak, Eftimov (2020), "A survey of named-entity recognition methods for food information extraction", IEEE Access.*

[5] *Mori, S., Sasada, T., Yamakata, Y., and Yoshino, K. (2012), "A machine learning approach to recipe text processing", In Proceedings of the 1st Cooking with Computer Workshop, pages 29-34.*

[6] *Rayson, P., Archer, D., Piao, S., and McEnery, A. (2004), "The UCREL Semantic Analysis System".*

[7] *Shelake, V. Honmane (2020), "Entity Recognition by Natural Language Processing and Machine Learning", International Research Journal of Engineering and Technology (IRJET), vol. 7, Issue 9.*

[8] *T. Eftimov, B. Korousic Seljak, and P. Korosec (2017), "A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations", PloS One, vol. 12, no. 6.*

[9] *Y. Sari, F. Hassan, N. Zamin (2010), "Rule-based Pattern Extractor and Named Entity Recognition: A Hybrid Approach", IEEE.*

[10] *Z. Kong, C. Yue, Y. Shi, J. Yu, C. Xie and L. Xie (2021), "Entity Extraction of Electrical Equipment Malfunction Text by a Hybrid Natural Language Processing Algorithm", IEEE Access, vol. 9.*

CLASSIFICATION OF WEATHER CONDITION FROM IMAGES BY USING CONVOLUTIONAL NEURAL NETWORK

May Phyu Phyu Thaw[1] and Thein Htay Zaw[2]

[1]Faculty of Computer Science, [2] Department of Information Technology Supporting and Maintenance, University of Computer Studies (Mandalay)
[1]mayphyuphyuthaw61@gmail.com, [2]theinhtayzaw.cu@gmail.com

ABSTRACT

In ordinary routine, weather phenomena are quite significant. The system is composed of three steps namely pre-processing step, train step and test step. This system train and test by using Convolutional Neural Network (CNN) method. By using the CNN approach, the crucial features are automatically extracted from the image. Convolutional Neural Network is a method of deep learning in computer vision and the most used technique for the classification of image dataset. The weather dataset available from the kaggle.com was utilized in this strategy. As a consequence, the performance of the designed and constructed model is presented in this study. The experimental result shows the highest accuracy 87% in Conv6 using Epochs12 and the lowest accuracy 70% in Conv3 using Epochs12 for testing images 20.

KEYWORDS

Image Classification, Deep Learning, Pre-processing, Convolutional Neural Network (CNN), Matlab

1. INTRODUCTION

The weather affects many visual systems, such like outdoor video surveillance and car assistant driving systems, in addition to having a significant impact on our daily activities through the solar energy system and outdoor athletic events, for example. In previous years, one of the most complicated issues in visual information indexing and retrieval systems is the automatic classification of images. Typically, each image is composed of a group of pixels, and each pixel serves as a space for data storage. More calculations, configurations, and computer power are required for image classification [2].

In this study, we provide CNN based feature extraction and classification method for weather classification system. Although the earlier research offer intriguing ways to classifying the weather, these methods are unappealing. The dataset is quite difficult because it was noted that the histogram of mean lightness of sunny and cloudy images greatly overlaps in [17]. We primarily blame the manufactured image features used in this method for the system's poor performance.

Various elements, including as illumination, reflection, scene, and shadow, affect weather classification from images differently than they do with ordinary image classification tasks. Due of the close relationships between these variables, the categorization manifold exhibits a significant degree of nonlinearity. Discrimination across weather classes is a challenging task due to the nonlinearity of the categorization manifold, which might meet certain desirable features and reduce other negative properties from these components but cannot be properly captured by previous constructed systems.

A neural network model called CNN is used to record nonlinear mapping between various spaces, such as feature space and label space. In comprehensive image representation and classification tasks, Deep CNN has proven its strong

discriminating ability. Additionally, CNNs are clear end-to-end convolutional architectures that are straightforward and may streamline the classification of the weather without the use of manufactured characteristics.

Section 2 presents the related works and section 3 outlines the dataset we used. Information on the experimental deployment of the suggested classification model is detailed in Section 4. Reviewing the experiment's observations, Section 5 analyses the CNN model using a variety of assessment metrics. The paper is concluded in Section 6, which further highlights the opportunities for subsequent research.

2. RELATED WORKS

The related works associated to weather classification system are highlighted in this section. Numerous studies have been concluded on the weather classification. Due to the weather conditions with multi labels and time conditions, there is still no ideal answer in this area.

Hand-crafted feature combined with SVM classifier for image classification was used in [3]. By computing attributes using data that is already present in each image, hand-crafted features aim to characterize each image. For each image, these manually created features are computed and then utilized as SVM inputs. By analysing 14 different statistical measures on the output of each image after going through various transformations, 308 characteristics are computed on each image.

The system applied regions of interest (ROI) and Support Vector Machine (SVM) for weather image classification [4]. ROI used for feature extraction and it gathers information about the overall effect of weather on the image. Strong features are indicated by small values, whereas discriminant traits are indicated by huge values. The ability to add kernel approaches within the algorithm is one of the benefits of SVMs. They contain non-linear decision limits. But SVM cannot extract features from images and it needs to combine other feature extraction methods.

Breast cancer was categorized using artificial neural networks in a study of [6]. Radial Basis Function Network is used to further improve the findings that were achieved. In terms of performance, the radial basis function network is faired better than the ANN.

Recurrent neural network (RNN) can also use for Satellite Image Classification. The authors demonstrated how RNN efficiently picks up an iterative technique that greatly enhances the accuracy of satellite image classification maps in [7]. RNNs are successful at learning iterative techniques for tasks involving classification improvement.

CNN is used to automatically generate features and combined it with the classifier to classify the data. The testing time of CNN is significantly faster even though the training time is very large [5]. The outcome of this study is expected to classify the conditions of the weather and display accuracy result.

Above the explanations, the algorithm classified the weather using convolutional neural networks (CNN). This method was used to demonstrate that it can conduct feature extraction and classification. The authors used CNN to prove that it can perform Feature Extraction and Classification in [1]. It may be inferred that having many feature maps enables them to search for various patterns in various parts of the input image. It is a kind of multilayer neural network with biological inspiration. Animals' visual occipital cortexes are known to be intricate cell networks.

How responsive a cell is would be determined by its tiny receptive field, which is a portion of the visual fields.CNN intends to simulate this structure by extracting the features from the input space in a prominent example before completing the classification, in contrast to usual approaches where features are manually extracted and then supplied to the model for classification. This methodology can deal with such situations by utilizing the ideas of local connectivity, parameter sharing, and pooling/subsampling.

3. WEATHER DATASET

In this paper, the system uses Weather Dataset by Vijay Gupta, Kaggle Expert, Delhi, India. The dataset has five categories such as cloudy, rainy, shine, sunrise and foggy. It consists of 1500 images which are classified into 5 weather conditions and all images are .JPG & .JPEG format. Among them, the system used 1155 images and each class has 231 images. Train set consists of 1105 images partitioned into 5 sub directories and each sub directory has about 221 train images. Test set consists of 50 images partitioned into 5 sub directories with 10 test images per conditions.

(a)Cloudy

(b)Foggy

(c)Rainy

(d)Shine

(e)Sunrise

Figure1. Sample Weather Images from Five Categories of the Weather Dataset: (a) Cloudy, (b) Foggy, (c) Rainy, (d) Shine and (e) Sunrise

4. SYSTEM ARCHITECTURE

This system performs the classification in three main steps: pre-processing step, train step and test step. The system takes the images dataset as input and produces the condition of the weather as output. The input dataset obtained from Kaggle.com.

4.1. Pre-processing Step

Pre-processing is a procedure that enables improved outcomes once additional images have been trained and warped. It can offer advantages and resolve issues that might help features be identified more accurately. To alter the number of pixels overall, an image should always be resized [9]. Whenever an image resizes, its pixels are updated. It is very important to extract features [10]. In this system, the images were resized 150*250*3 pixel.

After resizing the image, it is filtered using geometric mean filter to enhance image. One of the image processing approaches

that is frequently employed in a broad range of applications is image enhancement [11].A particular method of calculating the average of a series of numbers is the geometric mean. [12].The following can be deduced and calculated

$$f^{\wedge}(x,y) = [\prod_{(s,t)\in Sxy} g(s,t)]^{1/mn} \quad (1)$$

4.2. Train Step

Secondly, the system needs to build the network that trains the image dataset. The image of weather dataset is used in this stage to extract features. Features are detected by filters to obtain the important features for classification. The system uses Convolutional Neural Network not only in train step but also in test step.

Convolutional Neural Network (CNN) method instantly extracts important features from the image by itself. CNN is a sequence of layers and every layer performs some unique functions on the dataset [13]. The input, hidden, and output are the layers of CNN.

Ordinarily, an image is constructed of a matrix of pixels, and the input layer obtains these pixel values [14]. The output layer, which is typically used to assign an image to a certain class, is a fully connected layer. The hidden layers are convolutional and pooling layer.

In response to numerous feature detectors, such as filter, the convolutional layer condenses the input by extracting features of interest from it and generates feature maps [10]. Outputting volume of convolutional layer (W2*H2*D2) can be computed as following formula

$$W_2 = (n+2p-f)/s+1 \quad (2)$$
$$H_2 = (n+2p-f)/s+1 \quad (3)$$
$$D_2 = k$$

When the system gets the features from the convolutional calculation, they are included a various pixel values. And to avoid the explosive growth in the computing necessary to run the neural network, these signal matrix are constructed using an activation function termed as the Rectified Linear Units (ReLUs) function [15]. Pooling layer helps to reduce noise feature and increase processing speed [16]. And these layers can perform their processes repeatedly according to the author recommend to get good result. ReLU Activation Function is calculated with the equation

$$f(x) = \max(0,x) \quad (4)$$

Top layer of a network that gathers the far more sophisticated feature is the most recent layer of CNN, which is fully connected. Each neuron is connected in this layer, which reduces the data from the pooling layer to a single dimension [8]. And the classification process is carried out by using classifier called activation functions such as Softmax function. Softmax function tests the result by the trained data for the classification. Softmax function can be defined as

$$S(y_i) = \frac{\exp(y_i)}{\sum_{j=1}^{n} \exp(y_i)} \quad (5)$$

The result of softmax function is passed into loss function to detect the difference between actual value and prediction values. The cross-entropy loss function is generated by

$$L = -\sum_{j=1}^{M} y_j log(y_j^{\wedge}) \quad (6)$$

Then, it performs iterative backward passes which attempt to minimize the difference between the known correct label and the model prediction. In each backward pass, the weights move towards an optimum that minimizes the loss value and results in the most accurate prediction. Finally, the system obtains the trained model to test the result.

4.3. Test Step

This step is the final step of the system. It performs classification for the system. The tested data is tested by the trained data for the classification. Classification is the process of classify image into weather condition and give accuracy percentage to the user. The classifier demonstrates the predicted class with the highest probability. The output of the picture input is the class with the highest probability. End of the all

process, it produces the output that the input image to which class it belongs to.

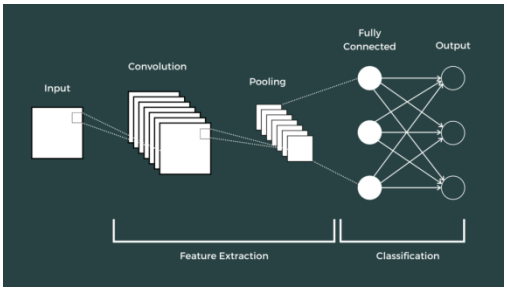


Figure2. Architecture of CNN

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.theclickreader.com%2Fintroduction-to-convolutional-neural networks%2F&psig=AOvVaw3XJZPCvpkOG2ldQWSEbrhw&ust=1668100005976000&source=images&cd=vfe&ved=2ahUKEwjw79vzyqH7AhVrMLcAHd-JAQIQr4kDegUIARDhAQ

5. EXPERIMENTS

5.1. Experimental Setup

The feature representations and CNNs on the weather image dataset, which has just been released in Kaggle.com are assessed in this paper. Five classes make up this dataset (total of 1,500 images). The training portion of the dataset for Weather-CNN can be randomly split into two subgroups. The first selection is used for optimizing/ fitting Weather CNN model and another one is used for validation. The classifier was trained and tested using images from each category, which comprises cloudy, foggy, rainy, sunshine, and sunrise images.

The system used Conv3, Conv4, Conv5 and Conv6. Each result of the convolutional layer performs ReLUs function and pooling layer function. So, Conv-3 calculates ReLUs function and pooling layer function three times. After the several runs of the experiment were completed, the mean Loss function was generated for every one of the trained models using the data from each of the categories.

5.2. Results and Discussion

The effectiveness of suggested system is analysed by measuring the evaluation criteria on the weather classification. It is a great performing model. These tables showed maximum, average and minimum accuracy results of five weather conditions in test set.

In the first part of experiments, the system used Conv3, five Fully Connected Layers, Epochs 12, Training 211images and Testing 20 images to classify weather. The result for experiments is shown in table 1.

Table1.The performance of CNN (Conv3)

	Cloudy	Foggy	Rainy	Shine	Sunrise
Cloudy	13	2	2	3	0
Foggy	4	7	6	1	2
Rainy	3	2	15	0	0
Sunshine	2	2	0	15	1
Sunrise	0	0	0	0	20
Accuracy	70%				

In the second part of experiments, the system used Conv4, five Fully Connected Layers, Epochs12, Training 211 images and Testing 20 images to classify weather. Table 2 displays the experiment results.

Table2. The performance of CNN (Conv4)

	Cloudy	Foggy	Rainy	Shine	Sunrise
Cloudy	6	8	3	3	0
Foggy	1	4	0	0	0
Rainy	2	4	12	1	1
Sunshine	1	0	0	19	0
Sunrise	0	1	0	0	19
Accuracy	73%				

In the third part of experiments, the system used Conv5, five Fully Connected Layers, Epochs 12, Training 211 images and Testing 20 images to classify weather. Table 3 presents the experiment results.

Table3. The performance of CNN (Conv5)

	Cloudy	Foggy	Rainy	Shine	Sunrise
Cloudy	6	6	3	5	0
Foggy	1	17	0	0	2
Rainy	1	9	10	0	0
Sunshine	0	2	0	17	1
Sunrise	0	0	0	0	20
Accuracy	77%				

In the last part of experiments, the system used Conv6, five Fully Connected Layers, Epochs 12, Training 211 images and Testing 20 images to classify weather. Table 4 states the experiment results.

Table4. The performance of CNN (Conv6)

	Cloudy	Foggy	Rainy	Shine	Sunrise
Cloudy	14	2	1	3	0
Foggy	2	15	2	0	1
Rainy	0	0	20	0	0
Sunshine	0	1	0	19	0
Sunrise	0	0	0	1	19
Accuracy	87%				

Above the experimental result, the system depends on the numbers of convolutional layer (Convs). If the system uses more convolutional layers, it will get more accuracy result. Moreover, Convolutional Neural Network (CNN) depends on the size of the dataset. It can give high performance result when it uses a huge dataset. This system applies the dataset included about 1500images because of hardware demands. But it got 87% high accuracy result.

6. CONCLUSION

This system states the implementation of Convolutional Neural Network (CNN) technique for weather classification. The overview of the constructed weather classification model is summarized in this paper. It displays convoluted results after all of the processing processes have been completed. This model was implemented using a dataset that was acquired from the Kaggle.com website. This image comprises a set of only original data. It generates results with 87% accuracy for Conv6, 77% accuracy for Conv5, 73% accuracy for Conv4 and 70% accuracy for Conv3. According to Convolutional Layer, the accuracy result can change. In this system, the highest accuracy result is 87% in Conv6. So, it can be concluded that the performance of the model is acceptable.

ACKNOWLEDGEMENTS

I want to express my gratitude to everyone who helped and supported this system.

REFERENCES

[1] J.D. Bodapati and N. Veeranjaneyulu, (2019) Feature extraction and classification using deep convolutional neural networks, Journal of Cyber Security and Mobility, pp.261-276.

[2] M. Elhoseiny, S. Huang and A. Elgammal, (2015) September, Weather classification with deep convolutional neural networks, (ICIP) (pp. 3349-3353), IEEE.

[3] W. Zhang, B. Pogorelsky, M. Loveland and T. Wolf (2021) Classification of Covid-19 X-ray Images Using a Combination of Deep and Handcrafted Features, arXiv:2101.07866v2 [eess.IV] 21 Jan 2021.

[4] M. Roser and F. Moosmann, (2008) Classification of Weather Situations on Single Color Images, IEEE, June 4-6, 2008.

[5] D.P. Mishra, T.K. Tripathy and S. Chakraborty, (2018) CNN for Butterfly Classification, International Conference on ICRIEECE.

[6] S. Kaymak, A. Helwan and D. Uzun, (2017) Breast cancer image classification using artificial neural networks, ICSCCW 2017, Budapest, Hungary.

[7] E. Maggiori, G. Charpiat, Y. Tarabalka and P. Alliez, (2017) Recurrent Neural Networks to Correct Satellite Image Classification Maps, arXiv:1608.03440v3 [cs.CV] 21 Apr 2017.

[8]https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiG3_Gmr7T6AhVdDrcAHeLUCfUQFnoECAcQAQ&url=https%3A%2F%2Ftowardsdatascience.com%2Fconvolutional-layers-vs-fully-connected-layers-364f05ab460b&usg=AOvVaw0TmzVuHvTHjsTrKZezfJ93

[9]https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwj19ZP6pbT6AhWA9XMBHft8DjgQFnoECAMQAw&url=https%3A%2F%2Fsisu.ut.ee%2Fimageprocessing%2Fbook%2F3&usg=AOvVaw0Uo15KA8Q6X8M9EwbXh_U2

[10]https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiy15MprT6AhUy9XMBHdRfD1sQFnoECAoQAQ&url=https%3A%2F%2Fwww.mygreatlearning.com%2Fblog%2Ffeature-extraction-in-image processing%2F&usg=AOvVaw1IXfLc4O4ZWikWmGifI577

[11]https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwi6qbOjprT6AhWGxnMBHSedCbIQFnoECAYQAQ&url=https%3A%2F%2Fwww.sciencedirect.com%2Ftopics%2Fcomputer-science%2Fimage enhancement&usg=AOvVaw1AXBESlWFKs5MDTgwoSvp4

[12]https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjp3fe6prT6AhXJyXMBHZFMACwQFnoECAQQAw&url=https%3A%2F%2Fbyjus.com%2Fmaths%2Fgeometric-mean%2F&usg=AOvVaw30vbGfRcrEmXOrfpts2mVd

[13]https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwj2i7X9q7T6AhVEDLcAHbvMDjcQFnoECAkQAQ&url=https%3A%2F%2Ftowardsdatascience.com%2Fapplied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2&usg=AOvVaw1chCNpqFeQkWk2wtL4R0AK

[14]https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwi_0v6mrLT6AhX70XMBHc8iBo4QFnoECBEQAQ&url=https%3A%2F%2Ftowardsdatascience.com%2Fconvolution-neural-network-for-image-processing-using-keras-dc3429056306&usg=AOvVaw2ri3xpFcTQeEGtaGbUukbC

[15]https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwj305nK1bf6AhWvCrcAHRPzBGIQFnoECAMQAw&url=https%3A%2F%2Fwww.baeldung.com%2Fcs%2Fml-relu-dropout-layers&usg=AOvVaw22w3aOIMAoKmpEYCBex3M5

[16]https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwju5eXjrrT6AhVOCLcAHbitCFoQFnoECAcQAQ&url=https%3A%2F%2Fanalyticsindiamag.com%2Fcomprehensive-guide-to-different-pooling-layers-in-deep-learning%2F&usg=AOvVaw1CDbhH0IZ8PQHOJjYVU94L

[17] Cewu Lu, Di Lin, Jiaya Jia, and Chi-Keung Tang, “Two-class weather classification,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 3718–3725

Authors

Biography

First A. Daw May Phyu Phyu Thaw was graduated B.C.Sc(Q) in 2020. Now, she prepares for (M.C.Sc-Thesis) and works at Faculty of Computer Science in University of Computer Studies (Mandalay). She is interested in Artificial Intelligence (AI) majored in Image Processing.

Second B. Dr. Thein Htay Zaw

PREDICTING STUDENTS' LEARNING EDUCATION RESULT

Ohnmar Aung[1], Aye Pyae Sone[2]

[1]Faculty of Information Science Department, Computer University (Mandalay)
[2]Faculty of Computer Science Department, Computer University (Hpa-an)
[1]mamalay2009@gmail.com, [2]maysonesone@gmail.com

ABSTRACT

Data Mining technology can determine useful information from low data. The prediction is data mining technique which can predict the situations in future from the data of current state. The prediction is the combination technique of classification and clustering. The main purpose of this research paper is to analyse and compare the performance evaluation of the data mining techniques for students' learning evaluation based on students' academic dataset using four popular data mining algorithms. This paper analyses the performance results of these techniques and then these have been used to build the models of the system. According to the performance of this system, Support Vector Machine (SVM) classifier achieved 90.3% average accuracy on the students' learning evaluation result that results are evaluated on the 80% training dataset on the total dataset. SVM is the better classifier that other classifiers which are used in this work.

KEYWORDS

Support Vector Machin, C4.5 Classifiers, Decision Tree, Naïve Bayes, Data Mining, Prediction

1. INTRODUCTION

Data Mining is a group of methods, which applies to complex and large datasets. Nowadays, data mining has attracted a great attention for information industry and multimedia society. There are many classification algorithms in Data Mining approaches to predict the result on the large amount of dataset. In the previous literature, many researchers approved the data mining algorithms in the wide area of real-world applications such as Finance, Healthcare, Telecommunication, Marketing, Education, E-commerce and etc.

In education world, education is deliberate and conscious effort to apply the process of learning techniques for leaners developing in their education. For many universities, higher education is important education area in which various education levels have required role to improve learners' knowledge in the successful of technology and science by applying and observing the humanities value such as human resource empowerment of individual interest field of their talent and interest.

In the management case of students' educational learning process, develop activities that improve every year and produces large amount of information such as students' profile and results of academic that to measure performance of education work to improve the students' learning. All collected information are being stored in the database of education because of the information amount that usually occurs hardly to find and analyse the patterns for knowledge discovery to be ignored.

Discovery of knowledge can be accepted through many difference predicting methods such as data mining which refers to the large quantities of data analysis that are collected in the database [1]. The various algorithms of data mining approaches divided as classification, clustering, association rules and selection which usually create the model that achieved from the training dataset related to the label of target for prediction on the label of target [2].

The purpose of this paper is to describe the comparison on the evaluation performance of applied data mining classifiers such as Naïve Bayes, Decision Tree, C4.5 and Support

Vector Machine classifiers. Moreover, this paper studies to describe the main outline of source dataset likes participation level of students in the educational organization prediction model creation.

2. Related Works

Data Mining techniques in the database of academic have been applied by researchers to improve, predict or understand the academic result of students. For example, standard process of cross-industry for mining on data and decision tree classification are used for prediction on the final marks of semester using variables from 40 dataset of graduated degree, master degree and other researchers create a model to predict the final average grade of students from 230 datasets on the under graduated students in the university [3].

Moreover, many researchers developed the research for the comparing on the different kinds of algorithms in data mining to predict the final marks of semesters and to analyse the evaluation result of each classification algorithms based on confusion matrix approach to discover the accuracy and precision of algorithm [4]. Other literatures developed in studying to find deeper on the dataset of students that are extracted from the different kind of academic dataset related with comparing and creating to get the optimal data mining approach of prediction on the educational work [5].

In the past, most researchers developed on the prediction of students' results performance based on the data mining approaches in the database of academic and then analysed on the various data mining classifiers for each research that was developed in their proposed literatures [6]. In this paper, the study will discovery on various models with large amount of dataset which does not usually create on the academicals dataset to perform the using of datamining techniques for the main goal of data mining in the field of education.

3. Classification In Data Mining

Every classifier has various quality which differential classifier. There are two steps in the classification process such construction a classification model and testing model for unknown dataset. The classification model creating is shown in figure 1. In this creation, training data is required to classify the testing unknown data. And then figure 2 shows the testing model for the testing of the unknown data in the classification step.

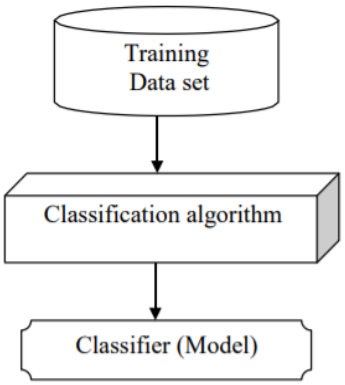


Figure 1. Construction of classification model

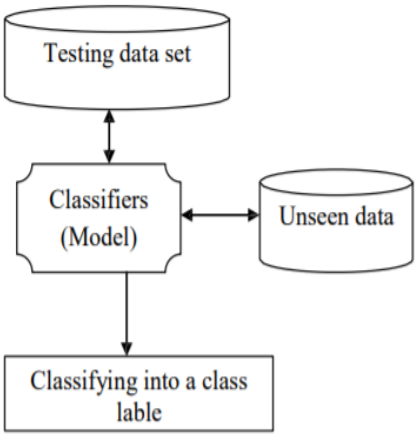


Figure 2. Construction of Testing model

Every classifier has properties that are known as the classification characteristics in the useful classifiers. The characteristics are correctness, time, strength, data size and extendibility.

Correctness: In the classifier, how classify the tuples to get the target accuracy depend on this characteristic. To verify, the accuracy, there are various numerical values techniques depend on the number of correctly classification tuples and wrong classification tuples.

Time: In the classifier how much time is needed to create the model? This fact also involves the time to apply by the crated model to classify the tuples that refers to the cost of computation.

Strength: It is the ability to divide tuple correctly even tuples consisting a noise. Noise can produce the missing value or wrong value for the classification results.

Data Size: Every classifier should be free format on the dataset size. Classification model should be scalable format and the performance of the classification model is not based on the database size.

Extendibility: Most new features can be combined whenever needed in the classification step. This feature is difficult to use in the implementation of the classifier.

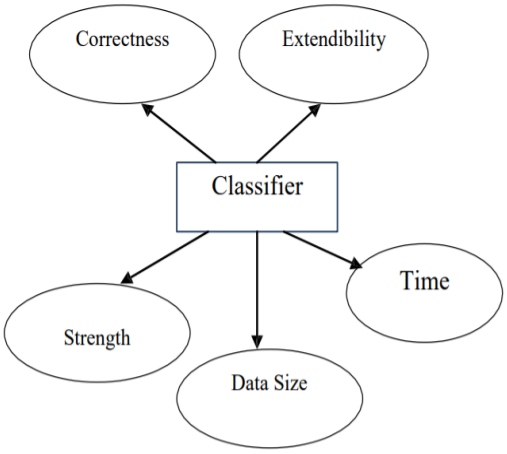


Figure 3. Characteristics of Classifiers

4. METHODOLOGY

In this system, the following important steps are used such as understanding, preparation, and transformation of data, modelling and evaluation on the applied datamining classifiers. The flow diagram of the proposed system is shown in figure 4. To implement the proposed system "Higher Education Students Performance Evaluation Dataset Data Set" is used for the evaluation result.

4.1. Data Understanding

Data understanding takes a closely related information from dataset and determine the data quality that applied on the process of data mining. Dataset of this system are collected from the information of students from 2016 to 2020 in the Faculty of Computer Science dataset and then interview organization of students and students in author's university. In this step, all data of this system are organized and collected in Microsoft 2016 excel sheet.

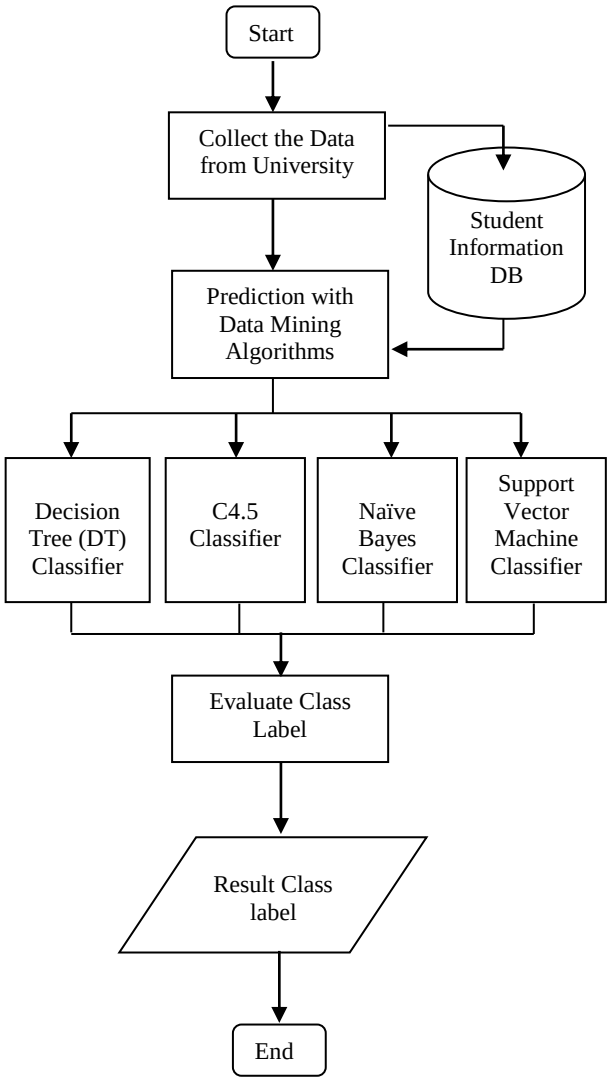


Figure 4. Proposed System Design

4.2. Data Transformation and Preparation

In this step, cleaning on data, reduction, selection and partition for attributes are applied to achieve the better performance model. To improve the performance of model based on data mining approaches, some irrelevant variables are eliminated from the collected dataset.

4.3. Models with Data Mining Approaches

In this paper, models for classification are created by using popular four classification methods and then these classifiers are described in the next section. The implementation of the system was created using Python Programming Language.

5. BACKGROUND THEORY

5.1. Decision Tree (DT) Classifier

Decision Tree algorithm is one of the approaches in predictive modelling, which was used in data mining and statistics approaches. To perform the decision making and to represent the decision, decision trees are used among the popular data mining approaches. The main goal of the decision tree is to build the prediction model that predicts the value of variables in target variables based on the various input variables [7].

The classification tree or decision tree is a tree in which each non leaf (internal node) is labelled using features of input. Each leaf node of the decision tree is labelled with probability distribution on the predefined classes, that classified by decision tree into the particular distribution probability or into the predefined classes. In the data mining approaches, decision tree can be used as the computational techniques and mathematical techniques for the categorization generalization and description on the collected data. The process of decision tree is shown in figure 5.

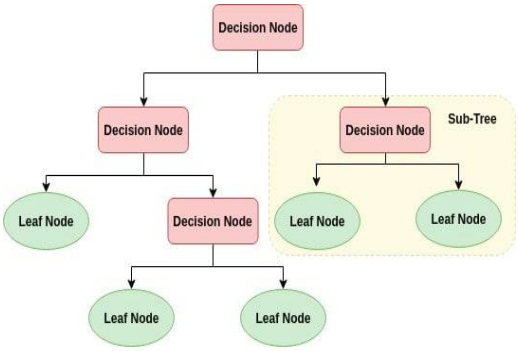


Figure 5. Decision Tree Classifier

5.2. C4.5 Classifier

The C4.5 classifiers is one of the most popular data mining approaches that applied to create a decision tree for a prior expansion calculation. C4.5 classifiers is the enhancement of ID3 classification algorithm. The classification model is created by C4.5 classifiers for grouping that represents for statistical classification method. It analyses the training dataset and create the classifier model that accurately manage for testing and training dataset [8].

C4.5 classifiers is better performance especially in various problems which are predefined as decision tree classifiers. However, C4.5 classifiers does not perform in all problems of classification C5.0 is improvement version of C4.5 on accuracy, speed and memory cost. In many research areas, C4.5 classification method is used to get the better classification model [9]. The structure of C4.5 classifiers is described in figure 6.

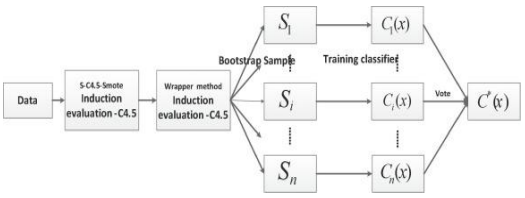


Figure 6. C4.5 classifiers

5.3. Naïve Bayes Classifier

Naïve Bayes classifiers is also one of the popular classification methods of data mining. It is not a single classification algorithm that based on Bayes' Theory but algorithms family

where all algorithms of them provide or common principle. It calculates the probability of occurring event with given probability of another event that has been already performed [10].

Naïve Bayes classifiers are used in analysis of sentiment calculation and recommended system etc. These classifiers are very easy and fast to develop but disadvantage of these classifiers is predictor requirements to be improved. In the real-world application areas, the dependent predictors are applied to get the better performance of the classification process [11]. The structure of Naïve Bayes classifier is shown in figure 7.

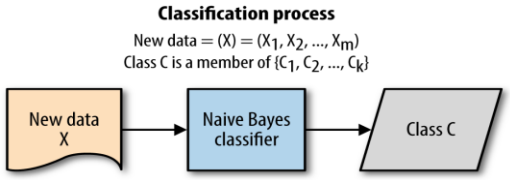


Figure 7. Naïve Bayes Classifier

5.4. Support Vector Machine (SVM) Classifier

Support Vector Machine is the most supervised learning classification algorithm among the popular data mining classification algorithm which can be applied for regression or classification challenges. The main goal of SVM classifier is to build the best decision boundary or line, that divided n-dimensional value into the predefined classes. This reason to define the new point of the dataset into the predefined categories in the future which optimal decision boundary is defined as hyper plane. That provide the best separation of the classes [12].

Thereby, the largest possible distance creating between the separated hyper plane. The effectiveness of SVM classifiers don't depend on the classified entities dimension. SVM is the best accurate and most robust classifier in the popular classification methods of data mining. In SVM classification, analyzation on data depended on convex quadratic programming. It requires large amount of operation matrix such as numerical computations time cost [13].

6. Evaluation Result

In this system, evaluation analysis results are performed on the various training dataset such as 90%, 80% and 70% training dataset on the all collected dataset. And then the accuracy result of each classifier based on the amount of training dataset.

Table 1 shows the accuracy results of each classifier based on the 70% training dataset of the total dataset. In this testing 70%of total dataset are used for the training dataset and 20% of total dataset are used testing dataset.

Table 1. Performance Analysis on (70% Training Dataset)

Classifiers	Accuracy Result
SVM	83.4
DT	78.2
NB	76.5
C4.5	74.2

In this analysis, the accuracy of SVM classifier achieved the better accuracy results than other classifiers. The bar chart of this analysis is described in the figure 8.

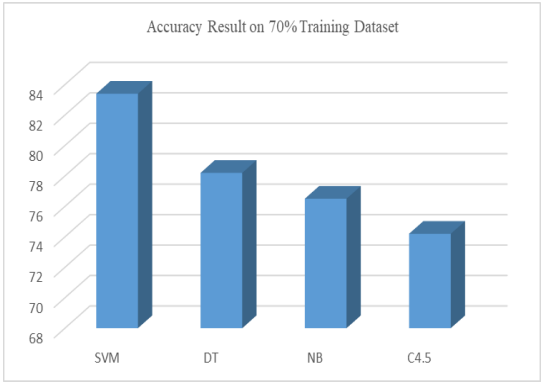


Figure 8. Accuracy Results on 70% Training Dataset

Table 2 shows the analysis of accuracy results on the used classifiers. In this analysis, 80% of total dataset are used as training dataset and 20% of total dataset are used as testing dataset. The accuracy of all classifiers is improved due to the change amount of training dataset. SVM classifier achieved better accuracy result in

this analysis. The bar chart of this analysis is shown in the figure 8.

Table 2. Performance Analysis on (80% Training Dataset)

Classifiers	Accuracy Result
SVM	90.3
DT	82.2
NB	79.85
C4.5	77.6

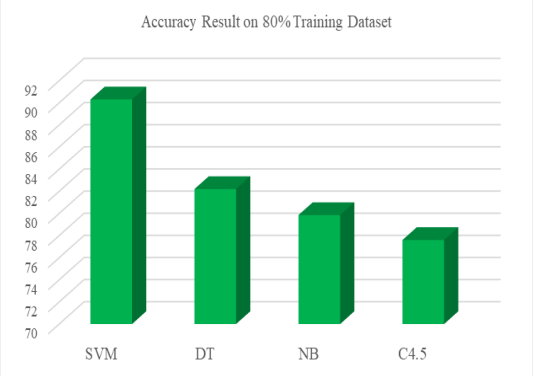


Figure 8. Accuracy Results on 80% Training Dataset

Table 3. Performance Analysis on (90% Training Dataset)

Classifiers	Accuracy Result
SVM	92.3
DT	84.7
NB	75
C4.5	78.2

In this analysis, the 90% training dataset is used to evaluate the results of the classifiers. According to the evaluation results, all classifier accuracy is improving but Naïve Bayes classifier's accuracy is decreased than other classifier. The improvement of the training data can't support for all classifier getting the higher accuracy. So, 80% training dataset is more suitable for all classifiers that are used in this system. The bar chart of this analysis is described in the figure 9.

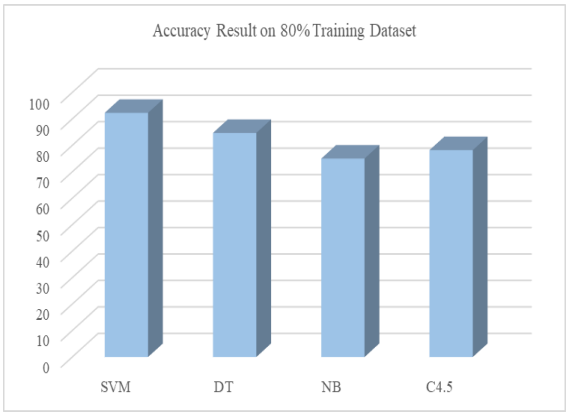


Figure 9. Accuracy Results on 90% Training Dataset

7. Conclusion

From the evaluation result of this system's work, SVM classifier results is optimal than the other classification methods that applied in this work. In the analysis of the evaluation results, this system analysed the performance result on the various training amount of dataset such as 90% training dataset, 80% training dataset and 70% dataset. According to the evaluation results, 80% training dataset and 20% testing dataset the most suitable situation for this system and then SVM classifier achieves the better and higher accuracy in this system. SVM classifier is also supervised learning classifiers, so its accuracy result is also improving depend on the amount of training dataset. By using this classification analysis, the student performance prediction work can improve in the future.

Acknowledgements

I am very grateful to Rector, Professor for fruitful discussion during the preparation of this paper and colleagues from Computer University (Mandalay), Myanmar.

REFERENCES

[1] Baradwaj, Brijesh Kumar, and Saurabh Pal, "Mining educational data to analyze students' performance." arXiv preprint arXiv:1201.3417, 2012.

[2] Patil, Tina R., and S. S. Sherekar. "Performance analysis of Naive Bayes and J48 classification algorithm for data classification." International Journal of Computer Science and Applications 6.2, 256-26, 2013.

[3] Al-Barrak, Mashael A., and Muna Al-Razgan. "Predicting student's final GPA using decision trees: a case study." International Journal of Information and Education Technology 6.7 (2016), 2016.

[4] Mueen, Ahmed, Bassam Zafar, and Umar Manzoor. "Modeling and Predicting Students' Academic Performance Using Data Mining Techniques." International Journal of Modern Education & Computer Science 8.11, 2016.

[5] Saa, Amjad Abu. "Educational Data Mining & Students' Performance Prediction." International Journal of Advanced Computer Science & Applications 1.7: 212-220.

[6] Nithya, P., B. Umamaheswari, and A. Umadevi. "A survey on educational data mining in field of education." International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) 5.1, 2016.

[7] Rokach, and et al. "Data Mining with Decision Tree: Theory and Application", ISBN: 978-9812771711, 2008.

[8] Karaolis, M.A. &Moutiris, J.A (2010), "Assessment of the Risk Factors of Coronary Heart Events Based on Data Mining with Decision Trees", IEEE Transactions on Information Technology in Biomedicine, Vol.14, No.3, May 2010, pp. 559-566.

[9] F.M. Javed Mehedi Shamrat, Rumesh Ranjan, Khan Md. Hasib, Amit Yadav, and Abdul Hasib Siddique, "Performance Evaluation among ID3, C4.5, and CART Decision Tree Algorithm", International Conference on Pervasive Computing and Social Networking [ICPCSN], Salem, Tamil Nadu, India, 19-20, March 2021.

[10] Nigam, K.; McCallum, A. K.; Thrun, S.; and Mitchell, T, "Text classification from labeled and unlabeled documents using em". Machine Learning 39(2-3):103–134,2000.

[11] Rigutini, L.; Maggini, M.; and Liu, B. 2005. An em-based training algorithm for cross-language text categorization. In Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence.

[12] NIU Yan1, and YE Shenglan, "Data Prediction Based on Support Vector Machine (SVM)", ICEMEE2019, IOP Conf. Series: Earth and Environmental Science 295 (2019) 012021.

[13] Gao Wei. "Research on incremental learning algorithm of support vector machine", Liaoning University of Science and Technology, 2018.

Authors

Biography

Ohnmar Aung: B.C.Sc.(25.11.2004), B.C.Sc(Hons:) (23.11.2005), M.C.Sc.(17.11.2008). Places of employ are Computer University (Lashio), Computer University (Mandalay), University of Computer Studies (Yangon), Myanmar Institute of Information Technology (Mandalay), Computer University (Myeik), Computer University (Mandalay). Paper and journal are DIANA Based Clustering for Interval-Scaled Variables, Second National Conference on Parallel and Soft Computing, PSC (December 24, 2008) Yangon, Myanmar, Stochastic Context Free Grammar for Statistical Parsing of Myanmar Natural Language Processing, Proceedings of 16th International Computer Conference on Computer Applications, Yangon, Myanmar, Proposed Framework for Stochastic Parsing of Myanmar Language, First International Conference on Big Data Analysis and Deep Learning Applications (Springer Journal), Japan, Fingerprint Image Retrieval Based on SURF and MSER Feature, First International Conference on Big Data Analysis and Deep Learning Applications (Springer Journal), Japan, Comparison on data Cyber Security, University Journal of Research and Innovation 2020. Research interests are Data Mining and Mathematics Computing.

WEB DOCUMENTS CATEGORIZATION BY TOPIC LABELING USING LBL2VEC

Pyae Sone

Faculty of Computer Science, University of Computer Studies (Mandalay)
dawpyaesone19@gmail.com

ABSTRACT

The rapid growth of the World Wide Web has led to the automatic classification of textual documents, driving technologies such as Data mining, NLP and Machine Learning to classify documents that cannot be classified by humans. With the high availability of information from diverse sources; Sorting activities were of utmost importance. Automatic text classification is considered as an important way to manage and process large amounts of documents in digital formats. In this paper, an unsupervised approach is used to classify the documents with predefined topics from an unlabeled web document dataset. Here, Lbl2Vec algorithm is applied to perform unsupervised text classification. The Lbl2Vec unsupervised approach requires only a small number of keywords describing the respective topics and no labeled document. The key idea of the algorithm is that many semantically similar keywords can represent a category.

KEYWORDS

the document classification, Machine Learning, unsupervised approach, Lbl2Vec algorithm, keywords

1. INTRODUCTION

With the proliferation of digital documents from diverse sources, text classification is becoming more popular every day. The mushrooming of digital data available over the last few years; Data discovery and data mining work together to extract meaningful data into useful information and knowledge [12]. Infeasibility of human beings to go through all the available documents to find the document of interest precipitated the rise of document classification. Automatically categorizing documents could provide people a significant ease in this realm. Text classification is the task of assigning a sentence or document an appropriate category. The categories depend on the selected dataset and can cover arbitrary subjects. Therefore, text classifiers can be used to organize, structure, and categories any kind of text [13]. The common approaches for text classification are supervised learning approaches. These text classification approaches usually require a large amount of labeled training data. However, an annotated text dataset for training state-of-the-art classification algorithms is often unavailable. The annotation of data usually involves a lot of manual effort and high expenses. Therefore, unsupervised text classification also referred to as zero-shot text classification offer the opportunity to run low-cost text classification for unlabeled data sets [9].

In this paper, the Lbl2Vec algorithm is used to provides the classification of documents on predefined topics from a large corpus based on unsupervised learning. The Lbl2Vec is an algorithm for unsupervised document classification. This algorithm enables to identify the relevant categories without having any annotated data. It automatically generates jointly embedded label, document and word vectors and returns documents of categories modeled. The key idea of the algorithm is that many semantically similar keywords can represent a category [6]. In the first step, it creates a joint embedding of document, and word vectors. Once documents and words are embedded in a shared vector space, the goal of the algorithm is to learn label vectors from previously manually defined keywords representing a category. Finally, the algorithm can predict the affiliation of documents to categories based on the similarities of the document vectors with the label vectors. This paper show that the

approach produces reliable results while saving annotation costs and requires almost no text preprocessing steps. To this end, the system applies to a publicly available and commonly used web documents dataset.

The web documents dataset obtained from https://www/kaggle.com/datasets/hetulmehta/website-classification and in this dataset contained 5000 documents with five difference categories. In this paper, only 1200 documents are classified into three difference categories, these are Travel, Sport and Food.

2. RELATED WORK

There has been much research done on text classification over the years, and also on different methods for performing categorization of web documents. In this section, some research papers on automatic web documents categorization are presented to get an idea of what work has been done in the past and what work should been done in the future. The keyword enrichment (KE) and subsequent unsupervised classification based on latent semantic analysis (LSA) vector cosine similarities was proposed by Haj-Yahia et al. (2019) [5]. In this approach, the entire unlabeled dataset are classified by using the similarity scores between documents and target categories. Another approach for mentioning in this context is Song and Roth (2014) [11] used the pure dataless hierarchical classification to evaluate different semantic representations.

Chang et al. (2008) [1] applies Wikipedia as source of world knowledge to compute explicit semantic analysis embeddings of labels and documents. Afterward, they identify the nearest neighbor classification to assign the most likely label to each document. Devlin, Chang, M.-W., (2019) [3] train a bidirectional encoder representation from transformers (BERT) model by using various public entailment datasets and then the pretrained BERT entailment model is used to directly classify texts from different datasets. Rao et al. (2006) [10] derived and assigned document labels based on a k-means word clustering. Besides, Chen et al. (2015) [2] proposed a method called descriptive latent Dirichlet allocation, which could perform classification with only category description words and unlabeled documents, thereby eradicating the need for a large amount of world knowledge from external sources. Gysel et al. (2018) [4] introduce a neural vector space model to learn document representation unsupervised. Neethu K S. (2016) [9] presented a technique that combines two machine learning techniques, K-Means and extreme learning machines (ELM).

3. LBL2VEC ALGORITHM

The system performs the web documents classification by using Lbl2Vec algorithm. The Lbl2Vec algorithm is applied to classify unlabeled documents and to retrieve the documents. It automatically generates embedded label, document and word vectors and return documents of categories modeled.

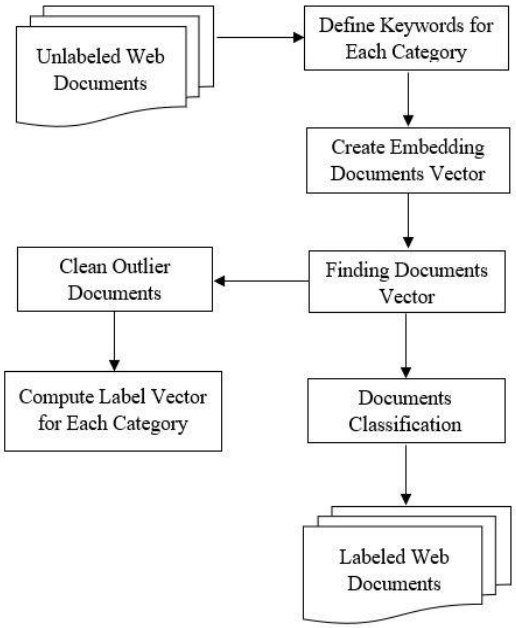


Figure 1. Documents Classification System

The algorithm comprise of five difference tasks and these tasks are: 1) Create jointly embedded document vectors. 2) Find document vectors that are similar to the keyword vectors of each classification category. 3) Clean outlier documents for each classification category. 4) Compute the centroid of the outlier cleaned document

vectors as label vector for each classification category. 5) Text document classification.

3.1 Define Keywords for each Category

The system needed to define keywords to describe each classification category of interest. In this process requires some degree of domain knowledge to define keywords that describe classification categories and are semantically similar to each other within the classification categories.

Table 1. Example keywords for 3 difference categories

Travel	Sport	Food
Hotel	Football	Delicious
Trip	Tennis	Ingredient
Beach	Basketball	Breakfast
Island	Hockey	Lunch
Vacation	Golf	Dinner

3.2 Create Embedded Document Vectors

An embedding vector is a vector that allows the system to represent a text document in multi-dimensional feature space. The embedded document vectors are created by using one of the paragraph vectors framework called the distributed bag of words version of paragraph vector (PV-DBOW) and the Skip-gram architecture [5]. The PV-DBOW and Skip-gram architectures enable to learn simultaneously word and document embedding that share the same feature space. The main idea of embedding vectors is that similar words or text documents will have similar vectors. Therefore, the embedded documents vectors have been created based on the documents are located close to other similar documents and close to the most distinguishing words.

3.3 Finding Documents Vectors Similar to the Keywords Vectors

In this step, firstly need to calculate a centroid of embedding keywords $\bar{k}$. In order to calculate the centroid of keywords embedding the following embedding equation is used.

$$\bar{e} = \frac{1}{n}\sum_{x=1}^{n} \bar{e}_x$$

After the centroid of the keywords embedding for each category is calculated, the similarity of this centroid and the documents vectors are computed. The cosine similarity is used to compute the similarity between the documents and the centroid of keywords embedding for each category. Documents that are similar to category keywords are assigned to a set of candidate documents of the respective category.

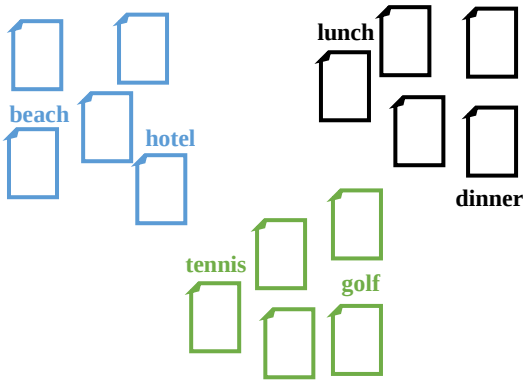


Figure 2. Defined Keywords with their respective set of candidate Documents

3.4 Clean Outlier Documents

Local Outlier Factor (LOF) algorithm [3] is applied to clean the outlier documents from each set of candidate documents that may be related to some of the descriptive keywords but do not properly match the intended classification category. LOF algorithm identifies the outliers by considering the density of the local neighborhood. If the density of a documents is not the same as its neighborhood, the document says that the outlier.

3.5 Compute Label Vector

In this paper, the label vectors are identified to get embedding representations of classification categories. The Lbl2Vec algorithm compute the label vectors by calculating the centroid of the respective cleaned document vectors rather than keyword centroids because it is more difficult to classify documents based on similarities to keywords only. The similarity

of the document vectors to the resulted label vectors will be used to classify the web documents.

3.6 Text Document Classification

The system needs to calculate the cosine similarity between document embedding and label embedding in order to decide whether the content of document is semantically similar to a single topic. The Lbl2Vec algorithm computes the label vector and document vector similarities for each label vector and document vector in the dataset. Finally, web documents are classified as a category with the highest label vector and document vector similarity.

4. PERFORMANCE EVALUATION

In this paper, the Lbl2Vec algorithm is used to classify text documents from the website document classification dataset that is obtained from Kaggle.com. It is a collection of approximately 5000 text documents, partitioned evenly across 15 different categories. However, this paper will focus on a subset of the website documents dataset consisting of the categories "Travel", "Sport" and "Food".

The performance of the system is evaluated by comparing the result of the documents classification with the annotations found in the original dataset. The system performance is evaluated by using various evaluation metrics, (i) Precision, (ii) Recall, (iii) F1-score. Here, the confusion matrix is used to compute Precision, Recall and F1-score of the system. In a confusion matrix, four metrics can be defined: true positive (TP), true negatives (TN), false positive (FP), false negative (FN).

$$\text{Precision (P)} = \frac{TP}{TP+FP}$$

$$\text{Recall (R)} = \frac{TP}{TP+FN}$$

$$\text{F1-Score (F1)} = \frac{2 * Precision * Recall}{(Precision + Recall)}$$

The following result show that the Lbl2Vec algorithm can predict the web documents categories correctly with the accuracy of 91%. This is achieved without even seeing the document labels during classification. Moreover, the algorithm can also predict the classification categories of documents that were not used to train the Lbl2Vec model and are therefore completely unknown to it.

Tabel 2. Prediction for web documents categorization

Actual / Prediction	Travel	Sport	Food	Total
Travel	367	34	23	424
Sport	28	329	52	409
Food	50	24	293	367
Total	445	387	368	1200

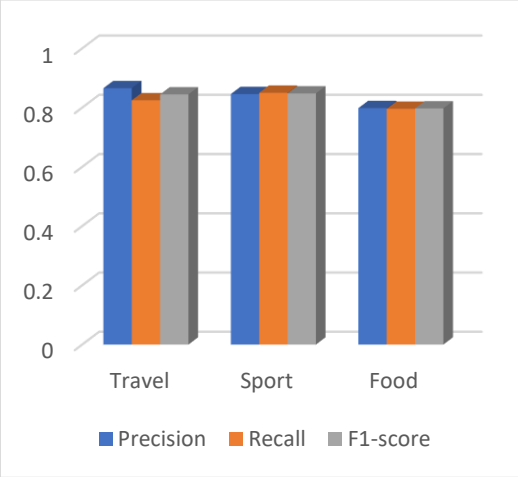


Figure 2. Performance Evaluation for Web Documents Categorization for each class

Tabel 3. Evaluation metrics for web documents categorization

Precision (P)	0.8366
Recall (R)	0.8237
F1-score (F1)	0.8230
Accuracy (A)	0.8242

5. CONCLUSIONS

This paper applies the Lbl2Vec algorithm that can be used for unsupervised text documents classification. Unlike other state-of-the-art approaches it needs no label information during training and therefore offers the opportunity to run low-cost- text classification for unlabeled datasets. Moreover, the Lbl2Vec algorithm yields better fitting models of predefined topics

than conventional topic modeling approaches. The web documents are classified by creating a joint embedding of document, and word vectors. Once documents and words are embedded in a shared vector space, the goal of the algorithm is to learn label vectors from previously manually defined keywords representing a category. Finally, the algorithm can predict the affiliation of documents to categories based on the similarities of the documents vecotrs with the label vecotrs.

6. REFERENCES

[1] *Chang, M.-W., Ratinov, L.-A., Roth, D., and Srikumar, V. (2008). Importance of semantic representation: Dataless classification. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence..*

[2] *Chen, X., Xia, Y., Jin, P., and Carroll, J. (2015). Dataless text classification with descriptive lda. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence.*

[3] D*evlin, J., Chang, M.-W., Lee, K., and Toutanova, K., "BERT: Pre-training of deep bidirectional transformers for language understanding". In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1.*

[4] *Gysel, C. V., de Rijke, M., and Kanoulas, E. (2018). "Neural vector spaces for unsupervised information retrieval", ACM Trans. Inf. Syst..*

[5] *Haj-Yahia, Z., Sieg, A., and Deleris, L. A. (2019), "Towards unsupervised text classification leveraging experts and word embeddings". In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy.*

[6] *https://github.com/sebischair/Lbl2Vec*

[7] *https://towardsdatascience.com/local-outlier-factor-lof-algorithm-for-outlier-identification]*

[8] *Le, Q. and Mikolov, T. (2014). Distributed representations ofsentences and documents. In Xing, E. P. and Jebara, T., editors, Proceedings of the 31st International Conference on Machine Learning, volume 32 of Proceedings of Machine Learning Research, Bejing, China. PMLR.*

[9] *Neethu K S, Jyothis T S, and Jithin dev, "Text Classification Using KM-ELM Classifier", in Proceeding of the International Conference on Circuit, Power and Computing Technologies [ICCPCT] 2016.*

[10] *Rao, D., P, D., and Khemani, D. (2006). Corpus based unsupervised labeling of documents. In Sutcliffe, G. and Goebel, R., editors, Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference, Melbourne Beach, Florida, USA.*

[11] *Song, Y. and Roth, D. (2014). On dataless hierarchical text classification. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence.*

[12] *T. Eriksson, "Automatic web page categorization using text classification methods", Degree project in Computer Science 2013.*

[13] *T. Schopf, D. Braun, and F. Matthes "Lbl2Vec: An Embedding-based Approach for Unsupervised Document Retrieval on Predefined Topics", in Proceeding 17th International Conference on Web Information Systems and Technologies (WEBIST 2021).*

Author

Biography

First A. Author All biographies should be limited to one paragraph consisting of the following: sequentially ordered list of degrees, including years achieved; sequentially ordered places of employ concluding with current employment; association with any official journals or conferences; major professional and/or academic achievements, i.e., best paper awards, research grants, etc.; any publication information (number of papers and titles of books published); current research interests; association with any professional associations. Author membership information, e.g., is a member of the IEEE and the IEEE Computer Society, if applicable, is noted at the end of the biography.

DATACENTER SELECTION FOR CLOUD-BASED APPLICATIONS

Kaung Myat Soe[1] and May Phyo Thu[2]

Faculty of Computer Science, University of Computer Studies (Mandalay)
[1]kaungmyatsoemdy1@gmail.com, [2]mayphyothu.mpt1@gmail.com

ABSTRACT

As cloud computing advances, organizations' IT infrastructure and application deployment processes are moving to the cloud because cloud computing provides everything as a service over the Internet. The performance of a cloud-based application is based on proper datacenter selection and workload distribution within the selected datacenter. Service broker policies are used for suitable datacenter selection, and load balancing algorithms are applied to distribute workloads. This paper is to evaluate the effect of a proposed service broker policy (PSBR) on the performance of cloud-based applications with load balancing algorithms. To achieve the objective, the behavior of the TikTok application was modeled using the worldwide users' statistics on the cloud simulation framework, namely CloudAnalyst. As a result, the average response time and datacenter processing time are measured. Next, the proposed service broker policy provides better results than the existing service proximity-based policy. This paper would support the benefits for not only cloud service providers from coordination between datacenter configuration, datacenter selection, and workload distribution but also cloud users in identifying the appropriate procedures for their organization or application.

Keywords

Cloud Computing, CloudAnalyst, Service Broker Policy.

1. INTRODUCTION

Cloud computing is a technology that uses virtualization and grid computing to provide flexible and unlimited computing resources to users. The context of why commercial and industrial users are migrating widely to the cloud can be seen in the five essential characteristics of cloud computing.

Cloud computing users can manage their own computing infrastructure and services without service provider interaction and can access computing resources from heterogeneous client platforms that are available on the Internet with broad connectivity options. In cloud computing, services can be scaled in or out, and the use of services can be measured. And users can share computing resources from a pool, such as data centers. Therefore, cloud computing has become widely used in building and hosting various application domains (e.g., social media, e-commerce) due to its cost effectiveness and operational efficiency. However, some problems, such as datacenter selection and load balancing, have to be solved to deploy the applications in the cloud computing environment. Cloud service brokers are responsible for datacenter selection, and they use service broker policies to select the appropriate datacenter. Some service broker policies are described as follows:

- Service Proximity Based Policy (SPR)
- Best Response Time Service Broker Policy
- Dynamic Service Broker Policy

Service Proximity Based policy selects the datacenter closest to the user request based on transmission parameters such as network latency and bandwidth. But this policy randomly selects the datacenter when there are one or more datacenters in the same region. The randomly selected datacenter can give the following unsatisfactory results:

- Higher response time of user-base
- Higher datacenter processing time

In order to solve these issues, the new service broker policy is proposed by extending the random selection part of the existing service proximity based policy (SPR).

In this paper, the effect of the proposed service broker policy (PSBR) on the performance of cloud-based applications is evaluated. Then, this proposed policy is implemented with different load balancing algorithms. The objective of this system is described as follows:

- To handle the issue of random selection of datacenter
- To reduce response time of user base
- To reduce datacenter processing time

For the system implementation, TikTok users' statistics [9] are used as a sample application using the CloudAnalyst simulation framework. The performance of this application is evaluated with PSBR and existing service proximity-based policies by applying three load balancing algorithms that are already available in the CloudAnalyst simulation framework. The proposed PSBR provides the appropriate datacenter based on the response time, processing time, and cost of each datacenter. This proposed policy will help cloud service providers and application designers successfully achieve an appropriate service broker policy and load balancing algorithm for cloud-based applications.

The rest of the document will be described as follows: Section 2 discusses the related works to the proposed policy. And the next Section 3, describes the background theory of the proposed policy. In Section 4, the proposed system is presented, and the simulation and performance results of the proposed policy are discussed in Section 5. Finally, the presentation of the system is concluded in Section 6.

2. RELATED WORKS

This paper aims to evaluate the effect of PSBR on the performance of cloud-based applications. As related works, datacenter selection, application workload distribution, and cloud simulation frameworks are presented.

Currently, cloud simulation frameworks are powerful tools in the field of cloud computing research because they allow for freely performing iterative experiments. Many studies have been conducted using simulation techniques to evaluate the behavior of cloud-based systems. Some useful simulation frameworks are CloudSim, CloudAnalyst, SPECI, GreenCloud, OCT, Open Cirrus, GroudSim, NetworkCloudSim, EMUSIM, DCSim, and iCanSim. These simulation frameworks are presented in a review by W. Zhao et al. in 2012 [8].

CloudSim [1] can simulate the scenarios of cloud infrastructure level because it offers basic components such as hosts and virtual machines to model the services. But CloudSim does not support a graphical user interface (GUI). Therefore, B.Wickremasinghe proposed CloudAnalyst based on CloudSim in 2009 [7]. CloudAnalyst makes it easy to simulate the cloud computing environment due to its GUI.

Meftah et al. [4] presented the effect of cloud-based application performance using service broker policies and load balancing algorithms. And, C.C. Agbo presented the simulation of a cloud-based application using CloudAnalyst.

CloudAnalyst is mainly used for load balancing algorithms and service broker policies in most research papers. Load balancing algorithms help select the virtual machine for user requests and balance the workload among virtual machines. There are three existing load balancing algorithms in CloudAnalyst. Service broker policies help to select the datacenter with the lowest response time, processing time, and cost.

H. V. Patel conducted a survey on CloudAnalyst's existing service broker policies in 2015 [6]. They presented a practical comparison of existing policies. In addition, various service broker policies were proposed to select the suitable datacenter.

Nandwani et al. [5] proposed a weight-based service broker policy to choose the appropriate datacenter based on the proportion of virtual machine weights for each datacenter. As a result, this policy reduced the response time, processing time, and overall cost. However, performance-aware policies lead to increased

costs, while cost-aware policies increase processing time.

Jyoti et al. [2] proposed a service broker policy for datacenter selection by focusing on the workload. They provided an extension of the weight-based service broker policy that was proposed by Nandwani. The result was only a trivial decrease in response time, processing time, and cost.

Additionally, Kofahi et al. [3] proposed a user priority-based service broker policy using vector space model and multi-objective optimization techniques. In a real cloud platform, most users do not want to know the specifications of datacenters.

Through the above three proposed policies, we have extended the existing service proximity-based policy as a new policy (PSBR). The effect of PSBR on cloud-based application performance was evaluated in this paper using three existing load balancing algorithms.

3. BACKGROUND THEORY

Cloud computing is a technology that provides computing resources as services over the internet. Instead of investing in and maintaining physical datacenters and servers, users can access cloud services, such as compute power, storage, and databases, with pay-as-you-go pricing from cloud service providers like Amazon Web Services (AWS), Microsoft Azure, etc. Cloud computing offers everything as a service to users, such as mailing, streaming, delivering software, developing and deploying applications, analyzing and retrieving data, backing up and restoring data, etc. But these services can generally be categorized into three main types: software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS). A cloud represents the internet, which can be public, or private, or both. The public cloud is provided by the general public for open use. A private cloud is one provided by an organization for exclusive use. The main difference between public and private is that the former uses shared infrastructure, while the latter uses its own dedicated infrastructure.

3.1. Service Broker Policy

Cloud service brokers assist cloud users in managing multiple cloud services within an organization and in negotiating the relationship between cloud service providers and cloud users.The three primary roles of a cloud service broker are aggregation, service intermediation, and service arbitrage. A cloud service broker significantly reduces process costs, increases flexibility, and reduces downtime, and it can help cloud users get computing resources efficiently. And cloud service brokers are also responsible for appropriate datacenter selection, and they decide datacenter selection for incoming user requests by using service broker policies such as:

Service Proximity Based Policy (SPR): It selects the datacenter that is closest to the user. When there is more than one datacenter in the closest region, it will select the datacenter randomly.

Best Response Time Service Broker Policy: It extends the closest datacenter policy. When the response time of the nearest datacenter begins to degrade, this policy estimates the current response time of the datacenters. Then it seeks the datacenter with the lowest response time. But the closest and fastest datacenter selection process is a 50:50 chance, and it does not consider the situation of more than one datacenter.

3.2. Load Balancing Algorithm

The role of load balancers is workload distribution among server within a datacenter to ensure that no server is underloaded, overloaded, or idle. Load balancers use load balancing algorithms to optimize resource utilization and to enhance execution time and response time. In cloud computing, server hubs use load balancing algorithms to balance workloads among available virtual machines. Currently, CloudAnalyst has implemented the following three load balancing algorithms:

Round Robin Load Balancer: It allocates the virtual machines in the form of a simple round-robin algorithm.

Active Monitoring Load Balancer: It balances the tasks among available virtual machines in

a way to even out the number of active tasks on each virtual machine at any given time.

Throttled Load Balancer: It performs only a pre-defined number of tasks that are allocated to a single virtual machine at any given time. If more requests are presented than there are available virtual machines, some of the requests will have to be queued until the next virtual machine becomes available.

4. PROPOSED SYSTEM

The proposed service broker policy PSBR extended the closest datacenter policy. When there are one or more datacenters in the region, the PSBR selects the datacenter based on the number of virtual machines in each datacenter and the cost of each datacenter. As a result, the PSBR selects the datacenter with the lowest response time, datacenter processing time, and total cost. Figure 1 shows the PSBR flow diagram.

According to the flow diagram, when the user sends a request to select a datacenter, PSBR collects a list of datacenters from the closest region. If there is only one datacenter in the list, it selects the datacenter according to the closest datacenter policy. When there are multiple datacenters in the list, the PSBR calculates the weight of each to find the one with the least response time. The weight of each datacenter is calculated based on the number of virtual machines in each datacenter with the following equation: 1.

$$W_i = \frac{no\ of\ VM\ on\ each\ DC_i\ from\ closest\ region}{no\ of\ VM\ in\ all\ DC\ from\ closest\ region} \quad (1)$$

where W_i refers to the weight of each datacenter, DC refers to the datacenter, DC_i refers to each datacenter, and VM refers to the virtual machine.

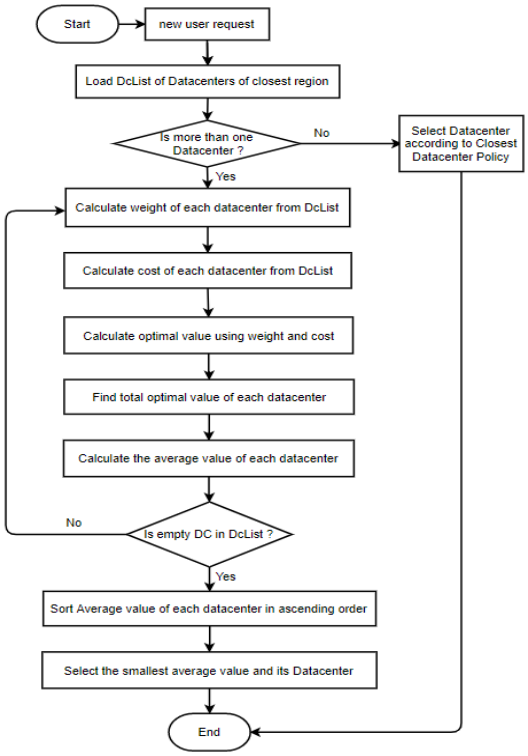


Figure 1. PSBR flow diagram

Then, the PSBR calculates the total cost of each datacenter based on the virtual machine cost and data transfer cost using equation: 2.

$$C_i = \left(\frac{MIPS_{total}}{VM_{MIPS}} \right) \times VM_{cost} + UDS \times DT_{cost} \quad (2)$$

where, C_i refers to the total cost of each datacenter that considers both data transfer and virtual machine processing cost, ($MIPS_{total}/VM_{MIPS}$) refers to the total number of instructions per average processing power assigned to the datacenter. VMcost denotes the virtual machine cost of each datacenter; UDS denotes the size of requests; and DT_{cost} denotes the data transfer cost. Then, the PSBR searches for the optimal values by using weight and total cost with equation 3.

$$Opt_i = w1 \times W_i + w2 \times C_i \quad (3)$$

where, Opt_i refers to the optimal value of each datacenter, W_i refers to the weight value of each datacenter, and C_i refers to the total cost value of each datacenter. According to the general formula of multi-objective

scalarization, the coefficient values will determine the solution of the fitness function and show the priority. The larger the coefficient value, the higher the priority, and the sum of coefficient values must be 1. Therefore, w1 and w2 are the coefficient values. Then the total optimal value and average values of each datacenter is calculated by using equations 4 and 5, respectively.

$$totalOpt_i = \sum_{i=1}^{N} Opt_i \qquad (4)$$

$$Avg_i = \frac{totalOpt_i}{N} \qquad (5)$$

Where, $totalOpt_i$ refers to the sum of optimal values of each datacenter, Opt_i refers to the optimal value for each datacenter, and N is the number of intervals. Avg_i refers to the average value of each datacenter. When the average values of all datacenters have been calculated, the PSBR selects the datacenter with the smallest average value.

Table 1. Notations Used in Proposed Service Broker Policy (PSBR)

Notation	Description
DC	Datacenter List
UB	User Request List
R	Region List
W	Weight Value
C	Cost Value
Opt	Optimal Value
totalOpt	Total Optimal Value
Avg	Average Value

Algorithm: Proposed Service Broker Policy

Input: Datacenter List DC_i = {DC_1, DC_2, …, DC_n}, User Request List UB_i = {UB_1, UB_2, …, UB_n}, Region List R_i = {R_1, R_2, …, R_6}
Output: Optimal Data Center List DC_i
Get a region proximity list
if there is new User Request **UB_i then** load
  Datacenter List **DC_i** of closest region
    if there is more than one Datacenter **DC_i** at that region **R_i then**
      for each Datacenter **DC_i** at that region **R_i do**
        Calculate the weight value **W_i** of each Datacenter **DC_i** using (1)
        Calculate the cost value **C_i** of each Datacenter **DC_i** using (2)
        Calculate the optimal value **Opt_i** of each Datacenter **DC_i** using (3)
        Find the total optimal value **$totalOpt_i$** of each Datacenter **DC_i** using (4)
        Calculate the average value **Avg_i** of each Datacenter **DC_i** using (5)
      end for
      Sort the average value **Avg_i** of each Datacenter **DC_i** in ascending order
      Select the smallest average value **Avg_i** and its Datacenter **DC_i**
      Return Datacenter **DC_i**
    else
      Return Datacenter **DC_i** according to Service Proximity Based Policy Policy
    end if
end if

5. EXPERIMENTS

The goal of the experiments is to demonstrate how the proposed service broker policy can affect cloud-based application performance with load balancing algorithms. TikTok has over 1.4 billion users around the world [9]. Table 2 shows the TikTok annual user statistics by region for 2018 to 2021. The CloudAnalyst simulation framework was used to evaluate the effects of the cloud-based application TikTok.

Table 2. TikTok Annual Users by Region 2018 to 2021(million)

Year	Asia	North America	Europe	South America
2018	62	28	21	3
2019	130	55	53	10
2020	198	105	98	64
2021	313	138	158	188

5.1. Experiment Setup

To simulate in CloudAnalyst, we assumed that 1/100 of the size of TikTok's annual user statistics in 2018 was considered simultaneous online users during off-peak hours, and we also assumed that 1/100 of the size of TikTok's annual user statistics in 2021 was considered simultaneous online users during peak hours. Then, we assumed that the hardware specifications of datacenters were the same.

Table 3. Userbase Configuration

Name	Region	Peak Hours (GTM)	Avg Peak Users	Avg Off-Peak Users
UB1	3	13-15	313000	62000
UB2	0	15-17	138000	28000
UB3	2	20-22	158000	21000
UB4	1	1-3	188000	3000

Table 4. Other CloudAnalyst Parameter Values

Parameters	Values
User Grouping Factor in Userbase	10
Request Grouping Factor	10
Executable instruction length per request	100
Simulation Duration	60 min
Image size	10000
Memory	512
Bandwidth	1000
Datacenter Architecture	X86
Datacenter processor	4
Datacenter Operating system	Linux
Datacenter virtual machine monitor	Xen

5.2. Simulation Scenario

The CloudAnalyst simulation framework was used to evaluate the performance of the cloud-based application TikTok as described in the following scenario using the proposed service broker policy (PSBR) with three existing load balancing algorithms. The scenario represents the configuration of datacenters within the same region, which is shown in Table 5.

Table 5. Datacenter configuration

Datacenter	Region	No. of virtual machines	Cost per virtual machines /hr	Data transfer cost/Gb
DC1	0	25	0.15	0.15
DC2	0	50	0.12	0.12
DC3	0	75	0.1	0.1
DC4	2	100	0.05	0.05
DC5	2	50	0.1	0.1
DC6	3	125	0.03	0.03
DC7	3	75	0.1	0.1

5.3. Result and Performance Analysis

During the simulation of the scenario, the response time and the datacenter processing time are evaluated, and the results are recorded. And the analysis of the results is shown in the following tables and figures.

Table 6. Experimental Results for Response Time

Service Broker Policy	Load Balancing Algorithms	Response Time of Application		
		Min	Max	Avg
PSBR	Round Robin	36.5	261.2	78.5
	Active Monitoring	37.3	260.6	75.1
	Throttled	36.6	259.9	78
SPR	Round Robin	37.4	260.6	84.5
	Active Monitoring	37.4	261.2	84.2
	Throttled	37.4	260	84

According to the experimental result table 6, the average response time of the proposed PSBR with active monitoring load balancing algorithm is the best in all. Next, the proposed PSBR with three load balancing algorithms is better than the existing SPR in all measures of average response time.

Table 7. Experimental Results for Processing Time

Service Broker Policy	Load Balancing Algorithms	Datacenter Processing Time		
		Min	Max	Avg
	Round Robin	0.18	10.7	5.72

PSBR	Active Monitoring	0.19	7.87	5
	Throttled	0.18	6.08	4.67
SPR	Round Robin	0.31	10.85	5.79
	Active Monitoring	0.19	11.25	5.04
	Throttled	0.18	5.77	4.8

The experimental result table 7 shows the datacenter processing time. In the result table, the proposed PSBR with throttled load balancing gives the best datacenter processing time. Then, the proposed PSBR with active monitoring load balancing algorithm also gives the better processing time than the existing SPR. Figs. 2 and 3 clearly show that the best values for average response time and average processing time are obtained in all scenarios.

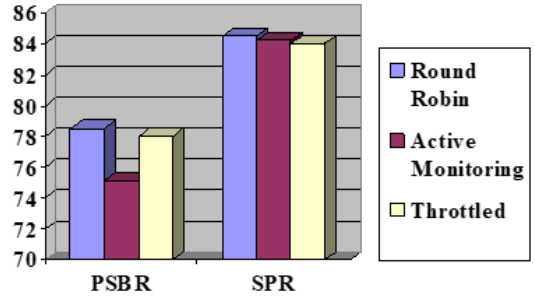


Figure 2. Average Response Time of the Proposed PSBR and Existing SPR

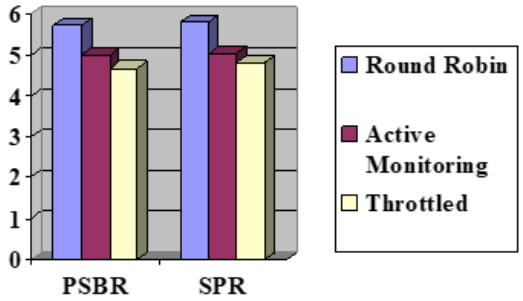


Figure 3. Average Processing Time of the Proposed PSBR and Existing SPR

6. CONCLUSION

This paper has presented the effect of the proposed service broker policy (PSBR) on the performance of cloud-based applications with different load balancing algorithms. The existing SPR randomly selects the datacenter without considering response time or datacenter processing time. So, the new service broker policy is proposed to handle the random datacenter selection part of the existing SPR. Next, the average response time and datacenter processing time are measured. According to the evaluation results, the response time of PSBR with the active monitoring load balancing algorithm is the best of all, and the processing time of PSBR with the throttled load balancing algorithm is better than the other since its result is shorter. Finally, the proposed PSBR gives better response time and less datacenter processing time than the existing SPR in all conditions.

REFERENCES

[1] R. N. Calheiros, R. Ranjan, C. A. F. De Rose, and R. Buyya, "CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services".

[2] A. Jyoti, Dr. R. K. Pathak, Dr. P. Singh, "Service Broker Algorithm for Datacenter Selection with light and heavy load in Cloud Computing", *International Journal of Advanced Trends in Computer Science and Engineering*, Vol.9, No.4, 2020. pp.6747–6751.

[3] N. A. Kofahi, T. Alsmadi, M. Barhoush and M. A. Al-Shannaq, "Priority-Based and Optimized Data Center Selection in Cloud Computing", *Arabian Journal for Science and Engineering*, 2019. pp. 9275–9290.

[4] A. Meftah, A. E. Youssef and M. Zakariah, "Effect of Service Broker Policies and Load Balancing Algorithms on the Performance of Large-Scale Internet Applications in Cloud Datacenters", *(IJACSA) International Journal of Advanced Computer Science and Applications*, Vol. 9, No. 5, 2018. pp. 219-227.

[5] S. Nandwani, M. Achhra, R. Shah, A. Tamrakar, K. Joshi, and S. Raksha, "Weight-Based Data Center Selection Algorithm in Cloud Computing Environment", *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, 394, 2017. pp.515–525.

[6] H. V. Patel, R. Patel, "Cloud Analyst: An Insight of Service Broker Policy", *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 4, Issue 1, 2015. pp. 122-127.

[7] B. Wickremasinghe, "CloudAnalyst: A CloudSim-based Tool for Modelling and Analysis

of Large-Scale Cloud Computing Environments", *MEDC Project Report*, 2009.

[8] W. Zhao, Y. Peng, F. Xie, and Z. Dai,"Modeling and Simulation of Cloud Computing: A Review", *IEEE Asia Pacific Cloud Computing Congress (APCloudCC)*, 2012.

[9] https://www.businessofapps.com/data/tik-tok-statistics/

ASPECT-BASED SENTIMENT ANALYSIS FOR HOTEL REVIEWS

Han Thi Phoo[1] and Yu Ya Win[2]

[1,2] Faculty of Computer Science, University of Computer Studies (Mandalay)
[1]akariphoo8@icloud.com, [2]yuyawin@gmail.com

ABSTRACT

Sentiment analysis is very hot research topic in natural language processing. The principal aim of sentiment analysis is for the extraction or capturing of the sentiment and subjectivity in natural language. Travelling services on online are very popular things of the travelling information sharing. Customer created information and content in reviews is precious for travelling organizations and this can possess a potential effect for providing decision. The online reviews on web has become an enormous deposit of data, to which online users add every new hotel information data. But the enormous amount of data makes impossible for a customer to read it all. This system will extract the aspects and opinion words from the online hotel reviews and analyze sentiments of each aspect from all these reviews. In this system, the hotels are rating according to positive, and negative score of each aspect in hotel reviews using SentiWordNet.

KEYWORDS

Sentiment Analysis, Natural Language Processing, SentiWordNet

1. INTRODUCTION

Nowadays, the Internet can provide the chances to everyone for the opinion expression to various fields with the relation of every things. People can express the opinions with various ways of customer reviews in an informal format. The deduced sentiment in those reviews is the concentration to important user who wants to buy the best goods in the market and to organizations in the user opinions analysis. The requirement to sentiment deduction in texts has provided the widely usage in natural language processing and computer science fields as the sentiment analysis [2].

The Web 2.0 services enhancement has occurred the potential impacts to the flow of travelling portion and provided the greatest innovations. Travelling web services are more popular and provides the useful information for the customers with the decision of choosing which the hotel and acquiring which the service. The development in the reviewing and feedback opportunities of the travelling web services have caused the internet as the major observation and obtaining the travelling information created by people. The customer created reviews and hotel contents on travelling services are developing by the rate of exponential and this can support many opinions of various people. By applying web portals, the preferences and opinions of customers can be shared and many reviews can be created by daily (Hu et al., 2017) [1].

Sentiment analysis analyzes feelings, attitudes, and opinions of the customer about the specific services and products. This exchanges the information applicable business information through unstructured text data. The classification of the opinions into negative, neutral, and positive is performed by the sentiment analysis. The travelling feelings, attitudes, and experiences are expressed on social media by posting reviews about views, restaurants, places, and hotels. In this paper, the reviews of hotels are collected to perform sentiment analysis. The classification of sentiment analysis into aspect-based, document level, and document level sentiment analysis can be performed. The positive reviewed document doesn't mean that the customer like everything. The negative reviewed document on a particular object doesn't mean that the customer has negative reviews on all features in the object. The sentence and document level analysis fail such information in this condition. Hence, aspect-based sentiment analysis is proposed for

achieving the detailed information for every aspect.

This paper proposes a model for aspect-based sentiment analysis for hotel's reviews and comments in English language using SentiWordNet. The division of aspect-based sentiment analysis is done into three parts, the prediction of how many aspects in every review, the extraction of those aspects and classification of sentiment polarity for those aspects. In this paper, the next section 2 describes about the related works of the proposed system and then section 3 presents about the background theory of the proposed system. In section 4, the proposed system is presented and the experimental results of the proposed system are discussed in section 5. Finally, the system is concluded in section 6.

2. RELATED WORKS

Kande Trupti V and H. P. Shah introduced that deep learning-based sentiment analysis for faster aspects classification with high accuracy [3]. Much experiments on real-time review sentiment analysis have been done for the determination of how the effectiveness of the framework to aid the tourists for indicating the good hotel, restaurant, and location at a region. This paper presented an aspect-based sentiment analysis for the review's classification among aspects into positive or negative. In this paper, the aspect deduction approach according to tree structure is introduced for the explicit and implicit aspect deduction through the reviews of tourists. The deduction of noun phrases and frequent nouns through opinions is performed and the similar nouns are grouped with WordNet. Convolutional neural networks are applied for sentiment analysis in that extracted nouns are utilized as the leaf of a tree and review words are utilized as internal nodes. Firstly, the removal of irrelevant and less opinion sentences is performed with the utilization of stanford basic dependency at every sentence. Then, the extraction of features through the remaining sentences is done by POS Tags and N-Grams for the training of classifiers. Finally, the extraction of features for the training of classifiers is taken by utilizing machine learning approaches.

M. Godakandage, and S. Thelijjagoda [4] presented a system for the analysis of hotel reviews supporting aspect-based personalized hotel recommendations in order to aid the customers in the searching of the best hotel based on the preferences, no having many reviews. In this paper, the system implementation is done by four steps: the collection data, the preprocessing of data, the extraction of aspects and the analysis of sentiment, and the visualization of the result. Firstly, the analysis of hotel reviews is done and then the deduction of all opinion reviews as the units of opinions by the respective aspects. The suggestion of good hotels is done by the applicability of customers for achieving the reviews on hotels according to the sentiment analysis of preferences and opinions of the customers.

P. S. Bhargav, G. N. Reddy, R. R. Chand, K. Pujitha and A. Mathur introduced the sentiment analysis system for hotel reviews. On the website, the preferences of hotels were put by the users according to the reviews relating with the rating of the hotel and the analysis by another hotels [5]. The system implementation was done by applying Naïve bayes machine learning approaches and opinion mining approaches according to natural language processing.

The authors presented a sentiment analysis system based on aspects for the comments and reviews of restaurants and hotels expressed by myanmar language applying LSTM neural network [6]. The sentiment analysis system based on aspects is performed with three steps: the forecasting of how many aspects in every review, the deduction of aspects, and the classification in polarity of sentiment to these aspects.

The framework for sentiment analysis system based on aspects was presented for the efficient identification of the aspects by high accuracy utilizing decision tree and naïve bayes machine learning approaches [7]. This framework performed the classification of aspects into negative and positive. This tree-based extraction approach deducts the implicit and explicit aspects through tourist reviews. This system deducted nouns and noun phrases through reviews and the similar nouns are categorized applying WordNet. This system aids the tourists for the searching of the optimal hotel, place, and restaurant in the

region and the experimental evaluation was performed with foursquare and yelp datasets.

3. BACKGROUND THEORY

Sentiment analysis is the analysis process of the user's sentiments and emotions on the services. Sentiment analysis is one field of natural language processing that performs the customer's opinion extraction and classification based on the polarity. The major aim of sentiment analysis is to construct an effective system supporting an interface to users. This has 3 levels of sentiment analysis: document, sentence, and aspect level.

Document level: At this level, the decision of all sentiments in a complete document is done. Example, when the service review is provided, the decision is taken whether it follows an overall positive or negative preference relating with this service. The validation job is performed whether the whole document is neutral, negative, or positive.

Sentence level: At this level, the limitation to sentences is performed and validate when every sentence follows a neutral, negative, or positive preference. The classification of sentence to subjective or objective is done and the categorization of sentences with subjective into neutral, positive, or negative is performed.

Aspect level: This phase is more challenging than the above two mentioned levels. At this level, the analysis of opinions is taken instead of the analysis of documents, sentences, paragraphs, or phrases. The finer grained analysis is done to every aspect. Opinion is a phrase containing a sentiment and subject on the target. The application of idea is utilized at this phase. It aids in comprehending sentiment analysis issue better.

In this paper, the sentiment analysis based on aspect is mainly focused. The overall polarity is classified by sentence and document phase. A sentence consists of various aspects and opinion. Instance, price is fair and food is decent however service is so bad, for aspects service is negative when food and price is positive. Hence, the connection between an aspect and the content of a sentence is explored. All aspects consisting in the review are deduced and the polarity on every aspect is computed.

4. METHODOLOGY

Methods of sentiment analysis can be categorized into three approaches such as lexicon-based approach, machine learning and deep learning-based approach and hybrid approach (combine machine learning and lexicon-based approach). In this paper, we use lexicon-based approach in our study.

Lexicon based approach search lexicons from the sentence and compare with the seed words. Lexicon consists of list of words and expressions. This approach is based on count the number of positive words and negative word from a given text. Two approaches are corpus-based approach and dictionary-based approach. Machine learning algorithms such as SVM, Naive Bayes can be addressed as a combination of methods to automatically detect the available pattern in the given set of data. There are supervised and unsupervised approaches in this approach.

4.1. SentiWordNet

The SentiWordNet is a document resource that includes a collection of english terms that has been attributed positive and negative scores. It supports the information that is deduced and matched for providing the overall score and the forecasting expression required in the document. It is composed with tens of thousands of words, and the representation of part of speech, meanings, and the degree of negativity and positivity of the word, ranging from 0 to 1. The derivation of this words is done through the WordNet 2.0 database that is a database of english words and their meanings in which the organization of terms is done based on semantic meanings or relations. The grouping of these words is done with synsets known as synonyms. The SentiWordNet is formed by the extension of the WordNet with the subjectivity information (+ or -) addition to each word in the database. SentiWordNet was developed with the ranking in all terms/synsets subjectivity based on the related part of speech as the same words can possess various meanings relating to the representative part of speech. The representative parts of speech of SentiWordNet are noun, verb, adjective, and

adverb that are respective represented as 'a', 'n', 'r', 'v'. This database owns five columns, the part of speech, the offset that is a numerical ID, that when matched with a specific part of speech, defines a synsets; negative score, positive score and synset terms which contains all terms relating with a specific synset.

4.2. Stanford Dependency Parser

The Stanford Dependencies provide a representation of grammatical relations between words in a sentence. Dependency Parsing is the process to analyze the grammatical structure in a sentence and find out related words as well as the type of the relationship between them. Each relationship has one head and a dependent that modifies the head.

5. THE PROPOSED SYSTEM

The proposed system design is described in Figure 1.

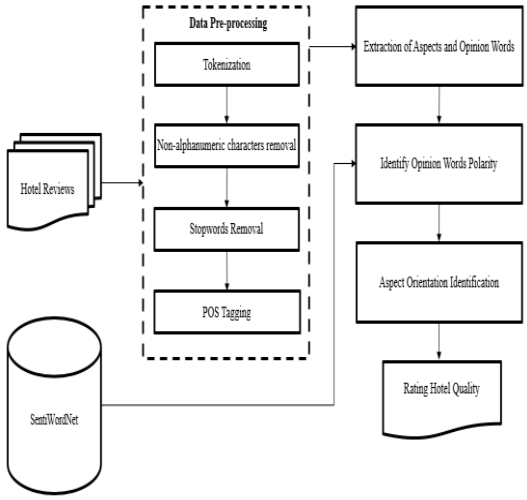


Figure 1. Proposed System Design

5.1 Data Collection

The review comments on various hotels from hotel review sites, TripAdvisor are collected. Opinions and feelings are expressed in different way, with different vocabulary, context of writing, usage of short forms and slang, makes data huge and disorganized. The text data includes positive, negative, reviews and mixed by writing with formal and informal writing style without segmentation. Customers show positive, negative and sometimes both positive and negative opinion in the review.

Table 1. Sample Hotel Reviews

Only the park outside of the hotel was beautiful	Positive
Good restaurant with modern design great chill out place Great park nearby the hotel and awesome main stairs	Positive
Location on the park with easy access to tram lines was very good	Negative

5.2 Data Preprocessing

Firstly, the conversion of data into lowercase letters is performed. The steps included in preprocessing steps are:

Tokenization: Tokenization is the division of the raw text into small chunks (words, sentences) known as tokens. The reviews are divided into separate sentences. Then the splitting of each separated sentence into words is performed. For instance, **the location on the park is excellent. the restaurant cafe in the lobby is wonderful**. After Tokenization:

['the', 'location', 'on', 'the', 'park', 'is', 'excellent', '.', 'the', 'restaurant', 'cafe', 'in', 'the', 'lobby', 'is', 'wonderful', '.']

Stopwords Removal: Stopwords removal is a common technique to eliminate noisy data in text classification. In this system, not only stopwords based on classification domain but also common stopwords are considered with the manual data examination. For instance, the most common words utilized in a text are “the”, “is”, “in”, “for”, “where”, “to”, “at”, etc. This can aid to improve the performance and increase the accuracy in task like text classification. For instance, **the restaurant cafe in the lobby is a wonderful environment and they make excellent coffee.** After Removing Stopwords:

restaurant cafe lobby wonderful environment make excellent coffee.

Non-alphanumeric characters removal: Non-alphanumeric characters are the other characters that are not letters or numbers such as commas, brackets, space, asterisks and so on. For example, **the breakfast @ this hotel was so good!!!** After removing non-alphanumeric characters:

the breakfast this hotel was so good

Part_of_Speech (POS) Tagging: In English language the main "parts of speech" are noun, adjective, verb, adverb, etc. "POS tagging" means assigning the labels (tags) to words in sentence according to its function in the sentence. In this system for assigning a label (tag) to each word, we use "Stanford POS (Parts of Speech) tagger". For example, **great hotel good staff**

POS Tagging for input Text:

[('great', 'JJ'), ('hotel', 'NN'), ('good', 'JJ'), ('staff', 'NN')]

All of these pre-processing steps are performed with Natural Language Toolkit (NLTK) in Python.

5.3 Extraction of Aspects and Opinion Words

Aspect extraction is an important step in aspect-based sentiment analysis. As user reviews of hotel contains various aspects described by many different users, these aspects are categorized into five collections of aspects groups: Price, Room, Facility, Staff, and Food. Each aspect group comprises a lot of related aspects. As the aspects may be nouns, searching of these nouns of the reviews are needed in this phase. This task is performed for each sentence of user reviews. The aspect extraction is done by using Stanford Dependency Parser. This system uses three types of rules to extract aspects:

Type 1 (R1): Using opinion words to extract aspects.

IF depends (amod, A, O) ∧ pos (A, nn) ∧ opinion word (O)

THEN aspect (A)

For example, from "The spacious room", we can extract the aspect "room" as there is an amod relation between "spacious" and "room", "spacious" is an opinion word, and "room" is a noun.

Type 2 (R2): Using aspects to extract aspects.

IF depends (conj, A1, A2) ∧ pos (A1, nn) ∧ aspect (A2)

THEN aspect (A2)

For example, from "great hotel and staff", if "hotel" has been extracted by a previous rule as an aspect, this rule can extract "staff" as an aspect.

Type 3 (R3): Using aspects and opinion words to extract new opinion words.

IF depends (amod, A, O) ∧ pos (O, jj) ∧ aspect (A)

THEN opinionword (O)

For example, from "comfy bed", if "bed" has been extracted as an aspect and "comfy" was not a seed opinion word, then "comfy" will be extracted as an opinion word by this rule.

Type 4 (R4): Using aspects to extract new aspects.

IF depends (compound, A1, A2) ∧ pos (A1, nn) ∧ aspect (A2)

THEN aspect (A1)

5.4 Identification of the Polarity of Opinion Words

In this system, the output of the dependency parser is sent to the SentiWordNet. Then it extracts the score values of opinion words from the SentiWordNet to calculate their polarity. And then, SentiWordNet categorizes the output of dependency parser into five groups like Price, Room, Facility, Staff, and Food using its similarity function. SentiWordNet is used for calculating the score of each word. "SentiWordNet" is a lexical resource in which each word is associated to positive, negative and neutral score. In this step, calculate the

polarity of the sentiments and classify them as positive, and negative.

5.5 Aspect Orientation Identification

After getting the score values for all opinion words, the summation is calculated for each aspect in each sentence. If the summation value is positive, then the aspect is good for the respective hotel. If the summation value is negative, then the aspect is bad for the respective hotel. Then the summation is calculated for total orientation of each sentence.

$$SPos = \sum_{i=1}^{n} Positive\_Text\_Score \quad (1)$$

$$SNeg = \sum_{i=1}^{n} Negative\_Text\_Score \quad (2)$$

where n=number of words, SPos is a variable which contains summation of all positive scores of text, SNeg contains summation of all negative score of text, Postive_Text_Score contains the number of all positive score words (Adj, Adv, Noun and Verb) of text, and Negative_Text_Score contains the number of all negative score words (Adj, Adv, Noun and Verb) of text.

5.6 Ranking Process Calculation

The number of count which has positive values of all comments in each hotel is extracted. Then, the computation of the percentage based on number of comments in each hotel is performed.

Rank $Value_{hotel}$= (number_of_count having positive values / number_of_comments having positive and negative values) * 100% (3)

After the calculation of rating values, the stars are provided for the hotels according to the rating values. Table 1 shows the rating values in terms of stars.

Table 1. Rating in Terms of Stars

Rating Score	**Rating Stars**
0-59.99%	1 star
60-64.99%	1.5 star
65-69.99%	2 stars
70-74.99%	2.5 stars
75-79.99%	3 stars
80-84.99%	3.5 stars
85-89.99%	4 stars
90-94.99%	4.5 stars
95-100%	5 stars

6. PERFORMANCE EVALUATION

We evaluated the system using publicly available customer review data from TripAdvisor hotel reviews [8]. In this implementation, we use 15000 reviews comments of 10 hotels: Apex Temple Court Hotel, KK Hotel George, Hotel Arena, The Park Grand London Paddington, The Hub Hotel, The Park Plaza County Hall London, Grand Royale London Hyde Park, Splendid Etoile, Hotel Trianon Rive Gauche, and Pullman Paris Centre Bercy.

Table 2. Aspects and its Polarity

Aspect	Sentiment	Polarity
Food	very good	positive
Price	expensive	negative
Staff	be cordial	positive
Facility	very bad	negative
Room	spacious	positive

Table 3. Ratings of Aspects

Hotel Name	Food	Price	Staff	Facility	Room
Hotel_Arena	0.88	0.86	0.9	0.9	0.9
KK_Hotel_George	0.92	0.9	0.9	0.89	0.88
Apex Temple Court	0.87	0.86	0.86	0.86	0.86
The Park Grand London Paddingtion	0.89	0.88	0.87	0.87	0.86
The Hub Hotel	0.76	0.81	0.77	0.79	0.8
The Park Plaza County Hall London	0.9	0.88	0.91	0.91	0.9
Grand Royale London Hyde Park	0.86	0.85	0.88	0.88	0.87
Splendid Etoile	0.9	0.95	0.91	0.91	0.91
Hotel Trianon Rive Gauche	0.9	0.9	0.85	0.85	0.83
Pullman Paris Centre Bercy	0.91	0.9	0.9	0.91	0.9

Table 4. Ratings of Hotels

Hotel Name	Hotel Rating	System Rating	Website Rating
Hotel_Arena	88.80%	4 stars	4 stars
KK_Hotel_George	89.80%	4 stars	4 stars
Apex Temple Court	86.20%	4 stars	4 stars
The Park Grand London Paddington	87.40%	4 stars	4.5 stars
The Hub Hotel	78.60%	3 stars	3 stars
The Park Plaza County Hall London	90%	4.5 stars	4.5 stars
Grand Royale London Hyde Park	86.80%	4stars	4 stars
Splendid Etoile	91.60%	4.5 stars	4.5 stars
Hotel Trianon Rive Gauche	86.60%	4 stars	4 stars
Pullman Paris Centre Bercy	90.40%	4.5 stars	4.5 stars

According to the rating values of these hotels, this system provides the same rating values of hotels in 90% as the TripAdvisor website to customers.

7. CONCLUSIONS

Sentiment analysis or opinion mining is the study that is used to analyze people emotions, sentiments towards the product. This system is used to perform evaluation measure on comments obtained from the customer. The POS tagging is used to extract the most relevant features to get better results in classifying the sentence as positive or negative. This positive and negative separation of comments is used to analyze the quality of the hotel. This system presented an aspect-based sentiment analysis of hotel reviews about aspects into positive or negative. These positive and negative separation of aspects in user comments is used to analyze the quality of the hotel. The system evaluates hotel ratings according to the quality of the hotel. According to the rating values of these hotels, this system provides the same rating values of hotels as the TripAdvisor website to customers.

REFERENCES

[1] I. Perikos, K. Kovas, F. Grivokostopoulou, and I. Hatzilygeroudis, “A System for Aspect-based Opinion Mining of Hotel Reviews”, In Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017), SciTePress, Porto, Portugal, April 25-27, 2017, pp. 388-394.

[2] V. Rybakov, and A. Malafeev, “Aspect-Based Sentiment Analysis of Russian Hotel Reviews”, Supplementary Proceedings of the 7th International Conference on Analysis of Images, Social Networks and Texts (AIST-SUP 2018), HSE, Moscow, Russia, July 5-7, 2018, pp. 75-84.

[3] Kande Trupti V, and H. P. Shah, “Recommendation System for Tourist Reviews using Aspect Based Sentiment Classification”, International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; Volume 10 Issue III Mar 2022.

[4] M. Godakandage, and S. Thelijjagoda, “Aspect Based Sentiment Oriented Hotel Recommendation Model Exploiting User Preference Learning”, 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS), 26-28 November 2020.

[5] P. Sanjay Bhargav, G. Nagarjuna Reddy, R.V. Ravi Chand, K. Pujitha, and Anjali Mathur, “Sentiment Analysis for Hotel Rating using Machine Learning Algorithms”, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-6, April 2019.

[6] S. Y. Maw, and M. A. Khine, “Aspect based Sentiment Analysis for travel and tourism in Myanmar Language using LSTM”, ICCA 2019, pp.

[7] P. B. Nehe, and A. N. Nawathe, “Aspect based sentiment classification using machine learning for online Reviews”, EasyChair Preprint no. 3051, March 26, 2020.

[8]https://www.kaggle.com/datasets/andrewmvd/trip-advisor-hotel-reviews

SPEAKER RECOGNITION SYSTEM USING COMBINED FEATURES

Ei Po Po Aung[1] and Thein Htay Zaw[2]

Faculty of Information Technologies Supporting and Maintenance, University of Computer Studies (Mandalay)

[1]eipopoaung503@gmail.com, [2]theinhtayzaw.cu@gmail.com

ABSTRACT

The process of recognizing a speaker based on unique information included in speech waves is known as speaker recognition. Speaker recognition is one of the most useful and well-liked biometric recognition methods available today, especially in contexts where security is a main priority. For speaker recognition, the system combined MFCC, Delta Features, Delta-Delta Features, LPCC, Zero crossing rate, and Energy. Using an artificial neural network (ANN) to train on speech data, this system will then assess the network's performance on a different set of data. To deal with a lot of speech data, efficient, highly accurate, and quick approaches are required. Utilizing the LibriSpeech Acoustic-Phonetic Continuous Speech Corpus, the methodology employed in this study was assessed. This system uses a Matlab artificial neural network to distinguish speakers based on a combination of features.

KEYWORDS

MFCC, Delta and Delta-Delta, Zero Crossing Rate, Short Time Energy.

1. INTRODUCTION

Based on the individual's speech, the aim of speaker recognition is to identify the real speaker among a set of other speakers [1]. Speaker identification and speaker verification are the two main subproblems that make up the speaker recognition problem [2]. The process of identifying the speaker from a set of recognized voices or speakers is known as speaker identification. The task of verifying whether a person is who they say they are (a yes/no decision) is known as speaker verification (also known as speaker authenticity or detection) [3].

A series of feature vectors can be used to represent the speech signal in order to use mathematical tools without losing generality [2]. Several of these features are applied in combinations in real-world systems. It's interesting to note that human beings identify speakers primarily based on prosodic features like intonation and length as well as source characteristics like glottal vibrations. Since it is challenging to accurately extract and create a speaker model using this data, little work has been put into developing speaker recognition system systems that use source data along with suprasegmental information, which in phonetics refers to speech features like stress, tone, and intonation [3].

In [4] provided a technique for creating a speaker recognition system employing source aspects of voice output. The source features are represented by the LP residual. An auto associative neural network model is used to extract the speaker information from the source features. This technique may offer robustness against channel and handset effects, which are known to impair the performance of a speech recognition system because the residual lacks any important spectrum information.

Based on MFCC and LPC characteristics for the Azerbaijani DataSet, Support Vector Machines were applied to the acoustic model of the Speech Recognition System in [5]. Demonstrate that SVM with a Radial Bases kernel on LPC features performs better than SVM with a polynomial kernel on MFCC features.

In [6], surveys a comparison of LPCC, MFCC and BFCC as feature extraction

approaches and ANN as a classifier. Hindi Isolated, Paired and Hybrid words are used for database purpose. The results of the series of studies show that MFCC surpasses the traditional LPCC and BFCC approaches.

SVM was used to complete the connected digit recognition problem in [5]. Show that SVM-based voice recognition performs somewhat better overall than ANN on the similar data set.

An innovative method for speaker and language recognition based on SVMs was introduced in [7]. The generalized linear discriminant sequence (GLDS) kernel is a brand-new sequence kernel that was created. It was established that this kernel was computationally effective and simple to integrate into common SVM software. The outcomes proved that the strategy was accurate and effective. Finally, a GMM system was compared to the SVM and combined with it. In terms of EER and min DCF performance, the SVM was demonstrated to perform similarly to the GMM. Additionally, it has been demonstrated that when used with a GMM system, the SVM offers complementary score data that significantly reduces error rates.

The structure of this paper is as follows: Section 2 of this article involves pre-processing. Features from this research are described in Section 3. In Section 4, the artificial neural network (ANN) model-based speaker recognition system is described. In Section 5, the findings of an evaluation of the performance of the recognition systems based on source and system properties are discussed. The paper is summarized in Section 6.

1.1 Speaker Recognition System Architecture

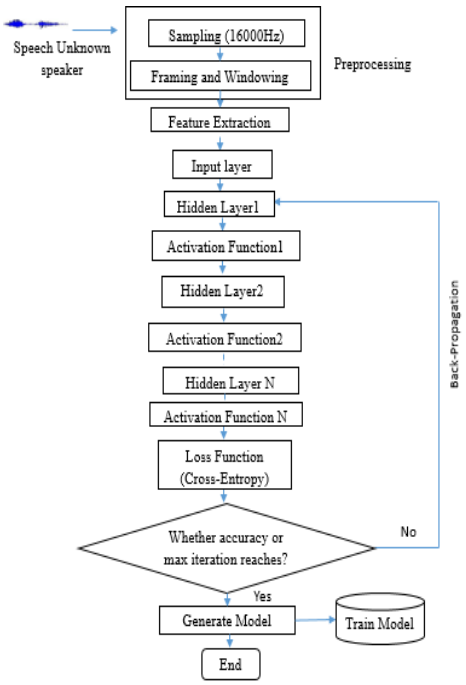


Fig 1: System Flow Diagram for Training

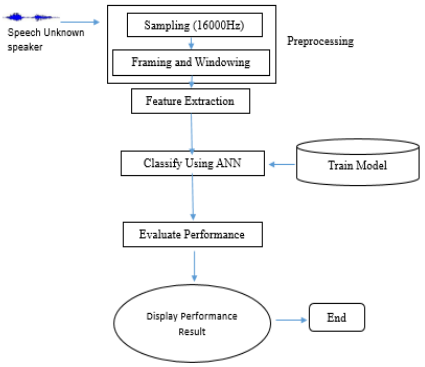


Fig2: System Flow Diagram for Testing

2. Pre-processing

The first and most crucial phase in the automatic speech recognition process is speech signal pre-processing. Sampling, framing, and windowing steps are included in the pre-processing stage. In audio signals, frame length or window length segmentation is crucial because it enhances classification performance. After pre-processing is complete, the audio output moves on to the feature extraction stage.

2.1. SAMPLING

Analog signals must first be transformed to sampled signals before being processed for classification. A time- and discrete-signal is the end outcome. The audio file was sampled by the system using 16000 Hertz.

2.2. FRAMING AND WINDOWING

Create fixed-length frames from the audio signal segments. Each frame is multiplied using a Hamming window in order to reduce spectral distortions when the speech signal is blocked [8].

The number of frame of the signal is calculated as the following equations,

$$no\ of\ frame = floor\left(\frac{s(fLength - wLength + 1)}{wLength}\right) + 1 \quad (1)$$

$$s = \frac{1}{1 - overlaprate} \quad (2)$$

Where,

- fLength = the length of signal
- wLengt = the length of window for signal
- overlaprate = the length of overlap rate on window for signal

3. FEATURE EXTRACTION

An emphasis is made on extracting elements in speaker independent speech recognition that are somewhat resistant to changes in the speaker. Therefore, feature extraction involves speech signal analysis [9]. The extraction of five features—MFCC, delta and delta-delta features, LPCC Features, Zero Crossing Rate, and Energy—is the focus of this study.

3.1. MEL FREQUENCY CEPSTRAL COEFFICIENTS

These steps are used to extract MFCC features (1) to convert from the spatial domain to the frequency domain, the discrete Fourier transform (DFT) is used [10]. (2) the power spectra are applied to the mel-filter-bank, which adds the energy in each filter.(3) the filter-bank energies' logarithm is calculated by the system.(4)it uses the log filter-bank energies' discrete cosine transform (DCT).

Mel-filter-bank

To convert the frequency in Hertz to mel, it is used the following formula

$$m = 2595 log10\left(1 + \frac{f}{700}\right) \quad (3)$$

where, f: the frequency in Hertz.

3.2. DELTA FEATURES AND DELTA-DELTA FEATURES

It is advantageous to have knowledge of how the features change over time for sound signals. MFCCs by themselves provide the power spectrum envelope for a single frame, and delta (differential) coefficients are derived to simulate the dynamics of the MFCCs as

$$d_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2} \quad (4)$$

Where, d_t is the delta coefficient vector of length N for the frame *t* and c_t is the MFCC vector for frame t. Delta-Delta (Acceleration) coefficients are calculated in the same way, but they are calculated from the deltas, not the static coefficients.

3.3 LINEAR PREDICTIVE CEPSTRAL COEFFICIENTS (LPCC)

Another typical method for obtaining spectral information is LPC. The Levinson-Durbin recursion is used to determine the coefficients in order to reduce residual error energy.

Using a linear mixture of its previous values, the signal u is predicted as

$$u_n = \sum_{i=1}^{p} a_i\, u_{n-i} \quad (5)$$

where, a_i is the linear predictor coefficients and *p* is the LPC order.

3.4. SHORT TIME ENERGY

The voice signal's amplitude obviously changes over time. In particular, the voiced signal often has a substantially higher

amplitude than the unvoiced signal [11]. Short-term energy, which provides an appropriate reflection, expresses short-term changes in amplitude.

$$E(n) = \sum_{i=1}^{N} x_{n(i)}^{2} \quad (6)$$

N is frame length, $x(i)$ is original speech signal and $E(n)$ is the short time energy of frame n.

3.5. Zero Crossing Rate (ZCR)

A zero crossing is defined to occur in the context of discrete-time signals when consecutive samples contain different algebraic signs. A simple method for evaluating a signal's frequency consistency is to evaluate at the frequency at which zero crossings occur. The number of times the amplitude of speech signals in a given time period or frame crosses a value of zero is known as the zero-crossing rate [12].

$$ZCR = \frac{1}{2N} \sum_{i=1}^{N} |\, sgn\,(u_i - u_{i-1})| \quad (7)$$

where, i = [1, 2, 3… N], N is the quantity of samples from the input, u_i is the discrete time input and *sgn*(.) is defined as

$$sgn\,(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases} \quad (8)$$

4. Artificial Neural Networks System

Artificial neural networks (ANNs) are computer systems made up of interconnected nodes that function substantially like human neurons. Data can be roughly divided into classes using neural networks. An input layer, many hidden layers, and an output layer make up the Artificial Neural Networks (ANN), which are multi-layer fully connected neural nets. Each node in one layer is linked to each node in the next layer [13]. A specific node applies a nonlinear activation function on the weighted sum of its inputs. This is the node's output, which later serves as the input for a different node in the next layer. Utilize a loss function to calculate the error by contrasting the final result with the real target from the training set. Using the Softmax activation function, this deep neural network is trained by learning the weights connected to all of the edges [14]. The output signal will only be a linear function in the absence of an activation function, making it impossible for neural networks to learn complicated data like speech, images, and other types of complex data. Weighted sum can be described mathematically using the following equation:

$$weight\ sum = \sum_{i=1}^{N} x_i\, w_i \quad (9)$$

where, x_i is input of each layer and w_i is weight.

5. Performance and Results

In this section the system present an experimental validation of the benefit of feature combination in speaker verification performance. The following section provides details about the experimental setup used throughout all the experiments.

5.1 Experimental Setup

The System recognize by three steps. They are Pre-processing step, Feature extraction step and Classification step (ANN).Create fixed-length frames from the audio signal segments .Frame size is 25ms.Overlapping size is 15ms. Each frame is multiplied using a Hamming window in order to reduce spectral distortions.

Three types of feature combination are used in this system. First type of combination is MFCC, Delta and Delta-Delta. Secondly, MFCC, Delta and Delta-Delta, LPCC. In Third, MFCC, Delta, Delta-Delta, LPCC, Zero Crossing rate and Energy. Total feature vectors for 6 feature set are 63 feature vectors. MFCC are 17 feature vectors, 17 for Delta, 17 for Delta-Delta, 10 for LPCC, 1 for ZCR and 1 for Energy. In ANN classification method hidden layers used are 2, 4, 6 layers respectively. Hidden node are 10 nodes. A specific node applies a nonlinear activation function (Softmax) on the weighted sum of its inputs.

Utilizing the LibriSpeech Acoustic-Phonetic Continuous Speech Corpus, the methodology employed in this study was assessed. This corpus comprises clean speech data and is mostly used for speech recognition applications. This system uses a Matlab artificial neural network to distinguish speakers based on a combination of features.

5.2 CORPUS

The LibriSpeech (train-clean-100) corpus of read speech was created to offer speech data for the development and testing of automatic speech recognition systems as well as for the acquisition of acoustic-phonetic knowledge. LibriSpeech contains a total of approximately 100 hours long by each of 125 female speakers and 126 male speakers. This system will use 5 female speakers, 5 male speakers and (480+100) audio files to train and test [14].

5.3 RESULT AND DISCUSSION

The performance of the speaker recognition system was evaluated on the LibriSpeech Corpus. Speaker recognition system detection results are as shown in tables. These tables displayed the maximum, average, and minimum accuracy scores for the test set of 10 speaker audios.

5.3.1 EVALUATION METRIC FOR SPEAKER RECOGNITION SYSTEM

$$\%Precision = \frac{TruePositives}{TruePositives + FalsePositive} \quad (10)$$

$$\%Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (11)$$

$$\%F = \frac{2*Precision*Recall}{Precision+Recall} \quad (12)$$

Table 1. Confusion matrix of Speaker Recognition System using 2 hidden layers with (Features=MFCC, Delta, Delta-Delta)

	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
Sp1	12211	250	253	237	32	161	1	380	45	252
Sp2	170	10284	377	145	512	387	100	18	315	155
Sp3	364	434	8123	169	11	523	8	551	254	309
Sp4	87	138	100	9015	36	279	199	427	503	404
Sp5	87	231	134	125	7398	294	779	42	162	25
Sp6	106	221	599	517	56	9703	991	64	284	150
Sp7	10	27	640	250	45	312	9609	122	103	41
Sp8	229	110	465	344	0	33	0	11437	317	130
Sp9	4	303	197	443	208	95	105	428	11642	51
Sp10	79	156	232	263	0	145	22	318	58	11685
Total	13347	12154	11120	11508	8298	11932	11814	13787	13683	13202
Accuracy	91.48	84.61	73.04	78.33	89.15	81.31	81.33	82.95	85.08	88.50
Total Accuracy										83.58

In the first part of experiments, the system used 2 hidden layers with three features (MFCC, Delta and Delta-Delta) and to detect ten speaker speech. The result for experiments is shown in table 1. In speaker recognition system, the three features combination accuracy is lower than four features and all six features combination.

Table 2. Confusion matrix of Speaker Recognition System using 2 hidden layers with (Features=MFCC, Delta, Delta-Delta and LPCC)

	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
Sp1	12310	194	188	181	14	185	2	417	46	223
Sp2	209	10468	364	212	504	306	418	38	305	141
Sp3	323	472	8476	132	13	365	4	653	295	252
Sp4	83	129	33	9062	11	294	345	314	514	398
Sp5	28	190	110	108	7488	154	251	38	115	17
Sp6	60	225	353	319	63	10226	561	45	176	163
Sp7	25	21	681	275	28	233	10128	111	90	26
Sp8	189	122	472	519	0	35	0	11392	382	184
Sp9	7	190	224	334	177	45	90	369	11703	27
Sp10	113	143	219	366	0	89	15	410	57	11771
Total	13347	12154	11120	11508	8298	11932	11814	13787	13683	13202
Accuracy	92.23	86.12	76.22	78.74	90.23	85.70	85.72	82.62	85.52	89.16
Total Accuracy										85.23

In the second part of experiments, the system used 2 hidden layers with four features (MFCC, Delta, Delta-Delta and LPCC) to detect ten speaker speech. The result for experiments is shown in table 2. Accuracy is a slightly rises in four features combination than three features combination.

Table 3. Confusion matrix of Speaker Recognition System using 2 hidden layers with (Features=MFCC, Delta, Delta-Delta, ZCR, Energy and LPCC)

	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
Sp1	12326	222	163	186	5	173	1	327	48	222
Sp2	190	10441	438	211	350	456	97	41	305	128
Sp3	264	424	8719	152	9	457	141	624	304	161
Sp4	111	73	65	9311	14	198	480	487	592	235
Sp5	37	244	52	74	7713	53	250	54	91	16
Sp6	33	225	379	291	51	10146	592	76	220	193
Sp7	14	15	494	313	19	262	10203	82	65	61
Sp8	244	119	421	382	0	23	1	11222	303	164
Sp9	5	201	230	343	137	80	41	404	11708	67
Sp10	123	190	159	245	0	84	8	470	47	11955
Total	13347	12154	11120	11508	8298	11932	11814	13787	13683	13202
Accuracy	92.35	85.90	78.40827	80.90	92.95	85.03	86.36	81.39	85.56	90.55
Total Accuracy										85.94

In the third part of experiments, the system used 2 hidden layers with all six features (MFCC, Delta, Delta-Delta, ZCR, Energy and LPCC) to detect ten speaker speech. The result for experiments is shown in table 3. The table demonstrates highest accurate result than three and four features combination methods in ten speaker.

Table 4. Confusion matrix of Speaker Recognition System using 4 hidden layer with (Features=MFCC, Delta, Delta-Delta)

	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
Sp1	12337	242	198	245	43	191	0	412	30	225
Sp2	219	10428	377	280	487	313	43	33	393	128
Sp3	229	388	8374	103	29	697	6	704	271	186
Sp4	73	129	40	9104	24	231	134	487	406	284
Sp5	99	265	164	74	7403	233	567	39	117	27
Sp6	55	241	574	504	95	9612	940	51	224	213
Sp7	12	15	556	270	12	414	9964	137	59	49
Sp8	260	86	484	376	0	55	0	11265	299	217
Sp9	3	173	174	365	205	78	155	315	11837	35
Sp10	60	187	179	187	0	108	5	344	47	11838
Total	13347	12154	11120	11508	8298	11932	11814	13787	13683	13202
Accuracy	92.43	85.79	75.30	79.11	89.21	80.55	84.34	81.70	86.50	89.66
Total Accuracy										84.46

The system used four hidden layers with three features (MFCC, Delta, and Delta-Delta) in the fourth trial to identify the speech of ten speakers. Table 4 displays the experiment results. In a speaker identification system, the accuracy of the three features combined is lower than that of the combinations of four and six features.

Table 5. Confusion matrix of Speaker Recognition System using 4 hidden layers with (Features=MFCC, Delta, Delta-Delta and LPCC)

	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
Sp1	12360	266	206	155	12	203	0	386	39	261
Sp2	104	10134	382	194	359	498	91	19	243	142
Sp3	309	621	8723	141	7	314	30	635	304	363
Sp4	144	149	68	9277	17	335	226	528	520	314
Sp5	45	247	98	136	7652	88	205	33	211	15
Sp6	63	256	382	355	54	10015	611	48	252	253
Sp7	15	10	459	220	9	283	10627	105	86	42
Sp8	257	66	459	507	0	18	0	11476	270	166
Sp9	4	195	193	304	188	105	21	296	11708	46
Sp10	46	210	150	219	0	73	3	261	50	11600
Total	13347	12154	11120	11508	8298	11932	11814	13787	13683	13202
Accuracy	92.60	83.37	78.44	80.69	92.21	83.93	89.95	83.23	85.56	87.86
Total Accuracy										85.78

The system used four hidden layers with four features (MFCC, Delta, Delta-Delta, and LPCC) throughout the five parts of the tests to identify the speech of ten speakers. Table 5 displays the experiment results. When four features are combined, accuracy somewhat improves over when three features are combined.

Table 6. Confusion matrix of Speaker Recognition System using 4 hidden layers with (Features=MFCC, Delta, Delta-Delta, ZCR, Energy and LPCC)

	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
Sp1	12398	203	173	185	20	178	2	477	33	219
Sp2	143	10602	330	70	326	471	82	34	285	209
Sp3	297	465	8917	202	21	356	17	732	380	184
Sp4	54	74	60	9422	30	265	181	424	414	273
Sp5	72	213	99	173	7724	75	71	32	153	5
Sp6	62	228	331	334	24	10149	814	68	213	119
Sp7	21	15	382	236	15	258	10579	79	101	63
Sp8	200	95	396	338	1	34	0	11076	366	149
Sp9	2	126	190	348	137	61	65	424	11683	59
Sp10	98	133	242	200	0	85	3	441	55	11922
Total	13347	12154	11120	11508	8298	11932	11814	13787	13683	13202
Accuracy	92.88	87.23	80.18	81.87	93.08	85.05	89.54	80.33	85.38	90.30
Total Accuracy										86.58

The system used four hidden layers with all six features—MFCC, Delta, Delta-Delta, ZCR, Energy, and LPCC—in the six parts of the experiments to identify the speech of ten speakers. Table 6 displays the experiment results. The table shows that using ten speakers, four features combination approaches produce the most accurate results.

Table 7. Confusion matrix of Speaker Recognition System using 6 hidden layers with (Features=MFCC, Delta, Delta-Delta)

	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
Sp1	12117	204	285	191	71	211	5	484	30	394
Sp2	274	10116	375	241	718	455	54	20	398	276
Sp3	278	655	8363	111	24	493	5	696	246	230
Sp4	83	83	23	9036	12	605	222	474	423	350
Sp5	114	277	241	79	7172	296	1417	27	98	15
Sp6	98	344	557	517	84	9097	1064	61	429	130
Sp7	52	14	509	341	37	488	8998	125	123	11
Sp8	278	72	404	298	0	20	0	11160	281	220
Sp9	1	220	195	420	180	167	49	335	11602	56
Sp10	52	169	168	274	0	100	0	405	53	11520
Total	13347	12154	11120	11508	8298	11932	11814	13787	13683	13202
Accuracy	90.78	83.23	75.20	78.51	86.43	76.24	76.16	80.94	84.79	87.25
Total Accuracy										81.95

In the seven parts of experiments, to detect the speech of ten speakers, the system employed six hidden layers and three features (MFCC, Delta, and Delta-Delta). Table 7 shows the results of the experiment. In a speaker identification system, the accuracy of the three features combined is lower than that of the combinations of four and six features.

Table 8.Confusion matrix of Speaker Recognition System using 6 hidden layers with (Features=MFCC, Delta, Delta-Delta and LPCC)

	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
Sp1	12460	193	287	169	11	163	0	316	24	275
Sp2	153	10465	294	156	378	591	16	14	299	206
Sp3	284	454	8647	201	14	442	29	780	349	218
Sp4	54	95	44	9154	20	305	262	430	366	443
Sp5	40	227	112	126	7576	102	399	38	187	13
Sp6	67	272	394	371	72	9887	457	52	207	269
Sp7	20	7	601	303	21	289	10569	111	121	21
Sp8	236	60	388	351	0	17	0	11485	292	286
Sp9	1	190	213	439	206	69	82	268	11789	23
Sp10	32	191	140	238	0	67	0	293	49	11448
Total	13347	12154	11120	11508	8298	11932	11814	13787	13683	13202
Accuracy	93.35	86.10	77.76	79.54	91.29	82.86	89.46	83.30	86.15	86.71
Total Accuracy										85.65

The system used six hidden layers with four features (MFCC, Delta, Delta-Delta, and LPCC) in the 8 sections of the tests to identify the speech of ten speakers. Table 8 presents the test results. When four features are combined, accuracy slightly improves over when three features are combined.

Table 9. Confusion matrix of Speaker Recognition System using 6 hidden layers with (Features=MFCC, Delta, Delta-Delta, ZCR, Energy and LPCC)

	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
Sp1	12475	282	253	242	35	147	1	455	40	215
Sp2	147	10462	294	197	397	448	25	28	311	147
Sp3	232	437	8587	128	9	572	2	647	289	180
Sp4	70	56	39	9064	8	267	224	435	466	339
Sp5	85	189	93	47	7533	97	355	43	126	20
Sp6	43	263	499	435	43	9865	685	41	335	220
Sp7	11	7	581	293	8	283	10378	94	86	68
Sp8	208	72	360	387	1	19	0	11220	255	161
Sp9	2	176	207	481	264	135	132	397	11739	24
Sp10	74	210	207	234	0	99	12	427	36	11828
Total	13347	12154	11120	11508	8298	11932	11814	13787	13683	13202
Accuracy	93.46	86.07	77.22	78.76	90.78	82.67	87.84	81.38	85.79	89.59
Total Accuracy										85.36

Finally, the system detected the speech of ten speakers using 6 hidden layers and all six features (MFCC, Delta, Delta-Delta, ZCR, Energy, and LPCC) .Table 9 summarizes the experiment. The table shows a slightly less accurate result than four feature combinations in ten speakers, but a greater accuracy result than three feature combinations.

The proposed approach in the speaker recognition system exceeded the other eight baseline methods in terms of accuracy by using the model created with the optimal subset of features. It is observed that feature combination methods that using all 6 features and 4 hidden layers play a significant role in this system.

The simplest option to view the effectiveness of neutral networks and the adding of hidden layers is as 3 phases. They are enhancing, decreasing, and adversely affecting performance. Increasing the number of layers (making it “deeper”) would increase test set accuracy, but eventually the test set accuracy would start to decline as the model began to over fit the training set.

6. CONCLUSION

This system implemented speaker recognition system using combined features on Artificial Neural Networks (ANN) method. This study analyses three types of combination methods to determine the more accurate output of speaker recognition system .The experimental result show that

the combination of all 6 features and 4 hidden layers gives the best accurate result between 9 experiment .Addition of hidden layers helps improve the model, but only up to a certain point, and further addition of layers can actually harm the model's performance.

REFERENCES

[1] Neha Chauhan, Tsuyoshi Isshiki, Dongju Li, "Speaker Recognition Using LPC, MFCC, ZCR Features with ANN and SVM Classifier for Large Input Database" 2019 IEEE 4th International Conference on Computer and Communication Systems.

[2] S. K. Singh (Roll No. 03307409)

"FEATURES AND TECHNIQUES FOR SPEAKER RECOGNITION" M. Tech. Credit Seminar Report, Electronic Systems Group, EE Dept, IIT Bombay submitted Nov 03.

[3] Douglas A. Reynolds "An Overview of Automatic Speaker Recognition Technology" MIT Lincoln Laboratory, Lexington, MA USA.

[4] B. Yegnanarayana, K. Sharat Reddy and S. P. Kishore, "SOURCE AND SYSTEM FEATURES FOR SPEAKER RECOGNITION USING AANN MODELS", Speech and Vision Laboratory Department of Computer Science and Engineering Indian Institute of Technology Madras Chennai-600036, INDIA.

[5] Kamil Aida-zade, Anar Xocayev, Samir Rustamov, William M. Campbell, "Speech Recognition using Support Vector Machines", 2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT).

[6] Taabish Gulzar, Anand Singh, Sandeep Sharma, Ph.D "Comparative Analysis of LPCC, MFCC and BFCC for the Recognition of Hindi Words using Artificial Neural Networks", International Journal of Computer Applications (0975 – 8887) Volume 101– No.12, September 2014.

[7] W.M. Campbell *, J.P. Campbell, D.A. Reynolds, E. Singer, P.A. Torres-Carrasquillo

"Support vector machines for speaker and language recognition", Computer Speech and Language 20 (2006) 210–229.

[8] Oday Kamil Hamid "Frame Blocking and Windowing Speech Signal" Journal of Information, Communication, and Intelligence Systems (JICIS) Volume 4, Issue 5, December 2018.

[9] Manish P. Kesarkar, "FEATURE EXTRACTION FOR SPEECH RECOGNITON", M.Tech. Credit Seminar.

[10] © The Author(s) 2017 K.S. Rao and Manjunath K.E., "Speech Recognition Using Articulatory and Excitation Source Features, Springer Briefs in Speech Technology", DOI 10.1007/978-3-319-49220-9

[11] Bachu R.G, Kopparthi S., Adapa B., Barkana B.D. "Separation of Voiced and Unvoiced using Zero crossing rate and Energy of the Speech Signal" arXiv:1808.00158v3 [eess.AS] 9 Aug 2019

[12] D.S.Shete 1 , Prof. S.B. Patil 2 ,Prof. S.B. Patil 3 "Zero crossing rate and Energy of the Speech Signal of Devanagari Script" IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) Volume 4, Issue 1, Ver. I (Jan. 2014), PP 01-05 e-ISSN: 2319 – 4200, p-ISSN No.: 2319 – 4197 www.iosrjournals.org

[13] Anett Antony, R. Gopikakumari, "Speaker identification based on combination of MFCC and UMRT based features" Selection and peer-review under responsibility of the scientific committee of the 8th International Conference on Advances in Computing and Communication (ICACC-2018)

[14] Arden dertat "Applied Deep Learning - Part 1: Artificial Neural Networks", Aug 8,2017.

[15] http://www.openslr.org/12.

MYANMAR CHARACTERS SEGMENTATION AND RECOGNITION USING TESSERACT ENGINE

Thiri Mon

University of Computer Studies (Mandalay)
thirimon8840@gmail.com

ABSTRACT

Optical character recognition, usually abbreviated to OCR, is the mechanical or electronic conversion of scanned or photographed images of typewritten or printed text into machine-encoded/computer-readable text.This system has presented Myanmar character recognition using open-source Tesseract OCR engine. In this system, there are four stages, namely, image acquisition, preprocessing, text regions segmentation, and recognition. In image acquisition, the system supports the image files such as .png, .jpg, .jpeg, .tiff, and .bmp format. In preprocessing, grayscale conversion, contrast enhancement, sharpening, binary image conversion, and deskewing operations are done to improve the accuracy of the system. In text regions segmentation, horizontal and vertical profiling method is used to detect text regions in the image. Finally, each text region is recognized with the open-source Tesseract engine and the results are displayed using Microsoft word file.

KEYWORDS

Optical Character Recognition, Tesseract, Myanmar Character Segmentation, Myanmar Character Recognition

1. INTRODUCTION

Optical Character Recognition **(**OCR**)** is the electronic conversion of images of text into digitally-encoded text using specialized software. OCR software enables a computer to convert a scanned document, a digital photo of text, or any another digital image of text into machine-readable, editable data [4, 7].

Optical character recognition system is used in many applications such as document reading, script recognition, postal processing, banking, security (i.e. passport authentication) and mail sorting, language identification, writer identification, smart card processing system, signature verification, license plate recognition system, bank cheque processing, automatic data entry, postal automation, money counting machine, address and zip code recognition etc. Many organizations are depending on optical character recognition system to eliminate the human interactions for better performance and efficiency [6,8]. There are four basic stages when converting data into an editable text format using optical character recognition. These stages are image acquisition, pre-processing; character recognition and output post-processing. Before recognizing the character, optical character recognition software usually pre-processes images to enhance their quality for capturing data accurately. Preprocessing removes or minimizes unwanted distortions and enhances specific image features. For ensuring higher accuracy of optical character recognition, post-processing is a popular error correction technique. Post processing usually includes restricting the output by a lexicon to identify the right words, numbers, and codes [9, 12].

2. THEORY BACKGROUND

There are several phases in our system, namely, Image acquisition, Preprocessing, Segmentation, and Recognition.

In Image acquisition, the recognition system acquires a scanned image as an input image. The image should have a

specific format such as PNG, JPEG, BMP etc. This image is acquired through a scanner.

Preprocessing should be done to enhance the quality of image. Preprocessing is the preliminary step which transforms the data into a format that will be more easily and effectively processed.

Preprocessing can be done through various ways, Grayscale Conversion, Contrast Enhancement, Enhance Text Details, Binary Image Conversion, Deskew Text Line, Segment Text Line.

In segmentation, the image is segmented into characters before being passed to classification phase. Segmentation is an integral part of any text-based recognition system. It assures efficiency of classification and recognition. Accuracy of character recognition heavily depends upon segmentation phase.

In recognition, we use the open-source Tesseract OCR engine. Tesseract OCR engine used LSTM (Long Short Term Memory) neural networks.

It was developed at HP in between 1984 to1994. It was modified and improved in 1995 with greater accuracy. In late 2005, HP released Tesseract for open source.

It is highly portable. It is more focused towards providing less rejection than accuracy. Currently only command base version is available. As of now Tesseract version 4.0 is released and available for use. HP never used it. Now it is developed and maintained by Google. It provides support for various languages [1].

3. ARCHITECTURE OF THE PROPOSED SYSTEM

Overview of the system is shown in figure 1. In this system, there are four stages, namely, image acquisition, preprocessing, text line segmentation, and recognition.

In image acquisition step, the system will accept the input image which is obtained by the scanning the printed Myanmar document using scanner.

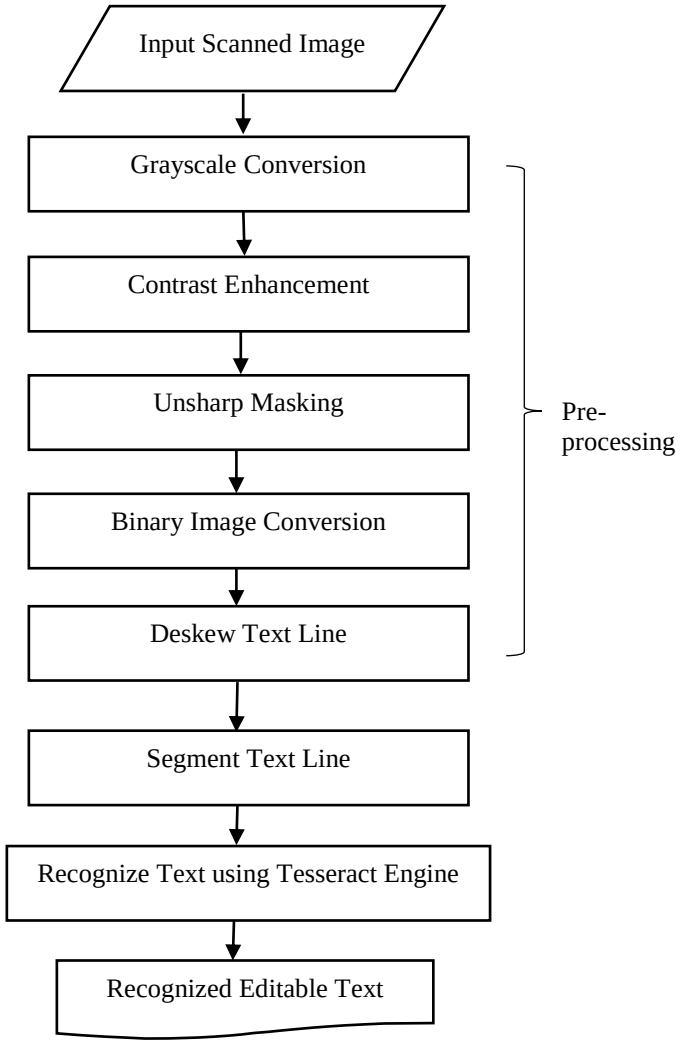


Figure 1. Overview of the system

3.1. Preprocessing

There are five steps in preprocessing, namely grayscale conversion, contrast enhancement, image sharpening, binary image conversion, and deskew the text line.

3.1.1. Grayscale Image Conversion

In this system, luminance algorithm is used to convert RGB color image into grayscale image. Gray scale value for each RGB image pixel can be computed as follows:

$$\text{Gray value} = 0.3R + 0.59G + 0.11B \quad (1)$$

3.1.2. Contrast Enhancement

In this system, Histogram Equalization is used to improve contrast of the overall objects presents in the processed document image. Let f be a given image represented as a m_r by m_c matrix of integer pixel intensities ranging from 0 to L − 1. L is the number of possible intensity values, often 256. Let p denote the normalized histogram of f with a bin for each possible intensity. So

$$P_n = \frac{number\ of\ pixels\ with\ intensity\ n}{total\ number\ of\ pixels} \quad (2)$$

The histogram equalized image g will be defined by

$$g_{i,j} = floor\left((L-1)\ \sum_{n=0}^{f_{i,j}} P_n\right) \quad (3)$$

where floor () rounds down to the nearest integer.

3.1.3. Unsharp Masking

Unsharp masking is used to sharpen the image to enhance text details in the image. Enhanced image can be obtained by the following equation:

Eimage= original + amt *(original – blurred) (4)

Where, original is original image and blurred is Gaussian smoothed image. In this system, amt parameter value 1.5 is used.

3.1.4. Binary Image Conversion

Otsu's thresholding technique is used for binary image conversion.

In Otsu's thresholding algorithm, the following steps are done.

1. Compute Within Classes Variance

$$\sigma_w^2 = W_b\ \sigma_b^2 + W_f\ \sigma_f^2 \quad (5)$$

2. Compute Between Classes Variance

$$\sigma_B^2 = W_b\ W_f\ (\mu_b - \mu_f)^2 \quad (6)$$

Where W_b is background weight, W_f is foreground weight, μ_b is mean of background pixels, and μ_f is mean of foreground pixels.

3. Choose the threshold value T that has minimum within classes variance or maximum between class variance.
4. The next step is binary image conversion. If g(x, y) is a threshold version of f(x, y) at some global threshold T, it can be defined as:

$$g(x,y) = \begin{cases} 1\ if\ f(x,y) \geq T \\ 0\ Otherwise \end{cases} \quad (7)$$

3.1.5. Deskew Text Line

When scanning the Myanmar document there may have slanted text line in the image sometimes. Because of slanted text lines, the accuracy of the recognition rate will be deduced. So, we need to deskew the text line in order to improve the accuracy of the recognition rate. In this system, we use Hough Transform algorithm to detect the degree of skew angle of the detected text line.

Hough Transform algorithm works as follows:

1. Extract Edge of the image

2. Create accumulator array and initialize to zero

3. For each edge pixel

    For each theta

        calculate rho = x cos(theta) + y sin(theta)

        increment accumulator at rho, theta

4. Find maximum values in accumulator (line)

5. Extract related rho, theta

When the skew angle is obtained, it is easy to correct the skew by using the rotation using the following formula.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (8)$$

Where θ is the detected skew angle.

3.2. Segment Text Line

Now, it is ready to detect the text line from the deskewed image. In this system, Horizontal and Vertical Profile method is used to detect text line from the image. In this method, each row in the image is checked whether the row is foreground or background using average value of the row. If the average value of the row is greater than predefined threshold value, the row is background and otherwise, the row is foreground. If the row is foreground, the system finds the next background row to determine the text region. Then, detected text regions are sorted according to their coordinates.

3.3. Recognition using Tesseract Engine

Finally, detected text regions from the image are recognized with the Tesseract Engine and recognized texts are obtained. Tesseract is an open-source optical character recognition engine which used long short-term memory neural networks.

Accuracy of the OCR system can be calculated by two ways, namely, character error rate (CER) and word error rate (WER). This system will use character error rate (CER) to compute the accuracy of the system.

Character error rate (CER) can be calculated as follows:

$$CER = \frac{S + D + I}{N} \quad (9)$$

Where,

S = number of substitutions

D= number of deletions

I = number of insertions

N = number of characters in reference text

4. IMPLEMENTATION OF THE SYSTEM

Firstly, the system accepts the input image that contains Myanmar characters. The system supports the image file types such as .png, .bmp, .tiff, .jpg, .jpeg file formats. By experimentation results, the image file should have at least 300 dpi resolution to improve recognition rate of the system. The system assumes the input image as a RGB color image. Figure 2 is the sample input image.

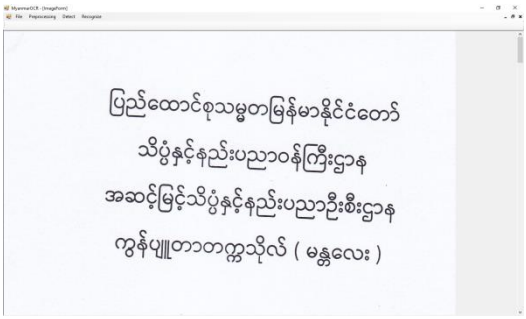


Figure 2. Sample Input Image

In this system, we use Otsu's thresholding algorithm to convert the grayscale image into binary image. Figure 3 is resulted image of Otsu's thresholding algorithm.

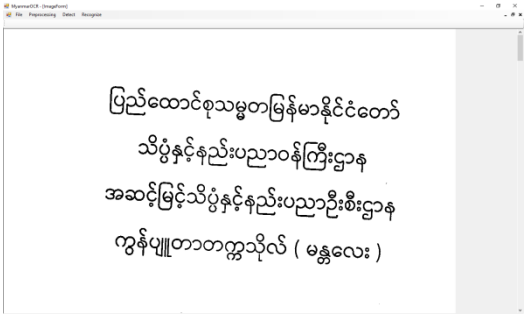


Figure 3. Binary Image

Straight lines are detected by using Hough transform algorithm. Skew angle can be calculated with the deviation of the line with horizontal axis.

When the skew angle is obtained, rotation can be done to deskew the text line. If the skew angle is negative then rotation can be done using positive degree. If the skew angle is positive then rotation can be done using negative degree. Figure 4 is the

resulted image of deskewing the slanted text line.

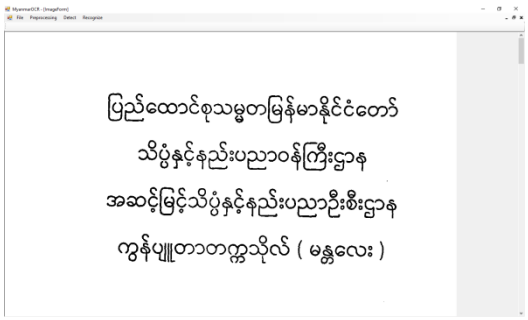


Figure 4. Deskewed Image

After preprocessing have been done, it is time to locate text regions in the image. Finding text regions will improve performance of OCR based solutions by reducing the number of pixels to be processed. In this system, horizontal and vertical profile method is used to detect text regions in the image. Figure 5 is the detected text regions with rectangle.

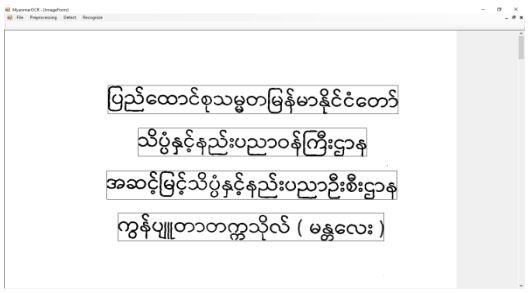


Figure 5. Detected Text Regions

Finally, detected text regions from the image are recognized with the Tesseract Engine and recognized texts are obtained. Tesseract is an open-source optical character recognition engine which used long short-term memory neural networks. Figure 6 is the recognized text displayed with Microsoft word file.

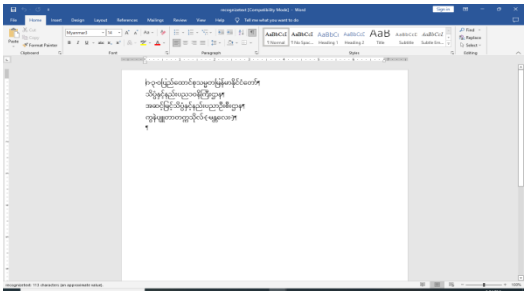


Figure 6. Recognized Text

5. Conclusions

In optical character recognition field, recognition of characters for language such as Myanmar characters still remain an open challenge.

This system is implemented for Myanmar characters recognition from the scanned image which contains Myanmar characters. Accuracy of the system is calculated by using character error rate (CER) method and if CER value is zero, the system recognized the characters with 100% accuracy rate. That is, the lesser the CER value, the higher the accuracy rate. In this system, CER value is obtained near 0.04 for the sample input image of figure 2. But for the other sample images which contains many text lines, the CER value is obtained about 0.2.

Acknowledgements

I would like to take this opportunity to express my sincere thanks to my all teachers who gave me many valuable advice and information. I am also graceful to all respectable people who directly or indirectly contributed towards the success of this thesis.

I would like to express my respectful gratitude to Dr. San San Tint, Pro-Rector of the University of Computer Studies (Mandalay), for allowing me to develop this thesis and giving me general guidance during the period of study.

I am deeply grateful to my supervisor, Dr. Saw Thandar Myint, Professor, Head of Department of Information Technology Supports and Maintenance, University of

Computer Studies, (Mandalay), for her close guidance, helpful advice, numerous invaluable suggestions, patience and support she has provided throughout my time.

Thanks, are also extended especially to all my teachers, my parents, seniors, friends and staff members of the University of Computer Studies (Mandalay) for the contribution towards the completion of this thesis.

REFERENCES

[1] Chirag Patel, Atul Patel, PhD., Dharmendra Patel, "Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study", International Journal of Computer Applications (0975 – 8887) Volume 55– No.10, October 2012.

[2] Christopher Kanan, Garrison W. Cottrell, "Color-to-Grayscale: Does the Method Matter in Image Recognition?", Department of Computer Science and Engineering, University of California San Diego, La Jolla, California, United States of America.

[3] Daniela O. Albanez, "Image Segmentation by Otsu Method", Anais do Encontro Anual de Computação (ENACOMP 2015). ISSN: 2178-6992.

[4] Karez Abdulwahhab Hamad, Mehmet Kaya, "A Detailed Analysis of Optical Character Recognition Technology", International Journal of Applied Mathematics, Electronics and Computers, ISSN: 2147-8228.

[5] Nikita Kotwal, Ashlesh Sheth, "Optical Character Recognition using Tesseract Engine", International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 10 Issue 09, September-2021.

[6] Noman Islam, Zeeshan Islam,Nazia Noor, "A Survey on Optical Character Recognition System", Journal of Information & Communication Technology-JICT Vol. 10 Issue. 2, December 2016.

[7] Prasanta Pratim Bairagi, "Optical Character Recognition for Hindi", International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 05 | May-2018, ISSN: 2395-0056.

[8] Ranjan Jana, Amrita Roy Chowdhury, Mazharul Islam, "Optical Character Recognition from Text Image", International Journal of Computer Applications Technology and Research Volume 3– Issue 4, 239 - 243, 2014, ISSN: 2319–8656.

[9] Ravina Mithe, Supriya Indalkar, Nilam Divekar, "Optical Character Recognition", International Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878, Volume-2, Issue-1, March 2013.

[10] Saravanan S, P.Siva kumar, "Image Contrast Enhancement Using Histogram Equalization Techniques: Review", International Journal of Advances in Computer science and Technology, Volume 3, No.3, March 2014, ISSN-2320-2602.

[11] Tetsuo Asano, "Variants for the Hough transform for line detection", Department of Applied Electronics, Osaka Electro-Communication University, Hatsu-cho, Neyagawa 572, Japan.

[12] Umal Patel, "An Introduction to the Process of Optical Character Recognition", International Journal of Science and Research (IJSR), India Online ISSN: 2319-7064.

[13] Zohair Al-Ameen, Alaa Muttar and Ghofran Al-Badrani, "Improving the Sharpness of Digital Image Using an Amended Unsharp Mask Filter", I.J. Image, Graphics and Signal Processing, 2019.