










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, 2025

JiTRi 2025

ကွန်ပျူတာတက္ကသိုလ် (မန္တလေး)

# JOURNAL OF INFORMATION TECHNOLOGY, RESEARCH AND INNOVATION

December, 2025

Organized by

University of Computer Studies

(Mandalay)

Volume-5, Issue-2

# JOURNAL OF INFORMATION TECHNOLOGY, RESEARCH AND INNOVATION


JITRI, 2025

Vol-5, Issue-2

UNIVERSITY OF COMPUTER STUDIES (MANDALAY)

# Journal of Information Technology, Research and Innovation (JITRI), 2025 Committee

## University of Computer Studies (Mandalay)

**Editor in Chief** *Prof. Dr. San San Tint, Rector*

**Technical Program Committee & Editorial Board**

- Prof. Dr. Thin Thin Naing
- Prof. Dr. Aye Aye Chaw
- Prof. Dr. Thandar Aung
- Prof. Dr. Mya Thida Kyaw
- Prof. Dr. Yu Ya Win
- Prof. Dr. Saw Thandar Myint
- Prof. Dr. Thiri Kyaw
- Assoc. Prof. Dr. Thein Htay Zaw
- Assoc. Prof. Dr. Yin Min Tun
- Assoc. Prof. Daw San San Lwin
- Dr. May Phyo Thu

## Acknowledgements for Reviewers and Organizing members of JITRI

The Journal of Information Technology, Research and Innovation (JITRI) 2025 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 (2025) is published as the ninth 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), 2025, Vol-5, Issue-2.

# *Table of Contents*

# HUMAN FACE SYNTHESIS USING CONDITIONAL LATENT DIFFUSION MODELS

Myat Kyaw Thu[1], and Aye Aye Chaw[2]

[1,2] Faculty of Computer Science, University of Computer Studies (Mandalay)
myatkyawthu770377608@gmail.com, ayeayechaw@cumandalay.edu.mm

## ABSTRACT

*Human face synthesis guided by textual descriptions is an essential capability for generative AI systems used in multimedia production, digital identity creation, and human–computer interaction. This system develops a text-to-image(T2I) face generation framework based on a Latent Diffusion Model (LDM) that learns to synthesize photorealistic human faces through iterative denoising of latent representations on CelebA dataset. A text encoder inspired by the (Contrastive Language–Image Pre-training) CLIP architecture transforms natural-language prompts and facial attribute descriptions into conditioning vectors, enabling semantic control over the generative process. All diffusion operations are performed by the system in a compressed latent space to reduce computational cost while maintaining high visual fidelity. In this system, the model is trained on the CelebA dataset, leveraging its extensive annotated facial attributes to support flexible generation of novel faces aligned with a user's textual input. This system performance is quantified using the CLIP score, which evaluates text–image consistency across generated samples.*

## KEYWORDS

*Latent Diffusion Model; Text-to-Image Generation; Variational Autoencoder; Human Face Synthesis; Deep Generative Models.*

## 1. INTRODUCTION

The generation of realistic human faces has gained importance as digital systems become more common in communication and media. Deep learning models now capture complex facial patterns, enabling the synthesis of images that appear natural and consistent with human features [1]. When text descriptions guide this process, the model must interpret both facial structure and semantic meaning. Using the annotated CelebA dataset, the system learns how facial attributes relate to one another, allowing it to produce faces that match user-specified descriptions [4].

The transformation of noisy latent representations into coherent images through iterative denoising has demonstrated the strong performance of diffusion-based models. This system follows a latent diffusion approach and incorporates a CLIP-inspired text encoder to interpret natural-language input, supporting both semantic alignment and visual realism [7,8]. As a result, the model can generate a wide range of synthetic faces shaped by descriptive cues.

In this system, to measure how well each output reflects the intended text, the CLIP Score is used as an indicator of semantic consistency, building on prior work demonstrating its reliability for text–image evaluation [7,5]. Together, these components highlight latent diffusion as an effective method for producing detailed, text-guided facial imagery.

## 2. RELATED WORK

The development of realistic human face generation has progressed through several major phases in computer vision, moving from early statistical representations to modern latent-space generative modeling. Early approaches relied on eigenfaces and handcrafted descriptors, which offered limited expressiveness and struggled to represent the high intra-class variability of real human faces (Turk & Pentland, 1991). A significant turning point was marked by the shift toward deep neural architectures, which enabled multi-level visual abstractions to be learned directly from data.

In particular, a breakthrough in image realism was delivered by Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), where sharp and coherent faces were synthesized by generators through adversarial training. However, GAN-based systems raised well-documented stability issues, including mode collapse, non-convergent dynamics, and poor semantic controllability (Arjovsky et al., 2017; Brock et al., 2019).

To overcome these limitations, Variational Autoencoders (VAEs) (Kingma & Welling, 2014) and hybrid latent-variable models introduced structured latent spaces that support interpolation and attribute manipulation, although the resulting outputs typically lacked the photorealistic fidelity of GANs. Meanwhile, contrastive language–image pretraining (CLIP) (Radford et al., 2021) provided a new mechanism for aligning text and image domains via large-scale supervision, enabling generative systems to condition outputs on natural-language prompts with far greater semantic consistency. Attribute-rich datasets such as CelebA (Liu et al., 2015) further enabled models to learn fine-grained facial statistics, supporting controllable editing across traits such as age, gender, lighting, and expression.

The introduction of diffusion models reshaped the generative modeling landscape by surpassing GANs in both output fidelity and training robustness. Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020; Sohl-Dickstein et al., 2015) formulate generation as an iterative denoising process that recovers structure from Gaussian noise, avoiding adversarial instability while capturing complex multimodal distributions. Latent Diffusion Models (LDMs) (Rombach et al., 2022) further improved scalability by performing diffusion within a compressed latent representation rather than pixel space, significantly reducing computational overhead while preserving detail.

Subsequent architectures such as Stable Diffusion demonstrated the effectiveness of combining latent diffusion with CLIP-based conditioning, enabling fine-grained text-guided control over synthesized faces. As a result, modern face-generation pipelines increasingly rely on latent diffusion frameworks that integrate semantic conditioning, efficient latent-space operations, and high-fidelity reconstruction, establishing a consistent trend toward models that balance realism, controllability, and computational efficiency.

## 3. METHODS AND MODEL

This system is applied on latent diffusion for text-guided human face generation requires a unified workflow that integrates dataset preparation, latent-space modeling, contrastive learning, and iterative denoising. The methodology summarized below outlines the steps needed to preprocess data, train each module, and evaluate the final text-to-image generation performance.

During training, the system periodically generates sample images and computes the CLIP Score on a held-out validation set, allowing visual quality and semantic alignment to be monitored across epochs and ensuring that the model converges toward producing coherent, text-consistent human faces [7].

### 3.1 Variational Autoencoder (VAE)

A VAE consists of two core components—an encoder and a decoder network. High-dimensional data is compressed into a compact latent representation by the encoder, parameterized by a mean (μ) and variance (σ) vector. The original data is reconstructed from this latent vector by the decoder, producing outputs that approximate the true data distribution [8,1].

An approximate posterior distribution $(z|x)$ is defined by the encoder, while a likelihood $p\theta(x|z)$ is defined by the decoder, with both being trained to maximize the likelihood of the observed data. The training objective of a VAE is derived from the Evidence Lower Bound (ELBO), which serves as an efficient approximation of the intractable data likelihood:

$$L(\theta, \varphi;\, x) \;=\; E\_\{q\varphi(z \mid x)\}[\, log\; p\theta(x \mid z)\,] \;-\; D_{\mathrm{kl}}(q\varphi(z \mid x) \,\|\, p(z))$$

To enable differentiable training, VAEs apply the reparameterization trick, which separates stochastic sampling from deterministic computation:

$$z \;=\; \mu\varphi(x) + \sigma\varphi(x) \odot \varepsilon, \qquad \varepsilon \sim N(0, I).$$

The overall loss function is typically expressed as:

$$L_{VAE} = L_{rec} + \beta \times D_{\mathrm{kl}}(q\varphi(z \mid x) \parallel p(z)).$$

### 3.2. Diffusion Models

A different perspective on generative modeling is provided by diffusion models, where a pair of stochastic processes is defined: a forward (diffusion) process, in which data is progressively corrupted with noise, and a reverse (denoising) process, in which this corruption is learned to be reversed.

The forward process incrementally destroys structure in the data distribution, while the learned reverse dynamics restore structure, enabling sampling from the data distribution starting from pure noise [9]. Denoising Diffusion Probabilistic Models (DDPMs) refine this idea by parameterizing the reverse conditionals with neural networks and training them using a variational objective closely related to denoising score matching [2].

#### 3.2.1 Forward Diffusion Process

In DDPM-style models, a Markov chain $\{x_t\}_{t=0}^{T}$ is used to define the forward process, in which Gaussian noise is increnetally injected into a clean data sample $x_0$. At each timestep $t$, the transition is expressed as:

$$q(x_t \mid x_{t-1}) = N\left(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t I\right),$$

where $\{x_t\}_{t=0}^{T}$ is a predefined noise schedule with small positive values [8,5]. This process admits a closed-form expression relating $x_t$ directly to the original data $x_0$:

$$q(x_t \mid x_0) = N\left(x_t; \sqrt{\bar{\alpha}_t}\, x_0, (1-\bar{\alpha}_t) I\right),$$

where $\bar{\alpha}_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. As $t$ increases, $x_t$ converges toward an isotropic Gaussian distribution, effectively removing semantic information from the image [9,2].

#### 3.2.2 Reverse Diffusion Process

The generative process samples in the reverse direction, starting from a Gaussian noise sample $x_T \sim N(0, I)$ and iteratively denoising it back to the data space. The reverse Markov chain is defined as:

$$p_\theta(x_{t-1} \mid x_t) = N(x_{t-1}; \mu_\theta(x_t, t), \sum_\theta (x_t, t)),$$

where $\mu_\theta$ and $\Sigma\theta$ are parameterized by a neural network with learnable parameters $\theta$ [8,2]. In practice, DDPMs simplify training by letting the network predict the noise $\epsilon$ added at each step and then deriving $\mu_\theta$ from that prediction. Sampling proceeds by repeatedly drawing $x_{t-1}$ from $p_\theta(x_{t-1}|x_t)$ until $x_0$ is obtained, which serves as the synthetic image. Later extensions such as Latent Diffusion Models (LDMs) perform this entire process in a compressed latent space to reduce memory and computation while maintaining high visual fidelity [2,8].

#### 3.2.3 Contrastive Language–Image Pre-training (CLIP)

To align natural language descriptions with visual representations, Radford et al. (2021) [5,3] introduced Contrastive Language–Image Pre-training (CLIP), a dual-encoder system that learns a shared multimodal embedding space for both text and images.

In CLIP, each image $I$ and its associated text description $T$ are encoded separately using dedicated encoders (typically a Vision Transformer for images and a Transformer-based model for text). Both encoders produce feature vectors f_I and f_T of equal dimension $d$, which are projected into a joint embedding space. During training, a contrastive objective is applied so that similarity is maximized for aligned pairs and minimized for unaligned pairs. The loss function is formulated as:

$$L_{CLIP} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{exp(sim(I_i, T_i)/T)}{\sum_j exp(sim((I_i, T_j)/T)} + log\frac{exp(sim(T_i, I_i)/T)}{\sum_j exp(sim((T_i, I_j)/T)}$$

### 3.3 U-NET

The U-Net architecture serves as the denoising backbone in most diffusion models due to its ability to integrate global semantic features with fine-grained spatial information. Structurally, U-Net consists of a downsampling encoder and a symmetrical upsampling decoder, connected through skip pathways that preserve spatial detail lost during resolution reduction [7]. Let $f_d^{(i)}$ denote the feature map at encoder level $i$, and $f_u^{(i)}$ the decoder feature map at the corresponding level. Skip connections concatenate the two:

$$f_u^{(i)} = UP(f_u^{(i+1)}) || f_d^{(i)},$$

where $Up(\cdot)$ is an upsampling operator and $\|$ denotes channel-wise concatenation. This formulation ensures that high-resolution details are retained during reconstruction, which is

essential in face-generation tasks. For diffusion models, the U-Net parameterizes the noise prediction function $\epsilon_\theta(x_t, t)$. To incorporate timestep information, a sinusoidal positional embedding is used:

$$Emb(t)_k = \begin{cases} \sin\left(\frac{t}{1000^{2k/d}}\right), if\ k\ is\ even, \\ \cos\left(\frac{t}{1000^{2k/d}}\right), if\ k\ is\ odd, \end{cases}$$

where $d$ is the embedding dimensionality [9]. By adding this embedding to the intermediate feature blocks, the denoising behavior is enabled to adapt to different levels of noise. In text-conditioned diffusion models, the U-Net is further augmented with cross-attention, allowing conditioning vectors (e.g., CLIP text embeddings) to influence the denoising trajectory. Given an image feature query set $Q$ and conditioning keys $K$ and values $V$, cross-attention computes:

$$Att(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V.$$

### 3.4 Loss Function

Generative models rely on different loss formulations depending on their underlying principles. VAEs optimize the evidence lower bound (ELBO), which consists of a reconstruction loss (often mean squared error or cross-entropy between input and reconstruction) plus a KL divergence regularizer that encourages the approximate posterior to remain close to a prior distribution over latent variables [3].

In contrast, GANs employ an adversarial loss in which the generator is trained to fool the discriminator, while the discriminator is trained to differentiate real samples from generated ones, typically through a cross-entropy or hinge loss [1].

Diffusion models employ a loss function that can be interpreted as a weighted variational bound but is often implemented in a simplified form as an $\ell_2$ loss between the true noise $\epsilon$ and the model's predicted noise $\epsilon_\theta(x_t, t)$:

$$L_{simple}(\theta) = E_{t,x_0,\epsilon}\left[\left|\left|\epsilon - \epsilon_\theta(x_t, t)\right|\right|^2\right],$$

with $x_t$ sampled according to the forward process [2]. This formulation has been shown to be sufficient for high-quality image synthesis and is widely adopted in practical systems, including latent diffusion models. Extensions may reweight timesteps, introduce perceptual or adversarial terms, or modify the prediction target (e.g., predicting velocity or $x_0$) to improve stability and sample quality [2,8].

## 4. Implementation and Experimental Setup

The implementation is built on PyTorch and HuggingFace Diffusers with Metal backend optimization, enabling efficient training on Apple M4 hardware. Experiments use the CelebA dataset (202,599 aligned face images with 40 annotated attributes), which provides rich semantic labels for text-guided generation.

The system consists of three components: a Variational Autoencoder (VAE) that compresses images into a low-dimensional latent space, a CLIP-style text encoder trained to align textual descriptions with latent visual features, and a Conditional Latent Diffusion Model (LDM) that learns to denoise latent representations into photorealistic human faces. Together, this pipeline enables semantic control over face synthesis while maintaining computational efficiency on local Apple-Silicon devices.

### 4.1 Image Pre-processing

All CelebA images are standardized prior to training to ensure consistency across the VAE, CLIP encoder, and diffusion model. Each image is first cropped and resized to 128×128 resolution, converted to floating-point tensors, and normalized to the $[-1, 1]$ range required by the VAE encoder. This preprocessing step produces compact latent tensors of approximately 16×16×4, reducing memory usage and enabling efficient training on Apple M4 hardware.

Text descriptions associated with each image are tokenized using a DistilBERT tokenizer, which performs lowercasing, punctuation removal, subword segmentation, and numerical ID mapping. These processed text sequences form the conditioning inputs for CLIP alignment and subsequent diffusion sampling. This unified preprocessing pipeline ensures consistent visual and textual representations across all stages of the model.

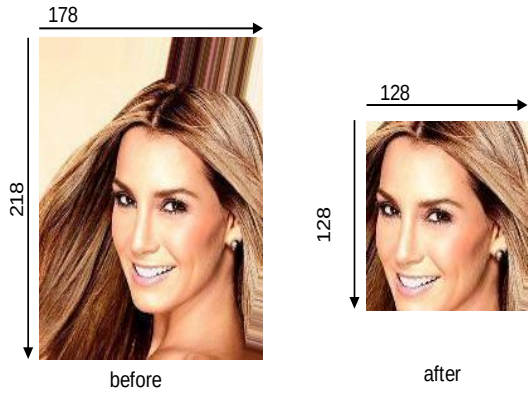



Figure 4.1. Example of CelebA image (000001.png) before preprocessing (left) and after preprocessing (right). Raw images are cropped, resized to 128×128, normalized to the [−1, 1] range, and converted to tensors suitable for latent-space encoding.

## 4.2 Vocabulary and Text Embedding Construction

Text associated with each CelebA image is transformed into a numerical representation so it can be used as a conditioning signal for generation. The preprocessing stage extracts all attribute labels and caption descriptions, standardizes them through tokenization, lowercasing, and punctuation removal, and records approximately 8,700 unique tokens across the training split. Each token is assigned an integer index, producing structured text sequences that are saved for efficient reuse during training.

These processed sequences are then encoded by a DistilBERT-based text model, which maps the input text into a 1280-dimensional embedding space. The resulting embeddings are trained to align closely with the VAE's latent image representations, enabling the generative model to interpret textual descriptions as meaningful semantic constraints during synthesis.

## 4.3 Model Training

Model training is conducted in three coordinated stages: VAE training, CLIP alignment, and conditional latent diffusion modeling.

| Raw Image | Raw CelebA Attribute Labels (Original) | Preprocessed Text Captions (Used for Train |
|---|---|---|
|  *(000001.jpg here)* | **Attributes = 1** (positive): Arched_Eyebrows, Attractive, Brown_Hair, Heavy_Makeup, High_Cheekbones, Mouth_Slightly_Open, No_Beard, Oval_Face, Pointy_Nose, Smiling, Straight_Hair, Wearing_Earrings, Wearing_Lipstick, Young | "good looking young woman"<br>"woman"<br>"young woman"<br>"woman smiling"<br>"attractive woman"<br>"young woman smiling"<br>"young woman brown hair"<br>"attractive young woman smiling" |
| | **Attributes = -1** (negative): 5_o_Clock_Shadow, Bags_Under_Eyes, Bald, Bangs, Big_Lips, Big_Nose, Black_Hair, Blurry, Bushy_Eyebrows, Chubby, Double_Chin, Eyeglasses, Goatee, Gray_Hair, Male, Mustache, Narrow_Eyes, Pale_Skin, Receding_Hairline, Rosy_Cheeks, Sideburns, Wavy_Hair, Wearing_Hat, Wearing_Necklace, Wearing_Necktie | |

Figure 4.2 Example of raw and preprocessed textual information for image 000001.jpg. CelebA attribute labels (middle) include both positive and negative binary annotations. Text cleaning and normalization convert these labels into natural-language captions (right), which are subsequently tokenized and embedded for conditioning the generative model.

### 4.3.1 VAE Training:

During 30 epochs of VAE training, the total loss decreased steadily from a maximum of 0.2100 to 0.1202, indicating smooth convergence of the reconstruction objective. The VGG perceptual loss remained relatively low, averaging 0.0943, which reflects strong preservation of high-level semantic structure in the reconstructed images. The MSE term also stabilized at a low average value of 0.0100, confirming accurate pixel-level reconstruction. The KL divergence converged to an average of 0.8697, with values ranging between 0.2282 and 0.9470, showing that the latent space maintained sufficient regularization without collapsing. These trends collectively demonstrate that the VAE successfully learned a compact and expressive latent representation suitable for downstream diffusion-based generation.

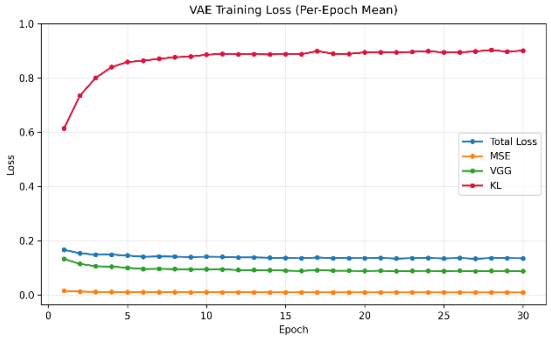



Figure 4.3 Training curves for the VAE across 30 epochs. The total reconstruction loss

decreases and stabilizes, MSE and VGG losses remain low, and KL divergence converges, confirming effective latent-space learning.

### 4.3.2 CLIP-Style Alignment Training:

The second stage of the system focuses on establishing a joint semantic space that aligns textual descriptions with the VAE's latent image representations. A DistilBERT encoder is fine-tuned together with an image projection module using a contrastive learning objective, where matching text–latent pairs are drawn closer in the embedding space while mismatched pairs are separated. Training is performed with a batch size of 64 for 10 epochs at a learning rate of $(1 \times 10^{-4})$. Based on the recorded logs, the contrastive loss begins around 4.16 in the early iterations of Epoch 1 and decreases consistently across training, reaching approximately 3.40 by Epoch 10. This downward trend indicates progressive strengthening of alignment between textual and visual features. Cosine similarity values follow an upward trajectory—from roughly 0.50 in early training to above 2.00 near the final epoch—demonstrating improved similarity between caption semantics and latent image attributes. These training dynamics confirm that the system's CLIP-style alignment module effectively learns a coherent shared embedding space that enables the diffusion model to generate images reflecting the intended textual descriptions.

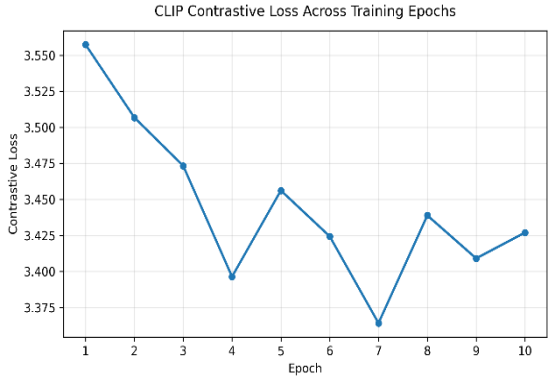



Figure 4.3.1 CLIP contrastive loss measured across 10 training epochs. The loss decreases during early training and stabilizes around 3.40 in later epochs, indicating effective alignment between textual embeddings and latent visual representations.

### 4.3.3 Conditional Latent Diffusion Modeling:

The final stage trains a conditional latent diffusion model that operates entirely within the VAE latent space. A clean latent code $z_0$ is progressively corrupted by Gaussian noise over $T = 400$ steps, producing a nearly noise-only latent $z_T$, as illustrated in Figure 4.3.1.

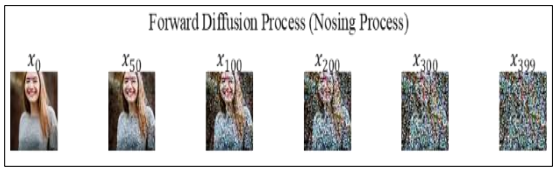



Figure 4.3.2 Forward Diffusion in Latent Space (Noising Process)

A U-Net denoiser conditioned on CLIP text embeddings learns the reverse process, iteratively predicting and removing noise so that a semantically meaningful latent can be reconstructed from $z_T$ (Figure 4.3.2).

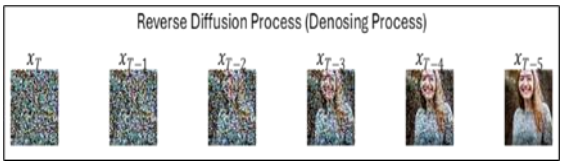



Figure 4.3.3 Reverse Diffusion in Latent Space (Denoising Process)

The model is trained on CelebA 128×128 faces using a batch size of 128 on Apple M4 hardware. During the initial eight epochs, using a learning rate of $1 \times 10^{-4}$, the diffusion loss decreases from approximately 0.29 to 0.19, indicating stable learning of global denoising behaviour. The model is then fine-tuned for the remaining epochs from the best checkpoint with reduced learning rates ($5 \times 10^{-5}$ and $1 \times 10^{-5}$), reaching a total of roughly 30 epochs. Throughout fine-tuning, the loss trends towards ($\sim 0.15$), and generated samples improve from blurry and distorted outputs to substantially sharper and more photorealistic faces. Together, the quantitative loss reduction and the qualitative forward/reverse diffusion visualisations demonstrate that the conditional latent diffusion model effectively learns to invert the noising process in a text-aware

manner while remaining computationally efficient on Apple Silicon hardware.

### 4.4 Model Evaluation

Model performance is evaluated using the CLIP Score, which measures the semantic consistency between each generated face and its conditioning text by comparing their embeddings in a shared vision–language space. For an image embedding $E_i$ and text embedding $E_t$, similarity is computed as

$$CLIPScore(E_i, E_t) = \frac{E_i . E_t}{||E_i||\ ||E_t||},$$

where higher values indicate stronger alignment. After 30 epochs of training, the model achieves an average CLIP Score of 0.354 on a held-out set of 2,000 CelebA samples, with the best batch reaching 0.381. These results indicate that the model is able to reliably map textual attribute descriptions—such as smiling, wearing glasses, hair color, or age group—onto the corresponding visual features in the synthesized images. In addition to quantitative evaluation, qualitative inspection confirms that the generated samples exhibit coherent facial structure, improved sharpness, and consistent attribute rendering, demonstrating that the conditional latent diffusion model effectively captures both semantic meaning and perceptual detail in the face synthesis task.

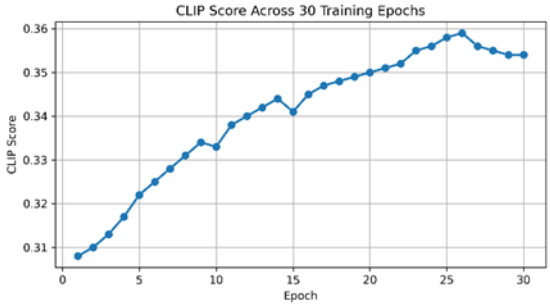



Figure 4.3.4 CLIP Score evaluated over 30 training epochs. The metric shows a gradual increase from 0.308 to 0.354, indicating improved semantic alignment between generated images and textual conditioning as training progresses.

## 5. Experimental Result

Figure 5 (a–f) illustrates the results produced by the conditional latent diffusion model for six representative text inputs, including “(a) young man smiling eyeglasses, (b) man wavy blond hair, sideburns, eyeglasses, smiling softly, (c) elderly man gray hair, small nose, high cheekbones, (d) young woman smiling brown hair, (e) woman straight hair, heavy makeup, pale skin, high cheekbones, wearing lipstick and earrings, and (f) old woman gray straight hair, arched eyebrows, wearing a necklace.The generated images demonstrate that the model is capable of synthesizing coherent facial structures while accurately reflecting multiple semantic attributes such as age group, hairstyle, facial features, makeup, accessories, and expressions. To quantitatively assess semantic alignment between the textual descriptions and the generated images, the CLIP Score was computed over the evaluation set.

In this sysetm, after 30 training epochs, the model achieved an average CLIP similarity of 0.354, with the highest batch reaching 0.381, indicating a meaningful level of correspondence between text inputs and visual outputs. The CLIP Score increases steadily during training before stabilizing, suggesting improved semantic consistency without performance degradation. The qualitative results shown in Figure 5.1 further confirm that the latent diffusion model produces perceptually coherent face images that closely align with their conditioning text, supporting the effectiveness of the proposed approach for text-guided face synthesis.

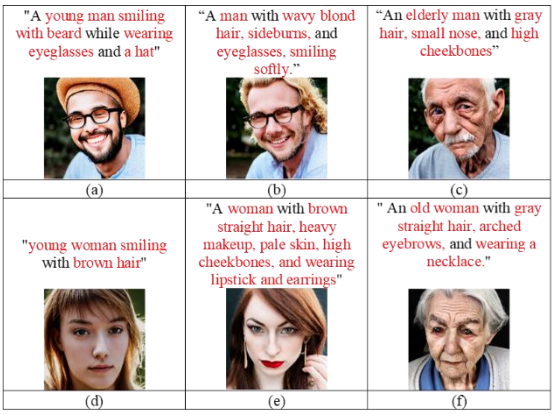



Figure 5.1 Example outputs generated from text inputs

## 6. Conclusion

This system is a conditional latent diffusion model was presented for text-guided face

synthesis, combining a VAE-based latent representation with CLIP-driven semantic conditioning. Training on the CelebA 128×128 dataset demonstrates that operating in latent space enables efficient optimization while maintaining coherent and attribute-consistent image generation. Quantitative evaluation using the CLIP Score yields an average similarity of 0.354 across 30 epochs, with a maximum batch score of 0.381, reflecting reliable alignment between textual descriptions and generated outputs. Qualitative samples further show accurate reproduction of key facial attributes, including smiling, hair characteristics, and eyeglasses. Remaining challenges include limitations in high-frequency detail and less common attributes. Future enhancements may focus on higher-resolution latent spaces, stronger conditioning mechanisms, and improved denoising architectures to increase realism and semantic precision.

## REFERENCES

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models from Natural Language Supervision," in Proc. Int. Conf. Mach. Learn. (ICML), 2021.

[2] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical Text-Conditional Image Generation with CLIP Latents," arXiv preprint arXiv:2204.06125, 2022.

[3] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in Proc. Int. Conf. Learn. Representations (ICLR), 2014.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2014.

[5] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2020.

[6] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep Unsupervised Learning using Nonequilibrium Thermodynamics," in Proc. Int. Conf. Mach. Learn. (ICML), 2015.

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in Proc. Int. Conf. Med. Image Comput. Comput. -Assist. Interv. (MICCAI), 2015, pp. 234–241.

[8] R. Robach, A. Blattmann, D. Lorenz, P. Esser, andB. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2022, pp. 10684–10695.

[9] Z. Liu,P. Luo, X. Wang, and X. Tang, "Deep Lerning Face Attributes in the Wild," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2015.

**Author's Biography**

Myat Kyaw Thu received the Bachelor of Computer Science (B.C.Sc.) degree in 2023 from the University of Computer Studies, Banmaw, Myanmar. He is currently pursuing his Master of Computer Science (M.C.Sc.) degree at the Mandalay university.

# NAME ENTITY RECOGNITION MODEL FOR RESUMES USING DEEP LEARNING

Su Hnin Aung[1], and Mya Thidar Kyaw[2]

[1,2] Faculty of Information Science, University of Computer Studies (Mandalay)
*suhninaung2000.sha@gmail.com, myathidarkyawmin@gmail.com*

## *ABSTRACT*

*The rapid growth of digital recruitment platforms has led to a substantial increase in the number of resumes submitted online, creating challenges for organizations in efficiently identifying suitable candidates. Manually reviewing unstructured resume text is labor-intensive and prone to inconsistencies, which points to the importance of automated and reliable information extraction techniques. This paper presents a transformer-based Named Entity Recognition (NER) system designed to extract essential information from resumes using a fine-tuned RoBERTa model. This system leverages RoBERTa's contextual embeddings to capture semantic and syntactic patterns within resume text, enabling accurate identification of key entities such as names, skills, education, designations, organizations, and experience. The Resume NER Training dataset is used for training and evaluation, and the experimental results indicate a precision of 89%, a recall of 91%, and an f1-score of 90%. The system demonstrates strong performance, as the RoBERTa-based NER model efficiently transforms unstructured resume documents into structured information, thereby enhancing the speed and reliability of the candidate screening process in recuriment systems.*

## *KEYWORDS*

*Resume Parsing, Name Entity Recognition, RoBERTa, Natural Language Processing, Deep Learning*

## 1. INTRODUCTION

The rapid increase in digital job applications has generated a critical need for automated systems that can effectively process and structure unstructured resume data, reducing manual effort and streamlining candidate selection. Resume parsing is particularly challenging due to the wide range of formats, inconsistent terminology, abbreviations, and the varied ways in which applicants present their details. Traditional methods, including rule-based approaches and basic machine learning techniques, often fail to handle these variations effectively, leading to incomplete or inaccurate extraction of important entities. This system addresses these issues by leveraging advanced natural language processing methods, specifically focusing on Named Entity Recognition (NER) with transformer-based models, to automatically detect and extract essential resume elements such as names, contact details, educational background, skills, and work experience. The main goal of this study is to build a reliable framework that can process resumes accurately in real-time, producing structured outputs suitable for direct use in recruitment management systems.

## 2. LITERATURE REVIEWS

Research on resume parsing has progressed from early rule-based methods to advanced deep learning approaches. Rule-based systems depend on fixed patterns and regular expressions to identify specific information, but they often fail when resumes vary in format or language use [9]. Machine learning techniques, such as Support Vector Machines, Hidden Markov Model (HMM), and Conditional Random Fields (CRF), improve adaptability by learning from annotated data; however, they typically require extensive feature design and struggle with capturing long-distance relationships within text [8]. More recently, transformer-based models like BERT and RoBERTa

have shown that they can do Named Entity Recognition (NER) better by using contextual embeddings that capture semantic connections, which makes entity extraction more accurate [7][10]. Despite these advancements, many existing systems are not equipped to handle the wide variety of resume formats or provide a fully integrated solution, highlighting the need for a robust framework that combines PDF text extraction, preprocessing, and RoBERTa-based NER to achieve comprehensive and reliable resume parsing [1][6].

## 3. BACKGROUND THEORY

Natural Language Processing (NLP) is a branch of artificial intelligence that enables machines to understand, interpret, and generate human language [3]. It aims to transform unstructured text into meaningful, structured information, facilitating applications like information retrieval, sentiment analysis, machine translation, and automated document parsing [2]. Early NLP methods primarily relied on rule-based techniques and statistical models, which depended on handcrafted features and specialized domain knowledge [3] [8]. Although these approaches performed well on structured or narrowly defined tasks, they often faced challenges when handling unstructured and diverse data such as resumes due to differences in writing style, formatting, and vocabulary [6].

### 3.1. Deep Learning

The emergence of deep learning transformed NLP by enabling models to automatically learn hierarchical and contextual representations from raw text [2]. Word embeddings, including Word2Vec and GloVe, encode semantic relationships between words into dense vector spaces, improving generalization across vocabulary [3]. Sequential models such as Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and Bi-directional LSTMs (BiLSTMs) were introduced to capture contextual dependencies in sequences [5] [8]. When combined with Conditional Random Fields (CRFs), BiLSTM-CRF architectures achieved strong performance on sequence labeling tasks, including Named Entity Recognition (NER), by capturing both token-level features and label dependencies without requiring extensive manual feature engineering [3]. Sequential models primarily rely on the current and past states. They may fail to fully incorporate bidirectional context unless specifically modified, which increases complexity.

### 3.2. Transformer Models

Transformer architectures marked a major advancement in NLP. Unlike sequential models, transformers rely on self-attention mechanisms to model relationships between all tokens in a sequence simultaneously, enabling efficient parallel processing and effective long-range context modeling [2]. State-of-the-art performance across a wide range of NLP tasks has been demonstrated by pretrained transformer models such as BERT and RoBERTa. BERT has been enhanced by RoBERTa through the removal of the next-sentence prediction task, the application of dynamic masking, and the use of larger datasets during pretraining [7] [11]. These improvements result in robust and context-rich embeddings that generalize effectively to domain-specific tasks like resume parsing [1] [10].

### 3.3. Name Entity Recognition

Entities such as names, skills, education, and work experience are identified and classified by Named Entity Recognition, a core NLP task [3]. In deep-learning-based NER, contextual token embeddings are generated by transformer models such as RoBERTa [11]. These embeddings are subsequently processed by a classification layer, where the BIO tagging scheme is employed to assign entity labels [1][3]. NER makes it possible to automatically extract structured information out of unstructured documents for resume parsing [4] [6]. Transformer-based models are particularly effective for this task, as they handle diverse layouts, fragmented text, and complex contextual relationships, providing accurate candidate information extraction [1] [10].

## 4. SYSTEM ARCHITECTURE

The overall workflow of the Resume Named Entity Recognition (NER) system is illustrated in Figure 1. The process begins with the Resume NER Training Dataset, which serves as the primary source of annotated resume samples for entity extraction. Before training, the dataset undergoes preprocessing, including cleaning overlapping spans, converting character-level annotations into BIO tags, tokenization using the BPE tokenizer, alignment of labels with tokenized outputs, and transforming tokens into input IDs.

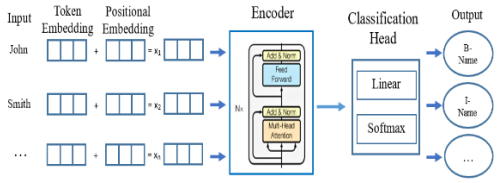



Figure 4.1: System Architecture

After preprocessing, the system splits the dataset into two subsets: the training set and the test set. The training set is used to optimize the parameters of the RoBERTa-based token classification model, while the test set is reserved for performance evaluation. The training phase involves feeding the preprocessed training data into the RoBERTa encoder, which converts tokens into contextualized representations, while the classification head predicts BIO labels for each token.

During this phase, the model learns to identify and classify resume entities based on the annotated dataset. At the end of training, a fine-tuned NER model is obtained. In the testing phase, the trained model is applied to the test set to predict BIO tags for unseen resume samples.

The predicted BIO tags are then processed through post-processing steps, including BIO entity merging and normalization, to generate complete and standardized entity spans. Model performance is evaluated using precision, recall, and F1-score, which measure the accuracy and completeness of the extracted entities compared to the reference annotations. Finally, the system produces structured resume information, marking the completion of the workflow.

### 4.1. Dataset

The dataset used in this system is the Resume NER Training Dataset, a large-scale corpus designed for Named Entity Recognition in resume parsing, consisting of 5,960 annotated resumes compiled from four sources: Kaggle ATS, HuggingFace NER, Resume Corpus, and Doccano Export. It includes 14 entity categories. The training dataset for resume parsing is composed of two key components. The first component contains the full resume text in plain-text format. The second component provides the corresponding entity annotations, structured as "annotation": {"label": ["entity_type"], "points": [{"start": start_index, "end": end_index, "text": "extracted_text"}]}.In this structure, "start" and "end" represent the character positions marking the beginning and end of the entity within the resume text, while "text" denotes the exact span of the annotated content.

## 5. METHODOLOGY

The methodology of this system combines advanced NLP techniques with transformer-based models to accurately extract entities from resumes. The approach begins with preprocessing raw text to handle inconsistent formatting and tokenize sentences for alignment with NER labels. Named Entity Recognition is performed using a RoBERTa encoder, which produces contextual embeddings for each token, capturing semantic relationships and dependencies across the resume text [6]. These embeddings are then passed through a fully connected layer with softmax activation to classify each token into predefined entity categories. The methodology emphasizes careful alignment of input tokens with BIO labels, fine-tuning of the transformer model on domain-specific resume data, and post-processing to generate structured outputs that preserve the originalinformation while providing a standardized format for downstream applications.

### 5.1. RoBERTa

In this system, a RoBERTa transformer encoder is employed, through which the input tokens are transformed into deep contextualized representations via multiple layers of self-attention and feed-forward transformations [2][11]. At each layer, the RoBERTa encoder processes the input sequence using multi-head self-attention followed by a position-wise feed-forward network [2]. The self-attention mechanism computes its internal operations as

$$\text{Attention (Q, K, V)} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \quad (1)$$

The attention mechanism takes three inputs:

Q = queries,

K = keys of dimension $d_k$, and

V = values (V) of dimension $d_v$

In multi-head attention, once all single-head attentions compute their resultant matrices, they will be concatenated into one big matrix.

$$\text{MultiHead (Q, K, V)} = \text{Concat (head}_1, \ldots, \text{head}_h) W_O \quad (2)$$

After the self-attention layer, the encoder applies a position-wise feed-forward network:

$$\text{FFN (z)} = \text{GELU (z}W_1 + b_1) W_2 + b_2 \quad (3)$$

Layer normalization and residual connections are applied after both the attention and feed-forward sublayers.

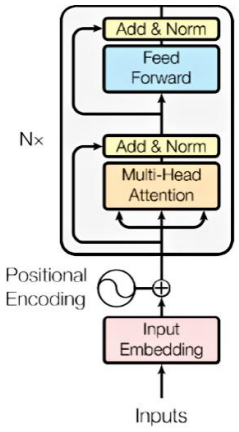



Figure 5.1: RoBERTa Encoder Architecture

Here, X is the embedded input token sequence, $W_Q$, $W_K$, $W_V$, $W_1$, $W_2$, $W_O$ are learnable weight matrices; $b_1$ and $b_2$ are bias terms; and GELU denotes the Gaussian Error Linear Unit activation function used in RoBERTa [2][11]. This self-attention mechanism enables the encoder to capture global dependencies across all tokens, making it highly effective for entity extraction in resume parsing tasks.

### 5.2. Classification Head

The classification head in the RoBERTa-based NER system maps token-level hidden representations to entity predictions. Once the input text has been encoded, a hidden state is produced by RoBERTa for each token. A linear (fully connected) layer then projects each hidden state $h_i$ into a set of logits representing all possible entity classes:

$$Z = Wh_i + b \quad (4)$$

where W and b denote the trainable parameters of the classification layer, and Z is the logit vector for token i. Each dimension of this vector corresponds to a specific NER category such as SKILL, PERSON, or EDUCATION. The logits are subsequently passed through a softmax function to obtain a probability distribution over the entity classes:

$$\text{Softmax } (x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \quad (5)$$

Where $X_i$ = the logit for the $i^{th}$ class, K = the number of classes, and $e^{x_i}$ = the exponential of the logit.

During inference, the class with the highest probability is selected as the predicted entity label for each token, enabling accurate token-level classification across all defined resume entity categories [1][10].

## 6. IMPLEMENTATION

For this system, establishing a comprehensive understanding of the underlying technologies is crucial. This involves transformer-based deep learning models, natural language processing (NLP), and the implementation of Python-based

frameworks such as Hugging Face Transformers and PyTorch within Google Colab. Knowledge of NumPy and Pandas for numerical operations and text preprocessing is equally important. All required libraries must be properly configured to ensure smooth execution of the system.

## 6.1. Data Pre-processing

Before training the resume parsing system, raw resume text needs to be cleaned and structured to ensure it is consistent and suitable for named entity recognition (NER). Resumes are often unstructured, with irregular formatting and overlapping entity annotations that can reduce extraction accuracy. To address these issues, preprocessing steps are applied to prepare the data for the model.In this system, preprocessing includes five steps:

- **Cleaning overlapping spans:** Duplicate or nested entity annotations are removed to avoid conflicts during training.
- **Convert offsets to BIO Tags:** Character-level entity offsets are converted into BIO tags. The BIO tagging scheme indicates to the model where entities begin, continue, and end.
- **Tokenization:** Text is split into tokens or subword units to match the transformer's vocabulary.
- **Label alignment:** When words are split into multiple subwords, the original BIO label is applied to all tokens.
- **Convert tokens to input IDs:** Numerical representations of tokens are generated to enable processing by the model.

By performing these preprocessing steps for training, the resume text is systematically cleaned, tokenized, and aligned with entity labels, enabling the NER model to extract structured information accurately and efficiently.

## 6.2. Creating Word Embeddings

After preprocessing, the tokenized resume text is converted into embeddings to enable the RoBERTa-based NER model to process it effectively. Dense numerical vector representations, known as embeddings, are used to encode the semantic information of each token within a high-dimensional space. By transforming textual tokens into embeddings, the model can interpret and learn from the data in a way that preserves both meaning and context. This embedding process is a crucial step in the system, as it allows raw tokens to be represented in a format suitable for deep learning models, bridging the gap between unstructured text and machine-readable input.

In the system, two types of embeddings are employed: token embeddings and positional embeddings.

### 6.2.1. Token Embeddings

Token embeddings provide a vector representation for each individual word or subword token. These vectors capture the semantic meaning of tokens based on patterns learned during pre-training of the RoBERTa model. For instance, the tokenized sequence ["John", "Doe", "Software", "Engineer", "Google"] is mapped to token IDs like {"John": 1012, "Doe": 4578, "Software": 1023, "Engineer": 200, "Google": 7689}, and each ID is converted into a corresponding vector in the embedding space. These embeddings allow the model to recognize semantic relationships, so that contextually related words such as "John" and "Doe" are closer in vector space than unrelated tokens like "John" and "Excel."

### 6.2.2. Positional Embeddings

Positional embeddings, on the other hand, capture the position of each token within the sequence. Unlike sequential models, transformer-based models like RoBERTa do not process tokens in order inherently. By adding positional information to the token embeddings, the model can understand the sequence of words and capture the relative or absolute position of each token. For

example, in the sentence “John Doe works at Google,” the token “John” at position 0 and “Google” at position 4 receive distinct positional vectors. The sum of token embeddings and positional embeddings forms the final input representation, providing both semantic meaning and sequential context to the model.After combining token and positional embeddings, they are passed through the transformer layers, where attention mechanisms are applied to enhance the representations by modeling dependencies among tokens across the sequence. This enables the system to accurately identify entities.

## 6.3. Training the Model

The resume parsing system is built on a RoBERTa-based named entity recognition (NER) model to identify structured information, such as skills, education, experience, and personal details, from unstructured resume text. Transformer models like RoBERTa are highly effective for this task because they capture complex contextual relationships and long-range dependencies in text, which are essential for extracting precise entity boundaries and semantic roles. The Hugging Face Transformers framework is used to build the model, which has a TokenClassification architecture. The layers process text from token embeddings to the final entity predictions.Training was configured with carefully selected hyperparameters to ensure stable learning and prevent overfitting. A batch size of 4 was used due to the high memory requirements of transformer models, and training was conducted for 20 epochs, allowing the entire dataset to be processed multiple times for optimal convergence. A learning rate scheduler was applied to automatically reduce the learning rate when validation performance plateaued, with warmup steps included to stabilize early training. Evaluation during training monitored validation loss and F1-score, enabling early stopping if no improvement was observed across consecutive epochs.The architecture begins with the pre-trained RoBERTa model, which provides deep contextual embeddings that encode semantic and syntactic information from large-scale unsupervised pretraining. These embeddings are fine-tuned rather than frozen, allowing the model to adapt to the domain-specific patterns of resume text. On top of RoBERTa, a token classification head is added, consisting of a dropout layer and a linear classifier that maps hidden states to BIO-tagged entity labels. This design enables the model to handle complex boundary detection and overlapping resume entities effectively. The system is trained using the AdamW optimizer, which provides adaptive weight decay regularization suitable for transformer models, while the loss function is token-level cross-entropy, and performance is evaluated using accuracy, precision, recall, and F1-score.

The system accepts resume data in the form of PDF documents and plain text as input. The system produces structured resume information such as personal details, skills, and education by applying a RoBERTa-based NER model. The final output is presented in a structured format to support automated recruitment and human resource management applications. The UI design of the system is shown as in figure 3.

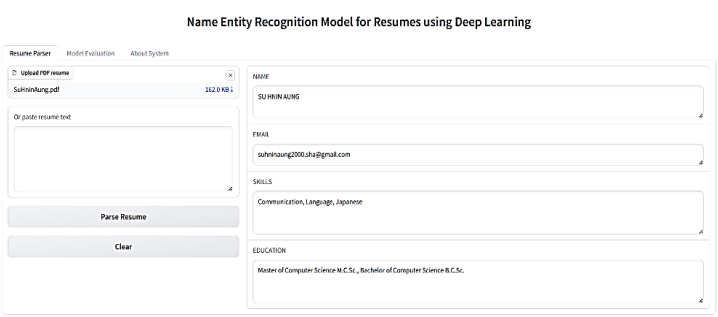



Figure 3: UI design for NER Resume Parsing System

## 7. EXPERIMENTAL RESULT

To evaluate the performance of the Resume NER system using a fine-tuned RoBERTa model, several metrics and visualization techniques were employed. The model was validated on a separate set of unseen resumes with diverse structures and writing styles to assess its generalization capability.

The system achieved a precision of 89%, a recall of 91%, and an F1-score of 90%, demonstrating consistent and reliable entity extraction across all defined categories.

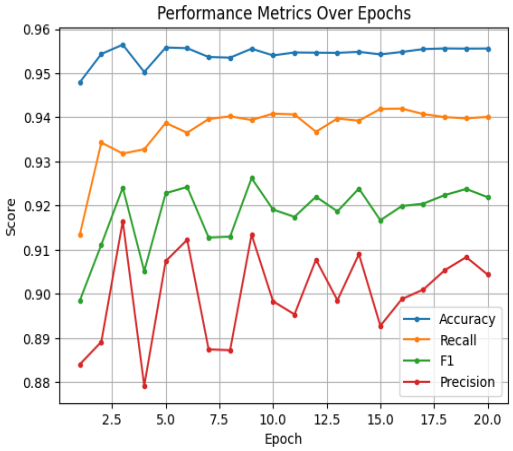



Figure 7.1 Evaluation Matrix

Figure 7.1 presents the evaluation metric trends observed during fine-tuning of the RoBERTa model, including precision, recall, F1-score, and accuracy. Across the 20 training epochs, all four metrics demonstrated consistent improvement, reflecting the model's growing ability to correctly identify resume entities.

Precision and recall steadily increased as the model learned more accurate token-level boundaries, while the F1-score exhibited a balanced rise, confirming strong performance across both major and minor entity classes.

Overall accuracy also improved throughout training, indicating that the model effectively generalized to unseen resume samples.

The smooth upward progression of these metrics, without sharp fluctuations, further suggests stable learning and the absence of overfitting.

Table 7.1: Accuracy, Precision, Recall, and F1-Score for NER system

| epoch | eval_accuracy | eval_precision | eval_recall | eval_F1 |
|---|---|---|---|---|
| 1 | 0.94 | 0.77 | 0.82 | 0.79 |
| 2 | 0.95 | 0.82 | 0.86 | 0.84 |
| 3 | 0.96 | 0.86 | 0.87 | 0.87 |
| 4 | 0.96 | 0.88 | 0.88 | 0.88 |
| 5 | 0.96 | 0.87 | 0.90 | 0.88 |
| 6 | 0.96 | 0.88 | 0.90 | 0.89 |
| 7 | 0.96 | 0.89 | 0.90 | 0.89 |
| 8 | 0.96 | 0.88 | 0.91 | 0.89 |
| 9 | 0.96 | 0.88 | 0.91 | 0.89 |
| 10 | 0.96 | 0.89 | 0.91 | 0.90 |
| 11 | 0.96 | 0.90 | 0.91 | 0.90 |
| 12 | 0.96 | 0.89 | 0.91 | 0.90 |
| 13 | 0.96 | 0.90 | 0.91 | 0.90 |
| 14 | 0.96 | 0.89 | 0.91 | 0.90 |
| 15 | 0.96 | 0.89 | 0.91 | 0.90 |
| 16 | 0.96 | 0.90 | 0.91 | 0.90 |
| 17 | 0.96 | 0.89 | 0.91 | 0.90 |
| 18 | 0.96 | 0.89 | 0.91 | 0.90 |
| 19 | 0.96 | 0.89 | 0.91 | 0.90 |
| 20 | 0.96 | 0.89 | 0.91 | 0.90 |

As shown in Figure 7.2, the loss curves depict the model's convergence throughout the training process. Training loss decreased from 0.15 at the first epoch to 0.02 by the twentieth epoch, while validation loss dropped from 0.19 to 0.18, remaining consistently lower than training loss. Regularization techniques such as dropout and careful learning rate scheduling contributed to stable convergence and effective error minimization.

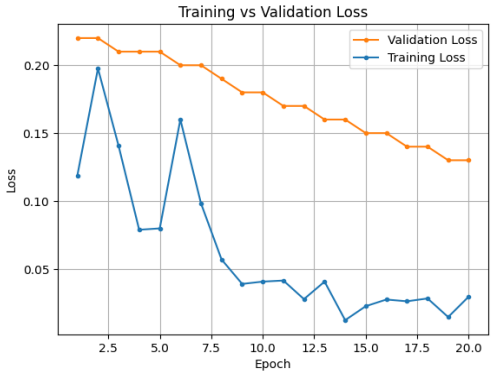



Figure 7.2: Training and Validation Loss

Overall, the experimental results confirm that the RoBERTa-based NER system provides highly reliable extraction of structured information from resumes. Token-level accuracy, precision, recall, and F1-score indicate robust performance across both frequent and rare entities. The

integration of pre-trained contextual embeddings with fine-tuned transformer architecture enables effective recognition of complex entities, capturing semantic and syntactic patterns in resume text. While the model performs strongly on structured and semi-structured resumes, challenges remain in handling extremely unstructured documents, rare entity types, and resumes in multiple languages. Future work could explore multilingual RoBERTa models, domain-specific pretraining on resumes, and integration with downstream candidate recommendation systems to further enhance parsing accuracy and applicability.

## 8. Conclusion

This system presents a comprehensive framework for automated resume parsing using Named Entity Recognition with a RoBERTa-based encoder. The system effectively addresses challenges such as heterogeneous resume formats, ambiguous terminology, and diverse entity representations, achieving high accuracy and robustness on real-world resumes. This framework integrates PDF text extraction, preprocessing, RoBERTa embeddings, and token-level classification to produce structured outputs suitable for use in recruitment applications. This study demonstrates the effectiveness of transformer-based models for domain-specific NER tasks and sets a foundation for future improvements, including multilingual support, expanded custom entity types, real-time processing, and integration with end-to-end applicant tracking systems.

## References

[1] A. A. Abdullah *et al.*, "NER-RoBERTa: Fine-tuning RoBERTa for Named Entity Recognition (NER) within low-resource languages," 2023.

[2] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[3] B. Jehangir, S. Radhakrishnan, and R. Agarwal, "A survey on named entity recognition—Datasets, tools, and methodologies," *Natural Language Processing Journal*, vol. 3, pp. 1–12, 2023.

[4] D. S. Chavare and A. B. Patil, "Resume parsing using natural language processing," *Grenze Int. J. Eng. Technol.*, Jan. 2023.

[5] H. Cho, Lee, H. "Biomedical named entity recognition using deep neural networks with contextual information," *BMC Bioinformatics*, vol. 20, p. 735, 2019

[6] I. J. A. Biol and C. C. Abalorio, "Utilizing named entity recognition for web-based resume scoring," *Int. J. Eng. Trends Technol. (IJETT)*, vol. 72, no. 7, pp. 381–387, 2024.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.

[8] N. Patil, A. Patil, and B. V. Pawar, "Named entity recognition using conditional random fields," in *Int. Conf. Computational Intelligence and Data Science (ICCIDS)*, 2019.

[9] P. Sroison and J. H. Chan, "Resume parser with natural language processing," King Mongkut's Univ. of Technol. Thonburi, Thailand.

[10] R. Rajesh, R. S. S. Harsha, B. Joseph, V. L. Duggineni, and J. R. Alapati, "Resume parsing using named entity recognition and Hugging Face model," *Int. J. Innov. Res. Sci. Eng. Technol. (IJIRSET)*, vol. 13, no. 3, Mar. 2024.

[11] Y. Liu *et al.*, "RoBERTa: A robustly optimized BERT pretraining approach," *arXiv preprint* arXiv:1907.11692, 2019.

**Author's Biography**

Su Hnin Aung completed the Bachelor of Computer Science (B.C.Sc) in 2023 at the University of Computer Studies, Meiktila. She is now pursuing a Master's degree at the University of Computer Studies, Mandalay.

# FACE EXPRESSION RECOGNITION USING ATTENTIONAL CONVOLUTIONAL NETWORK

Su Hlaing Tun[1], and Saw Thandar Myint[2]

[1,2]Department of Information Technology Supporting and Maintenance, University of Computer Studies (Mandalay)
suhlaing258654@gmail.com, sawthandar26@gmail.com

## ABSTRACT

*Facial expression recognition (FER) is one of the great significances in fields such as human-computer interaction, intelligent monitoring and emotional computing. One key challenge in expression recognition is to extract the discriminative features which focus on expressions of a face. Various deep learning methodologies, particularly the widespread use of convolutional neural networks (CNNs), have been affected for FER. This system develops a deep learning model using on attentional convolutional network to extract the relevant features for FER. Publicly available Cohn Kanade (CK+) dataset is used for experiments. The input of facial expression recognition system is an image and the output is one of seven expression classes describing the emotion of the face. Experimental results demonstrated that the presented model achieved higher classification accuracy (90%) on CK+ dataset. The results of the experiment indicated that the CNN model performs well in terms of precision, recall, and F1-score on CK+ dataset.*

## KEYWORDS

*Facial expression recognition, Convolutional neural network, Attention mechanism, Classification accuracy*

## 1. INTRODUCTION

Facial expression recognition (FER) is an interesting topic for extracting facial features which are discriminative of small changes in expressions. In recent, expression recognition systems have been studied many feature extraction techniques and classifiers.

FER with deep learning technologies has become an interesting topic in fields such as emotional computing, intelligent monitoring, medical diagnosis, and human-computer interaction. The DL techniques need a large number of methods and architectures and require appropriate parameter tuning and optimization. However, no perfect DL model which identifies FEs accurately is available as of now.

Facial expression recognition (FER) in real-world conditions remains a challenging task due to unconstrained facial variations and complex environments. One of the key challenges lies in extracting discriminative features from salient facial regions. Motivated by this limitation, this work proposes a deep learning–based CNN attention framework to effectively capture important facial features and address major shortcomings of existing FER systems. The proposed system aims to classify facial expressions into seven basic emotions: angry, disgust, fear, neutral, happy, sad, and surprise, and its effectiveness is validated through experimental evaluation on facial image datasets.

In the following sections, Section II provides an overview of related works in FER. The proposed framework and model architecture are explained in Section III. Then Section IV discusses the experimental results, overview of databases used in this work, and also model visualization. Finally, the conclusion of the paper is stated in Section V.

## 2. LITERATURE REVIEWS

A typical FER consists of three steps, namely face detection, feature extraction, and classification of FEs. In simple terms, the face is detected and cropped from a scene or a video frame in the first step. In the second step, features are extracted from the cropped face region and concatenated to form a feature vector. Then the feature vector is fed into a machine learning algorithm to identify FE. Some of most known adopted DL techniques for FER are described in the followings.

The authors in [1] focused on extracting useful features from gray-scale face images. Convolution neural network (CNN) model is used for feature extraction, to distinguish facial expressions (FEs) efficiently with the help of the Softmax classifier. The performance is tested on five benchmarking datasets, namely FER2013, Japanese Female Facial Expressions, Extended CohnKanade (CK+), Karolinska Directed Emotional Faces, and Real-world Affective Faces. Seven FEs, namely neutral, anger, disgust, fear, happiness, sadness, and surprise, are considered in this paper. The average accuracies on these datasets are 78.9%, 96.7%, 97.8%, 82.5% and 81.68%, respectively.

Recent studies have demonstrated that incorporating attention mechanisms into convolutional neural networks can significantly enhance facial expression recognition performance. In this context, DeepEmotion [2] introduces an attention-based CNN architecture that automatically focuses on emotion-relevant facial regions, leading to improved recognition accuracy across multiple benchmark datasets, including FER-2013, CK+, FERG, and JAFFE.

According to [3] a two-stage Deeply-Supervised Attention Network (DSAN) for facial expression recognition. The method uses attention mechanisms and multi-scale deep supervision to capture important facial features. Experimental results on RAF-DB, CK+, Oulu-CASIA, and AFEW show improved performance and robustness across different gender, age, and race variations.

Although existing emotion recognition methods show notable performance improvements, many lacks a simple and effective mechanism to attend to critical facial regions. To address this gap, this work proposes a lightweight attention-based CNN that focuses on salient facial areas and achieves robust performance even on small-scale datasets. The overall framework of the proposed FER system is described in the following sections.

## 3. MODEL ARCHITECTURE

The overall procedure of the facial expression recognition system is illustrated in Figure 1. The process begins with the input image. Before training, the dataset undergoes preprocessing, including resizing, detection and augmentation.

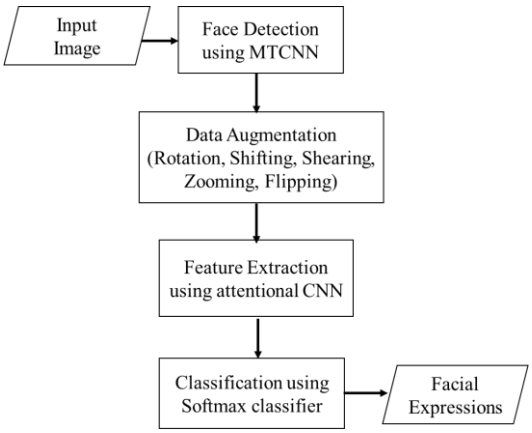



Figure 1. Framework for FER model

After preprocessing, the dataset is split into three subsets: the training set, validation set and the test set. The training set is used to optimize the parameters of the CNN model with attention, while the test set is reserved for performance evaluation. The convolutional layers produce feature maps, which denote high-level features such as edges, corner points, and color from the face region automatically. The training phase involves feeding the training data into a FER model with an attention mechanism. The input image is detected using MTCNN pre-train model, and

generates detected face. After that, the feature extraction part consists of seven convolutional layers, each two except the last one followed by max-pooling layer and rectified linear unit (ReLU) activation function. Then followed by a dropout layer which leads to the interpolation point. In the attention mechanism, the regression of transformation parameters takes place, then the input is transformed to the sampling grid to produce the warped data. At the end of training, a trained model is obtained.

In the testing phase, the trained model is applied to the test set to generate classified facial emotions. Model performance is then evaluated using the accuracy, precision, recall and F1-score. Finally, the system produces the one of seven expression classes (angry, disgust, fear, happy, neutral, sad, and surprise) describing the emotion of the face.

## 4. METHODOLOGY

This section describes the methods and materials used in these experiments.

### 4.1. Multitask Cascade Convolutional Neural Network (MTCNN)

Face detection is a critical technology in computer vision and has numerous applications. It often includes preprocessing steps such as face alignment and normalization. It locates and crops the face region from the input image. By accurately detection and cropping the face, the system can extract more effectively the facial features. If the face is not correctly detected, the features extracted may be incorrect or incomplete, leading to poor classification performance. Deep learning-based face detection techniques are Multitask Cascade Convolutional Neural Network (MTCNN) and Faster R-CNN. In the study, MTCNN pre-trained model is used to detect and align the face from the input images.

MTCNN uses a cascaded structure with three convolutional neural networks to refine the detection process. P-Net (Proposal Network) identifies potential face regions. R-Net (Refine Network) filters candidate regions and performs bounding box regression. O-Net (Output network) identifies face regions and five landmark points of left eye, right eye, and nose, left corner of mouth and right corner of mouth. It gives the pre-trained weights and biases for all layers of the three networks. It is more accurate for real-world facial images than other face detection methods. Figure 2 displays the step by step diagram of the MTCNN face detector.

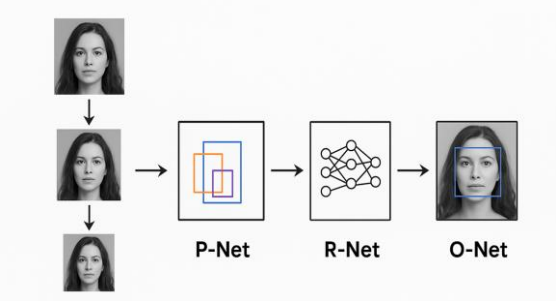



Figure 2. Step-by-step diagram of the MTCNN face detector

### 4.2. Data Augmentations

Data augmentation is a technique used to increase the diversity of data available for training models. The augmentation techniques prevent overfitting and enhances the performance on diverse data. The different augmentation methods for facial expression recognition. This system used the rotation (30 degree), shifting (0.2), shearing (0.2), zooming (0.2), flipping (Boolean), and brightness (range [0.5, 1.5]) augmentations. The system saved 8 images (1 original + 7 augment) for model training. These transformations change the positions of objects and angles. It is implemented by using Keras ImageDataGenerator package.

### 4.3. Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are deep learning models specifically designed for image-based tasks and are widely used in facial expression recognition. CNNs utilize convolutional layers with local receptive fields to automatically learn hierarchical and spatially local features from facial images. In this work, CNN is employed to extract

and classify discriminative facial expression features. The convolutional layer applies multiple filters to the input image to generate feature maps through element-wise convolution operations, enabling the extraction of essential local patterns such as edges and textures. The mathematical operation is defined as:

$$Feature\ map = X \times W + b \quad (1)$$

where, × is convolution, $X$ = input image, $W$ = kernel, $b$ = bias. The activation function is generally applied to the Ratified Linear Unit ReLU. The ReLU layer will apply an element wise activation function. The activation function can be defined as:

$$f(x) = max(0, x) \quad (2)$$

Pooling layer will perform a downsampling operation along the spatial dimensions. After several convolution and pooling layers, features are flattened into a vector passed through dense layers for classification. Three different types of pooling operations are mainly available such as max-pooling, min-pooling, and average-pooling. The maximum pooling is used for the FER system. Fully Connected (FC) layer will combine extracted features into global decisions. The output layer usually uses Softmax activation (multi-class) or Sigmoid (binary) to classify one of seven expression classes. The softmax function is shown in following equation:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad (3)$$

where, σ is soft-max function, $\vec{z}$ input vector, $e^{z_i}$ is standard exponential function and $K$ is the number of classes. A loss function is then optimized using Adam optimizer to train this model. In this study, the classification loss (cross-entropy) and regularization term, which is the l2 normalization of the weights in the last four fully connected layers, are simply added to form the loss function. The test results for CK+ dataset improved as a result of the proposed model.

## 5. Experimental Analysis

Dataset is divided into three parts mainly for training sets, validation sets, and testing sets separately. The ratio of dividing datasets for training, validation, and testing is 70%, 10%, and 20%, respectively. Then FER system is trained using the train set; moreover, while training the model too many epochs may lead to overfitting, too few epochs may lead to underfitting the model. The model is trained for 20, 30 and 50 epochs, with training and validation loss monitored at each epoch.

### 5.1. Dataset

The Cohn Kanade (CK+) dataset is widely used in major facial expression classification methods. It comprises both posed and unposed (spontaneous) expressions. It contains 123 different subjects ranging in age from 18 to 50 years old, with different genders and a total of 981 sequences. Figure (3) displays six representative pictures from this collection. The dataset contains seven basic expressions, namely neutral (NE), anger (AN), disgust (DI), fear (FE), happiness (HA), sadness (SA), and surprise (SU).

### 5.2. Parameter Settings

The parameters during training, which include the learning starting at a rate of 0.0005, a reduced learning threshold of 0.0001, a batch size set to 16, an Adam optimizer, and different iterations. All the images of datasets are resized from 256×256 to 224×224 pixels in the preprocessing step. PyTorch framework, TenserFlow and Python language are considered for the implementation of the proposed method. The specifications of the system are 16GB GPU RAM, 2560 Cuda cores, 256-bit memory interface, GDDR5X as memory type, 288.5 GB/s band width, NVIDIA Quadro P5000 as the graphic processor, and python 3.9.5 is used.


Figure 3. Sample images of the CK+ dataset

### 5.3. Evaluation Measures

The performance of the proposed system has been evaluated by four metrics, accuracy, precision, recall and F1-scores derived from confusion matrix shown in Figure (4).

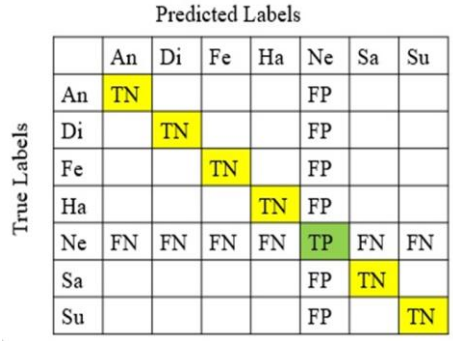



Figure 4. Confusion Matrix

These metrics are calculated by these following equations.

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

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

$$F1\,Score = 2 * \frac{Precision*Recall}{Precision*Recall} \quad (7)$$

The model is trained with a batch size of 16, a learning rate of 0.001 using the Adam optimizer, and a dropout rate of 0.5 to mitigate overfitting.

### 5.4. Results and Discussion

The experimental analysis demonstrated that the FER model with attention effectively addressed a wide range of face region. The model performed particularly well in cases requiring long-distance dependencies, where attention helped the decoder focus on relevant regions regardless of their position in the input image.

CK+ dataset has more samples and a more exact section of the facial image, which is ideal for training the model. As Table 1 results, an average validation accuracy of 96.43% is obtained on Epoch 50. Any additional augmentation techniques did not test because this was producing nearly flawless results on the main dataset. However, the CK+ dataset will yield the best results for this model.

Table 1. Comparison of Validation Accuracy on CK+ with different Epochs

| No. of Epochs | Average Validation Accuracy |
|---|---|
| 20 | 93.83 % |
| 30 | 95.82 % |
| 50 | 96.43 % |

According to the Figure 5, 6 and 7, the confusion matrices illustrate the performance of the proposed model on the CK+ dataset, achieving a high overall test accuracy of 96.88%, 100% and 97.92% on Epochs 20, 30 and 50, respectively.

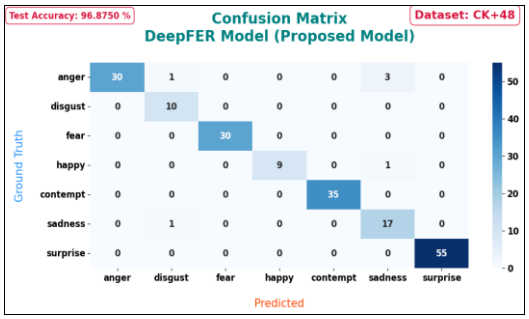



Figure 5. Confusion Matrix of Presented Model on Epoch 20

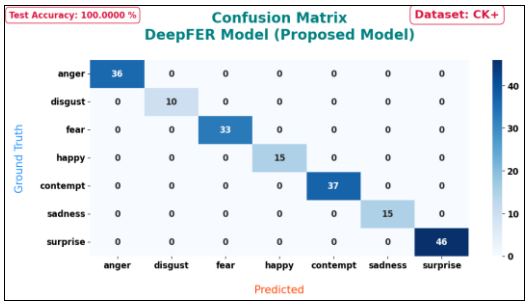



Figure 6. Confusion Matrix of Presented Model on Epoch 30

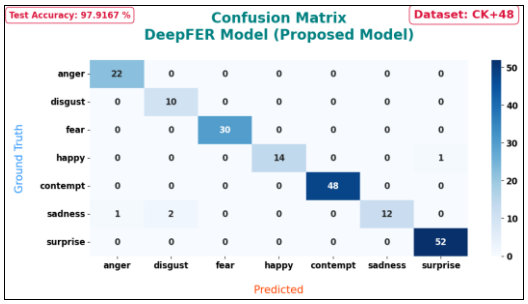



Figure 7. Confusion Matrix of Presented

Most emotion classes, such as anger, fear, contempt, and surprise, show perfect or near-perfect classification, indicating strong model capability in recognizing distinct facial expressions. A few minor misclassifications occur, for example, one instance of anger is predicted as disgust and one sadness image is incorrectly classified as disgust in Figure 5 confusion matrix. Despite these small errors, the model demonstrated excellent precision and generalization across all emotion categories.

According to the Figures 8, 9 and 10, the training and validation curves demonstrated that the model achieves excellent learning performance. In the accuracy plot, both training and validation accuracies rapidly increase during the initial epochs and stabilize near 100% after approximately 30 epochs, indicating that the model successfully captures the key patterns in the data. Similarly, the loss plot shows a steep decline in both training and validation losses, converging close to zero with minimal fluctuations. The close alignment between the training and validation curves in both plots suggests strong generalization ability and no significant overfitting. Overall, the model demonstrates efficient convergence, high accuracy, and stable performance across both training and validation phases. Overall, the presented model achieves robust and reliable performance in facial emotion recognition on CK+ dataset in facial expression recognition performance. The next section displays the implementation of the FER system.

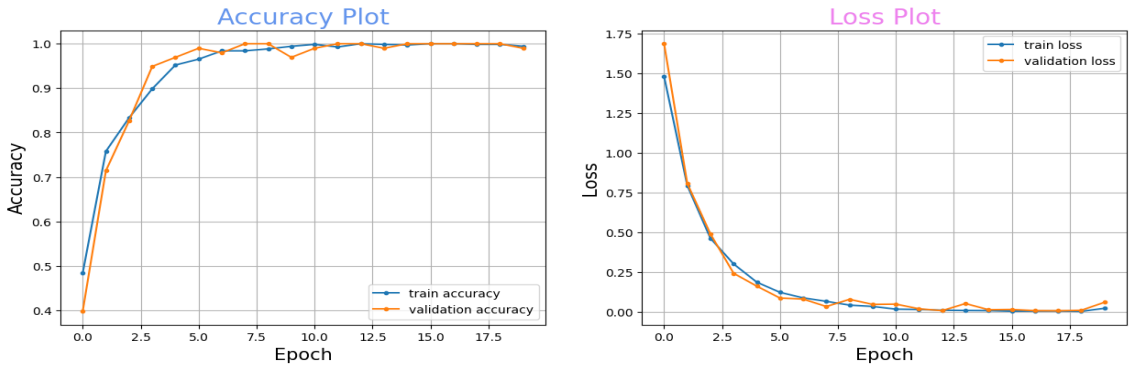



Figure 8. Accuracy and Loss Plot Results of Presented Model on Epoch 20

## 6. IMPLEMENTATION

The FER model is implemented in Python using the PyTorch deep learning framework. To examine the effect of training period on model performance, three independent experiments were carried out. Throughout training, a batch size set to 16 was continuously maintained, and a learning rate of 0.0001 was deliberately chosen to reduce the possibility of overfitting. The testing phase was performed on various images to comprehensively evaluate the model's detection accuracy and generalization capability.

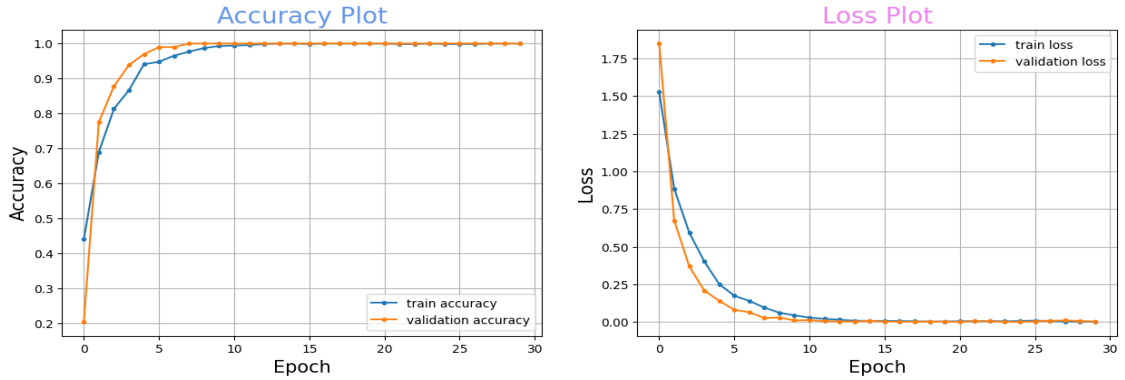



Figure 9. Accuracy and Loss Plot Results of Presented Model on Epoch 30

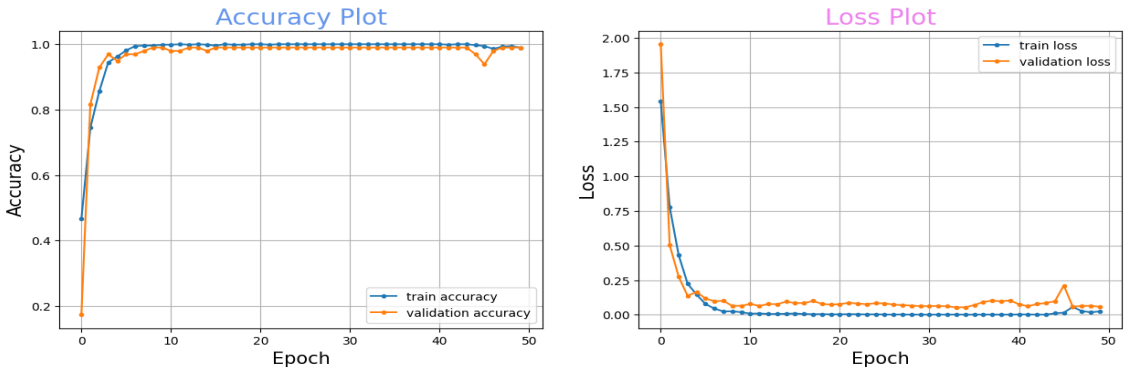



Figure 10. Accuracy and Loss Plot Results of Presented Model on Epoch 50

The step by step implementations of the FER system are shown from Figure 11 to 15.

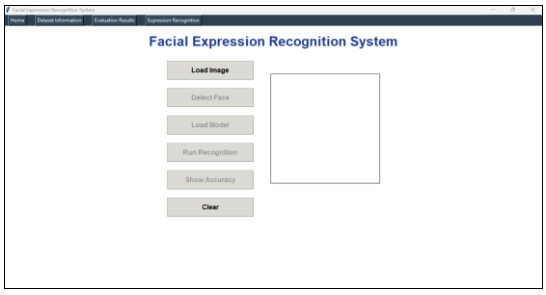



Figure 11. First Page of FER System to load the image


Figure 12. Load Image for FER System

To upload the image, the Load Model button can be clicked shown in Figure 12. After loading the image, the Detect Face button is clicked to detect the image using MTCNN displayed in Figure 13.


Figure 13. Detect Face using MTCNN

After the model is loaded, click Run Recognition button shown in Figure 14. After clicking the Run Recognition button, click the Show Accuracy button to view the results with prediction probabilities displayed in Figure 15.

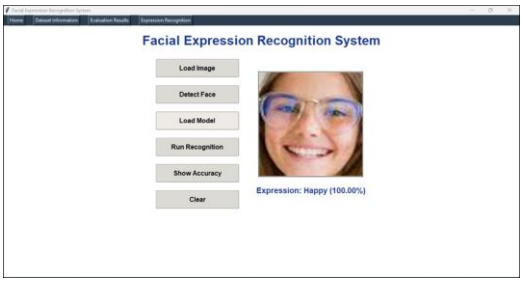



Figure 14. Run Recognition

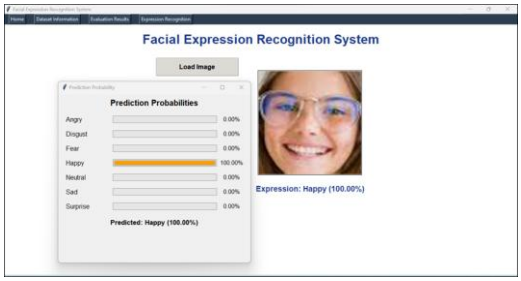



Figure 15. Show Results Accuracy

## 7. CONCLUSION

Facial expressions are a significant point of human non-verbal communication. This study provides an efficient method for using Convolutional Neural Network (CNN) to recognize facial expressions. By incorporating an attention through convolutional networks, the model is better able to identify facial regions and increasing the accuracy of emotion recognition. To improve performance, face detection techniques is applied by using MTCNN pre-train model. The CK+ dataset is used to analyze the experiments of the presented FER model. The experimental results demonstrated that the presented model performs well in terms of accuracy, precision, recall, and F1-score on CK+ datasets. Future work could further improve performance by incorporating larger, real-world datasets, and using transformer-based architectures.

## REFERENCES

[1] Mohan, K., Seal, A., Krejcar, O. et al. FER-net: facial expression recognition using deep neural net. Neural Comput & Applic 33, 9125–9136 (2021). https://doi.org/10.1007/s00521-020-05676-y

[2] Minaee, S., Minaei, M., & Abdolrashidi, A. (2021). Deep-emotion: Facial expression recognition using attentional convolutional network. Sensors, 21(9), 3046.

[3] Fan, Y., Li, V. O., & Lam, J. C. (2020). Facial expression recognition with deeply-supervised attention network. IEEE transactions on affective computing, 13(2), 1057-1071.

[4] Xu, C., Du, Y., Zheng, W., Li, T., & Yuan, Z. (2025). Facial expression recognition based on YOLOv8 deep learning in complex scenes. International Journal of Information and Communication Technology, 26(1), 89-101.

[5] Ren, S., Sun, M., Wang, B., Liu, M., & Men, S. (2025). High Precision Infant Facial Expression Recognition by Improved YOLOv8. IEEE Access.

[6] Lucey, Patrick, et al. “The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression.” Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Computer Society Conference on. IEEE, 2010

[7] Lyons, Michael J., Shigeru Akamatsu, Miyuki Kamachi, Jiro Gyoba, and Julien Budynek. “The Japanese female facial expression (JAFFE) database.” third international conference on automatic face and gesture recognition, pp. 14-16, 1998

**Authors**

**Biography**

Su Hlaing Tun received the Bachelor of Computer Science (B.C.Sc.) degree in 2019 from the University of Computer Studies, Mandalay, Myanmar. She is currently pursuing her Master of Computer Science (M.C.Sc.) degree at University of Computer Studies, Mandalay.

# OPTIMIZING SMART PARKING SYSTEM

Ei Phyo Thwe[1], and Saw Thandar Myint[2]

[1,2] Department of Information Technology Supporting and Maintenance, University of Computer Studies (Mandalay)
eiphyothwe71124@gmail.com, sawthandar26@gmail.com

## ABSTRACT

*Efficient utilization of parking infrastructure is one of the important aspects in the development of smart cities. It is inefficient to monitor the parking availability manually, and it includes many errors, hence calling for an automated process based on vision. This paper is concerned with the detection of vacant parking using the YOLO11 architecture of object detection, which has been enhanced through transfer learning and adjustment of hyperparameters. A custom dataset of parking-slot images is created, while its model is trained to detect vacant as well as occupied parking lots. The most crucial hyperparameters, learning rate, batch size, optimizer type, and input image resolution are elaborately adjusted to enhance the accuracy of detection and improve the stability of convergence. Experimental results are presented to show that the fine-tuned YOLO11 can realize high precision, recall, and mAP, beating the results of the baseline configuration. In general, the system can be applied to the intelligent management of parking.*

## KEYWORDS

*Smart parking, vacant slot detection, YOLO11, fine-tuning, hyperparameter optimization, deep learning*

## 1. INTRODUCTION

Urbanization at a rapid rate has led to an increase in demand for parking spaces, resulting in traffic jams, increased fuel consumption, and environmental degradation. It is evident that the major contributor to urban traffic is drivers searching for parking spaces. Thus, intelligent parking systems have received considerable attention as an integral part of smart city technology. With considerable progress in deep learning, particularly in convolutional neural networks (CNN), object detection in real-world environments has become possible. Within this framework, one widely popular object detection algorithm is the You Only Look Once (YOLO) series. YOLO11 is the most recently developed variant of this algorithm and is appropriate for dense object detection applications like parking

slot analysis. The study will center on uncovering vacant parking slot

spaces based on YOLO11 models and enhancing their accuracy via transfer learning.

The contributions made by this work are:

1. Construction of a parking space vacancy detection system using YOLO11.
2. Use of Transfer Learning with Pre-trained Weights.
3. Hyperparameter optimization using systematic tuning for optimal detection results.
4. Experimental Evaluation Based on Standard Object Detection Metrics.

## 2. RELATED WORK

The problem of empty slot detection in a parking lot has slowly transitioned from conventional image processing methods to more learning-based methods. Conventional methods used background subtraction and knowledge-based features. However, these methods performed poorly in varying lighting conditions, shadowy areas, and partially hidden cars. These drawbacks were manifested in the PKLot dataset, which was put forward by De Almeida _et al._ in [1], a benchmark dataset for a typical outside parking lot scenario. Tests on PKLot proved that CNN models performed better than conventional methods. However, slot classification methods remain prone to errors caused by variations in the viewpoint of the cameras.

These issues have been addressed in recent studies that have employed the strategy of

object detection in their approaches. Unlike other models that focused on the classification task of Slots, those that employed object detection detect cars directly in the images and derive parking slot occupation based on their spatial positioning relative to each other. In recent studies that employed detectors among detectors, the YOLO series, commencing from YOLOv1 through YOLOv8, was predominantly employed owing to its real-time processing and multi-scale representation abilities [3]-[7].

In addition, simulation platforms such as CARLA and AirSim have been used to evaluate detection models under controlled lighting and weather conditions, improving robustness when combined with real datasets like PKLot [8], [9]. Despite these advances, relatively little attention has been given to feature-map analysis for understanding how detectors interpret parking scenes. This gap motivates the exploration of YOLO11, whose enhanced multi-scale feature extraction offers a strong foundation for deeper analysis and improved vacant parking-slot detection.

## 3. METHODOLOGY

### 3.1. YOLO11 Model Architecture

YOLO11: This is a single-stage object detection model capable of performing object localization and classification within a single network. This model has three major parts:

- **Backbone**: It is used to extract the hierarchical feature representation of the image by utilizing convolution layers.
- **Neck**: It aggregates the multi-scale features through the use of feature pyramid networks to enhance the feature learning process for objects of various sizes.
- **Detection Head:** This makes predictions about the boxes and the probability for every slot.

YOLO11 is chosen because it is an optimal combination of detection precision and efficiency, making it suitable for smart parking in real time.

### 3.2. Feature Extraction Techniques

The YOLOv11 model used in this research adopts a convolutional neural network backbone to extract hierarchical feature representations from the input images automatically. The initial levels of the convolutional backbone focus on extracting features of low levels, which include edges and textures, while the higher levels focus on the extraction of higher-level semantic features associated with vehicles and the marking of parking slots. The feature maps obtained from the convolutional backbone are boosted by neck modules with the capability of extracting multi-scale features essential for the detection of vehicles of varying sizes and orientations.

### 3.3. Occupancy Classification

The occupation state of the slots is based on the outcome of the detection done by the YOLOv11 model. For each of the slots that have been detected, a verification is conducted to determine whether there is an overlap between the detected slot and the bounding box of a vehicle. If the overlap is beyond a set threshold, the slot is referred to as occupied; otherwise, it is considered to be empty.

### 3.4. Performance Metrics

The measured performance of the proposed system uses a set of standard metrics for object detection and classification, which are:

- Precision: representing the ratio of the correctly detected occupied slots.
- Recall is short for the system's ability to detect all occupied slots.
- F1-Score, which is the harmonic mean of precision and recall
- Mean Average Precision (mAP), which measures the overall detection performance.
- Accuracy: This refers to the correct classification based on occupancy.

Standard object detection metrics are used to quantitatively validate the proposed vacant parking-slot detection system.

There are similarities in both, with the results of IoU, Precision, Recall, and mAP metric measurements.

#### 3.4.1. Intersection over Union (IoU)

Intersection over Union IoU computes the overlap between the predicted bounding box and its ground-truth bounding box. For detection, predictions whose IoU is higher than a certain threshold-usually 0.5-are considered true positives.

$$IoU = \frac{A_{pred} \cap A_{gt}}{A_{pred} \cup A_{gt}}$$

A detection is considered true positive if the IoU value exceeds a predefined threshold (commonly 0.5).

### 3.4.2. Precision

Precision evaluates the correctness of positive predictions regarding how many of the slots spotted are actually filled.

$$Precision = \frac{TP}{TP + FP}$$

### 3.4.3. Recall

It is the capacity of a system that can detect and identify correctly the actual spots that are occupied.

$$Recall = \frac{TP}{TP + FN}$$

### 3.4.4. Mean Average Precision (mAP)

Mean Average Precision (mAP) is a more refined way of calculating detection performance across several confidence thresholds.

$$AP = \int_0^1 Precision(R)\, dR$$

The mean Average Precision is then calculated by averaging AP over all detection classes:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i$$

## 3.5. System Overview

Overall, the architecture of the entire system is a pipelined one that consists of five separate modules, including data acquisition, processing, extraction, classification, and evaluation of performance. Images of parking lots or videos are obtained from public databases and virtual environments. They are subsequently processed to have the same resolution and improve image clarity.

The resultant images are then used as inputs for the YOLOv11 detection model, where the objects and the spatial features of the image are simultaneously extracted. Based on the outputs of the detection, the parking slots and the vehicles are classified as being filled or empty.

Ultimately, the outputs are tested for performance concerning standard detection and classification measures.

### 3.5.1. System Flow Diagram

In this paper, the parking slot detection is done with the YOLO11s object detection model.

The model consists of three main components:

- Backbone: extracts discriminative spatial features from the input images.
- Neck: This combines multi-scale feature maps to improve detection performance.
- Head: Outputs the predictions of bounding boxes, confidence scores, and class labels.

Transfer Learning

- Pre-trained YOLO11s weights for model initialization.
- The model is fine-tuned on the Parking Dataset to adapt it to the Vacant and Occupied Slot Detection task.

Model Training

- The model gets trained on the training set.
- The validation set is used for the tuning of hyperparameters.
- Overfitting is watched out for during training.

Inference and Generation

- The test set is used to evaluate the trained model.
- The system outputs bounding boxes and class labels pointing to vacant or occupied parking slots.

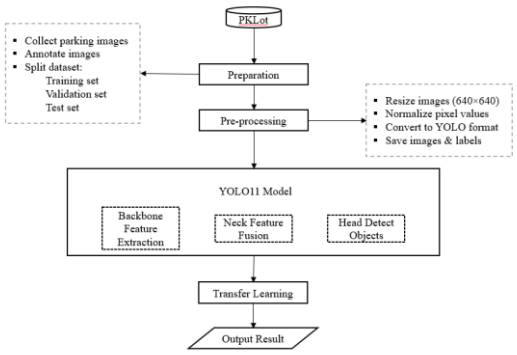



Figure 1: System Flow Diagram

## 4. IMPLEMENTATION

It is implemented solely in a simulated setup, where the system is developed, set up, and tested through the use of recorded images and video data rather than in a real-time setup.

This chapter aims to show how the proposed approach is carried out and verified by computational experiments.

### 4.1. Development Environment and Tools

Simulation framework implementation is done by utilizing the Python programming language, based on its strong support for deep learning as well as computer vision. Utilization of the Ultralytics YOLO framework is done based on its strong support for the base implementation of the YOLO11 model architecture. Training and evaluation are done utilizing a workstation together with a GPU.

The main software components include:

- Python programming language
- Ultralytics YOLO11 framework
- PyTorch deep learning library
- OpenCV for image and video processing

### 4.2. Model Configuration and Fine-Tuning

The YOLO11 is initialized with weights trained on pre-existing data, hence promoting faster convergence and resulting in accurate object detection. Fine-tuning is achieved through the replacement of the final layer based on the number of target classes of the parking dataset.

Instead, YOLO11 is trained with weights from pre-existing data; hence,

The major training parameters set in the simulation are:

- Input image size
- Batch size
- Learning rate
- Number of training epochs

To train the network, stochastic gradient descent with the use of back-propagation is employed. While training, loss with respect to classification, localization, and objectness is calculated.

### 4.3. Inference and Vacant Slot Detection

After the completion of the training process, the YOLO11 model is used for inference on the test dataset by examining every image separately. It provides information on bounding box positions, confidence levels, and class predictions related to whether the parking slot is empty or occupied. Non-maximum suppression is employed for removing any overlap between the results.

## 5. EXPERIMENTAL RESULTS

### 5.1. Quantitative Results

The results have been calculated on the basis of precision, recall, IoU, and mAP. The mAP@0.5 is 0.984 and the COCO mAP@0.5:0.95 is 0.8192, which is a very accurate estimate that includes various levels of IoU threshold value. The total IoU is 0.8886, which estimates the accurate localization, and the precision (0.9585) and recall (0.9699) are very high that estimates the proper classification with less false negatives. Generally, the mAP@0.5 serves the best estimate for this categorization.

Table 1. Overall Test Dataset Performance Metrics

| Metric | Value |
|---|---|
| mAP@0.5 (IoU ≥ 0.5) | 0.9840 |
| mAP@0.5:0.95 | 0.8192 |

| | |
|---|---|
| (COCO) | |
| Mean IoU | 0.8886 |
| Average Precision (P) | 0.9585 |
| Average Recall (R) | 0.9699 |

Table 2. Per-Class mAP Performance

| Class | mAP@0.5:0.95 |
|---|---|
| Space–Empty | 0.8228 |
| Space–Occupied | 0.8156 |

Table 3. Per-Class IoU Performance

| Class | IoU |
|---|---|
| Space–Empty | 0.8880 |
| Space–Occupied | 0.8892 |

### 5.2. Detection Results of Vacant and Occupied Parking Spaces

The detection output of the proposed YOLO11s-based model on a test parking image. Empty spaces are marked in green and occupied spaces in red, with 19 detections (6 empty and 13 occupied), demonstrating accurate localization and classification in real-world scenes.


Figure 2: Detection results with empty and occupied

### 5.3. Training and Validation Loss Curves

The training and validation loss trends across epochs, including bounding box, classification, and distribution focal losses. The decreasing training losses and stable validation losses indicate effective learning and good generalization without significant overfitting.

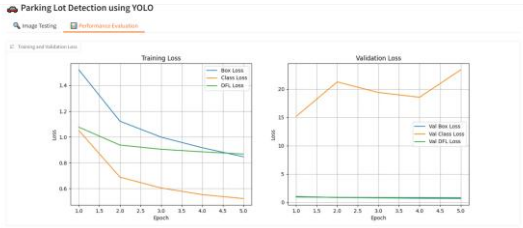



Figure 3: Loss Curves of Training and Validation

### 5.4. Validation Performance Metrics at Final Epoch

The following figure represents the final validation results of the model on mAP@0.5, mAP@0.5:0.95, precision, and recall values. The reason for such high values for all parameters is to ensure accurate detection results, tight localization, and balanced precision and recall, which justify the effectiveness of the proposed model for the vacant parking slot detection task.

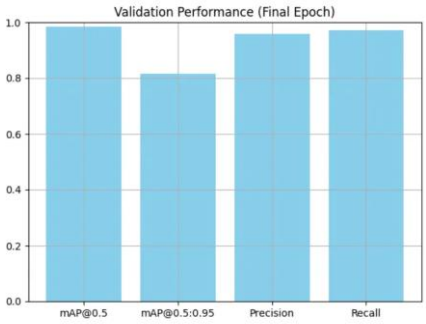



Figure 4: Validation Performance Metrics (Final Epoch)

### 5.5. Confidence-Based Performance Analysis

The performance of the proposed static image–based parking space classification model. The F1–confidence curve demonstrates stable and high performance across a wide range of confidence thresholds, with a peak F1-score of approximately 0.96. The precision–recall curve shows strong discriminative capability for both empty and occupied parking spaces, achieving a mAP@0.5 of about 0.985.

The precision–confidence and recall–confidence curves indicate that higher confidence thresholds improve precision while

maintaining high recall until very strict thresholds are applied. The confusion matrix and its normalized version confirm high classification accuracy, with approximately 95% accuracy for empty spaces and 97% for occupied spaces. Overall, the results validate the robustness and effectiveness of the proposed approach for parking space occupancy detection using static images.

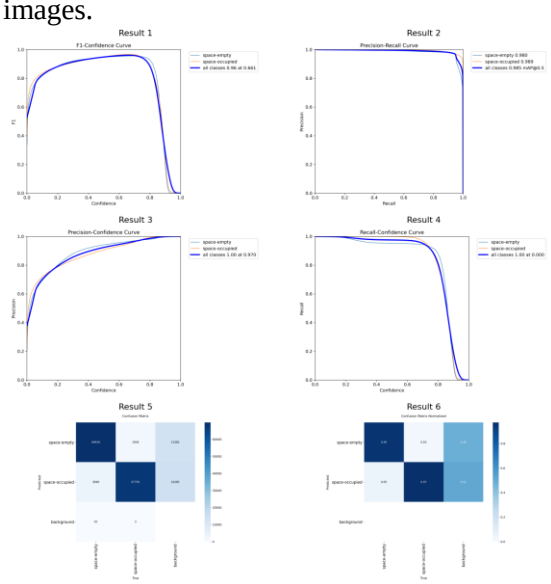



Figure 7: Performance Evaluation of the Static Image–Based Parking Space Classification Model

## 6. CONCLUSIONS

In this paper, a parking slot vacancy detection system using YOLO11, an object detection model, is introduced and analyzed based on the PKLot dataset. The proposed algorithm is capable of detecting whether a parking slot is vacant or not, irrespective of weather, light, and the presence of occlusion, due to efficient feature extraction and post-processing techniques. Experimental analysis reveals that it achieves higher accuracy and can process objects instantly, proving that YOLO11 can be utilized for intelligent and effective parking and transport systems. Future research will be pursued for smaller and multi-level parking systems.

## REFERENCES

[1] J. R. De Almeida, L. S. Oliveira, A. S. Britto, E. A. Silva, and A. L. Koerich, "PKLot – A robust dataset for parking lot classification," Expert Systems with Applications, vol. 42, no. 11, pp. 4937–4949, 2015.

[2] G. Amato, F. Carrara, L. Falchi, and C. Gennaro, "Deep learning for decentralized parking lot occupancy detection," in Proc. IEEE International Conference on Big Data, 2016, pp. 3262–3267.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-time object detection," in Proc. IEEE CVPR, 2016, pp. 779–788.

[4] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in Proc. IEEE CVPR, 2017, pp. 6517–6525.

[5] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv:1804.02767, 2018.

[6] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv:2004.10934, 2020.

[7] G. Jocher et al., "YOLOv5 by Ultralytics," 2020.

[8] A. Dosovitskiy et al., "CARLA: An open urban driving simulator," in Proc. CoRL, 2017, pp. 1–16.

[9] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-fidelity visual and physical simulation for autonomous vehicles," in Field and Service Robotics, Springer, 2017.

### Authors Biography

Ei Phyo Thwe received the Bachelor's degree in Computer Science (B.C.Sc) in 2021 and Master of Computer Science (M.C.Sc) from the University of Computer Studies, Myanmar. She is currently pursuing postgraduate research in the field of computer vision and artificial intelligence. Her research interests include deep learning, object detection, and intelligent transportation systems, with a particular focus on smart parking systems using YOLO-based architectures. Her recent work involves the analysis of feature maps in neural networks and the application of object detection models for vacant parking slot detection.

# NAMED ENTITY RECOGNITION IN OLD ENGLISH: DIRECT, TRANSLATION-BASED AND RETRIEVAL-AUGMENTED APPROCHES

Shine Khant Aung[1], and Khotskina Olga Valerievna[2]

[1,2]Department of Mechanic and Mathematics, Novosibirsk State University
shine76225@gmail.com

## ABSTRACT

*Named Entity Recognition (NER) [1] in Old English [2] remains a challenging task due to the scarcity of annotated resources, complex morphology, and inconsistent spelling conventions. In this work, we present a comprehensive comparison of three distinct approaches to Old English NER: direct supervised learning, translation-based NER, and retrieval-augmented large language model methods. For the direct approach, we construct a manually annotated dataset of 905 Old English sentences focusing on person names and fine-tune a RoBERTa [3] model. For the translation-based approach, we compile a parallel corpus of 8,832 sentences by combining Beowulf and the Old English dataset from Hugging Face, and fine-tune an mBART [4] model for Old English to Modern English translation. The translated text is then processed using a pretrained BERT-based NER model.[5]*

*In addition, we explore retrieval-augmented NER using GPT-based [6] methods, including zero-shot, few-shot, sentence similarity retrieval, and entity-level embedding retrieval with self-verification. All approaches are evaluated on a shared out-of-domain test set derived from the Old English ChronicleA corpus. Experimental results demonstrate that retrieval-augmented methods, particularly entity-level retrieval combined with self-verification, consistently outperform both direct and translation-based approaches in terms of F1-score. Our findings highlight the limitations of supervised learning with limited data and error propagation in translation-based pipelines, while emphasizing the effectiveness of retrieval mechanisms in low-resource historical language settings. This study provides the first unified evaluation of multiple NER paradigms for Old English and offers practical insights for future research in historical and low-resource NLP.*

## KEYWORDS

*Old English, Named Entity Recognition, Low resource NLP, RoBERTa, mBART, GPT-NER*

## 1. INTRODUCTION

Named Entity Recognition (NER) [1] is a fundamental task in Natural Language Processing (NLP) that aims to identify and classify named entities such as person names, locations, and organizations within text. While NER has achieved strong performance in high-resource languages such as English, its application to historical and low-resource languages remains largely underexplored. In particular, Old English poses significant challenges due to its complex morphology, lack of standardized spelling, and limited availability of annotated data.

Unlike Modern English, Old English exhibits substantial orthographic variation and rich inflectional forms, making it difficult for existing NER models to generalize effectively. Furthermore, the scarcity of labeled datasets limits the performance of supervised learning approaches. Even transformer-based architectures such as RoBERTa [3] may struggle in such low-resource settings when training data is insufficient.

To address these challenges, alternative approaches have been explored in low-resource NLP. One common strategy is translation-based NER, where text is translated into a high-

resource language and processed using pretrained models. Recent advances in multilingual sequence-to-sequence models such as mBART [4] have made this approach more feasible. However, translation pipelines can introduce errors such as semantic shifts and entity distortions, which may negatively affect downstream NER performance.

In addition, large language models (LLMs) [7] have demonstrated strong capabilities in zero-shot and few-shot learning scenarios, enabling NER without requiring extensive labeled data. Retrieval-augmented approaches further enhance this process by incorporating relevant contextual examples, allowing models to leverage external information dynamically. These methods have shown promising results in low-resource and domain-specific tasks.

In this paper, we present a comparative study of three approaches to Old English NER: (1) direct supervised NER using a fine-tuned transformer model, (2) translation-based NER leveraging Modern English resources, and (3) retrieval-augmented NER using GPT-based methods [6]. For the direct approach, we construct a manually annotated dataset of 905 Old English sentences focusing on person names and fine-tune a transformer model. For the translation-based approach, we compile a parallel corpus of 8,832 sentence pairs and fine-tune a translation model, followed by NER using a pretrained BERT model [5] on the translated text. For the retrieval-based approach, we explore zero-shot, few-shot, sentence similarity retrieval, and entity-level embedding retrieval.

All methods are evaluated on a shared out-of-domain test set derived from the Old English ChronicleA [8] corpus, ensuring a fair comparison across approaches. Through this study, we aim to answer the following research questions: (1) How effective are different approaches for NER in Old English? (2) Which method performs best under low-resource conditions? and (3) What are the strengths and limitations of each approach?

The contributions of this work are as follows:

- We provide a systematic comparison of multiple NER approaches in Old English.
- We construct both labelled and parallel datasets for experimentation.
- We demonstrate the effectiveness of retrieval-augmented methods in low-resource settings.

## 2. Related Work

Named Entity Recognition (NER) has been extensively studied in high-resource languages, with recent research focusing on improving performance in low-resource and domain-specific settings.

Liu et al. (2021) [9] proposed NER-BERT, a domain-specific pre-training approach designed to address the limitations of standard BERT in low-resource scenarios. Instead of relying on small annotated datasets, they constructed a large-scale corpus from Wikipedia using hyperlink anchors and DBpedia ontology, resulting in millions of labeled examples. The model replaces the masked language modeling objective with an entity tagging task, enabling direct learning of entity recognition. Their results show significant improvements, especially when only limited labeled data is available.

To incorporate entity-level knowledge into pre-trained models, Jia et al. (2020) [10] introduced an entity-enhanced BERT model for Chinese NER. Their method combines unsupervised entity discovery with a modified transformer architecture (Char-Entity-Transformer), which integrates entity embeddings into the attention mechanism. This approach improves performance by explicitly modeling entity information, particularly in morphologically complex languages.

For multilingual NER, Tavan and Najafi (2022) [11] proposed a transformer-based architecture using mT5 embeddings. Their model addresses challenges in multilingual settings by handling subword tokenization through a subtoken alignment mechanism and capturing contextual dependencies with multi-head self-attention. The model achieves competitive performance across multiple languages, demonstrating the effectiveness of multilingual pre-trained representations.

In the context of historical languages, Zhang et al. (2025) [12] developed a GujiRoBERTa-LSTM-CRF model for ancient Chinese NER. Their approach combines domain-specific pre-training with sequential modeling and a CRF layer to enforce label consistency.

Experimental results show strong performance, highlighting the importance of domain-adapted models for historical text processing.

Recent advances in large language models have introduced new approaches to NER. Wang et al. (2023) [6] proposed GPT-NER, which reformulates NER as a text generation task using prompt-based learning. The method uses special tokens to mark entity boundaries and incorporates retrieval-based demonstration selection and self-verification to improve accuracy. Their results demonstrate that retrieval-augmented methods can achieve performance close to supervised models, particularly in low-resource settings.

To further improve few-shot NER, Wang et al. (2022) [13] introduced InstructionNER, which converts NER into an instruction-based text-to-text task. The model leverages structured prompts and multi-task learning to improve both entity boundary detection and type classification, achieving strong generalization with limited training data.

Cui et al. (2021) [14] proposed a template-based NER framework that reformulates NER as a sequence-to-sequence ranking problem. By using natural language templates, the model can handle different entity types without modifying the output layer, enabling effective cross-domain transfer and improved few-shot performance.

For domain-specific NER in non-English languages, Ivanin et al. (2020) [15] developed RuREBus, a system for Russian e-Government texts. Their work highlights the challenges of domain adaptation and shows that combining pre-trained models with domain-specific features is necessary for handling specialized corpora.

Finally, Zaratiana et al. (2023) [16] introduced GLiNER, a model designed for zero-shot NER across arbitrary entity types. The model represents both text spans and entity types in a shared embedding space, allowing flexible and efficient extraction without retraining. Their results demonstrate strong performance across multiple datasets and languages.

Overall, these studies highlight three main directions in NER research: domain-adapted supervised learning, multilingual and translation-based approaches, and prompt-based retrieval-augmented methods using large language models. However, these approaches have not been systematically compared in the context of Old English, which motivates our work.

## 3. METHODOLOGY

In this work, we investigate Named Entity Recognition (NER) in Old English using three different approaches: direct supervised NER, translation-based NER, and retrieval-augmented NER. This section describes the task formulation, datasets used for training, and the implementation details of each approach.

### 3.1 Task Definition

The goal of this study is to perform Named Entity Recognition on Old English text, focusing specifically on person names.

Given an input Old English sentence, the task is to identify and extract all person entities present in the sentence. This is formulated as a sequence labeling problem for supervised models and as a text generation or extraction task for large language models.

### 3.2 Direct NER

The direct NER approach applies supervised learning directly on Old English data.

For model training, we fine-tune a pretrained transformer-based model, RoBERTa [3], for token classification. The dataset is converted into a structured format of tokens and labels, and subword tokenization is applied with label alignment.

The model is trained using an 80/20 split for training and validation, with evaluation performed at each epoch. The best-performing model is selected based on the F1-score on the validation set.

The trained model is then applied to unseen Old English sentences from ChronicleA [8] for entity extraction. This approach serves as a baseline representing supervised NER in a low-resource setting.

Figure 1 illustrates the overall pipeline of the direct NER approach.

### 3.3 Translation-Based NER

The translation-based approach leverages resources from high-resource languages to improve NER performance.

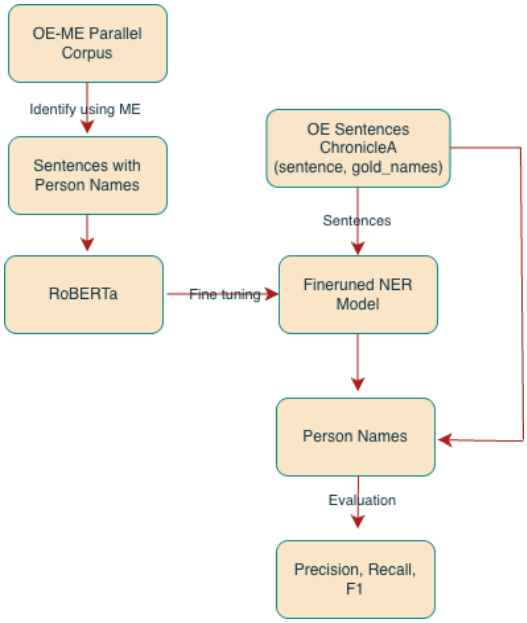



Figure 1: Pipeline of the Direct Finetuned NER approach. Modern English translations are used to assist annotation, followed by manual labeling of Old English sentences. The labeled data is then used to fine-tune a RoBERTa model for person name recognition.

We fine-tune a multilingual sequence-to-sequence model, mBART [4] , for Old English to Modern English translation. The trained model is then used to translate Old English sentences into Modern English.

The translated sentences are processed using a pretrained NER model (Jean-Baptiste/roberta-large-ner-english) to extract person names.

This pipeline allows us to leverage well-established NER models in Modern English. However, the overall performance is highly dependent on translation quality, as errors such as incorrect or missing entity translations may negatively affect downstream NER results.

Figure 2 illustrates the overall pipeline of the translation-based NER approach.

## 3.4 Retrieval-Augmented NER

To address the limitations of supervised and translation-based approaches, we explore retrieval-augmented NER using large language models. Our approach follows the GPT-NER framework (Wang et al., 2023) [6] , where NER is formulated as a text generation task.

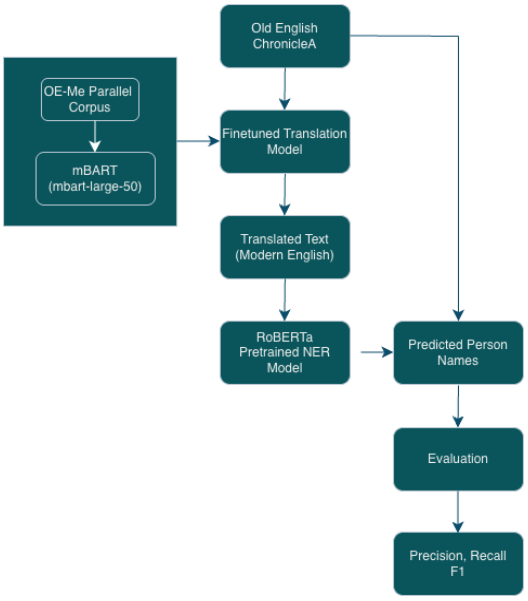



Figure 2: Pipeline of the translation-based NER approach. Old English sentences are translated into Modern English using a fine-tuned mBART model trained on an OE–ME parallel corpus. The translated text is then processed by a pretrained RoBERTa NER model to extract person names, and the predictions are evaluated using standard metrics.

### 3.4.1 Dataset and Setup.

We use a labeled Old English dataset consisting of 930 sentences, including 25 sentences without person entities. The dataset is divided into 5 folds for cross-validation. In each fold includes a balanced distribution, ensuring that 5 sentences without person names are included. The remaining folds are used as the retrieval pool.

### 3.4.2 Prompt Structure

For all retrieval-based methods, we construct prompts consisting of three components: (1) a task description, (2) optional demonstration examples, and (3) the input sentence. The model is instructed to extract person names by generating a labeled sentence using special markers (e.g., @@entity##).

### 3.4.3 Self-Verification.

To improve prediction reliability, we apply a self-verification step across all methods. Each predicted entity is re-evaluated by prompting the model with a yes/no question to determine whether it belongs to the target entity type.

Only verified entities are retained as final predictions.

### 3.4.4 Retrieval Strategies.

We evaluate multiple retrieval strategies, differing only in how demonstration examples are selected:

• Zero-shot: No demonstration examples are provided.

• Random Retrieval: k examples are randomly sampled from the training data.

• Sentence Similarity Retrieval: k examples are selected based on sentence-level similarity using embedding models.

• Entity-Level Retrieval: k examples are selected based on similarity between entity representations extracted from a fine-tuned NER model.

For retrieval-based methods, we experiment with different values of k ∈ {4, 8, 16, 32} to analyze the impact of the number of retrieved examples.

All methods share the same prompting format and self-verification process, ensuring that performance differences arise solely from the retrieval strategy.

In the zero-shot setting, no demonstration examples are provided in the prompt. The model directly predicts entities based on the task description and input sentence, followed by self-verification..

In random retrieval, k demonstration examples are randomly sampled from the training dataset and included in the prompt. These examples provide guidance for the model, and self-verification is applied to refine predictions.

In sentence similarity retrieval, k demonstration examples are selected based on the semantic similarity between the input sentence and training sentences.

We use a pretrained sentence embedding model, SimCSE (princeton-nlp/sup-simcse-bert-base-uncased), to encode both the input sentence and candidate sentences into dense vector representations.

The similarity between sentences is computed using cosine similarity. Given an input sentence embedding x and a candidate sentence embedding y, the similarity score is defined as:

$$Sim(x, y) = \frac{(x.y)}{||x||\ .\ ||y||}$$

Based on this similarity score, the top-k most similar sentences are selected as demonstration examples and included in the prompt.

This approach provides more contextually relevant examples compared to random retrieval, leading to improved entity extraction performance.

In addition to sentence-level retrieval, we explore entity-level embedding retrieval.

In entity-level retrieval, k demonstration examples are selected based on the similarity between entity representations rather than full sentence similarity.

We first use a fine-tuned RoBERTa [3] NER model to extract entity representations from the training dataset. Each entity is encoded into a dense vector representation along with its surrounding context, forming an entity-level embedding datastore.

Given an input sentence, candidate entity representations are compared with stored entity embeddings to identify the most relevant examples. The similarity between embeddings is computed using cosine similarity. The top-k most similar entity examples are selected and used as demonstrations in the prompt.

## 4. DATASET

In this study, we utilize multiple datasets for training and evaluation across different approaches to Old English Named Entity Recognition (NER). [1]All approaches are evaluated on a shared out-of-domain test dataset derived from the Old English ChronicleA [8] corpus. This dataset contains 237 sentences and is not used during training or validation, ensuring an unbiased evaluation of model generalization.

The person name annotations in ChronicleA [8] are manually extracted by an expert, providing high-quality ground truth labels for evaluation. Model performance is measured by comparing predicted entities with these manually annotated references using precision, recall, and F1-score.

### 4.1 Direct NER Dataset

For the direct NER approach, we construct a labeled Old English dataset using a parallel corpus obtained from Hugging Face [17] . The Modern English translations are used as annotation support to identify person names, which are then manually labeled in the corresponding Old English sentences. Using this process, we obtain a dataset consisting of 905 Old English sentences with token-level annotations for person name entities. The dataset is split into training and validation sets with a ratio of 80% and 20%, respectively.

We fine-tune a RoBERTa-based NER model [3] on the training set and select the best-performing model based on validation performance. The trained model is then applied to unseen Old English sentences of ChronicleA for entity extraction.

### 4.2 Translation Dataset

For the translation-based approach, we construct a parallel corpus by combining the Beowulf [18]dataset , the Old English dataset obtained from Hugging Face [17] and The Homilies of the Anglo Saxon Church [19]. The resulting corpus contains 8,832 sentence pairs, consisting of aligned Old English and Modern English sentences.

This parallel dataset is used to fine-tune a multilingual sequence-to-sequence model (mBART) [4] for Old English to Modern English translation. The diversity of the corpus allows the model to learn translation patterns across different linguistic styles and contexts.

The constructed dataset serves as the foundation for the translation-based NER pipeline, enabling the use of high-resource language models for downstream entity recognition.

### 4.3 Retrieval Dataset

For retrieval-based methods, we construct the retrieval dataset based on the labeled Old English NER dataset used in the direct approach. The dataset consists of 905 sentences with person name annotations, supplemented by an additional 25 sentences that do not contain any person entities.

The final retrieval dataset contains a total of 930 sentences and serves as the source pool for selecting demonstration examples in retrieval-based NER methods.

This dataset is used to retrieve relevant or random examples depending on the retrieval strategy, including random sampling, sentence similarity, and entity-level retrieval.

## 5. EXPERIMENTS

This section describes the experimental setup, including implementation details, training configurations, and evaluation methodology used to compare different approaches for Old English Named Entity Recognition (NER).

### 5.1 Implementation Details

All experiments were implemented using the PyTorch framework and executed on Google Colab with GPU acceleration.

For the direct NER approach, a pretrained RoBERTa [3] model was fine-tuned on the manually annotated Old English dataset. The model was adapted for token classification to identify person name entities.
For the translation-based approach, a multilingual sequence-to-sequence model, mBART, was fine-tuned on 8,832 Old English–Modern English sentence pairs. The fine-tuned translation model was used to translate Old English sentences into Modern English. The translated output was then processed using a pretrained BERT-based NER model [5] to extract person names.

For retrieval-augmented NER, GPT-4.1 was used for all experiments. A consistent prompt structure was applied across all methods, consisting of a task description, optional demonstration examples, and the input sentence.

The differences between methods arise from how demonstration examples are selected, including zero-shot (no examples), random sampling, sentence similarity retrieval using SimCSE embeddings, and entity-level retrieval based on representations extracted from a fine-tuned NER model.

In all retrieval-based methods, a self-verification step was applied to validate predicted entities and reduce false positives.

### 5.2 Training Configuration

For the direct NER approach, the RoBERTa-based token classification model was fine-tuned

using the Hugging Face Trainer framework. The model was trained for 5 epochs with a learning rate of $2 \times 10^{-5}$ , a batch size of 32 for both training and evaluation, and a weight decay of 0.01. A warmup ratio of 0.1 was applied during training.

Evaluation and model checkpointing were performed at the end of each epoch. The best-performing model was automatically selected based on the validation F1-score. Mixed-precision training (FP16) was enabled when GPU resources were available to improve training efficiency. In addition, the random seed was fixed to 42 to ensure reproducibility.

For the translation-based approach, the translation model was fine-tuned using (facebook/mbart-large-50-many-to-many-mmt ) with the Hugging Face Seq2SeqTrainer framework. The model was trained for up to 10 epochs with a learning rate of $2 \times 10^{-5}$ , a batch size of 32 for both training and evaluation, and a weight decay of 0.01. To improve generalization, label smoothing with a factor of 0.1 and a warmup ratio of 0.06 were applied. Evaluation and checkpoint saving were performed at the end of each epoch, and the best-performing model was selected based on the BLEU score. Early stopping was also used with a patience of 2 epochs. Mixed-precision training with BF16 was enabled, and the random seed was fixed to 42.

For the NER step, the translated Modern English sentences were processed using the pretrained English NER model (Jean-Baptiste/roberta-large-ner-english) through the Hugging Face pipeline interface. Entity extraction was performed sentence by sentence with the aggregation_strategy set to simple, and only entities predicted as **PER** were retained. Duplicate person names within the same sentence were removed before saving the final extracted results.

### 5.3 Evaluation Setup

All approaches were evaluated on a shared out-of-domain test dataset derived from the Old English ChronicleA corpus. This dataset was not used during training and serves as a benchmark for evaluating generalization performance.

For the direct NER and translation-based approaches, the dataset was split into training and validation sets with a ratio of 80/20. The best-performing model was selected based on validation performance and then evaluated on the ChronicleA test dataset.

For retrieval-augmented methods, 5-fold cross-validation was applied. In each fold, a portion of the dataset was used as the test set, while the remaining data served as the retrieval pool. Performance was averaged across all folds to obtain the final evaluation results.

### 5.4 Evaluation Metrics

We evaluate model performance using standard metrics for NER:

- Precision (P): the proportion of correctly predicted entities among all predicted entities
$$P = \frac{TP}{TP + FP}$$
- Recall (R): the proportion of correctly predicted entities among all ground truth entities
$$R = \frac{TP}{TP + FN}$$
- F1-score (F1): the harmonic mean of precision and recall
$$F1 = 2.\frac{P \,.\, R}{P + R}$$

These metrics provide a comprehensive evaluation of both accuracy and completeness of entity extraction.

## 6. RESULTS

This section presents the performance of different approaches for Old English Named Entity Recognition (NER), including direct NER, translation-based NER, and retrieval-augmented methods. All approaches are evaluated on the ChronicleA test dataset using precision, recall, and F1-score. For retrieval-augmented NER, all methods were first evaluated using 5-fold cross validation on the labeled dataset to ensure a fair comparison across different retrieval strategies.

Based on the cross-validation results, sentence similarity retrieval with self-verification achieved the best performance among all retrieval methods. Therefore, this method was selected for further evaluation on the out-of-domain ChronicleA test dataset.The results on ChronicleA demonstrate the generalization ability of the best-performing retrieval-based approach.

### 6.1 Overall Performance

Table 1 presents the performance of different retrieval-based NER methods evaluated using 5-fold cross-validation. The table reports precision, recall, and F1-score for each method, including zero-shot, random retrieval, sentence embedding retrieval, and entity embedding retrieval, all combined with self-verification.

Table 1: Cross-validation performance of retrieval-based NER methods using precision, recall, and F1-score

| Retrieval Method | Precision | . | F1 |
|---|---|---|---|
| Zero-Shot + SV | 91.05 | 82.52 | 86.57 |
| Random Retrieval + SV | 96.93 | 90.99 | 93.86 |
| Sentence Embedding Retrieval+SV | 97.06 | 93.24 | 95.11 |
| Entity Embedding Retrieval+SV | 96.80 | 91.91 | 94.29 |

Table 2 presents the performance comparison between the direct NER approach, the translation-based NER approach, and the best-performing retrieval-based method (sentence similarity retrieval with self-verification) on the ChronicleA test dataset. The table reports precision, recall, and F1-score for each method.

Table 2: Performance comparison of direct NER, translation-based NER, and the best retrieval-based method on the ChronicleA dataset

| Method | Precision | Recall | F1 |
|---|---|---|---|
| Direct NER | 59.57 | 66.67 | 62.92 |
| Translation-Based NER | 68.83 | 59.15 | 63.63 |
| Sentence Embedding Retrieval+ SV | 90.59 | 88.46 | 89.51 |

### 6.2 Comparison Across Methods

The results show a clear performance gap between traditional approaches and retrieval-augmented methods. In the cross-validation setting, all retrieval-based methods achieve significantly higher performance, with F1-scores ranging from 86.57 to 95.11. Among them, sentence embedding retrieval with self-verification performs best, reaching an F1-score of 95.11.

When evaluated on the ChronicleA [8] dataset, the same method (sentence similarity retrieval with self-verification) maintains strong performance with an F1-score of 89.51, outperforming both direct NER (62.92) and translation-based NER (63.63) by a large margin.

In contrast, direct NER and translation-based NER show relatively lower performance, indicating limitations in handling low-resource Old English data and domain shift. These results demonstrate that retrieval-augmented approaches are more effective and robust, particularly when combined with semantically relevant example selection and self-verification.

### 6.3 Effect of Retrieval and Self-Verification

The results demonstrate the significant impact of retrieval strategies and self-verification on NER performance. Compared to the zero-shot setting, incorporating retrieved examples leads to substantial improvements across all evaluation metrics. In particular, random

retrieval already provides a strong performance boost, indicating that even non-specific examples can help guide the model.

More importantly, semantically informed retrieval methods further enhance performance. Sentence embedding retrieval achieves the best results, suggesting that selecting contextually similar examples is highly effective for improving entity recognition. Entity-level retrieval also shows strong performance, highlighting the usefulness of leveraging entity-specific representations.

In addition to retrieval, the self-verification step plays a crucial role in improving prediction reliability. By filtering out incorrect predictions, self-verification helps reduce false positives and leads to higher precision across all methods. Overall, the combination of retrieval and self-verification significantly improves both accuracy and robustness in low-resource NER.

### 6.4 Generalization to Out-of-Domain Data

All methods are evaluated on the Chronicle-

A [8] dataset, which is not used during training, ensuring that the results reflect true generalization performance.

The results show that the retrieval-based approach (sentence similarity retrieval with self-verification) maintains strong performance with an F1-score of 89.51, significantly outperforming direct NER (62.92) and translation-based NER (63.63). This indicates that retrieval-augmented methods are more robust to domain shifts and can better generalize to unseen Old English data.

These findings suggest that incorporating relevant contextual examples is particularly effective for low-resource and historical language tasks.

## 7. Analysis

The experimental results reveal clear performance differences across the three approaches. The direct NER model achieves moderate performance, with an F1-score of 62.92, which is limited by the small size of the labeled dataset and the challenges of modeling Old English directly. Similarly, the translation-based approach achieves a comparable F1-score of 63.63, benefiting from pretrained English NER models but suffering from translation errors that negatively affect entity recognition.

In contrast, retrieval-augmented methods demonstrate significantly stronger performance. In the cross-validation setting, all retrieval-based approaches outperform the traditional methods, with F1-scores exceeding 86. Notably, sentence embedding retrieval with self-verification achieves the highest performance (F1 = 95.11), indicating the effectiveness of selecting semantically relevant examples.

When evaluated on the ChronicleA dataset, the same method maintains strong generalization performance, achieving an F1-score of 89.51. This substantial improvement over direct and translation-based approaches highlights the robustness of retrieval-based methods in handling domain shifts.

The results also show that the choice of retrieval strategy plays an important role. Sentence-level similarity outperforms entity-level retrieval, suggesting that broader contextual information is more beneficial than isolated entity representations in this setting. Additionally, the inclusion of self-verification consistently improves prediction reliability by reducing false positives.

Overall, these findings demonstrate that retrieval-augmented approaches are more effective for low-resource NER tasks, particularly when combined with semantically informed retrieval and verification mechanisms.

## 8. Conclusion and Future Work

In this paper, we investigated Named Entity Recognition (NER) in Old English by comparing three approaches: direct supervised NER, translation-based NER,

and retrieval-augmented NER using large language models.

The experimental results show that both direct and translation-based approaches achieve moderate performance, with F1-scores of 62.92 and 63.63, respectively. These results highlight the challenges of low-resource data and the limitations of relying on translation quality for downstream tasks.

In contrast, retrieval-augmented methods significantly outperform the traditional approaches. Among them, sentence embedding retrieval with self-verification achieves the best performance, reaching an F1-score of 95.11 in cross-validation and maintaining strong generalization with an F1-score of 89.51 on the ChronicleA dataset.

Overall, our findings demonstrate that retrieval-augmented approaches are highly effective for NER in low-resource and historical language settings. By leveraging relevant contextual examples and verification mechanisms, these methods provide a robust alternative to conventional

There are several directions for future work. First, this study focuses only on person name recognition; extending the approach to additional entity types such as locations would provide a more comprehensive evaluation.

Second, expanding the size and diversity of the annotated Old English dataset could further improve the performance of supervised models. Larger and more diverse training data may help reduce the limitations observed in direct NER.

Third, more advanced retrieval strategies and embedding models could be explored to improve the quality of retrieved examples.

Finally, combining sentence-level and entity-level retrieval may lead to better performance.

## REFERENCES

[1]. P. Yadav, V. Pandey, R. Goyal, and M. Saini, "A survey on recent advances in named entity recognition," *arXiv preprint arXiv:2401.10825*, Jan. 2024.

[2]. P. S. Baker, Introduction to Old English, 3rd ed. Chichester, UK: Wiley-Blackwell, 2012.

[3] Y. Liu, M. Ott, N. Goyal, et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv:1907.11692, 2019.

[4] Y. Liu, J. Gu, N. Goyal, et al., "Multilingual Denoising Pre-training for Neural Machine Translation," arXiv:2001.08210, 2020.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv:1810.04805, 2019.

[6] X. Wang, Y. Zhang, Z. Gan, et al., "GPT-NER: Named Entity Recognition via Large Language Models," arXiv:2304.10428, 2023.

[7] T. B. Brown, B. Mann, N. Ryder, et al., "Language Models are Few-Shot Learners," in Advances in Neural Information Processing Systems (NeurIPS), 2020.

[8] 5. Jebson, T. (Ed.). (2007). The Anglo-Saxon Chronicle (MS A): Electronic composite edition. https://asc.jebbo.co.uk/a/a-L.html

[9] Z. Liu, F. Jiang, Y. Hu, C. Shi, and P. Fung, "NER-BERT: A Pre-trained Model for Low-Resource Entity Tagging," arXiv:2112.00405, 2021.

[10] C. Jia, Y. Shi, Q. Yang, and Y. Zhang, "Entity Enhanced BERT Pre-training for Chinese Named Entity Recognition," in Proc. EMNLP, 2020, pp. 6384–6396.

[11] E. Tavan and M. Najafi, "MarSan at SemEval-2022 Task 11: Multilingual Complex Named Entity Recognition Using T5 and Transformer Encoder," in Proc. SemEval, 2022, pp. 1639–1646.

[12] Y. Zhang, M. Liu, H. Tang, S. Lu, and L. Xue, "Simple Named Entity Recognition System with RoBERTa for Ancient Chinese," in Proc. Ancient Language Processing Workshop, 2025, pp. 129–136.

[13] X. Wang, W. Zhou, C. Zu, H. Xia, T. Chen, Y. Zhang, R. Zheng, J. Ye, and Q. Gu, "InstructionNER: A Multi-Task Instruction-Based Generative Framework for Few-Shot Named Entity Recognition," arXiv:2203.03903, 2022.

[14] L. Cui, Y. Wu, J. Liu, S. Yang, and Y. Zhang, "Template-Based Named Entity Recognition Using BART," in Findings of ACL-IJCNLP, 2021, pp. 1835–1845.

[15] V. Ivanin, E. Artemova, T. Batura, V. Ivanov, V. Sarkisyan, E. Tutubalina, and I. Smurov, "RuREBus: A Case Study of Joint Named Entity Recognition and Relation Extraction from E-Government Domain," in Proc. COLING, 2020, pp. 2013–2023.

[16] U. Zaratiana, N. Tomeh, P. Holat, and T. Charnois, "GLiNER: Generalist Model for Named Entity Recognition Using Bidirectional Transformer," arXiv:2311.08526, 2023.

[17] "The Old English dataset," Hugging Face. [Online]. Available: https://huggingface.co/datasets/apssg96/the-old-english-dataset.

[18] "Beowulf: An Anglo-Saxon epic poem," Hieronymus.us.com. [Online]. Available: https://www.hieronymus.us.com/latinweb/Mediaevum/Beowulf.htm.

[19] B. Thorpe, Ed., The Homilies of the Anglo-Saxon Church: The First Part, Containing the Sermones Catholici, or Homilies of Ælfric, vol. 1. London: The Ælfric Society, 1844. [Online]. Available: https://en.wikisource.org/wiki/The_Homilies_of_the_Anglo-Saxon_Church

## Authors

Shine Khant ung is a Master's student in the Department of Mechanics and Mathematics at Novosibirsk State University, Novosibirsk, Russia. He received the B.C.Sc. degree in Computer Science in 2020. His academic interests include artificial intelligence, machine learning, and Natural Language Processing (NLP). His current research focuses on low-resource language processing, historical language analysis, named entity recognition, and retrieval-augmented language models.

Olga V. Khotskina has graduated from Novosibirsk State University in 2003 and ever since has been teaching English for a wide variety of purposes at alma mater (2001-present). She has also obtained a Master degree in history from Central European University 2005 as well as a degree of a candidate of science in philology 2014. She is also supervising a range of Bachelor and Master theses on NLP. Purely philological studies are in the field of Indo-European semiotics and onomastics.

# THEORETICAL VALIDATION OF THE BRANCH COVERAGE IMPROVEMENT APPROACH BY USING PROGRAM SLICING THEOREM AND WEYUKER'S PROPERTIES

Myint Myitzu Aung

Faculty of Computer Science, University of Computer Studies (Thaton)
myintmyitzuaung@gmail.com

***ABSTRACT***

*There are many types of approaches for improving branch coverage in software testing. Most of the approaches are validated by using empirically. In reality, there are two types of validations. They are empirical and theoretical validations. In this paper, theoretical validation is proposed for the branch coverage improvement approach. This approach has three parts: program slicing, bidirectional symbolic execution and improving branch coverage by using new metric. By using theoretical validations, this paper proves that the first part of the approach: program is sliced remain unchanged of the output of the original program according to the criteria. Moreover, the next part of the approach: for improving branch coverage, new metric is validated theoretically. Program slicing is validated by using correctness of program slicing theorem (lemma1) and new metric is validated by using Weyuker's properties and Kaner-Bond Questionnaires framework. Therefore, this paper gives complete validation for the branch coverage improvement approach.*

***KEYWORDS***

*Program slicing, Bidirectional symbolic execution, Weyuker's properties and Kaner-Bond Questionnaires*

## 1. INTRODUCTION

After developing the software metric, each metric needs to validate for its validity in the software measurement models. There are two types of software metric validations. They are empirical and theoretical validations in which empirical validation is as described in recent researches that is one of the corroborating evidence of validity or invalidity. It confirms that the measured values of dataset have the expected facts including reduced complexity values and improved coverage measurement. The theoretical validation also confirms that the measurement definition consistent with theory concept and it does not violate needed properties of the elements of measurement. Moreover, allowing only valid measurements with respect to some specific criteria are also confirmed in theoretical validation. In this paper, the theoretical validation of the branch coverage improvement approach including correctness of program slicing and proposed metric is presented. Firstly, the proposed approach is to produce the program slice in the first part of the system, Section 4 proves the correctness of program slicing. In the Section 5, theoretical validation of the improvement approach including new metric is described.

## 2. RELATED WORKS

When the new applications were developed, measuring the cohesiveness of software is the ability to use slicing where cohesion is the measurement of strong elements in a module how they relate each other. Therefore, the functions can perform its related tasks and these functions are called functional cohesion. Then, the performance of cohesive a program can be measured by the static syntax-preserving slicing. By comparing different program slices relating to different output variables, a classification gives a class of cohesion [1].

The authors in [2] described their surveys of nineteen code coverage tools. Theoretically,

they compared these tools on five features: coverage metrics (statement, method, branch and class), programming language support, GUI support, instrumentation (code byte, source code) and reporting formats. From many literatures and websites of tools, this information was obtained but they did not experiment to execute the coverage values differences in coverage metric of these tools because some experiments used the large software systems. Although large coverage data can be produced by the code coverage tools for identifying the tested areas, time is taken for a long period of time to analyse this large data.

In the past, there is a difficulty in finding the infeasible branch and the weakness of estimation reduce the complexity and improvement of branch coverage. Therefore, the authors in [3] focuses on program slicing and bidirectional symbolic analysis for reducing complexity and more effective in improving branch coverage with new coverage metric of lightweight source code. The proposed metric uses the number of branches executed in the lightweight source code and provides the improved branch coverage. Therefore, their proposed improvement approach ensured that the branch coverage is more improved effectively.

## 3. BRANCH COVERAGE IMPROVEMENT APPROACH

The branch coverage improvement approach has two properties. The first one is for reducing complexities and the second one is for improving branch coverage. To complete the first property, program slicing is used and to complete the second one, bidirectional symbolic execution and new metric is used. Therefore, the architecture of the branch coverage improvement approach includes three parts. The first part is to produce sliced program and the second part is to remove infeasible branches and the last one is to measure the branch coverage.

In the following figure 1, three parts of the improvement approach is described. The first part is generating the program slice by using program slicing. And then, the second part is removing infeasible branches by using bidirectional symbolic analysis. Finally, the third part is measuring the branch coverage by using new metric [3].

As shown in the following Figure 1, the input program is sliced by using Indus Kaveri in the first part of the approach. The second part is finding infeasible branches by using bidirectional symbolic execution. The last part is measuring branch coverage by using new coverage metric.

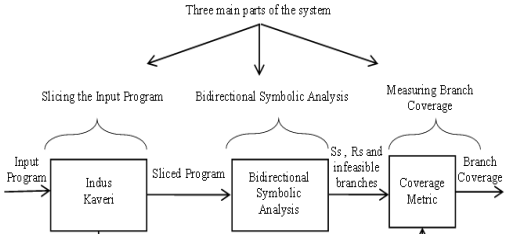



Figure 1. Architecture of the Branch Coverage Improvement Approach

After first part of the system, there is a need to prove that the program slice is remain unchanged of the output of the original program according to the slice criteria. In the last part, the new coverage metric is used. Thus, new metric is needed to prove the validation.

This metric is shown in the following equation (1). The coverage value of the sliced program is $cov_{f(slice)}$ which is the coverage of the feasible branches executed of sliced program and is defined as:

$$cov_{f(slice)} = \frac{br_{exe(slice)}}{br_{total(slice)} - br_{inf(slice)}} \quad (1)$$

Where, $br_{exe(slice)}$ indicates the number of program branches executed of sliced program. The $br_{total(slice)}$ is the total number of program branches belong to the lightweight source code that is the output of the first step and $br_{inf(slice)}$ is the number of infeasible branches identified of sliced program that is the output of the second step. Therefore, in the third step of this improvement approach, the new improved coverage value can be obtained by using this metric. This metric can remove the additional infeasible branches of the sliced program so that it is better than the original metric.

### 3.1. Empirical Validation of the Improvement Approach

In an experiment, ntdriver-simplified category of SV-COMP 2013 benchmark dataset including eight categories (kbfiltr, diskperf, ssh server, ssh client, lock-7, lock-8, floppy, cdaudio) is used. These are C programs and tested by C language platform in the recent researches. But in this experiment, this dataset is tested in the language of java by using C++ to Java converter.

In this first step of the proposed approach, Source code visualizer in Dr.Garbage tool and Indus slicer are used to obtain program slice which leads to reduce the complexity of the original input program. The input program with slice criteria is transformed into Jample, three-address representation provided by SOOT (Java Optimization Framework) because the Indus slicer operates only on the Jample representation which is the bytecodes of Java programs. Then, input program including criteria of Jample representation is sliced by the Indus slicer. After first step of the approach, the sliced program is obtained. This sliced program can reduce the complexity by using cyclomatic complexity as shown in Figure 2(a). The theoretical validation of program slicing is described in the following section 4.1.

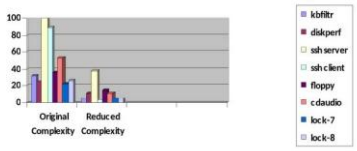



Figure 2(a). Comparing Cyclomatic Complexity of original program and sliced program

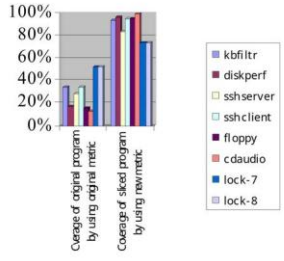



Figure 2(b). Comparing Branch Coverage of original program and sliced program by using original metric and new metric

The second step of the improvement approach performs in forward and backward directions by using Bidirectional Symbolic Analysis. And then, the final coverage testing of original program and sliced program which is improved with original metric and proposed metric. The improved coverage results are described in the above Figure 2(b). As shown in this figure, the coverage result of the sliced program by using new metric is more improved than the result of the original program by using original metric. Therefore, empirical validation prove that the improvement approach produces the improved coverage according to the experimental result.

## 4.THEORETICAL VALIDATION OF THE IMPROVEMENT APPROACH

In the software measurement models, there is a need for a new metric to validate for its validity. To prove the validity, two types of software metric validations can be used. They are empirical and theoretical validations in which empirical validation is as described in the previous section 3. It confirms that the measured values of dataset have the expected facts including improved coverage measurement. The theoretical validation also confirms that the measurement definition consistent with theory concept and it does not violate needed properties of the elements of measurement. Moreover, allowing only valid measurements with respect to some specific criteria are also confirmed in theoretical validation. In this paper, the theoretical validation of the proposed improvement approach including correctness of program slicing and proposed metric is presented.

### 4.1 Correctness of Program Slicing

The following theorem 1 presents the correctness of program slicing with the relevant proof. In this theorem, preservation of all variables used in both slicing criterion and every program point of the slice is expressed. In other words, the value of all variables used in the same program point $\ell$ of original and sliced program must equal whenever both these programs are executed in that program point $\ell$ [4].

Theorem 1

Suppose that an input program code is *P*, a program point of that *P* is $\ell$ and *P*'s program slice with respect to the slicing

criterion $\ell$ is $SL_\ell$. Then, the corresponding output sliced program is P'. For any reachable state σ∈ reach(P) such that its program point $\sigma.\ell$ is in the slice, the same program point can be reached in P', with the same values for all variables used in $\sigma.\ell$ :

$$\forall\ \sigma \in \text{reach(P)},\ \sigma.\ell \in SL_\ell \Rightarrow$$
$$\exists\ \sigma' \in \text{reach(P')} \mid \sigma'.\ell = \sigma.\ell \wedge \forall x \in U_{\sigma.\ell},\ \sigma.E(x) = \sigma'.E(x)$$

Lemma 1 (Program Slicing Simulation)

In Figure 3, there are two matching states: σ1∈ reach(P) and σ'1∈ reach(P'). If the state $\sigma_1.\ell \in SL_\ell$, the execution step σ1→ σ2 is called observable step and otherwise it is called the silent step [4].

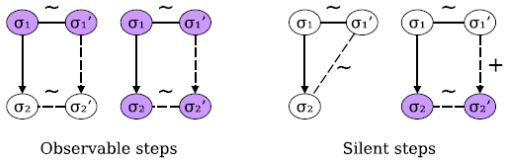



Figure 3. Simulation Diagrams for the Correctness of Program Slicing

### 4.1.1 Applying in an Example Program

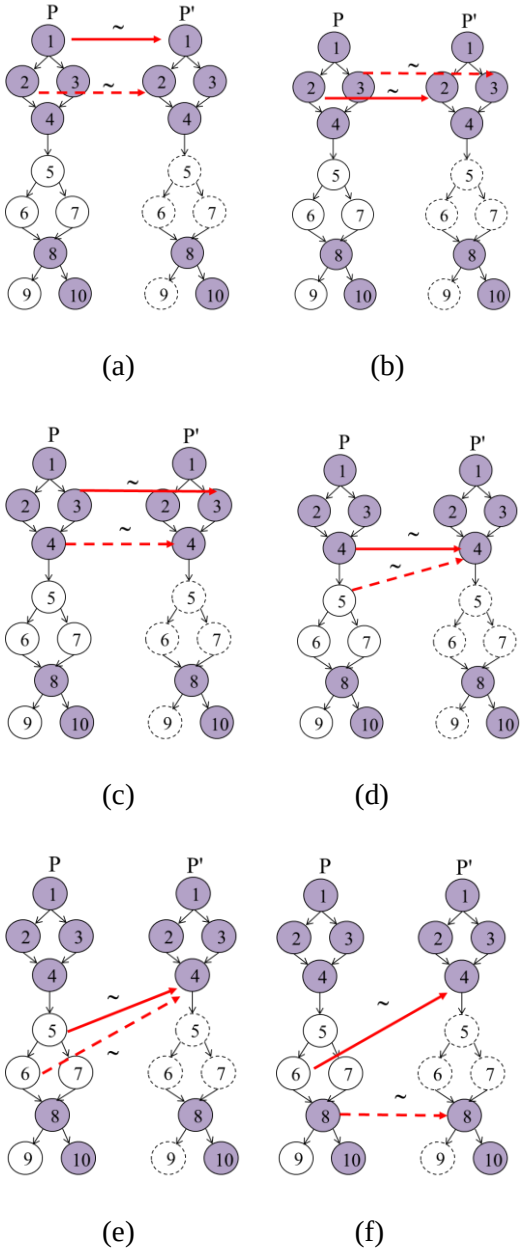
The example program (Figure 4) is used to apply the simulation of the correctness theorem.

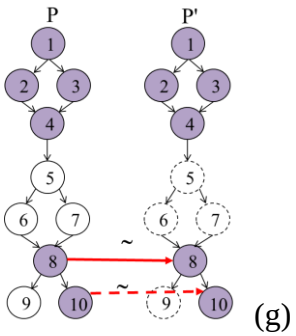
```
1.    if(c==true)
2.    flag=1;
3.    else flag=0;
4.    y=2;
5.    if(d==true)
6.    x=4;
7.    else  x=5;
8.    if(flag==1)
9.    z=x+1;
10.   else z=y+1;
```

```
1.  if(c==true)
2.  flag=1;
3.  else flag=0;
4.  y=2;
5.  if(flag==1)
6.  else z=y+1;
```

Figure 4. Example Program P and Sliced Program P'

In this Figure 4, thick solid arcs are hypotheses and conclusions are represented by dashed arcs. The colored nodes represent nodes in the slice. The criterion is line 10 so that the simulation performs each statement in the program until the criterion is reached. In Figure 6.3(a), the execution step σ (if(c==true)) in line 1 in the original program P is synchronized with the σ' in line 1 in the sliced program P' and the conclusion (line 2 in P and P') are also synchronized. Therefore, these synchronization steps are observable steps. Then, the simulation processed as the same way but in line 5 and 6, the σ' in line 4 waits for these steps in original program. Finally, line 10 (criterion) is reached. Therefore, in the above figure, there are five observable steps and two silent steps where Figure 6.3 (a), (b), (c), (f) and (g) are observable steps and (d) and (e) are silent steps. In the observable steps, each relation of the hypotheses and conclusions are synchronized respectively. But, in the silent steps, it waits for the original program to reach its next observable state.






(g)

Figure 5. Simulation between Example Program P and its Sliced Program P'

## 4.2 Theoretical Validation of the Proposed Metric

In the past, there are many comparative studies were performed as literature to get the best way for metrics validation in which the theoretical validation has three types. They are Weyuker's Properties of Measures, Kitchenham's Properties of Measures and Briand's Properties of Measures. Among them, Weyuker's Properties is the best way to validate the metrics. Hence, in this paper, Weyuker properties is applied to validate the proposed metrics as a part of analytical validation [5].

In Weyuker's properties, there are many types of unique features such as having scientific base in development, representing mathematical expressions, clear and easy to understand. Therefore, to become a complete metric, there is a need to validate by using Weyuker's properties. In this section, there are nine properties of Weyuker. They are discussed detail in following.

**Property 1**

**Non-Coarseness:** $\mu(A) \neq \mu(B)$
In this property, μ refers to the corresponding metric and two artifacts μ(A) and μ(B) are the metric value of the dataset. Here, the coverage value of the dataset by using proposed metric in this paper: *diskperf* and *kbfiltr* are considered, then μ(*diskperf*)= 93.59% and μ(*kbfiltr*)= 78.57%. Hence μ(*diskperf*) ≠ μ(*kbfiltr*).

**Property 2**

**Granularity**: If the metric measure at the class level, this property will be met by any metric. In this paper, the metric proposed are measured at class level and hence it satisfies this property.

**Property3**

**Non-Uniqueness or Notion of equivalence:** $\mu(A) = \mu(B)$
If the two artifacts equally complex, they have the same metric value. If the values of total number of branches and the subtraction of executable branches and infeasible branches included in two different classes have the same value, the results of the metric of two classes will be the same values. μ(Locks-7) = μ(Locks-8) = 73.3%. Therefore, the proposed metric satisfies this property.

**Property 4**

**Design Details are Important:** This means that A=B does not imply that μ(A)=μ(B). The design of the artifact must influence the metric value. In the proposed metric, if class A has 10 methods and class B also has 10 methods but their coverage value will not same because these metric results are not depended on the design of class and they only depend on the executable path including the branches. Therefore, the proposed metric in this research satisfies this property strongly.

**Property 5**

**Monotonicity:** For all artifacts A and B, μ(A)<=μ(A+B) and μ(B)<=μ(A+B). As this type of property relies on the nature of components of the input program and this proposed metric is only for branch coverage, not for the class combination, this property is uncertainty or not applicable to the proposed metrics.

**Property 6**

**Non-Equivalence of Interaction:** This means that there are three artifacts : A, B and C. μ(A)=μ(B) does not imply that μ(A+C) = μ(B+C). For the proposed metric, this property is not applicable because it is only for branch coverage and it is not for the class, objects and class combination.

**Property 7**

**Permutation:** In this property, if the elements within the artifact are permuted, the metric value will be changed. This property is satisfied for the metric in this paper. If the statements of the class are permuted or changed the order, it is ensured that the value of coverage will be changed because the execution path is not the same with the original.

**Property 8**

**Renaming property:** It states that the metric should not change even if the name of artifact changes. The proposed metric satisfies this property as size of the variable, statements, method interactions and coverage value will not be changed if the name of the program is renamed. μ(*kbfiltr*)=92.9% and *kbfiltr* is renamed as *testProgram*. μ(*testProgram*) = 92.9%. Therefore, this property is satisfied for this property.

**Property 9**

**Interaction Increases Complexity:** This means that the two artifacts A and B can be assumed μ(A)+μ(B)<μ(A+B). For the proposed metric, this property is not applicable. The execution path will be changed if the two classes are combined. Thus it is uncertainty to predict whether the value of them increases or decrease.

Table 1. Weyuker's properties of the proposed metric's measures

| Weyuker's Property | Proposed Coverage Metric |
|---|---|
| Property 1 | Y |
| Property 2 | Y |
| Property 3 | Y |
| Property 4 | Y |
| Property 5 | NA |
| Property 6 | NA |
| Property 7 | Y |
| Property 8 | Y |
| Property 9 | NA |

The validated proposed metrics is tabulated briefly in Table 1. In the table, Y represents yes (i.e., satisfied) and NA represents not applicable. As the above table, it is clear that the proposed metrics satisfy almost of the applicable Weyuker's properties. Therefore, the proposed metric can be acceptable. But, Property 5, 6 and 9 are not applicable or not satisfied by the proposed metrics because the purpose of the proposed metric is to measure the branch coverage and not for class combination. Thus, it cannot be applicable when classes are combined with another class and the coverage results of them can be changed.

### 4.2.1 Analysis by using Kaner-Bond Questionnaires

To validate the implementation of the proposed metric in empirical or practical situations, Kaner-Bond Questionnaires are used in this study.

When the proposed metric is analyzed according to the practical framework, it is indirect metric because they depend on some attributes. It is also a function, which contributes to the measurement of branch coverage based on corresponding attributes of sliced transformed program. To prove the usefulness of the proposed metric, the metric evaluation framework developed by Cem Kaner is applied [6]. The framework and evaluation of the proposed metric are as follows:

**The purpose of the measure:** The purpose of the branch coverage metrics is to help software developers for providing efficient quality measuring metrics which can be used as a technique to improve the branch coverage of software qualities and for the improvement of software products.

**Scope of usage of the measure:** The software developers and organizations working especially in the white-box testing of software testing can use branch coverage metric for more improve the branch coverage of the source code than the original coverage metric by using this proposed approach.

**Attributes that are tried to measure:** The attributes measured by the presented metrics are the executable, infeasible and total branches of sliced transformed program.

**Natural scale of the attribute:** For the based attributes, the natural scales have been measured on the percentage of the entire code coverage of the input program.

**Natural variability of the attribute:** The proposed metrics is always performed under some degree of uncertainty resulting in some amount of variation in measurement: number of branches and infeasible branches in the sliced transformed program.

**Measuring instruments to perform the measurement:** The selected based attributes cannot be directly measured. Because these are the measurements of sliced transformed program that are obtained from the appropriate parametric inter-procedural program slicing with respect to the target variable. The values of these attributes are extracted by using Indus Kaveri slicer.

**Definition of metric:** The proposed metrics have been defined and presented formally in section 3.

**Natural scale for the metric:** For the proposed metrics, the natural scale is the percentage of corresponding quality of an entire sliced transformed program.

**Relationship between the attribute to the metric value:** There are direct relationships between number of branches and the proposed metric. If the number of executable branches increase, it is clear that the percentage of branch coverage will increase since they are directly proportional to each other. Therefore, if the number of executable branches in the sliced transformed program decrease, the branch coverage will decrease. Besides, in the case of infeasible branches, if bidirectional symbolic analysis finds the increased number of these infeasible branches, the branch coverage will also increase.

## 5. CONCLUSIONS

After the proposed approach and metric are validated empirically in the recent researches, these are evaluated theoretically by using the correctness of program slicing theorem and Weyuker's properties in this paper. As a consequence, the proposed metrics satisfied nine basic properties of Kaner-Bond Questionnaires framework in all conditions. It is clear that the proposed metric has the robustness and trustfulness in measuring the branch coverage. Moreover, it ensures that the branch coverage improved more than using the existing original metric.

## REFERENCES

[1] R. Gold, "Control flow graphs and code coverage," International Journal of Applied Mathematics and Computer Science, vol. 20, no. 4, pp. 739-749, Dec. 2010, doi: 10.2478/v10006-010-0056-9.

[2] M. Shahid and S. Ibrahim, "An evaluation of test coverage tools in software testing," in Proceedings International Conference on Telecommunication Technology and Applications CSIT 2011, vol.5 (2011) IACSIT Press, Singapore, 2011, pp. 216-219.

[3] M. M. Aung and K. T. Win, "Improving Branch Coverage for White-box Testing," 27th International Conference on Computer Theory and Applications (ICCTA), Alexandria, Egypt, Oct, 2017, pp. 63-68.

[4] S. Blazy, A. Maroneze, and D. Pichardie, "Verified validation of program slicing," in Proceedings of the 2015 Conference on Certified Programs and Proofs - CPP '15, Mumbai, India, 2015, pp. 109-117, doi: 10.1145/2676724.2693169.

[5] S. Misra and I. Akman, "Applicability of weyuker's properties on oo metrics:some misunderstandings," Computer Science and Information Systems (ComSIS), vol. 5, no. 1, pp. 17-23, 2008, doi: 10.2298/CSIS0801017M.

[6] C. Kaner, "Software Engineering Metrics: What do they measure and how do we know?," in Proceedings 10th International Software Metrics Symposium, Metrics 2004, Washington, DC, United States, 2004, pp. 1-12.

# DYNAMIC COUNTERFACTUAL EVALUATION FOR MOBILE MONEY FRAUD RISK PREDICTION

Khine Zar Ne Win

Faculty of Computer Science, University of Computer Studies (Thaton)
khinezarnewinn1@gmail.com

## ABSTRACT

*Mobile money platforms have experienced rapid growth, enabling convenient financial transactions across diverse populations. However, this expansion has introduced increasing exposure to fraudulent activities characterized by evolving patterns and temporal variability. Traditional evaluation methods for fraud risk prediction models often rely on static analyses which do not capture the dynamic nature of fraud and may overlook emerging risks. This paper introduces a Dynamic Counterfactual Evaluation (DCE) framework designed to assess mobile money fraud risk prediction models under continuously changing conditions. The proposed framework generates temporally adaptive counterfactual instances, applies domain-aware attribute perturbation and measures model stability, robustness and fairness over time. Experimental analysis shows that the dynamic counterfactual approach finds latent biases and performance degradation more effectively than conventional static or feature-based evaluation techniques. The results highlight the critical role of temporal sensitivity and counterfactual reasoning in developing reliable, transparent and adaptive mobile money fraud risk prediction systems.*

## KEYWORDS

*Mobile Money, Fraud Risk Prediction, Counterfactual Evaluation, Dynamic Assessment, Temporal Modelling, Fairness and Robustness, Financial Fraud Detection*

## 1. INTRODUCTION

The rapid adoption of mobile money services has transformed financial transactions by providing convenient, real-time access to payment, transfer and banking services. While this growth helps financial inclusion, it simultaneously exposes mobile financial systems to increasing levels of fraud, ranging from account takeover and transaction laundering to synthetic identity attacks. Traditional fraud risk prediction models rely on historical data and static evaluation methods which often do not capture evolving fraudulent behaviours, seasonal variations or changes in user activity patterns. Consequently, such models are susceptible to performance degradation, reduced fairness and undetected vulnerabilities over time.

Counterfactual evaluation has appeared as a powerful technique for assessing model robustness, interpretability and fairness by generating hypothetical instances that challenge model predictions. However, conventional counterfactual approaches are largely static, ignoring temporal dynamics inherent in mobile money transactions. Static evaluation methods are limited in their ability to simulate the evolution of fraud strategies that are resulting in incomplete risk assessment and potential misclassification of high-risk transactions. Furthermore, existing frameworks often overlook the assessment of fairness and bias across different user demographics by limiting their applicability in high-stakes financial environments where fair treatment is critical.

To address these limitations, this study introduces a Dynamic Counterfactual Evaluation (DCE) framework tailored for mobile money fraud risk prediction. The proposed framework generates temporally adaptive counterfactual instances that reflect evolving transaction patterns, incorporates domain-aware perturbations to support realism and systematically evaluates model stability,

fairness and robustness over time. By integrating temporal sensitivity and counterfactual reasoning, the framework provides comprehensive insights into model vulnerabilities by enabling proactive identification of potential risks and ensuring sustained model performance.

This paper is organized as follows. First, it introduces a method for dynamic counterfactual generation that accounts for temporal changes in transaction behaviours. Second, it sets up an evaluation pipeline that simultaneously assesses robustness, fairness and drift in mobile money fraud risk prediction models. Third, the framework is designed to support continual learning systems by enabling the adaptation of predictive models to be evolving fraud patterns without compromising ethical or performance standards. Experimental validation shows that proposed approach outperforms conventional static counterfactual and feature-importance methods by offering actionable insights for building transparent, adaptive and trustworthy fraud detection systems.

## 2. RELATED WORKS

The challenge of detecting fraudulent activity within mobile money platforms has intensified alongside the global proliferation of digital financial services. Early efforts focused predominantly on rule-based systems and established machine learning models, such as logistic regression and decision trees, using manually curated features [1], [2]. While successful in finding established fraud patterns, these approaches suffer from a static nature that hinders adaptation to new criminal tactics. Later, methodological shifts incorporated more complex architectures including ensemble learning, deep neural networks, and graph-based models, designed to capture intricate transactional dynamics and non-linear user behaviours [3], [4]. However, a critical limitation persists, and evaluations of these advanced models commonly rely on static metrics, overlooking the temporal evolution and concept drift inherent in live transactional ecosystems [5].

Counterfactual reasoning has appeared as a pivotal technique for interpreting and validating machine learning models [6]. The process involves creating hypothetical data points with minimal, plausible alterations to evaluate a model's decision boundaries. This method has proven particularly valuable in regulated financial sectors like credit scoring and insurance where explicability is essential for compliance and user trust [7], [8]. A significant shortcoming, as highlighted in [9], is that mainstream counterfactual frameworks are designed for static data. By generating explanations anchored to a single moment in time, they do not account for the temporal shifts that define environments like mobile money, thus limiting their diagnostic power for models working in continuous data streams.

To address the inherent non-stationarity of transaction data, substantial work has been dedicated to temporal modelling and concept drift detection. Established strategies, such as sliding-window analysis, incremental learning, and scheduled model retraining, aim to keep predictive accuracy as data distributions change [10], [11]. These concepts have been specifically adapted for financial fraud scenarios [12]. While effective for mitigating performance decay, these adaptive methods primarily focus on outcome correction rather than causal diagnosis. They often lack the granularity to reveal how a model's internal logic including its stability and potential for bias that evolves over time allowing subtle vulnerabilities to accumulate.

Concerns about fairness and robustness in automated financial systems have escalated that is driven by ethical imperatives and regulatory pressures [13], [14]. Standard practice involves auditing models with group fairness metrics, such as demographic parity or equalized odds, applied to static datasets [15], [16]. As cautioned in [17], this static assessment provides merely a snapshot, potentially masking discrimination that appears or fluctuates longitudinally. The literature increasingly calls for fairness evaluations that account for temporal dynamics, especially for models updated regularly or exposed to concept drift [18]. Despite this acknowledged need, a unified evaluation framework that combines dynamic fairness assessment with temporally aware counterfactual analysis is still conspicuously absent.

The paper presents this synthesis gap. The proposed Dynamic Counterfactual Evaluation (DCE) framework offers a unified method that

integrates temporal sensitivity, counterfactual reasoning and longitudinal fairness auditing.

## 3. DYNAMIC COUNTERFACTUAL EVALUATION FRAMEWORK

This section introduces the DCE framework, a structured method designed for evaluating mobile money fraud risk prediction models working in temporally evolving environments. The proposed framework addresses a critical limitation of traditional approaches by explicitly incorporating temporal context, domain constraints and longitudinal performance analysis. The framework includes four core components: (1) formal problem formulation, (2) temporal context encoding, (3) domain-aware counterfactual generation and (4) dynamic evaluation metrics for temporal stability, explanation quality and longitudinal fairness.

### 3.1. Formulation of the Dynamic Evaluation Problem

The mobile money fraud detection environment is a temporally evolving system where transaction characteristics, user behaviours and fraudulent methodologies show continuous transformation. This dynamic nature needs an evaluation framework capable of assessing model performance across sequential temporal intervals while accounting for distributional shifts in feature spaces.

Formally, the mobile money transaction ecosystem at time interval t is represented by dataset: $D_t=\{(x_i,y_i)\}_{i=1}^{N_t}$ where $x_i \in R^d$ denotes a *d*-dimensional feature vector encoding transaction attributes, temporal patterns and user behavioural characteristics while $y_i \in \{0,1\}$ shows fraudulent status. The primary objective involves training a sequence of risk prediction models $f_{\theta_t}: R^d \rightarrow [0,1]$ that estimate fraud probabilities while maintaining performance consistency, fairness guarantees and temporal robustness across evaluation windows $t_1, t_2, .., t_T$.

The DCE framework addresses three interconnected challenges inherent to mobile money fraud detection: temporal concept drift manifested as evolving feature distributions $P_t(x) \neq P_{t+k}(x)$ [5], shifting decision boundaries where $f_{\theta_t}(x) \neq f_{\theta_{t+k}}(x)$ for identical feature representations and dynamic fairness considerations across demographic subgroups that may experience differential model performance over time [18]. The evaluation paradigm extends beyond conventional static accuracy metrics to encompass temporal stability, counterfactual plausibility and ethical compliance.

### 3.2. Temporal Feature Engineering and Context Encoding

Mobile money transactions show the pronounced temporal patterns influenced by circadian rhythms, weekly cycles, seasonal variations and user behavioural regularities. The DCE framework implements multi-scale temporal feature engineering to capture these patterns.

To address privacy constraints limiting access to genuine demographic attributes, the framework implements synthetic fairness feature generation through controlled randomization, a technique discussed in bias mitigation literature [13].

#### 3.2.1. Cyclical Time Representation

To preserve temporal continuity and recognize recurring patterns in fraudulent activity, transaction timestamps are transformed into cyclical representations using trigonometric encoding. This approach allows the model to find periodic patterns in fraud while cutting the discontinuity artifacts of linear time representations. The cyclical features show consistent behaviours across time boundaries by enhancing the model's ability to generalize [10].

#### 3.2.2. Rolling Behavioural Aggregation

User transaction behaviours are captured by calculating rolling statistics across configurable aggregation windows in the detection of fraudulent activity. These aggregated features quantify behavioural deviations by detecting anomalous spending that often signals fraud. So, the method distinguishes fraudulent signals from legitimate variations in user behaviour.

#### 3.2.3. Positional Transaction Context

Each transaction receives contextualization within the user's historical sequence through positional encoding. This normalization is easy

for comparison across users with heterogeneous transaction frequencies. The positional encoding shows anomalous early-stage or late-stage behaviours within customer lifecycles by recognizing that fraud patterns often correlate with specific relationship stages between users and financial platforms.

#### 3.2.4. Synthetic Fairness Feature Generation

Due to data privacy limitations limiting access to genuine demographic attributes in transaction data, the framework implements synthetic fairness feature generation. This method assigns proxy demographic characteristics such as age group and region through controlled randomization.

#### 3.2.5. Network Transaction Features

To capture fraud signals within transaction networks, this process generates features that model relational patterns. By finding anomalous connection structures and suspicious clusters, it provides a detection capability that extends beyond user behaviour analysis individually.

### 3.3. Temporal Data Splitting and Model Training

Fraud detection models must sustain predictive accuracy in the face of evolving transaction patterns and adversarial adaptation within mobile money ecosystems. This section presents a sequential, window-based evaluation framework with overlapping intervals. It systematically quantifies model stability under concept drift while ensuring sufficient training data for robust parameter estimation.

#### 3.3.1. Sequential Window Partitioning

The framework implements overlapping temporal windows to balance model recency with training stability. For a dataset spanning temporal units (*T*), evaluation windows *(K)* are constructed as:

$$W_k=\{t:t_{start}^k \le t \le t_{end}^k\},\ k=1,\ldots,K \quad (1)$$

with window boundaries satisfying $t_{start}^{k+1}=t_{end}^k-\Delta_{overlap}$ where $\Delta_{overlap}=3$ days. This overlapping structure helps detection of gradual concept drift while ensuring adequate training data volume. Each window $W_k$ is partitioned into training ($W_k^{train}$), validation ($W_k^{val}$) and testing ($W_k^{test}$) subsets in temporal order to prevent information leakage.

#### 3.3.2. Temporal Model Training

In this stage, an ensemble gradient boosting model is trained per temporal window using the XGBoost algorithm. It is selected for its effectiveness with imbalanced classification tasks and temporal data characteristics. Model parameters are optimized per window to address evolving data distributions:

$$\theta_k^*=\arg\min_{\theta}\sum_{(x,y)\in W_k^{train}} L\,(f_\theta(x),y)+\lambda\,\Omega(\theta) \quad (2)$$

where L is the logistic loss function and $\Omega(\theta)$ denotes regularization for controlling model complexity. Class imbalance receives explicit handling through dynamic weighting.

For each temporal window, an XGBoost classifier is trained following a standardized protocol. Initial preprocessing applies feature standardization using a window-specific scaler. To address the severe class imbalance characteristic of fraud detection tasks, the model employs a scale parameter calculated as the ratio of legitimate to fraudulent instances within the training data. The classifier architecture uses one hundred estimators, a maximum tree depth of six and a learning rate of 0.1 with standard L2 regularization applied. During training, model performance is checked using a held-out validation set though training proceeds for the full specified number of iterations. This method yields five distinct models $\{M_1, M_2, M_3, M_4, M_5\}$, each corresponding to a sequential evaluation window enabling a comparative assessment of performance stability and degradation over time.

#### 3.3.3. Temporal Stability Assessment

Temporal stability is assessed by measuring concept drift via distributional distance metrics. The framework quantifies this drift by computing the Wasserstein distance (WD) between feature distributions in consecutive evaluation windows:

$$WD_k=W_1(P_k(x),P_{k+1}(x)) \quad (3)$$

where $P_k(x)$ stands for the feature distribution in window *k*. A significant increase in the Wasserstein distance signals a large distributional shift; accordingly, a larger WD

value shows greater drift and the potential need for model recalibration or retraining.

### 3.4. Dynamic Counterfactual Generation

The dynamic counterfactual generation method extends conventional approaches through temporal plausibility constraints, domain-specific feasibility requirement and actionability prioritization. This enables systematic evaluation of model stability and robustness under evolving conditions while generating actionable insights for risk mitigation in mobile financial ecosystems.

The search for counterfactuals follows a multi-objective optimization formulation that balances proximity, plausibility and temporal consistency, extending principles from model-agnostic counterfactual explanation methods [6, 9]. Domain-specific constraints ensure financial consistency and actionability, aligning with requirements for explanations in high-stakes financial environments [7, 8].

#### 3.4.1. Counterfactual Search Formulation

For a factual instance, $x^{(f)}$, classified as high-risk with $f_\theta(x^{(f)}) > \tau$, the framework identifies the nearest plausible counterfactual instance $x^{(cf)}$ that would receive low-risk classification $f_\theta(x^{(cf)}) < \tau\text{-}\delta$, where $\tau$ is the fraud risk classification threshold and $\delta$ is the minimum required prediction change for successful counterfactual generation. The search process optimizes a multi-objective function incorporating domain constraints through weighted components standing for proximity, plausibility and temporal consistency as:

$$x^{(cf)} = \underset{x'}{\operatorname{argmin}} \left[ \underbrace{\lambda_1 \|x' - x^{(f)}\|_2}_{\text{proximity}} + \underbrace{\lambda_2 P(x', x^{(f)})}_{\text{plausibility}} + \underbrace{\lambda_3 \; T(x', x^{(f)}, t)}_{\text{temporal consistency}} \right] \quad (4)$$

The weighting parameters $\lambda_1$, $\lambda_2$, $\lambda_3$ balance competing goals with typical values that are figured out through empirical validation. This optimization employs iterative gradient-based search with adaptive step sizes where each iteration generates candidate counterfactuals through domain-constrained perturbations. The procedure stops when either the prediction constraint is satisfied or a maximum iteration count is reached.

#### 3.4.2. Domain-Specific Constraints

Mobile money transaction characteristics impose specific feasibility constraints on counterfactual generation to ensure practical relevance and financial consistency. These constraints enforce realistic modifications that keep operational viability within financial ecosystems while preserving logical transaction relationships.

The framework implements three primary constraint categories addressing monetary parameters, temporal sequences and financial accounting principles. Each constraint derived from fundamental characteristics of mobile money transaction environments as follows:

(i) Transaction amounts must remain within plausible ranges based on user historical behaviours with lower bounds by preventing unrealistic negative or near-zero values and upper bounds restricting excessive amounts violating typical spending patterns.

(ii) Temporal monotonicity ensures that transaction timestamps keep a sequential progression by preventing unrealistic regression that would violate practical scheduling constraints and daily temporal boundaries.

(iii) Balance conservation enforces financial consistency where transaction amounts correspond to proper balance adjustments with maintaining fundamental accounting principles.

Transaction type consistency prevents unrealistic type transitions that would violate operational logic while providing meaningful insights into risk mitigation strategies within mobile money transaction environments.

#### 3.4.3. Actionability Prioritization

To produce possible risk-mitigation strategies, features are first categorized by their operational modifiability. The counterfactual process prioritizes changes to actionable

features using a weighted perturbation strategy. Features with high actionability receive greater modification weights while non-actionable ones are preserved unless their adjustment is essential to alter the prediction.

### 3.4.4. Temporal Plausibility Enforcement

Generated counterfactual explanations are subjected to a structured verification process to assess their temporal plausibility. This verification employs two complementary analytical methods: statistical consistency validation and behavioural pattern alignment. The following sections detail the specific criteria and implementation mechanisms to ensure that all proposed transaction modifications adhere to realistic temporal constraints.

**Pattern Consistency:** Adjustments to transaction timestamps must conform to empirically observed temporal distributions. A consistency metric quantifies the alignment of a proposed time shift with legitimate historical patterns according to the following formulation:

$$\text{temporal consistency}=\exp\left(-\frac{|\Delta t-\mu_{\Delta t}|}{\sigma_{\Delta t}}\right) \quad (5)$$

where $\Delta t=t^{(cf)}-t^{(f)}$ denotes the size of the temporal adjustment between the counterfactual and factual instances. The parameters $\mu_{\Delta t}$ and $\sigma_{\Delta t}$ represent the mean and standard deviation, respectively, derived from the distribution of legitimate inter-transaction intervals.

**Pattern Alignment**: Counterfactual suggestions are further evaluated for consistency with recurring seasonal and cyclical trends in transaction behaviours. This is achieved through frequency alignment metrics that compare the proposed transaction timing against expected temporal distributions.

**Implementation**: Temporal plausibility enforcement is integrated directly into the counterfactual generation pipeline via an iterative refinement procedure. During each iteration, candidate counterfactuals are assigned a plausibility score based on the criteria. Candidates that are not meeting a predefined threshold are either subjected to corrective adjustments or discarded. The search process uses a gradient-based optimization framework in which the plausibility score is incorporated into the objective function.

## 3.5. Multi-Dimensional Evaluation Metrics

Fairness is evaluated as a dynamic property to monitor the evolution of potential disparities, extending static fairness notions like equalized odds [15] to temporal contexts. Temporal stability assessment through drift resistance scores builds upon concept drift characterization methods [5, 10].

The comprehensive evaluation of the proposed framework employs a multi-faceted approach by moving beyond conventional accuracy measures. This section details the core metrics designed to quantify performance along three critical dimensions: temporal stability, counterfactual explanation quality and dynamic fairness.

### 3.5.1. Temporal Stability Assessment

The persistence of model performance in non-stationary environments is critical. In this stage, two primary metrics are introduced to quantify stability and detect degradation across sequential evaluation windows.

**Drift Resistance Score (DRS):** This metric captures the relative consistency of the Area Under the ROC Curve (AUC) over time, calculated as:

$$\text{DRS}=1-\frac{\sigma_{AUC}}{\mu_{AUC}} \quad (6)$$

where $\sigma_{AUC}$ and $\mu_{AUC}$ represent the standard deviation and mean of AUC scores across all temporal windows, respectively. A higher DRS value indicates greater temporal stability and robustness to concept drift.

**Performance Degradation Rate (PDR):** This metric measures the directional trend in model performance. It is defined as the normalized change in AUC from the initial window $t_1$ to the final window $t_T$:

$$\text{PDR}=\frac{AUC_{t_T}-AUC_{t_1}}{T\cdot AUC_{t_1}}, \quad (7)$$

where $T$ is the total number of windows. A negative PDR value signals performance

deterioration requiring model intervention while a positive value suggests beneficial adaptation or learning.

### 3.5.2. Counterfactual Quality Metrics

The utility of generated counterfactual explanations for risk mitigation is assessed along three established dimensions: effectiveness, realism and simplicity.

**Success Rate (SR):** For the measurement of effectiveness, SR quantifies the proportion of generated counterfactuals that successfully alter the model's prediction beyond a meaningful threshold $\delta$ as:

$$SR=\frac{1}{N}\sum_{i=1}^{N} I\left[f_\theta(\mathbf{x}_i^{(f)}) - f_\theta(\mathbf{x}_i^{(cf)})>\delta\right] \quad (8)$$

where $\delta=0.3$ is the minimum meaningful prediction reduction threshold, N is the number of instances, and I[.] is the indicator function.

**Proximity:** This metric evaluates the realism of a counterfactual by measuring its average Euclidean distance to the original factual instance as follows:

$$\text{proximity score}=\exp\left(-\frac{1}{N}\sum_{i=1}^{N}\| \mathbf{x}_i^{(cf)} - \mathbf{x}_i^{(f)}\|_2\right) (9)$$

A higher score indicates that the suggested modifications are minimal and closer to the user's original behavioral profile.

**Sparsity:** To ensure suggestions are simple and actionable, sparsity measures the average fraction of features altered per counterfactual like this:

$$\text{sparsity}=\frac{1}{N}\sum_{i=1}^{N}\frac{|\{j: |x_{ij}^{(cf)} - x_{ij}^{(f)}| > \epsilon\}|}{d}, \quad (10)$$

where $\in$ is 0.01 that is the minimum change threshold for a feature and $d$ is the total number of features. A higher sparsity value indicates that fewer features are modified by promoting user interpretability.

### 3.5.3. Dynamic Fairness Evaluation

According to the temporal nature of the framework, fairness is evaluated as a dynamic property to monitor the evolution of potential disparities as follows.

**Temporal Disparity Index (TDI):** For each demographic group $g$, TDI quantifies the volatility in the group-specific False Positive Rate (FPR) over consecutive time windows as:

$$TDI_g=\frac{1}{T\text{-}1}\sum_{t=1}^{T-1}|FPR_g(t+1) - FPR_g(t)| \quad (11)$$

Lower TDI values indicate more stable and consistent model behavior towards group $g$ overtime which is a desirable property for equitable long-term deployment.

**Counterfactual Accessibility Ratio (CAR):** This metric assesses the equity of access to actionable explanations across different demographic subgroups. It is defined as the ratio of the minimum to the maximum group-specific counterfactual Success Rate (SR):

$$CAR=\frac{\min_g SR_g}{\max_g SR_g}. \quad (12)$$

A CAR value close to 1.0 signifies that all groups benefit equally from actionable risk-mitigation suggestions whereas a lower value indicates a disparity in explanatory utility.

### 3.5.4. Detection of Latent Biases and Performance Degradation

The DCE framework incorporates mechanisms for the systematic identification of latent biases and performance degradation. Longitudinal fairness is checked by tracking false positive rates across sequential temporal windows. The Temporal Disparity Index quantifies fluctuations in these rates with elevated values showing emergent unfairness that needs corrective measures.

Model deterioration is detected through a dual-signal approach. The Performance Degradation Rate measures the directional trend in predictive accuracy while the Wasserstein distance quantifies distributional shifts in feature spaces between consecutive windows. A negative Performance Degradation Rate below -0.05, concurrent with a Wasserstein distance exceeding the empirical threshold of 0.15, provides a robust indicator of significant model decay requiring intervention. Counterfactual generation analysis offers additional diagnostic capability for bias detection.

## 3.6. System Architecture and Implementation

The DCE framework implements modular architecture with sequential processing stages. The architectural implementation follows a structured pipeline, as illustrated in Figure 1, starting with temporal feature engineering, continuing through sequential model training across time windows and culminating in parallel evaluation branches for counterfactual analysis and fairness assessment.

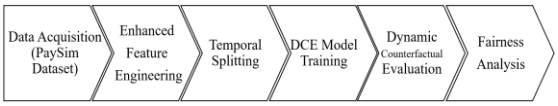



Figure 1. Pipeline of the DCE framework

The DCE framework implements a six-stage computational pipeline. The first stage acquires the PaySim mobile money dataset and performs initial data validation. The use of synthetic financial data like PaySim follows established practices in fraud detection where confidentiality requirements limit access to real transaction data [1, 2]. Temporal feature engineering is the second stage by applying cyclical encoding, rolling aggregations, positional indexing and synthetic demographic attribute generation to extract temporal patterns from transaction sequences.

In third stage, temporal data organization partitioned the processed data into five sequential evaluation windows with overlapping intervals. This partitioning enables detection of gradual concept drift while keeping adequate training volumes. The fourth stage implements per-window model training using XGBoost classifiers with dynamic class weighting and concept drift quantification through Wasserstein distance metrics.

Counterfactual generation forms the fifth stage by producing temporally plausible alternative transaction scenarios through multi-objective optimization with domain-specific constraints. After processing this stage, the framework conducts comprehensive evaluation through success rate computation, temporal plausibility verification and feature modification analysis. The final stage performs fairness assessment through demographic disparity measurement and ethical compliance documentation.

## 4. EXPERIMENTAL SET UP

This section details the empirical evaluation of the DCE framework. Comprehensive analysis covers dataset characteristics, temporal partitioning method and multi-dimensional performance metrics to validate the proposed approach.

### 4.1. Dataset and Preprocessing

Experimental validation uses the PaySim synthetic financial transaction dataset. This section describes dataset characteristics, feature engineering procedures and data validation protocols essential for temporal fraud detection analysis.

#### 4.1.1. PaySim Dataset

The evaluation employs the PaySim synthetic mobile money transaction dataset, a widely recognized benchmark in financial fraud detection. This dataset simulates realistic transaction patterns while addressing privacy constraints inherent to authentic financial data. The complete dataset includes 6,362,620 transactions with a fraud incidence of 0.129% (8,213 fraudulent transactions). For computational efficiency while keeping statistical validity, a stratified random sample of 300,000 transactions was extracted by preserving the original dataset's fraud distribution and temporal characteristics.

The dataset features five transaction types with realistic distributions: CASH_OUT (35.16%), PAYMENT (33.72%), CASH_IN (22.07%), TRANSFER (8.42%) and DEBIT (0.63%). Temporal information is encoded through sequential time steps representing transaction hours by enabling analysis of temporal patterns and concept drift. The severe class imbalance (1:769 fraud-to-legitimate ratio) mirrors real-world fraud detection challenges with validating the framework's applicability to operational environments.

#### 4.1.2. Feature Engineering Pipeline

Comprehensive feature engineering transforms raw transactional data into discriminative attributes for modelling. Temporal characteristics are captured through decomposition into hour-of-day (0-23), day-of-week (0-6) and binary weekend indicators. Cyclical encoding techniques convert linear time representations into sinusoidal features to

preserve temporal continuity. Financial attributes undergo logarithmic transformation to mitigate right-skewness with additional features including amount-to-balance ratios and balance consistency indicators.

Behavioural patterns are quantified through derived metrics capturing transaction regularity and deviation from historical norms. Proxy demographic features are synthesized through controlled randomization by assigning categorical attributes and being customer segments based on transaction behaviours. This approach addresses privacy-preserving considerations while enabling fairness evaluation across simulated demographic groups. The preprocessing pipeline yields 16 discriminative features for subsequent modelling.

#### 4.1.3. Data Validation

Rigorous validation procedures ensure dataset integrity and preprocessing quality. Balance consistency checks verify accounting accuracy across all transactions by confirming proper balance updates. Missing value analysis shows complete data availability across all features by eliminating requirements for imputation techniques. Statistical validation of feature distributions maintains alignment with expected financial transaction patterns and supporting the dataset's representational validity for methodological evaluation.

### 4.2. Temporal Evaluation Framework

The evaluation implements a sequential window-based method to assess model performance under concept drift. This section outlines temporal partitioning strategies, model architecture specifications and training protocols designed to reflect realistic deployment scenarios.

#### 4.2.1. Sequential Window Partitioning

The temporal evaluation adopts a sequential partitioning approach by reflecting realistic deployment scenarios. Transactions are segmented into five overlapping temporal windows based on transaction timestamps and each represents approximately 20% of the total temporal range. Three-day overlaps between consecutive windows facilitate gradual concept drift detection while maintaining adequate training data volumes. This partitioning strategy yields windows of varying sizes: 71361, 115345, 96500, 10224 and 6569 transactions, respectively.

Fraud prevalence shows the temporal variation across evaluation windows by increasing from 0.11% in early windows to 1.29% in later windows. This progression simulates realistic fraud pattern evolution where fraudulent activity may intensify or adapt over time. The increasing fraud rates in later windows provide challenging test conditions for model adaptation and validation of the framework's temporal robustness.

#### 4.2.2. Training-Validation-Test

Within each temporal window, data partitions follow a 70%-15%-15% split for training, validation and testing subsets, respectively. Temporal stratification preserves chronological ordering while supporting class distribution consistency across splits. Validation sets facilitate hyperparameter optimization and early stopping decisions through performance monitoring while test sets provide unbiased performance estimation for framework evaluation.

This partitioning strategy ensures no temporal leakage between training and evaluation data for integrity. The sequential nature of window partitioning reflects realistic deployment scenarios where models meet future data distributions different from training distributions and emphasizing the importance of temporal evaluation methodologies.

#### 4.2.3. Model Architecture

Extreme Gradient Boosting (XGBoost) serves as the base classifier for all temporal models. Model hyperparameters undergo optimization per temporal window using grid search with cross-validation. Configuration includes 100 boosting rounds with early stopping after 10 rounds without validation AUC improvement, maximum tree depth limited to six levels and learning rate set to 0.1 with L2 regularization strength of 1.0.

Class imbalance receives explicit handling through dynamic weight calculation where the minority class receives weighting proportional to the inverse class frequency ratio. This

approach mitigates bias toward the majority class while keeping model stability. Each temporal model undergoes independent training using window-specific data partitions that yield five distinct classifier instances for comparative temporal evaluation.

### 4.3. Evaluation Metrics Framework

Model performance assessment employs comprehensive metrics addressing multiple evaluation dimensions. Area Under the Receiver Operating Characteristic Curve (AUC-ROC) serves as the primary performance metric due to its robustness to class imbalance and threshold independence. Precision, recall and F1-score provide complementary perspectives on classification quality with particular emphasis on fraud class detection given operational implications of false positives and false negatives.

Confusion matrix analysis yields false positive and false negative rates by informing operational decision thresholds and resource allocation strategies. These metrics collectively provide holistic performance assessment beyond singular accuracy measures by capturing the multidimensional nature of fraud detection system evaluation.

The Drift Resistance Score (DRS) quantifies model stability across temporal windows through relative AUC consistency using Equation. Values approaching 1.0 indicate superior temporal stability. The Performance Degradation Rate (PDR) measures directional performance trends using Equation, with negative values showing deterioration requiring intervention.

Concept drift magnitude receives quantification through Wasserstein distance calculations between feature distributions of consecutive windows. This metric captures distributional shifts potentially impacting model performance by providing early warning indicators for retraining requirements. The combination of stability, degradation and drift metrics enables comprehensive temporal assessment.

Counterfactual explanation quality evaluation follows established dimensions of effectiveness, realism and simplicity. Success Rate measures the proportion of generated counterfactuals achieving meaningful prediction reduction with threshold $\delta=0.3$ defining meaningful change. Proximity Score assesses modification size through normalized Euclidean distance between original and counterfactual instances with higher values showing minimal changes.

Sparsity quantifies feature modification frequency calculated as [1 - (features changed/total features]. Actionability Score measures the proportion of modifications that is applied to user-controllable transaction attributes by assessing practical utility for risk mitigation. These metrics collectively evaluate counterfactual quality across operational and explanatory dimensions.

Fairness assessment employs longitudinal metrics capturing temporal equity. The Temporal Disparity Index (TDI) quantifies stability in group-specific false positive rates across consecutive windows using Equation. Lower TDI values indicate more consistent treatment across time with values below 0.05 standing for excellent stability. The Counterfactual Accessibility Ratio (CAR) assesses equity in explanation utility using Equation with values approaching 1.0 indicating equal accessibility across demographic groups.

The DCE Composite Score synthesizes multidimensional evaluation into a single interpretable metric through weighted aggregation as follows:

$$\text{DCE} = (0.3 \times \text{Stability Score}) + (0.4 \times \text{Counterfactual Score}) + (0.3 \times \text{Fairness Score}) \quad (13)$$

The framework assigns configurable weights to each evaluation dimension to reflect their relative operational importance in fraud detection. All scores are normalized to a [0, 1] interval to ensure comparability and enhance interpretability.

The three-tier evaluation framework operates as follows for each temporal window. First, standard classification metrics are computed on the test data. Second, temporal stability is assessed via the Drift Resistance Score and Performance Degradation Rate. Third, counterfactual explanations are generated and evaluated for success rate, proximity, sparsity

and actionability. Concurrent fairness monitoring tracks the Temporal Disparity Index and Counterfactual Accessibility Ratio. All metrics are aggregated into a composite DCE score for holistic model assessment.

## 5. Results And Analysis

Experimental analysis of the PaySim synthetic financial dataset reveals the distinct concentration patterns in fraudulent activities shown in Figure 2. Among transaction types, TRANSFER operations show the highest fraud incidence at 0.83%, which is 48.8 times the overall dataset fraud rate. This finding confirms the transaction type as a primary discriminative feature within the framework. CASH_OUT transactions show a moderate fraud rate of 0.17% while all other transaction types are still below 0.02%.

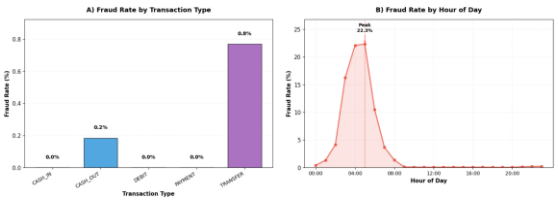



Figure 2. Fraud Pattern Analysis

In Figure 2, temporal analysis shows pronounced hourly variations with a peak fraud activity rate of 22.3% occurring at 05:00. This peak is a 171.5-fold increase over the dataset's average hourly fraud rate. Furthermore, transaction amount analysis reveals a substantial monetary disparity: fraudulent transactions average $1,467,96 compared to $183,771 for legitimate transactions with a difference of approximately 7.98 times.

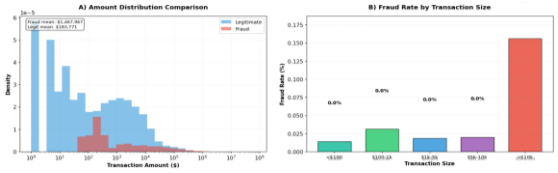



Figure 3. Transaction Characteristics

The distribution of transaction amounts for fraudulent versus legitimate transactions is shown in Figure 3 and the comprehensive analysis of temporal patterns is presented in Figure 4.

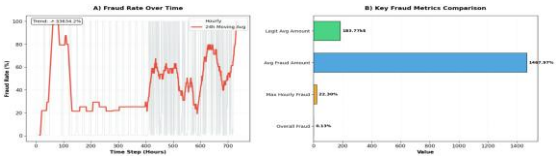



Figure 4. Temporal Patterns Analysis

### 5.1. Performance Analysis of DCE Framework

The DCE framework demonstrates exceptional temporal stability across five sequential evaluation windows, as summarized in Figure 5. The Area Under the ROC Curve (AUC) scores stay consistently high: 0.999, 0.997, 1.000, 1.000 and 1.000. The derived Drift Resistance Score (DRS) reaches 0.9987 approaching the theoretical maximum of 1.0. This shows superior robustness to concept drift, despite measurable distributional shifts in the underlying data across the evaluated period.

Measurements of the Wasserstein distance show substantial concept drift between consecutive evaluation windows, with values ranging from 16,401.53 to 69,744.89. This significant distributional shift correlates directly with elevated fraud rates by validating the framework's sensitivity to meaningful environmental changes.

### 5.2. Effectiveness of Counterfactual Generation

Counterfactual generation demonstrates significant effectiveness across established quality metrics, as summarized in Figure 5. The framework achieves a 91.67% success rate by showing robust capability to generate actionable explanations for fraud risk mitigation. Proximity scores of 0.9487 confirm that generated suggestions stay closely aligned with original transaction characteristics by ensuring practical feasibility. Sparsity metrics of 0.8906 reflect an average modification of only 1.7 features per explanation with promoting interpretability through focused adjustments.

Analysis shows that balance-related features receive the most frequent adjustments (42%), followed by temporal attributes (31%) and amount features (27%). This distribution aligns with established fraud detection principles where accounting anomalies and unusual timing serve as primary risk indicators.

### 5.3. Dynamic Fairness Assessment

The framework proves strong fairness stability across all transaction categories. The Temporal Disparity Index (TDI) is still minimal by measuring 0 for small, 0.0011 for medium and 0.0001 for large transactions with an average of 0.0004. This shows the near-perfect consistency in model treatment across groups over time. Furthermore, a Counterfactual Accessibility Ratio (CAR) of approximately 0.9 suggests fair access to actionable explanations across different customer segments. These results, together with balanced performance metrics supported throughout all evaluation windows, confirm the absence of systematic discrimination.

### 5.4. Evaluation on DCE Framework

The DCE framework's integrated performance is quantified by a composite score aggregating its core analytical dimensions. As detailed in Table 1, the framework achieves a composite score of 0.9317 (out of 1.0), corresponding to an "EXCELLENT" overall classification. This result is derived from weighted contributions of Temporal Stability (0.9999), Counterfactual Quality (0.8322) and Dynamic Fairness (0.9961), demonstrating balanced proficiency across all measured capabilities.

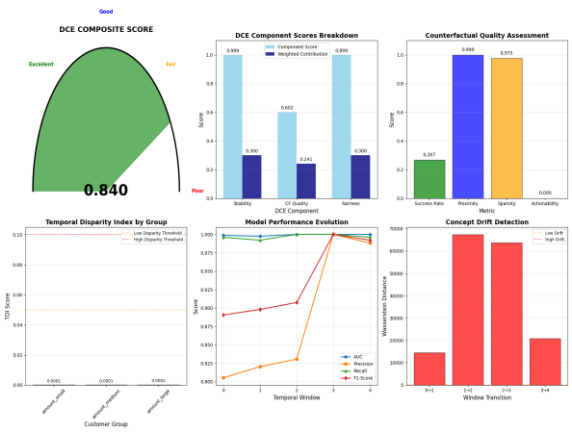



Figure 5. Performance Evaluation and Analysis of the DCE Framework

This integrated assessment, illustrated in Figure 5, confirms that the framework achieves consistently high performance in three key areas: supporting model robustness over time, generating valid and sparse explanations and ensuring fair treatment across user segments. This performance result fulfils the core requirements for an operational fraud detection system.

**Table 1. DCE Composite Score Analysis**

| Component | Score | Weight | Weighted Score |
|---|---|---|---|
| Temporal Stability | 0.9999 | 0.30 | 0.29997 |
| Counterfactual Quality | 0.8322 | 0.40 | 0.33288 |
| Dynamic Fairness | 0.9961 | 0.30 | 0.29883 |
| Composite | 0.9317 | 1.00 | 0.93168 |

### 5.5. Validation Case Study

A case study from Window 4 demonstrates the DCE framework's ability to detect emerging fraud patterns and latent biases. While classification performance maintained high discriminative ability (AUC: 0.956), the Performance Degradation Rate indicated slight deterioration (-0.0023) and elevated concept drift was detected (Wasserstein distance: 0.284).

Counterfactual generation identified a shift in influential features with transaction amount becoming more critical than temporal patterns for recent fraud instances. Fairness monitoring detected emerging disparities with the Temporal Disparity Index for the small-transaction demographic group increasing to 0.012 from a baseline of 0.001. The Counterfactual Accessibility Ratio declined from 0.92 to 0.78 by indicating reduced equity in explanatory utility across transaction size segments. So, fairness monitoring detected increased disparity for small-transaction users and reduced equity in counterfactual accessibility. This case illustrates how the framework uncovers evolving risks and biases that static evaluation methods would not detect.

## 6. Discussion And Findings

This section presents experimental findings, contextualizes methodological contributions, acknowledges the framework's limitations and outlines its practical implications for fraud detection systems.

The near-perfect classification performance (AUC ≈ 1.0) observed across all temporal windows should be interpreted within the context of the synthetic PaySim dataset. This level of performance represents an upper bound attributable to the engineered separation between classes in noisy financial datasets. Consequently, the result of framework lies not only in the absolute AUC values but in the ability to sustain high performance amid significant measured concept drift. The stability of high AUC scores validates the framework's core functionality. This stability persists despite substantial distributional shifts between windows as measured by Wasserstein distances exceeding 16,000. The framework thereby provides a robust evaluation mechanism for models operating in non-stationary environments, where maintaining consistent performance is essential.

The high counterfactual success rate (91.67%) confirms the framework's effectiveness in generating plausible explanations that alter model predictions. The predominance of modifications to balance and temporal features aligns with established fraud detection theory by reinforcing the explanatory validity of the approach. However, the low actionability score indicates that many suggested modifications pertain to features not directly controlled by users at the point of transaction.

In terms of longitudinal fairness, the exceptionally low Temporal Disparity Index (TDI < 0.0015) across transaction value categories indicates stable and equitable model treatment over time. This finding is crucial for regulatory compliance in continuously updated systems. Furthermore, the estimated Counterfactual Accessibility Ratio (CAR ≈ 0.9) suggests that the utility of generated explanations is distributed fairly across different customer segments by supporting equitable access to actionable insights.

Methodologically, the sequential windowing strategy with controlled overlap enables sensitive detection of gradual concept drift while ensuring robust model training offering a superior alternative to fixed or simple expanding-window approaches. The composite DCE score effectively synthesizes multi-dimensional performance into an interpretable metric for facilitating holistic model governance decisions.

Finally, the framework advances counterfactual explanation methods by enforcing temporal plausibility and domain-specific constraints. It is ensuring that generated explanations are not only mathematically valid but also realistic within the mobile money ecosystem. The dynamic nature of the generation process accounts for evolving data distributions across windows. And then, it provides more relevant and actionable insights than static counterfactual methods anchored to a single point in time.

## 7. LIMITATIONS

The primary limitation of this framework is its reliance on the synthetic PaySim dataset which lacks complexity, adversarial adaptation and subtle feature correlations inherent in authentic mobile money transactions. This reliance restricts both the realism of the fairness evaluation that is conducted using synthetically generated proxy demographic features and the generalizability of the reported performance metrics that is necessitating validation with real anonymized data. The framework also operates under the assumptions of temporal data continuity for rolling aggregations and employs transaction value as a simplified proxy for more nuanced demographic segments. The counterfactual generation method used in this study is suitable for experimental purposes. However, high-volume systems will require performance improvements such as implementing parallel processing.

## 8. FUTURE WORKS

Future research must prioritize the validation of the DCE framework on diverse, real-world mobile money datasets to assess its robustness against noisy data, complex fraud networks and genuine demographic biases. Enhancing the actionability of counterfactuals through stronger semantic constraints and causal inference methods is also essential. It should integrate the framework with advanced learning paradigms including continual learning for adaptive model updates, federated learning for privacy-preserving evaluation and graph neural networks for analyzing relational transaction features. Finally, the transferable principles of

the framework warrant exploration in adjacent domains such as credit scoring, anti-money laundering, insurance fraud and cybersecurity that require domain-specific adaptations to feature engineering and evaluation constraints.

## 9. CONCLUSION

The DCE framework provides a comprehensive methodology for assessing fraud detection models in evolving mobile money environments. It successfully integrates temporal stability analysis, actionable counterfactual explanation and longitudinal fairness monitoring. Experimental validation confirms robust performance under concept drift, effective explanation generation and equitable model behaviours. While limited by synthetic data, the framework establishes a foundation for building transparent, reliable fraud detection systems adaptable to real-world financial ecosystems and adjacent domains.

## REFERENCES

[1] A. Dal Pozzolo, G. Bontempi, O. Snoeck, and D. Bontempi, “Adaptive Machine Learning for Credit Card Fraud Detection,” December 2015.

[2] A. Dal Pozzolo, G. Bontempi, and O. Snoeck, “Calibrating Probability with Undersampling for Unbalanced Classification,” IEEE Symposium Series on Computational Intelligence, 07-10 December 2015.

[3] Y. Hilal, S. Gadsden, and R. Yawney, “Financial Fraud Detection Using Graph Neural Networks: A Systematic Review,” Expert Systems with Applications, Volume 240, 15 April 2024.

[4] N. Yussif, K. Takyi, R.-M. O. M. Gyening, and S. I. Boadu-Acheampong, “Advanced Mobile Money Fraud Detection Using CNN-BiLSTM and Optimized SGD with Momentum,” Online Academic Journal of Information Technology, vol. 16, no. 60, 2025.

[5] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, “A survey on concept drift adaptation,” ACM Comput. Surv., vol. 46, no. 4, Art. 1, Jan. 2013.

[6] S. Wachter, B. Mittelstadt, and C. Russell, “Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR,” Harvard Journal of Law & Technology, Volume 31, Number 2 Spring 2018.

[7] A. H. Karimi, G. Barthe, B. Balle, and I. Valera, “Model-Agnostic Counterfactual Explanations for Consequential Decisions,” in Proc. 23rd Int. Conf. Artificial Intelligence and Statistics (AISTATS), 2020, pp. 895–905.

[8] K. Sokol and P. Flach, “Counterfactual Explanations of Machine Learning Predictions: Opportunities and Challenges for AI Safety,” in Proceedings of the AAAI Workshop on Artificial Intelligence Safety 2019.

[9] R. K. Mothilal, A. Sharma, and C. Tan, “Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations,” in Proc. Conf. Fairness, Accountability, and Transparency, 2020, pp. 607–617.

[10] G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, and F. Petitjean, “Characterizing Concept Drift,” Data Mining and Knowledge Discovery, vol. 30, no. 4, pp. 964–994, Jul. 2016.

[11] A. Tsymbal, “The Problem of Concept Drift: Definitions and Related Work,” Department of Computer Science, Trinity College Dublin, Ireland, April 29, 2004.

[12] O. A. Okunola, "Comparative Review of Credit Card Fraud Detection using Machine Learning and Concept Drift Techniques," International Journal of Computer Science and Mobile Computing, vol. 12, no. 7, pp. 24–48, 2023.

[13] V. Iosifidis and E. Ntoutsi, “Dealing with bias via data augmentation in supervised learning scenarios,” Journal of Artificial Intelligence Research, vol. 24, no. 11, pp. 1–22, 2018.

[14] D. Pessach and E. Shmueli, “A review on fairness in machine learning,” ACM Computing Surveys, vol. 55, pp. 1–44, 2022.

[15] M. Hardt, E. Price, and N. Srebro, “Equality of opportunity in supervised learning,” in Advances in Neural Information Processing Systems (NeurIPS), vol. 29, 2016.

[16] M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi, “Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment,” in Proc. 26th Int. Conf. World Wide Web (WWW), 2017, pp. 1171–1180.

[17] A. D’Amour et al., “Underspecification presents challenges for credibility in modern machine learning,” J. Mach. Learn. Res., vol. 23, no. 226, pp. 1–61, 2022.

[18] E. Ntoutsi, P. Fafalios, U. Gadiraju, V. Iosifidis, W. Nejdl, M.-E. Vidal, S. Ruggieri, et al., “Bias in data-driven artificial intelligence systems—An introductory survey,” Wiley Interdiscip. Rev.: Data Mining Knowl. Discov., vol. 10, no. 3, p. e1356, 2020.

# DESIGN AND IMPLEMENTATION OF A BLUETOOTH-BASED HOME AUTOMATION SYSTEM USING ARDUINO

Chit Myo Naing[1], and Zan Mo Mo Aung[2]

[1] Department of Engineering Physics, Myanmar Aerospace Engineering University
[2] Faculty of Computer Science, University of Computer Studies (Mandalay)
chitmyonaing1992@gmail.com,zanmomoaung@gmail.com

## ABSTRACT

*Home automation systems play a vital role in enhancing comfort and convenience in modern households. This paper presents the design and implementation of a Bluetooth-based home automation system controlled via an Android smartphone. The proposed system is utilized an Arduino Nano microcontroller, a Bluetooth module, and an channel relay to enable wireless control of household appliances. Devices such as lights, fans, air conditioners, and door locks can be switched ON or OFF remotely through a mobile application. In this work, the HC-06 Bluetooth module provides wireless communication between the Arduino Uno and the smartphone. The system software is developed using the Arduino programming language (C/C++) and uploaded to the microcontroller via the Arduino IDE. Experimental results demonstrate that the proposed system provides a reliable, low-cost, and user-friendly solution for remote appliance control.*

## KEYWORDS

*Bluetooth-based home automation, Wireless, Arduino Nano, microcontroller*

## 1. INTRODUCTION

Home automation systems have gained significant attention over recent decades due to their ability to enhance user comfort and overall quality of life. A home automation system can be designed using a centralized controller capable of supervising and operating multiple interconnected household devices, including electrical outlets, lighting units, temperature and humidity sensors, smoke and gas detectors, fire alarms, and security or emergency systems [3].

One of the key benefits of a home automation system is its ability to be conveniently monitored and controlled using a wide range of devices, including smartphones, tablets, desktop computers, and laptops [1]. Nowadays, these systems typically rely on the integration of a smartphone interface with a microcontroller to manage and control household appliances. A mobile-based application enables users to manage and observe home appliances using diverse communication techniques [2].

A communication medium is required to interact with home automation systems. Smartphone-based home automation systems often utilize various wireless technologies to connect with microcontrollers, such as Bluetooth, Global System for Mobile Communication (GSM), ZigBee, Wi-Fi, and EnOcean.

The proposed system is implemented using an Arduino microcontroller, with communication established through a smartphone over Bluetooth. An Arduino control application installed on the smartphone provides the user interface for managing and monitoring connected devices.

Bluetooth is chosen as the communication medium due to its low power requirements, affordability, reliability, and ease of integration with microcontrollers such as Arduino [1], [3] and [4]. This combination ensures a user-friendly, cost-effective, and efficient solution for home automation.

## 2. METHODOLOGIES OF HOME AUTOMATION SYSTEMS

Many systems rely on microcontroller-based architectures, where a central controller such as Arduino or Raspberry Pi manages interconnected devices including lighting, appliances, and security sensors [3] and [4]. Wireless communication protocols like Bluetooth, Wi-Fi, ZigBee, GSM, and EnOcean are commonly used to establish reliable connections between user interfaces (e.g., smartphones or tablets) and the control unit. Voice- or audio-based control is another prominent methodology, enabling users to interact with the system through natural language commands, thereby increasing accessibility and convenience. Some advanced systems integrate IoT platforms and cloud computing, allowing real-time monitoring, remote access, and data analytics for energy efficiency and security management. Overall, these methodologies aim to provide user-friendly, cost-effective, and flexible solutions that enhance comfort, convenience, and safety in modern smart homes.

### 2.1. Bluetooth Based Home Automation System

R. Piyare and M. Tazil [3] developed an Arduino-based automation platform where Bluetooth connectivity allowed seamless communication between the microcontroller and a smartphone application, effectively controlling lighting and electrical devices. In [7], the authors proposed the smart home automation system that is designed by using Bluetooth technology. This system allows users to monitor water levels using an ultrasonic sensor and employs a soil moisture sensor to automate the plant irrigation process. The Bluetooth-based automation system is cost-effective and allows users to control appliances within the Bluetooth network's range.

### 2.2. Voice Recognition Based Home Automation

In [8], Sen et al. implemented a micro controller based voice controlled home automation system using smartphones. This system allows users to operate all household appliances using just their voice. An Arduino Uno microcontroller that interprets the user's commands and manages the switching of devices. Communication between the smartphone and the microcontroller is facilitated through Bluetooth, a widely used wireless technology for data transmission.

### 2.3. ZigBee Based Wireless Home Automation System

The authors presented a ZigBee-enabled home automation system designed for the remote control and monitoring of household electrical appliances. To compute the overall power usage of connected loads, the proposed system operates in two selectable modes: one utilizing the CS5490 energy-measurement integrated circuit and the other based on the WCS2705 current-sensing IC. Real-time power consumption data are visualized through a PC-based graphical user interface developed using the NetBeans Java platform. In addition, the system performance is evaluated by examining key parameters, including communication latency, round-trip delay (RTD), and received signal strength indicator (RSSI) [9].

### 2.4. GSM Based Home Automation System

A smart home automation system developed using the Global System for Mobile Communication (GSM). The system's hardware structure includes a GSM modem, a PIC16F887 microcontroller, and a smartphone. Electrical appliances are operated by sending control commands via SMS through the GSM modem. The findings indicate that integrating GSM technology into home automation systems offers enhanced reliability and strong security [10].

### 2.5. Internet of things (IoT) Based Home Automation System

The proposed system employed a NodeMCU microcontroller as a Wi-Fi gateway to connect various sensors and upload their data to the Adafruit IO cloud server. Data obtained from multiple sensors, such as ultrasonic, temperature, humidity, gas, and motion sensors, can be accessed from If This Then That (IFTTT). Then that (IFTTT) platform on

user devices such as smartphones or laptops via the Internet, regardless of the user's location.

Relay modules are employed to interface the NodeMCU with household appliances for controlled switching operations. The system is designed as a portable control unit that can be easily installed in a real home environment for monitoring and control purposes. This IoT-based home automation system allows efficient remote appliance management while also supporting home safety through automated operation [11].

### 2.6. EnOcean Based Home Automation System

EnOcean is an emerging energy-harvesting technology applied in transportation, building, and home automation systems. Moreover, this technology is widely adopted in logistics and industrial applications because of its high energy efficiency and ease of installation. A home automation system using EnOcean technology can be developed by integrating the Internet, a router, an automation controller, a Duckbill 2 EnOcean gateway, and various EnOcean-compatible devices [12].

## 3. PROPOSED SYSTEM

To explain the functionality of the proposed system as presented in Figure 1, its software, hardware, and operational components are discussed in the subsequent subsections. Home automation is a system of interconnected, controllable devices designed to improve comfort, personalization, efficiency, and security in a home.

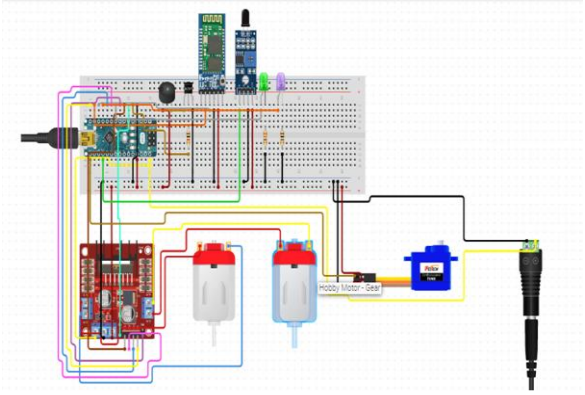


Figure1.Circuit Diagram of the Proposed System

This system mainly comprises four key components: an Arduino Nano, a Bluetooth module, relay drivers, and an Android smartphone application. The purpose of this proposed system is designed to allow users to control multiple home appliances via a smartphone.

### 3.1. Hardware System Design and implementation

The proposed system uses an Arduino 12microcontroller, HC-06 Bluetooth Module, Buzzer Module, channel relay module ,Flame Sensor, LED, DC motor, Breadboards, L298N motor driver, Switch, 2*4.2V Power supply and Jumper wires.

#### 3.1.1. Arduino Microcontroller

The Arduino microcontroller acts as the central processing unit of the home automation system, managing all connected components. It includes digital and analog input/output (I/O) pins that interface with expansion boards, breadboards (shields), and other circuits. This board as shown in Figure 2 is programmed using the Arduino programming language (C/C++) through the Arduino IDE. The Arduino Nano is a compact, breadboard-compatible microcontroller board based on the protocols, including UART, I2C (TWI), and SPI. Hardware serial communication is available on pins 0 (RX) and 1 (TX) through the on board FTDI FT232RL USB interface, while software serial can be configured on other digital pins. The Arduino IDE provides tools such as the Serial Monitor and libraries like Wire for I2C communication.

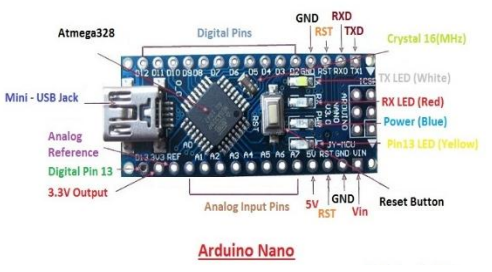



Figure.2 Arduino Nano Board

ATmega328P as shown in Figure 2. It offers functionality similar to the Arduino

Duemilanove but lacks a DC power jack, connecting instead via a Mini-B USB port.

### 3.1.2. Bluetooth Module

The Bluetooth module receives commands from the user and transmits them to the microcontroller (Arduino Nano) as shown in Figure 3. This wireless technology allows data exchange over short distances using short-wavelength radio signals. It was originally developed to replace cables in consumer electronics and to support short-range data communication. The module operates on the Bluetooth 2.0 protocol and functions exclusively as a slave device.

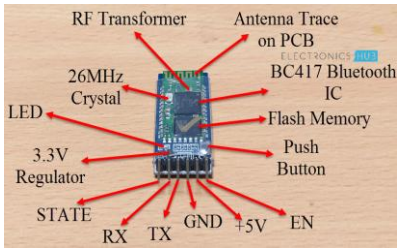



Figure 3. Bluetooth Module

This method provides a cost-effective solution for wireless data transmission and offers greater flexibility compared to other techniques, supporting file transfer speeds of up to 2.1 Mb/s.

The HC-06 module utilizes the Frequency Hopping Spread Spectrum (FHSS) technique to reduce interference with other devices and enable full-duplex communication [3].It operates within a frequency range of 2.402 GHz to 2.480 GHz. This module typically has six (or four) pins, of which only four are essential: VCC, GND, RX, and TX. VCC provides the power supply (either 5 V or 3.3 V), while GND represents the ground (0 V) as presented in Figure 3. The RX pin receives data, and the TX pin transmits data. In practice, the RX pin of the module connects to the TX pin of the Arduino Nano, and the TX pin of the module connects to the RX pin of the Arduino Nano.

The HC-06 module is an excellent choice for short-range wireless communication, typically suitable for distances under 100 meters. Among available wireless communication solutions, the module is one of the most cost-effective options. It also has low power consumption, making it ideal for battery-operated devices. Additionally, the HC-06 can be connected to nearly all controllers or processors because it utilizes a UART interface [14] and [15].

### 3.1.3. Buzzer Module

The buzzer, illustrated in Figure 4, is an audio signalling device that generates different sound tones based on the input frequency.

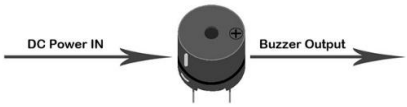



Figure 4. Buzzer Module

### 3.1.4. Channel Relay Module

In this study, an 8-Channel Relay Module is employed to control load operations, such as switching devices on and off. The module is a compact board containing eight 5 V relays and is capable of handling high-voltage or high-current loads, including motors, solenoid valves, lamps, and AC appliances. It features switching and isolation components, which facilitate easy interfacing with microcontrollers or sensors using minimal additional components and connections [16].

### 3.1.5. Flame Sensor

The flame sensor module in Figure 5 is designed to detect infrared (IR) light in the wavelength range of 700–1000 nm, with a detection angle of 60°. It uses an LM393 comparator to produce a digital output and operates on a DC voltage range of 3.3 V to 5.2 V. The module's sensitivity can be adjusted via an integrated potentiometer.

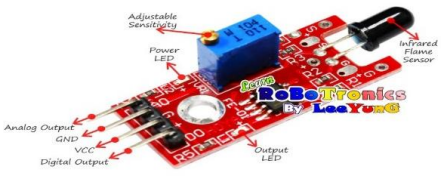



Figure 5. Flame Sensor

### 3.1.6. LED (Light Emitting Diode)

LEDs as shown in Figure 6, typically run on under 5V and handle up to 5V reverse voltage. A 5V source (like Arduino or USB) is safe. Higher voltages (e.g., 9V) require a larger resistor but may risk damaging sensitive parts.

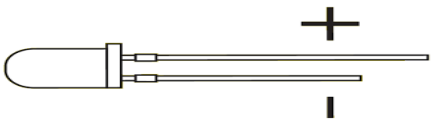


Figure 6. LED

### 3.1.7. DC Motor

A DC motor as shown in Figure 7, converts electrical energy into mechanical rotation using the principle of electromagnetism. The coils inside the motor generate a magnetic field that produces rotational motion.

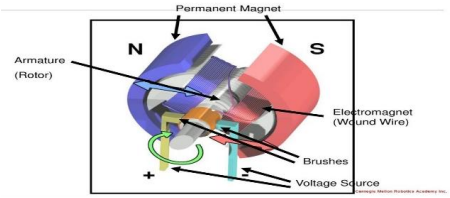



Figure 7. DC Motor

### 3.1.8. Switch

A switch turns a circuit on or off by connecting or disconnecting two terminals. When on, current flows to the device and returns via the neutral wire as shown in Figure 8.

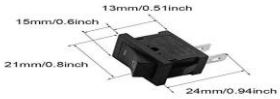



Figure 8.Switch

### 3.1.9. Breadboards

Breadboards are used for building and testing circuits with through-hole components, which have long leads meant for insertion into plated holes on PCBs, allowing them to be soldered securely as shown in Figure 9.

Figure 9. Breadboard

### 3.1.10. L298N motor driver

A motor driver amplifies low-current control signals to drive motors. The L298N as shown in Figure 10, can control two DC motors in speed and direction, with terminals for motor outputs, power, and a configurable 5V pin.

Figure 10. L298N motor driver

### 3.1.11. Jumper Wires

Jumper wires connect circuit points to carry current. They are mostly copper due to its good conductivity and low cost as displayed in Figure 11.

Figure 11. Jumper Wires

### 3.1.12. Power supply (4.2V)

Li-ion batteries as shown in Figure 12, are rechargeable and commonly power portable electronics, electric vehicles, and are expanding into military and aerospace uses.

Figure 12. 4.2V battery

### 3.1.13. Servo Motor

A servo motor precisely controls rotation or movement, allowing objects to move to exact angles or positions using a motor with a servo system as presented in Figure 13.

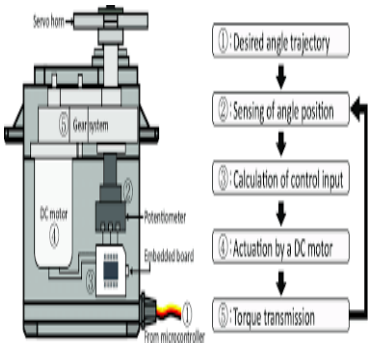



Figure 13. Servo Motor

## 3.2. Software Design and implementation

This section briefly describes the programming language and the associated software libraries used in the development of the proposed system.

### 3.2.1. Arduino Integrated Development Environment (IDE)

The Arduino Integrated Development Environment (IDE) is an open-source platform that enables users to write, compile, and upload code directly to a microcontroller. In this study, version 1.8.9 of the IDE was used.

## 3.3. Android Smart Phone

Android is an operating system mainly used for smartphones and tablets. Android applications are created using Java and the Android Software Development Kit (SDK). These applications run in a virtual machine. In this system, the ATMega328P microcontroller communicates wirelessly with an Android Bluetooth control application using the HC-06 Bluetooth module. The embedded system's input and output ports are connected to home appliances for control [16].

# 4. Results and Discussion

This research presents a practical and widely applicable Arduino-based home automation system that uses Bluetooth technology to wirelessly control various electronic devices. It is one of the most significant and beginner-friendly projects in Arduino development, offering both educational value and real-world applicability. The system is built around an Arduino Nano microcontroller, which functions as the central controller. Communication is established using an HC-06 Bluetooth module, enabling wireless data exchange between the Arduino and a user's Android smartphone through the device control application or any similar Bluetooth-enabled app.

The Arduino is connected to a main project board that includes a single-channel relay module, allowing safe control of external AC or DC loads. The relay acts as an electronic switch controlled by the Arduino. In this implementation, the connected devices are defined as Device 1 (buzzer), Device 2 (fan), and Device 3 (light). These devices can be replaced or expanded to include appliances such as water pumps, televisions, or other household electronics.

When the user opens the smartphone application and selects a device, tapping the ON button for Device 1 sends a Bluetooth command to the Arduino, which activates the buzzer. Pressing the same button again turns it off. Similarly, activating Device 2 switches the fan on or off, while Device 3, typically a light or LED, operates in the same manner. The Arduino receives these commands through the HC-06 Bluetooth module and controls each device using digital output pins interfaced with the relay module. The system is powered by two 4.2 V batteries, which may be connected in series or parallel depending on the required voltage and current. Final outcome of the research project is given below in Figure 14.

This power configuration makes the system portable and suitable for use in environments without continuous AC power, such as demonstrations or remote locations. The advantages of the proposed system include:

- Smart home control without the need for internet or Wi-Fi
- Energy savings through remote switching of unused devices

- Improved accessibility for elderly or physically challenged users
- Cost-effective automation using readily available components
- Easy expandability for additional relays and devices

Figure 14. Home Automation System

## 5. CONCLUSION

The proposed system presents the design and development of a Bluetooth-enabled home automation system. This system is designed to be economical, easy to maintain, and simple to operate, particularly benefiting elderly individuals and people with physical challenges. The primary objective of this work is to establish a centralized control mechanism that allows household appliances to be operated using an Android smartphone, thereby reducing the time and effort required for manual operation. Additionally, the system enables users to remotely monitor the operational status (ON/OFF) of connected devices. The proposed concept is scalable and can be adapted for automation in larger environments such as industrial facilities, shopping centers, and healthcare institutions. Furthermore, the system aims to enhance energy efficiency while contributing to improved comfort and quality of life.

An Arduino-based device control system using Bluetooth and a smartphone can be further enhanced to include speed control for a fan or volume adjustment for a buzzer. This concept can be expanded into a home automation system using Internet of Things (IoT) technology. Instead of Bluetooth, a GSM modem can be integrated to allow device control through SMS messages. This enables remote operation of appliances via mobile networks, offering greater flexibility and range compared to Bluetooth.

## ACKNOWLEDGEMENTS

The authors would like to acknowledge the Myanmar Aerospace Engineering University's ERC committee for granting permission to submit this paper.

## REFERENCES

[1] Ramlee, R. A., Othman, M. A., Leong, M. H., Ismail, M. M., & Ranjit, S. S. S. (2013, March). "Smart home system using android application" , *International Conference of Information and Communication Technology (ICoICT)* (pp. 277-280).

[2] M. Asadullah and A. Raza, "An overview of home automation systems," *2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI)*, Nov. 2016, doi: https://doi.org/10.1109/icrai.2016.7791223.

[3] Piyare, R., & Tazil, M. ,(2011, June), "Bluetooth based home automation system using cell phone", *IEEE 15th international symposium on consumer electronics (ISCE)* (pp. 192-195).

[4] Bolimera, R., Kambhampati, M. R., Bhargavi, U., Pavani, M., & Srikanth, K. ,(2025).,"Wireless Home Automation Using Arduino And Bluetooth", *International Journal of Data Science and IoT Management System*, *4*(4), 343-346.

[5] Jebarani, M. E., Lakshmi, S., & Kavipriya, P., (2022, November), "Implementation of smartly controlling home devices for physically challenged people using Raspberry Pi", In *AIP Conference Proceedings* (Vol. 2452, No. 1, p. 090008), AIP Publishing LLC.

[6] Jamil, M. M. A., & Ahmad, M. S. (2015, March), "A pilot study: Development of home automation system via raspberry Pi", *2nd International Conference on Biomedical Engineering (ICoBE)* (pp. 1-4).

[7] Asadullah, M., & Ullah, K., (2017, April), "Smart home automation system using Bluetooth technology", *International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)*, (pp. 1-6).

[8] Sen, S., Chakrabarty, S., Toshniwal, R., & Bhaumik, A., (2015), "Design of an intelligent voice controlled home automation system", *International Journal of Computer Applications*, *121*(15).

[9] Baviskar, J., Mulla, A., Upadhye, M., Desai, J. and Bhovad, A., 2015, January, "Performance analysis of ZigBee based real time Home Automation system",*International*

*conference on communication, information & computing technology (ICCICT)* (pp. 1-6).

[10] Teymourzadeh, R., Ahmed, S. A., Chan, K. W., & Hoong, M. V.,(2013, December), "Smart GSM based home automation system", *IEEE conference on systems, process & control (ICSPC)* (pp. 306-309).

[11] Jabbar, W. A., Kian, T. K., Ramli, R. M., Zubir, S. N., Zamrizaman, N. S., Balfaqih, M., ... & Alharbi, S.., (2019), "Design and fabrication of smart home with internet of things enabled automation system", *IEEE access*, *7*, 144059-144074.

[12] Kuzlu, M., Pipattanasomporn, M., & Rahman, S., (2015, November), "Review of communication technologies for smart homes/building applications". In *2015 IEEE Innovative Smart Grid Technologies-Asia (ISGT ASIA)* (pp. 1-6).

[13] Lai, T. W., Oo, Z. L., & Than, M. M. ,(2018), "Bluetooth based home automation system using android and arduino", *University of Computer Studies (Sittway). Availible at: https://www.researchgate.net/publication/333044694_Bluetooth_Based_Home_Automation_System_Using_Android_and_Arduino*.

[14] Amoran, A. E., Oluwole, A. S., Fagorola, E. O., & Diarah, R. S. ,(2021), "Home automated system using Bluetooth and an android application", *Scientific African*, *11*, e00711.

[15] Tyagi, C. S., Agarwal, M., & Gola, R., (2016), "Home Automation using voice recognition and Arduino", *IJRTER*.

[16] Raheem, A. K. K. A.,(2017), "Bluetooth based smart home automation system using Arduino UNO microcontroller",*Al-Mansour Journal*, *27*(1), 139-156.

**Authors**

**Biography**

**Chit Myo Naing** received the M.Sc. degree in Physics from Banmaw University and a Diploma in Computer Science (D.C.Sc.) from the University of Computer Studies, Banmaw. He later earned a Master of Business Administration (MBA) degree from Monywa University of Economics. He is currently serving as a Lecturer in the Department of Engineering Physics at Myanmar Aerospace Engineering University (MAEU). His research interests include in embedded systems, machine learning, artificial intelligence, and security.

**Zan Mo Mo Aung** received the Ph.D. degree in Information Technology from University of Computer Studies, Mandalay (UCSM). She is currently serving as an Associate Professor in the Faculty of Computer Science at the University of Computer Studies (Mandalay). Her research interests include in natural language processing, machine learning, artificial intelligence, network engineering and data mining.

# A DEEP LEARNING-BASED FRAMEWORK FOR BUILDING FOOTPRINT EXTRACTION AND GIS-DRIVEN URBAN SPATIAL ANALYSIS USING HIGH-RESOLUTION SATELLITE IMAGERY

Kyaw Zin Myat[1], Soe Thant[2], and Zayar Tun[3]

[1,2,3]Department of Computer Science, Defence Services Academy (Pyin Oo Lwin)
*kyawzinmyat980@gmail.com, perpetualsoul25@gmail.com, blueskybluesky9154@gmail.com*

## *ABSTRACT*

*Accurate building footprint information is essential for many urban applications, including urban planning, land-use monitoring, disaster management, and infrastructure development. Traditional manual digitization of building footprints from satellite imagery is time-consuming and not suitable for large-scale urban areas. This study proposes a deep learning-based framework for automatic building footprint extraction and GIS-driven urban spatial analysis using high-resolution satellite imagery. Satellite images of a selected urban area are obtained using the QGIS platform and manually annotated to generate ground truth masks. The images and masks are preprocessed and used to train a U-Net based segmentation model. The trained model is applied to detect building areas from satellite images. The predicted building masks are further analyzed and visualized within a GIS environment using spatial overlay techniques. The results demonstrate that the proposed approach can effectively detect building structures and support urban spatial analysis. The integration of deep learning and GIS technologies provides an efficient solution for automated building mapping in modern urban environments.*

## *KEYWORDS*

*GIS, Urban planning, QGIS, Satellite imagery*

## 1. INTRODUCTION

Urban areas are expanding rapidly due to population growth and economic development. As cities continue to grow, accurate spatial information becomes increasingly important for effective urban planning and infrastructure management. Among different types of spatial data, building footprint information plays a critical role in understanding urban structure and land-use patterns.

Building footprints represent the geometric outlines of buildings on the ground. These data are widely used in geographic information systems for various applications, including urban growth analysis, population estimation, disaster risk assessment, and environmental monitoring. Accurate building maps allow planners and decision-makers to analyze spatial patterns and make informed decisions.

Traditionally, building footprints are extracted manually from aerial photographs or satellite images through visual interpretation. Although manual digitization can produce accurate results, it requires significant time and effort. For large cities and rapidly growing urban areas, manual mapping becomes inefficient and impractical.

In recent years, remote sensing technology has provided high-resolution satellite images that contain detailed information about urban environments. At the same time, advances in artificial intelligence and deep learning have improved the ability to

automatically extract features from images. Deep learning models, particularly Convolutional Neural Networks (CNNs), have shown strong performance in image classification, object detection, and image segmentation tasks.

Despite recent advances in deep learning-based segmentation, challenges remain in accurately extracting building footprints in complex urban environments and integrating the results into GIS platforms for spatial analysis. Many existing approaches focus primarily on segmentation accuracy without addressing geospatial integration and practical usability. Therefore, this study aims to develop an end-to-end framework that combines deep learning-based building extraction with GIS-based spatial analysis.

## 2. LITERATURE REVIEW

Ji et al [6]. introduced a Siamese Fully Convolutional Network designed to process multisource aerial and satellite images. Their architecture used dual input branches to capture both original and down-sampled representations, enabling better handling of scale variation in large buildings. Experiments on the WHU dataset demonstrated improved segmentation accuracy compared to conventional U-Net models. However, the study revealed that performance declined when transferring across datasets with different resolutions and geographic characteristics. This limitation highlights the challenge of developing models that generalize well across diverse urban environments. This gap motivates the development of localized segmentation frameworks tailored to specific urban conditions, which directly informs the proposed research.

Maggiori et al. [7] introduced a deep convolutional neural network for large-scale urban mapping from high-resolution aerial imagery, designed to learn multi-scale features of complex urban morphology. Their study demonstrated that semantic segmentation models can accurately extract building structures across diverse urban scenes by capturing variations in building size, density, and spatial arrangement. The framework was validated on a large and geographically extensive dataset, highlighting its scalability and effectiveness for wide-area urban analysis. However, performance decreased when applied to unfamiliar regions, with different morphological characteristics, indicating limited cross-region generalization. This limitation highlights the need for localized deep learning frameworks. Their findings support the motivation for designing segmentation systems tailored to specific urban areas for reliable building footprint extraction and urban expansion assessment.

Buyukdemircioglu et al [5]. extracted building footprints using high-resolution orthophotos and normalized digital surface models from an urban dataset in Izmir, Turkey. Their deep learning segmentation framework demonstrated that integrating height information improves reliability in dense urban regions. The fusion of RGB and elevation data enhanced both boundary accuracy and building completeness. However, the absence of augmentation strategies and pretrained initialization limited model generalization across varying urban conditions. These findings reveal that robustness and domain adaptability remain open challenges. This research gap directly motivates the proposed study, which develops a localized deep learning framework optimized for building footprint extraction in a specific urban area.

Li et al [8]. presented a deep learning framework for urban building footprint extraction that reflects the growing shift from manual mapping toward automated semantic segmentation in remote sensing. They introduced a multi-scale convolutional neural network designed to capture both fine boundary details and large structural patterns. The architecture integrated hierarchical feature learning to improve segmentation consistency in dense urban scenes. Their experiments were conducted on high-resolution aerial datasets that included diverse building shapes and urban layouts. The dataset evaluation emphasized generalization across varying

spatial resolutions and illumination conditions, demonstrating the importance of training diversity in deep learning segmentation models. The results showed improved footprint completeness and boundary precision compared to traditional machine learning approaches. However, the study also revealed that model performance remained sensitive to regional differences in building morphology and imaging conditions. This limitation is particularly significant for localized urban expansion assessment, where accurate adaptation to region-specific urban characteristics is essential. Therefore, it highlights the need for adaptable segmentation frameworks that ensure consistent accuracy in specific urban environments, which is the focus of this research.

## 3. THEORICAL BACKGROUND

Building footprint extraction from high-resolution satellite images is an important task in remote sensing and urban studies. Accurate building information supports applications such as urban planning, infrastructure management, disaster assessment, and spatial analysis. Recent advances in deep learning, especially Convolutional Neural Networks, enable more efficient and accurate extraction of building footprints from satellite imagery.

### 3.1 Semantic Segmentation

Semantic segmentation is a computer vision technique that classifies every pixel in an image into a specific category. Unlike image classification, which assigns one label to the whole image, semantic segmentation provides detailed information about object locations and boundaries [1]. This method allows different regions such as buildings, roads, vegetation, and water bodies to be identified within an image. Because of its pixel-level prediction capability, semantic segmentation has become widely used in fields such as autonomous driving, medical image analysis, and remote sensing image interpretation [3].

In remote sensing, semantic segmentation is an effective approach for analyzing high-resolution satellite images. Satellite imagery often contains complex scenes with many objects and diverse textures. Segmentation models analyze these images and assign each pixel to a specific land-cover category. This process helps detect objects such as buildings, roads, and vegetation more accurately. Semantic segmentation has become an important method for urban mapping and for extracting building footprints from satellite imagery [3].

Recent advances in deep learning have greatly improved semantic segmentation performance. Convolutional Neural Networks (CNNs) are widely used because they can automatically learn spatial features from images. Many segmentation models adopt an encoder-decoder architecture, where the encoder extracts image features and the decoder reconstruct spatial information for pixel-level prediction [2]. Well-known models such as Fully Convolutional Networks (FCN), U-Net, and DeepLab have demonstrated strong performance in segmentation tasks [4].

### 3.2 U-Net Architecture

U-Net is a deep learning architecture developed for semantic segmentation tasks. It was first introduced for biomedical image analysis but is now widely used in many segmentation applications. The main objective of U-Net is to classify each pixel in an image and produce accurate object boundaries. Because of its ability to perform precise pixel-level prediction, U-Net has become an effective model for segmentation tasks in areas such as medical imaging, remote sensing, and object detection [4].

Figure 1 shows the U-Net semantic segmentation architecture. The structure of U-Net follows an encoder-decoder architecture. The encoder, also called the contracting path, extracts important features from the input image using convolution and pooling operations. During this process, the spatial size of the feature maps decreases while the number of feature channels

increases, allowing the network to capture high-level contextual information [4]. The decoder, or expanding path, gradually restores the spatial resolution using up sampling or transposed convolution layers to generate a detailed pixel-level segmentation map.

Another important characteristic of U-Net is the use of skip connections between encoder and decoder layers. These connections transfer feature information directly from the encoder to the decoder, which helps preserve spatial details that may be lost during down sampling. As a result, U-Net can accurately detect object boundaries and small structures in images [4]. Due to this efficient design, U-Net has become one of the most widely used models for semantic segmentation, including applications such as building footprint extraction from high-resolution satellite images.

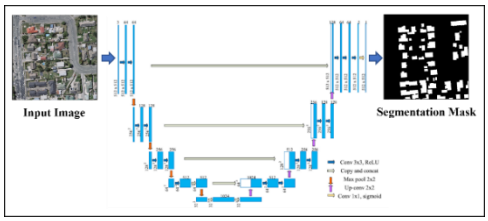



Figure 1. U-Net Architecture for Semantic Segmentation

### 3.3 Evaluation Metrics

Model performance is evaluated using several standard metrics: Precision, Recall, F1-score, and Intersection over Union (IoU). These metrics provide a comprehensive evaluation of segmentation accuracy [6]. Precision indicates how many of the pixels predicted as buildings are correct. A high precision value means that the model produces fewer false positive predictions.

Recall evaluates how well the model identifies all building pixels present in the ground truth data. A high recall value means that most of the actual buildings are successfully detected by the model. However, a model with high recall may still produce some incorrect predictions [6].

The F1-score is the harmonic mean of precision and recall. It provides a balanced evaluation of the model's performance by considering both false positives and false negatives. A higher F1-score indicates that the model performs well in both detecting buildings and avoiding incorrect predictions.

Intersection over Union (IoU) is commonly used in image segmentation tasks. It measures the degree of spatial overlap between the predicted building mask and the ground truth mask. The IoU value ranges from 0 to 1, where a higher value indicates better segmentation accuracy.

Together, these evaluation metrics provide a reliable assessment of the segmentation performance of the proposed deep learning models. These metrices determine how accurately the models detect building footprints and how closely the predicted results match the reference data [6].

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

$$F1 = \frac{2(Precision * Recall)}{Precision + Recall}$$

$$IoU = \frac{TP}{TP+FP+FN}$$

## 4. PROPOSED SYSTEM DESIGN

The overall workflow of the proposed system for building footprint extraction and GIS integration is illustrated in figure 2. First, high-resolution satellite images are used as the primary input data. The framework automatically generates accurate building footprint maps from satellite imagery using a series of processing stages. The final stage of the proposed workflow involves the integration and visualization of the validated building footprints within a GIS pipeline. After the segmentation stage, the extracted building footprint masks are transformed into georeferenced spatial data for visualization, management, and spatial analysis within the GIS environment.

The proposed system is built upon the U-

Net architecture, a convolutional neural network specifically designed for high-precision semantic segmentation. In this framework, the model performs pixel-wise classification to distinguish building regions from non-building areas. This approach enables accurate delineation of building boundaries in complex urban satellite imagery.

The system consists of four major stages: data acquisition, model training, inference, and geographic visualization.

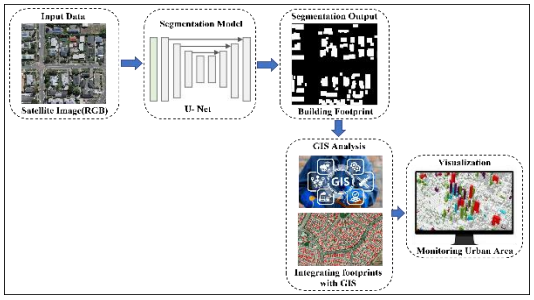



Figure 2. Overview of Proposed System

## 4.1 Data Acquisition

The data acquisition stage establishes the basis for the proposed framework for extracting building footprints. Satellite imagery serves as the primary data source for model development, training, and evaluation. Urban environments exhibit complex structural patterns that differ significantly from rural environments. Consequently, dataset quality, spatial resolution, and geographic characteristics play a critical role in determining segmentation performance.

The study area is defined as Pyin Oo Lwin Township (Ward-21), Myanmar. The geographic boundary of the study region is defined in QGIS to ensure consistent spatial coverage throughout the process of preparing the dataset. Satellite imagery with a spatial resolution of 0.3 metres is acquired from Google Earth and exported as a GeoTIFF file to preserve the geospatial metadata. The dataset covers an area of approximately 22.10 km² of urban terrain. It represents the diverse building typologies and urban layouts that are typical of the region. Figure 3 shows a satellite image of Pyin Oo Lwin (Ward-21) in QGIS.

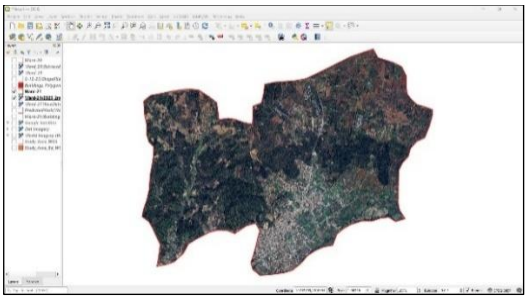


Figure 3. Pyin Oo Lwin (Ward-21) Satellite Image in QGIS

The ground truth building masks are generated through a controlled manual annotation process within the QGIS environment after downloading the satellite image. A new vector layer in shapefile format is created and then overlaid on the satellite image. Using polygon annotation tools, each visible building footprint is manually delineated. This step creates a labelled vector dataset that accurately depicts the boundaries of buildings across the study region. Figure 4 illustrates manually annotated building footprints in the study area using the QGIS application. Manual annotation ensures a precise, pixel-by-pixel correspondence between the satellite imagery and the reference mask. Precise delineation reduces labelling noise and improves the reliability of model supervision during training.

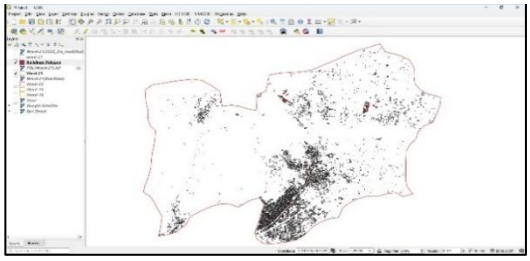


Figure 4. Labelled mask image of Pyin Oo Lwin (Ward-21) in QGIS

After annotation, the labelled vector shapefile is converted to a raster format (GeoTIFF) in order to generate binary mask images. These raster masks preserve the spatial structure of the annotated buildings. To standardize the dataset for deep learning, both satellite images and their

corresponding masks are cropped into fixed patches of 512 × 512 pixels. Using fixed-size tiling ensures that the input dimensions are uniform and supports efficient batch training. Figures 5 and 6 show satellite images and masks that were used as training input for extracting building footprints. The dataset reflects the architectural characteristics and spatial structure of Pyin Oo Lwin Township. Local building patterns include variations in roof materials, layout density, and urban morphology. This localized dataset enables realistic evaluation of segmentation models within a specific suburban environment.

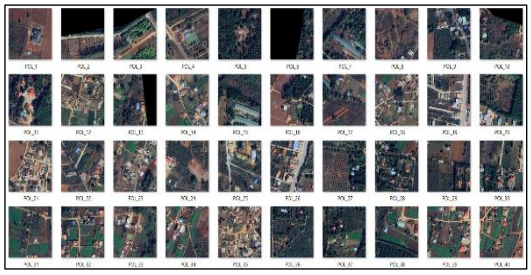


Figure 5. Satellite images used as training input for extracting building footprints

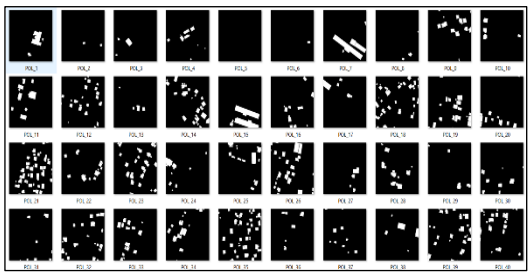


Figure 6. Masks used as training input for extracting building footprints

## 4.2 Flowchart of Building Footprints Extraction System

The system flowchart illustrates the proposed framework for building footprint extraction from satellite imagery. Figure 7 illustrates the flowchart providing an automated process for accurate building footprint extraction.

The workflow begins with preparing satellite imagery and vector data in QGIS. Ground truth building masks are manually annotated to achieve accurate pixel-level labelling. The images and masks are then divided into fixed-size patches of 512 × 512 pixels to standardize the dataset and improve computational efficiency. During pre-processing, the image patches are normalized to reduce radiometric variations and support stable model training.

The prepared dataset is used to train the segmentation model, which learns the mapping between input images and building masks. Model performance is evaluated using the loss function that measures pixel-level differences between predicted and reference labels. If validation results are not satisfactory, the training process is iteratively refined through parameter adjustment and optimization. Once stable performance has been achieved, the finalized model is ready for use.

The input images are resized and normalized using the same settings as during training, ensuring consistency. In the inference stage, new RGB satellite images are processed using the trained model. The model then predicts the probability of each pixel belonging to a building and generates segmentation masks. The output is a raw segmentation mask that highlights the predicted location of buildings in the image.

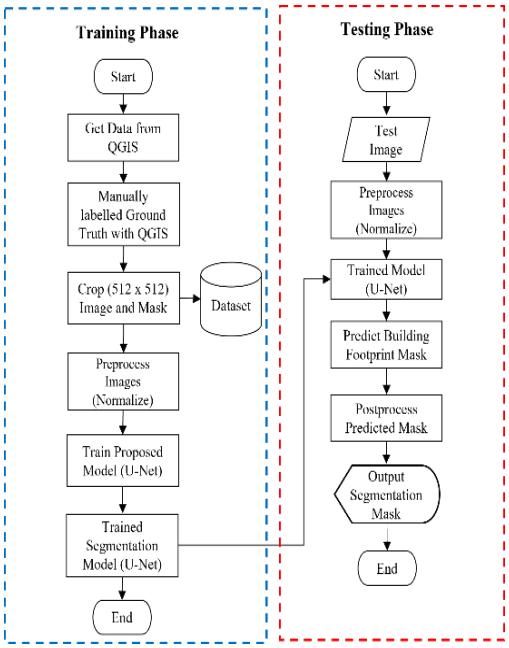



Figure 7. Flowchart of Building Footprints Extraction System

The predicted masks often contain imperfections, including false positives, holes in building regions, or fragmented shapes. Post-processing techniques are then applied to enhance the quality of predicted masks, involving the removal of noise and correction of fragmented regions. The final output is clean building footprint masks that can be visualized, integrated into GIS systems, or used for urban planning and analysis.

### 4.3 Geospatial Integration of Predicted Building Footprints

The output of the deep learning segmentation model consists of raster masks that represent predicted building regions extracted from satellite imagery. Although these masks contain valuable spatial information, they cannot be directly used in geographic information systems without proper spatial referencing and format standardization. Deep learning frameworks primarily focus on pixel-based prediction and often ignore geospatial metadata.

As a result, segmentation outputs lack coordinate reference information that is required for GIS-based analysis. To transform model predictions into usable geospatial products, a structured integration workflow is necessary. This workflow ensures spatial consistency, format compatibility, and accurate geographic alignment.

Figure 8 shows the workflow for validating and integrating predicted segmentation masks into a GIS environment. The workflow emphasizes verification, georeferencing, format conversion, and spatial overlay. Each step is designed to preserve geographic accuracy while converting raster predictions into structured building footprint datasets. This integration bridges the gap between machine learning outputs and practical urban analysis tools.

The predicted segmentation masks generated by the deep learning model are recombined to reconstruct the complete spatial output. The merged prediction results are then converted into TIFF format, which is compatible with Geographic Information System (GIS) applications for further spatial analysis. Without spatial referencing, the segmentation output cannot be aligned with satellite imagery or external GIS layers.

Therefore, georeferencing is a critical step in ensuring that extracted building footprints correspond to real-world coordinates. The merged raster mask is first verified to determine whether spatial metadata is present. If the output lacks projection information, it is converted into GeoTIFF format. The GeoTIFF standard supports the storage of coordinate reference systems, pixel resolution, and geographic extents. Embedding this metadata ensures compatibility with GIS platforms and prevents spatial misalignment. This alignment ensures that each pixel corresponds to a real geographic location. Once georeferenced, the segmentation mask becomes a valid spatial dataset that can be integrated with other geospatial layers. Accurate georeferencing ensures reliable subsequent urban footprint measurement and expansion analysis.

The raster segmentation masks are suitable for pixel-based analysis but are limited for advanced GIS operations. Vector data structures provide greater flexibility for spatial querying and measurement. Therefore, the georeferenced raster mask is converted into vector polygons through a polygonization process.

Polygonization identifies connected pixel regions representing individual buildings. Each region is converted into a closed polygon feature. These vector footprints enable area calculation, perimeter measurement, and topology analysis.

This step forms a critical bridge between deep learning outputs and structured spatial information systems. By converting raster predictions into vector footprints, the segmentation results become compatible with advanced GIS analysis and urban spatial modeling.

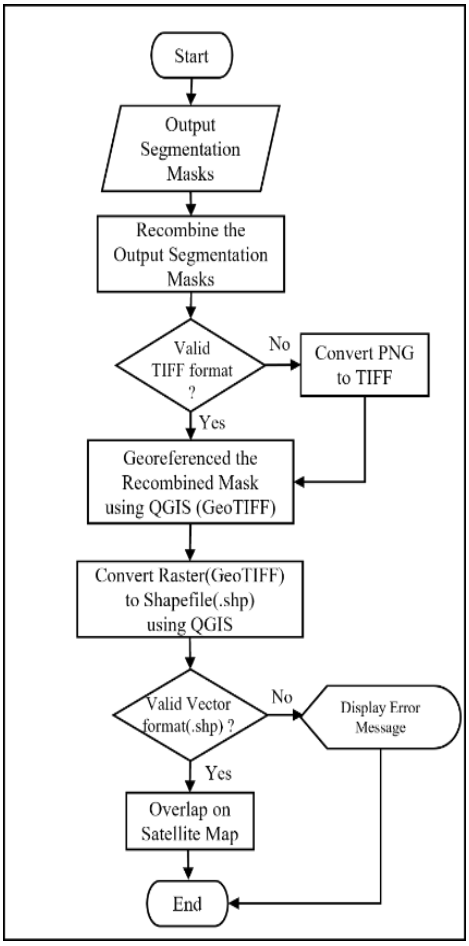



Figure 8. Flowchart for validating and integrating predicted segmentation masks into GIS

The vectorized building polygons are overlaid onto satellite base maps to assess prediction accuracy and spatial consistency. This overlay enables comparison between predicted building regions and visible urban structures within the same geographic coordinate system. When applied across multi-temporal imagery, the workflow facilitates detection of urban expansion patterns.

This integration step converts segmentation outputs into structured geospatial data for long-term urban monitoring. Figure 9 shows a spatial overlay analysis with vectorized building footprints predicted from the segmentation model overlaid on the Google Satellite Map.

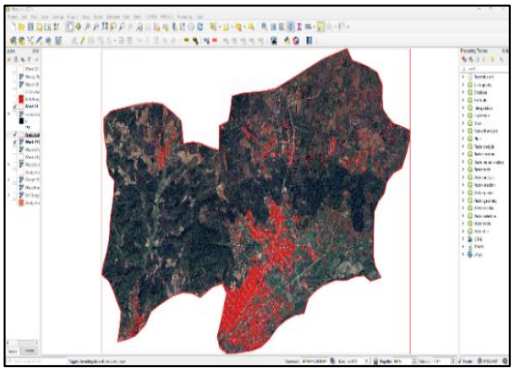


Figure 9. Overlaying the Predicted Building Footprints on the Satellite Map

## 5. EXPERIMENTAL RESULTS

A total of 1530 image mask pairs were tested, and performance was measured using four standard metrics: Intersection over Union (IoU), F1-score, Precision, and Recall. This setup allowed the effect of the enhancement step to be isolated while keeping the network, inference settings, and evaluation consistent.

All experiments were conducted on a GPU-accelerated workstation to expedite convolutional operations and large matrix computations. The system utilized an Intel (R) Core (TM) i7-12650H processor, 8 G RAM, and NVIDIA GeForce RTX 3050 Laptop GPU with CUDA support, which enabled efficient parallel processing during training. The training environment was implemented using Python 3.10 and a deep learning framework compatible with GPU acceleration. Figure 10 shows the results of testing images using the trained U-Net segmentation model.

The segmentation results demonstrate that the proposed model produces geometrically consistent building footprints. The performance of the segmentation model is assessed using standard pixel-level metrics, including Intersection over Union (IoU), precision, recall, and F1-score confusion matrix analysis. Each metric measures a different aspect of segmentation quality. IoU evaluates the spatial overlap between predicted masks and ground truth references.

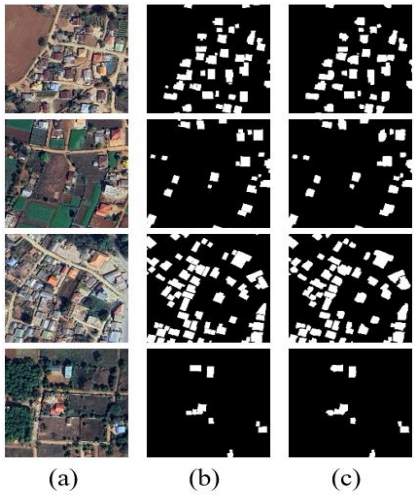



Figure 10. The experimental results (a) Satellite Images (b) Ground Truth masks (c) Predicted masks

The model achieves an IoU of 88.06%, which indicates a high level of spatial agreement. Precision measures the ability to reduce false detections, with a value of 93.42%, while recall reflects the ability to correctly identify building regions, reaching 93.88%. The F1-score provides a balanced evaluation of precision and recall, and the model achieves an F1-score of 93.65%, demonstrating overall robust performance. Figure 11 shows the quantitative performance analysis of the U-Net model for semantic segmentation of building footprints. Furthermore, the confusion matrix analysis shows that the proposed segmentation model can accurately separate building and background regions with low false positive and false negative values. This result indicates that the model is reliable and effective for urban spatial analysis using high-resolution satellite images.

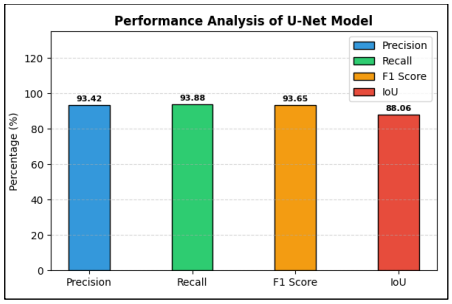



Figure 11. Performance analysis of U-Net segmentation model

Confusion matrices provide deeper insight into classification behavior beyond aggregate accuracy metrics. The matrix summarizes true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The model achieves high accuracy, with a 93.88% true positive rate and a 99.86% true negative rate, demonstrating effective classification of building and background pixels. The false positive rate is 0.14%, showing that very few background pixels are incorrectly labeled as buildings. The false negative rate is 6.12%, which represents missed building pixels. In urban analysis, false negatives are particularly critical because they lead to underestimation of building density. Therefore, reducing false negatives is essential for reliable spatial analysis of urban expansion. Figure 12 shows the results obtained by using a confusion matrix to evaluate the performance of the segmentation model.

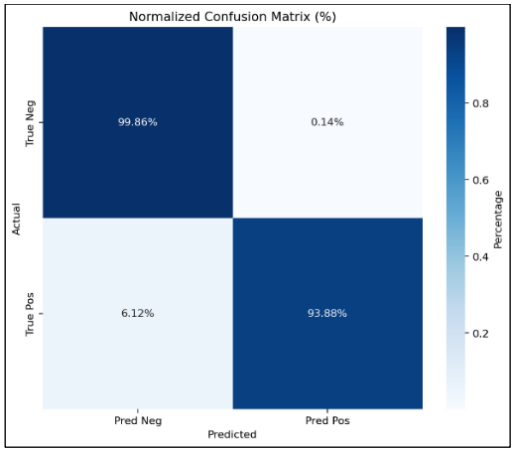



Figure 12. Evaluating performance of the Segmentation Model using confusion matrix

## 6. CONCLUSIONS

This study presented a deep learning-based framework for building footprint extraction from high-resolution satellite images and their integration into a GIS environment for urban spatial analysis. Satellite images were collected using QGIS, and ground truth building masks were generated through manual labelling. After pre-processing the dataset, a U-Net based semantic

segmentation model was trained to automatically detect building regions from satellite imagery. The experimental results demonstrated strong segmentation performance, achieving 93.42% precision, 93.88% recall, 93.65% F1-score, and 88.06% IoU. These results indicate that the proposed framework can accurately identify building structures and generate geometrically consistent building footprints. The extracted building masks were then integrated into a GIS environment for spatial visualization and analysis. The integration of deep learning and GIS technologies provides an efficient and reliable solution for automated urban mapping and spatial assessment. Future work may focus on improving segmentation accuracy, applying advanced neural network architectures, and extending the approach to larger and more complex urban environments.

## ACKNOWLEDGEMENTS

I would prefer to express my sincere gratitude and deep appreciation to Dr. Soe Win Myint, professor and Head of Department of Computer Science at Defence Services Academy (Pyin Oo Lwin). I wish to extend my thankfulness to my supervisor, Dr. Soe Thant, for valuable advice, their close supervision, kind guidance, support, and cooperation. Their energetic and enthusiastic supports are valuable encouragements during a period of study for this paper. Finally, many thanks to all persons who directly and indirectly contributed toward the success of this paper.

## REFERENCES

[1] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. (2018). "*DeepLab: Semantic Image Segmentation with Deep Convolutional Nets",* IEEE TPAMI.

[2] Long, J., Shelhamer, E., & Darrell, T. (2015). "*Fully Convolutional Networks for Semantic Segmentation",* CVPR.

[3] Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2021). "*Image Segmentation Using Deep Learning: A Survey",* IEEE TPAMI.

[4] Ronneberger, O., Fischer, P., & Brox, T. (2015). "*U-Net: Convolutional Networks for Biomedical Image Segmentation",* MICCAI.

[5] M. Buyukdemircioglu, R. Can, S. Kocaman, and M. Kada, *"Deep learning-based building footprint extraction from very high resolution true orthophotos and nDSM,"* ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. V-2-2022, pp. 211-218, 2022.

[6] S. Ji, S. Wei, and M. Lu, *"Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set,"* IEEE Transactions on Geoscience and Remote Sensing, vol. 56, no. 10, pp. 1-13, 2018.

[7] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, *"Convolutional neural networks for large-scale remote sensing image classification,"* IEEE Transactions on Geoscience and Remote Sensing, vol. 55, no. 2, pp. 645-657, Feb. 2017.

[8] W. He, J. Li, W. Cao, L. Zhang, and H. Zhang, *"Building extraction from remote sensing images via an uncertainty-aware network,"* arXiv:2307.12309, 2023.

Kyaw Zin Myat is a Ph.D. candidate at the Department of Computer Science at Defence Services Academy. He received his B.C.Sc degree in 2011 and M.C.Sc degree from Defence Services Academy in 2018.

Dr. Soe Thant is an Associate Professor at the Department of Computer Science at Defence Services Academy. He received his Ph.D degree from the Russian Federation's National Research University of Electronic Technology (MIET).

Zayar Tun is a Ph.D. candidate at the Department of Computer Science at Defence Services Academy. He received his B.C.Tech degree in 2011 and M.C.Tech degree from Defence Services Academy in 2018.

# ABSTRACTIVE TEXT SUMMARIZATION FOR NEWS ARTICLES WITH TRANSFORMER T5 MODEL

May Myat Noe[1], and Thiri Kyaw[2]

[1,2] Faculty of Computer Science, University of Computer Studies (Mandalay)
maymyattnoe6@gmail.com, thirikyawucsm@gmail.com

## ABSTRACT

*Automatic text summarization has become increasingly important as the volume of digital news content continues to grow rapidly. This paper presents a web-based abstractive text summarization system for news articles using the Text-to-Text Transfer Transformer (T5) model. The system employs a T5 model trained on the CNN/Daily Mail dataset, which contains approximately 300,000 news articles commonly used for large-scale summarization tasks. Users can input data either by pasting a news paragraph or uploading a PDF document through an interactive Streamlit interface. The system preprocesses the input by extracting raw text, removing unwanted characters with NLTK tools, and applying SentencePiece tokenization to convert subword units into token IDs. The T5 encoder–decoder architecture then transforms these tokens into vector embeddings and generates coherent summary tokens, which are converted back into readable text. The model produces concise, novel sentences that preserve the original meaning, and the final summary is presented within the Streamlit interface for viewing or downloading.*

## KEYWORDS

*Text Summarization; Text-to-Text Transfer Transformer; Streamlit; Deep Encoder–Decoder.*

## 1. INTRODUCTION

Natural Language Processing (NLP) plays a crucial role in enabling computers to understand and respond to human language. As the volume of digital text continues to increase across industries, organizations rely on NLP software to automatically process this information, analyze intent or sentiment, and respond in real time. Modern applications of NLP include business chatbots and virtual assistants, educational text-summarization tools, information-retrieval systems for question answering, financial fraud detection mechanisms, and machine translation platforms such as Google Translate. These advancements demonstrate the growing need for intelligent systems capable of interpreting and generating human-like language.

Among the various NLP applications, text summarization has become increasingly important. Text summarization techniques automatically generate shortened versions of longer texts, which may come from documents, conversations, news articles, or other written sources. Popular examples of summarization tools include ChatGPT (GPT-3.5 and GPT-4), Google's Gemini (LaMDA/PaLM 2), and QuillBot Summarizer (BERT-based). Text summarization methods fall into two main categories: extractive and abstractive summarization. Extractive summarization selects key sentences from the original text, whereas abstractive summarization generates new sentences that preserve the meaning but may use different wording [1].

Abstractive summarization is considered more advanced and suitable for tasks requiring fluent, natural-sounding output. It is commonly used in chatbot responses, rewriting content to reduce plagiarism, and improving readability for informal learning. Extractive summarization, on the other hand, remains widely used for applications such as customer feedback analysis, meeting minutes, and financial reporting. The evolution of large-scale transformer-based models has greatly improved the performance of abstractive summarization systems, making them more coherent and context-aware [2].

This system introduces a web-based abstractive text summarization tool specifically designed

for news articles using the Transformer-based Text-to-Text Transfer Transformer (T5) model. Trained on the CNN/Daily Mail dataset, the T5 model effectively generates concise and meaningful summaries from lengthy news content. The system aims to provide readers with essential insights without requiring them to read the entire article, thereby reducing misunderstanding and improving information accessibility. By integrating a user-friendly interface dedicated to abstractive summarization of news articles, the system ensures efficient text processing and an enhanced user experience for fast, accurate, and coherent news summarization.

## 2. RELATED WORK

BART [3] is a denoising autoencoder designed for pretraining sequence-to-sequence models by corrupting input text and training the system to reconstruct the original version. Built on a Transformer architecture, it generalizes elements of both BERT's bidirectional encoding and GPT's left-to-right decoding. The model performs best when trained with sentence shuffling and a novel text infilling method that replaces spans with a single mask token. BART excels in text generation tasks but also performs strongly in comprehension benchmarks, matching RoBERTa on GLUE and SQuAD with similar training resources. It achieves state-of-the-art results in abstractive dialogue, question answering, and summarization tasks, improving ROUGE scores by up to six points. Additionally, BART improves machine translation quality by 1.1 BLEU over back-translation systems using only target-language pretraining. Ablation studies further show how different pretraining components contribute to overall task performance.

This paper [4] examines transfer learning in NLP by proposing a unified text-to-text framework that reformulates all language tasks into the same input–output format. The authors conduct a comprehensive comparison of different pre-training objectives, model architectures, types of unlabeled data, and transfer strategies across numerous NLP benchmarks. Using insights gained from these evaluations—together with large-scale training on their newly developed Colossal Clean Crawled Corpus (C4)—the study achieves state-of-the-art performance on tasks such as summarization, question answering, and text classification. The work highlights the power and flexibility of the text-to-text approach in enabling robust transfer learning across diverse NLP problems.

PEGASUS [5] introduces a new self-supervised pre-training objective for large Transformer-based encoder–decoder models by masking or removing the most important sentences from a document and training the model to generate them as a single output sequence. This approach resembles an extractive summary while enabling strong abstractive capabilities. The model is pre-trained on massive text corpora and evaluated across 12 diverse summarization tasks, including news, scientific papers, stories, instructions, emails, patents, and legislative documents. PEGASUS achieves state-of-the-art ROUGE scores on all datasets and demonstrates exceptional performance in low-resource settings, outperforming prior models on six tasks with only 1,000 training examples. Human evaluation further confirms that the generated summaries are comparable to human-written summaries on multiple benchmarks.

This work [6] addresses key limitations of neural sequence-to-sequence models for abstractive summarization, specifically their tendency to generate inaccurate factual details and produce repetitive text. The authors introduce a novel architecture that enhances the standard attentional model in two complementary ways. First, a hybrid pointer-generator mechanism allows the model to copy specific words directly from the source text, improving factual accuracy while still enabling the generation of new, novel phrases. Second, a coverage mechanism tracks which parts of the source have already been summarized, effectively reducing unnecessary repetition. When applied to the CNN/Daily Mail dataset, this combined approach significantly improves summarization quality, outperforming previous abstractive models by more than two ROUGE points.

This paper [7] investigates the limitations of neural abstractive summarization models, finding that they frequently generate hallucinated content that does not accurately reflect the input document. Through a large-scale human evaluation of multiple summarization systems, the study reveals that

substantial amounts of unfaithful information appear across all model-generated summaries. Despite these issues, the analysis shows that pretrained models perform better than earlier approaches, not only in terms of standard metrics like ROUGE but also in producing more factual and faithful summaries according to human judgment. The authors further demonstrate that textual entailment scores correlate more strongly with summary faithfulness than traditional evaluation metrics, highlighting their potential for improving automatic assessment methods and guiding future training and decoding strategies.

This work [8] addresses a major limitation in current summarization evaluation metrics, which fail to measure whether generated summaries are factually consistent with their source documents. The authors introduce a weakly supervised, model-based method for detecting factual consistency and identifying specific conflicts between a summary and its source text. Training data is automatically constructed using rule-based transformations applied to original sentences, enabling scalable supervision. The model is trained jointly on three tasks: determining whether a transformed sentence is factually consistent, identifying supporting evidence spans in the source, and detecting inconsistent spans within the summary when present. When applied to summaries generated by state-of-the-art systems, the approach significantly outperforms prior methods, including models trained with full supervision on natural language inference and fact-checking datasets.

## 3. DATASET

The “Newspaper Text Summarization (CNN-DailyMail)” dataset available on Kaggle is a comprehensive English-language news corpus widely used in natural language processing research, especially for training and evaluating summarization models. It consists of over 300,000 news articles written by professional journalists from CNN and the Daily Mail, each paired with corresponding human-written summary highlights that serve as ground-truth targets for supervised learning. The data is typically split into training, validation, and test subsets with roughly 287,000 articles for training, about 13,000 for validation, and around 11,000 for testing, providing a robust resource for both model training and performance evaluation in diverse experimental setups.

Each instance in the dataset includes an identifier, the full article text as the source input, and the highlights which act as the summary target. These paired texts enable models to learn to map long news documents to concise, meaningful summaries suitable for both extractive and abstractive summarization tasks. Due to its large scale and real-world content, this dataset has become a standard benchmark for sequence-to-sequence and transformer-based models such as T5, BART, and PEGASUS, and is frequently used to compare model architectures, training strategies, and evaluation metrics in summarization research.

## 4. SYSTEM

This flowchart illustrates the technical architecture of a Natural Language Processing (NLP) system designed for text summarization, likely based on a Transformer-based model. The workflow begins with the CNN/Daily Mail dataset, which is partitioned into "Articles" and their corresponding "Target Summaries." The raw data flows into a Data Preprocessing stage, where the text is cleaned and then processed through SentencePiece Tokenization. This step converts human-readable subwords into numerical Token IDs that the machine can understand. Additionally, a Positional Encoding component is indicated, which is essential for providing the model with information regarding the sequence and order of words in the input text.

The core of the architecture features an Encoder-Decoder structure typical of sequence-to-sequence models. The processed data enters the Encoder to create a context representation, which the Decoder then uses to generate an Output Summary. To refine the model's accuracy, an Optimizer compares the generated output against the original "Target Summary" from the dataset, creating a feedback loop for training. Finally, the quality of the generated text is quantitatively measured through a ROUGE Evaluation block, which calculates the overlap between the machine-generated summary and the human-written reference to determine the system's performance.

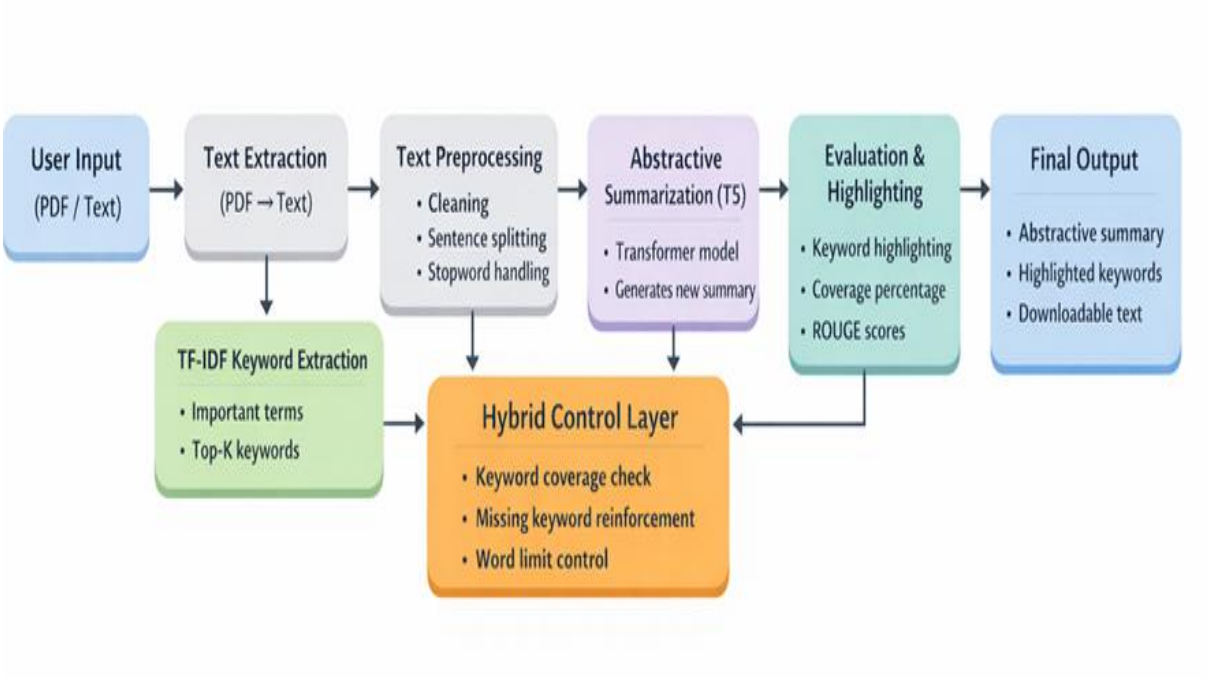



Figure 1. Abstractive Text Summarization for News Article with Transformer T5 Model

The proposed system is a hybrid abstractive text summarization framework specifically designed for news articles, combining statistical keyword extraction techniques with deep learning–based summarization to produce concise, informative, and contextually accurate summaries. The system begins by accepting user input in the form of either a PDF file or pasted text. If a PDF is provided, a dedicated PDF processing module extracts the textual content and converts it into a format suitable for further processing.

Once extracted, the text undergoes light preprocessing, which includes basic cleaning and sentence splitting to structure the text appropriately. Importantly, stop words are retained during this initial stage to preserve the full grammatical structure of the input, which is critical for maintaining sentence coherence and for downstream processing in the abstractive summarization module. This careful preparation ensures that the input data retains its natural linguistic characteristics while eliminating only superficial inconsistencies or formatting artifacts, thereby creating a high-quality foundation for the summarization workflow.

Following preprocessing, the system workflow diverges into two parallel processes that operate in tandem to ensure both semantic coverage and linguistic fluency. In the keyword extraction path, the system applies the term frequency–inverse document frequency (TF-IDF) method to identify the most important keywords in the text. During this stage, stop words are intentionally removed, as they contribute little semantic value and can dilute the significance of meaningful terms. Each term is scored based on its frequency within the document relative to its frequency across a larger corpus, allowing the system to rank and select the top-K keywords that are most indicative of the article's content. These keywords form a reference set that guides the generation of summaries and ensures that key concepts and important phrases are emphasized in the final output, thereby improving the informativeness and relevance of the summaries.

Simultaneously, the original text, including all stop words, is fed into the T5 transformer model within the abstractive summarization path. The model employs an encoder-decoder architecture, wherein the encoder converts the input tokens into contextual embeddings that capture the meaning of each token in relation to the entire article, while the decoder generates the summary token by token using cross-attention mechanisms to focus on the most relevant parts of the text. Preserving stop words in this stage is crucial for grammatical correctness and contextual understanding, enabling the model to produce fluent, coherent sentences that may paraphrase or restructure the input rather than copying it verbatim.

This process allows the system to generate summaries that are not only concise but also linguistically natural and semantically accurate, providing users with outputs that are readable and meaningful.

Once the initial abstractive summary is produced, a hybrid control layer integrates the outputs from both the keyword extraction and abstractive summarization paths. This layer verifies whether the essential TF-IDF–extracted keywords are adequately covered in the generated summary and reinforces any missing keywords if necessary. It also enforces user-defined minimum and maximum word limits to control summary length, ensuring that the summaries are concise yet complete.

Finally, the system highlights the identified keywords in both the original article and the generated summary, calculates keyword coverage metrics, and, if a reference summary is available, evaluates the quality of the generated summary using ROUGE metrics. The final output is a refined, keyword-aware, and contextually accurate abstractive summary that can be viewed by the user or downloaded for further use, effectively combining statistical keyword importance with deep learning–based sentence generation to deliver high-quality, informative summaries for news articles.

## 5. SYSTEM IMPLEMENTATION

The evaluation of experimental results for the abstractive text summarization system using the Transformer-based T5 model focuses on measuring how effectively the model generates coherent, informative, and concise summaries of news articles. This assessment examines both quantitative performance, using standard evaluation metrics such as ROUGE-1, ROUGE-2, and ROUGE-L, and qualitative performance, by analyzing the readability, fluency, and semantic accuracy of the produced summaries. By comparing the generated outputs with human-written references and observing how the model handles varying article lengths, structures, and writing styles, the evaluation aims to determine the strengths, limitations, and overall reliability of the T5 model for real-world news summarization tasks.

The system categorizes news articles into three length-based groups—300–500 words, 700–900 words, and 1000–1300 words—to ensure that each type of content is processed with an appropriate summarization strategy for system implementation. For shorter 300–500-word articles, the system uses lightweight keyword extraction and a smaller T5 model to produce fast, concise summaries. For medium-length articles of 700–900 words, it applies both keyword and entity extraction with a T5 Base model to preserve context, background information, and important details. For long articles ranging from 1000–1300 words, the system uses chunking techniques, topic modeling, and a larger T5 model to handle extended context and merge multiple sub-summaries into a coherent final summary. This three-tier design allows the system to adapt to different content sizes and maintain accuracy, clarity, and relevance across all types of news articles.

When comparing the three averaged performance results calculated by this system, it becomes clear that news articles around the 800-word range consistently achieve the highest overall ROUGE average score among the tested length categories. This indicates that the system can capture the main ideas, maintain contextual accuracy, and preserve important relationships within the text more effectively than it does with shorter or longer articles.

The strong ROUGE score shows that the generated summary aligns closely with the essential content of the original article, reflecting better overlap in keywords, phrases, and meaningful semantic structures. Because this mid-range length produces noticeably better performance, it can be concluded that this system works most efficiently with news articles in the 700–900-word range, where the information is well-balanced—not too brief and not overly extensive.

In this range, the article provides enough detail for the model to extract valuable patterns and insights, yet remains manageable enough for the summarization process to maintain coherence and relevance. Therefore, based on the averaged ROUGE evaluation, this system performs with the highest accuracy, stability, and summarization quality when processing medium-length news articles between 700 and 900 words.

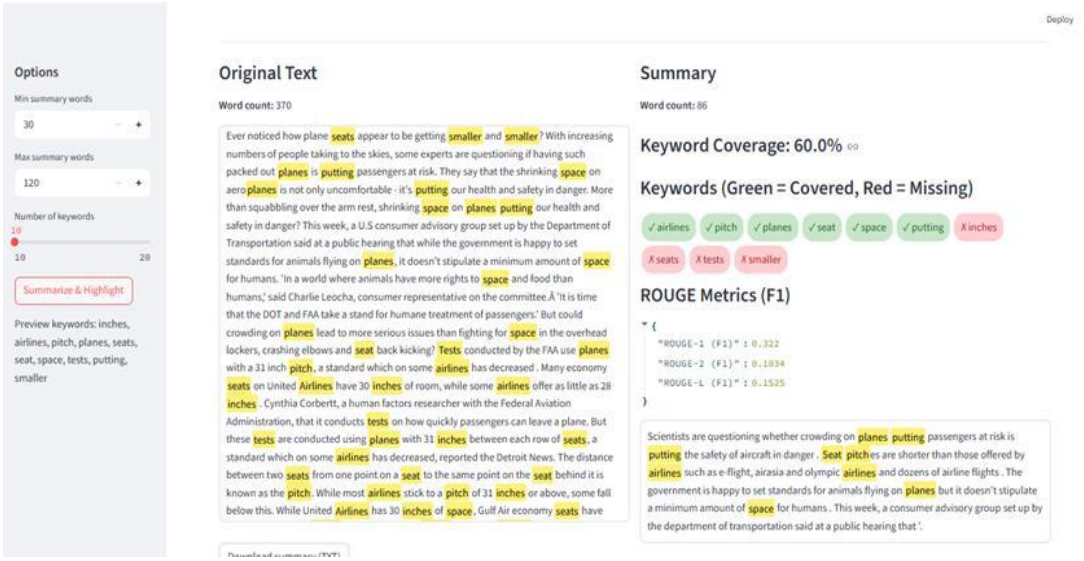



Figure 2. Results for Lengths 300-500

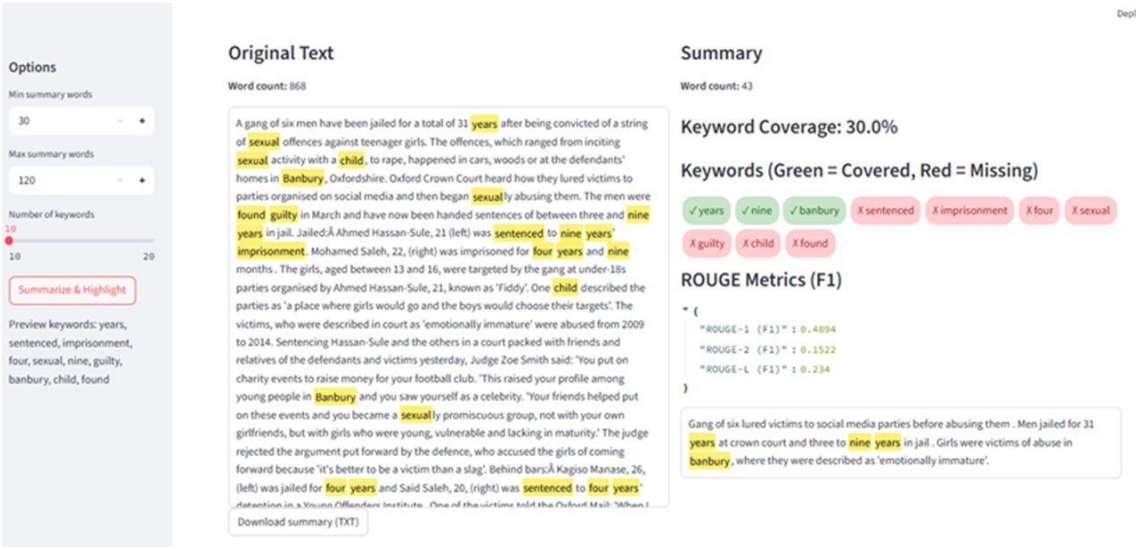



Figure 3. Results for Lengths 700-900

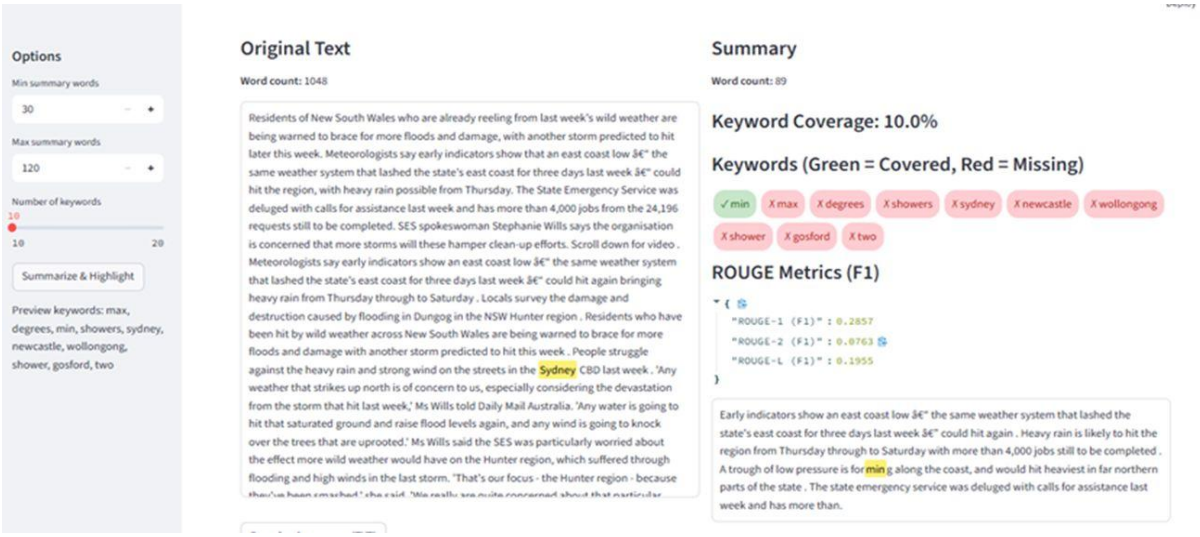



Figure 4. Results for Lengths 1000-1300

## 6. CONCLUSION

In conclusion, the Transformer T5 model demonstrates a highly effective approach to abstractive text summarization for news articles, emphasizing the generation of concise, coherent, and human-like summaries. Unlike extractive methods that rely on directly copying sentences or phrases from the original text, the T5 model creates summaries that rephrase and restructure content while preserving the essential meaning. This capability ensures that the output is not only shorter but also more readable and natural, which is particularly valuable in news contexts where clarity and fluency are critical. The encoder-decoder architecture of the T5 model plays a central role in this process, as it allows the system to fully capture contextual relationships between words and sentences in the source text. The encoder encodes the input article into a rich, contextual representation, while the decoder leverages this representation to generate summaries that maintain semantic coherence and relevance throughout. As a result, readers receive summaries that highlight the main points of the article in a form that is easy to comprehend, reducing cognitive load and enabling faster information processing. Furthermore, the T5-based abstractive summarization system provides flexibility in style and tone, allowing for summaries that can be tailored to different audiences or levels of detail. By transforming complex or lengthy news content into digestible summaries, this approach contributes significantly to information accessibility, especially in today's fast-paced digital environment. Overall, the application of the T5 model in news summarization represents a significant advancement in natural language processing, combining the strengths of deep learning with linguistic understanding to create summaries that are not only informative but also fluent, contextually accurate, and reader-friendly, ultimately enhancing the efficiency and quality of news consumption.

## REFERENCES

[1] M. Lewis et al., "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in Proc. 58th Annu. Meeting Assoc. Comput. Linguistics (ACL), 2020, pp. 7871–7880, doi: 10.18653/v1/2020.acl-main.703.

[2] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," J. Mach. Learn. Res., vol. 21, no. 140, pp. 1–67, 2020.

[3] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization," in Proc. 37th Int. Conf. Mach. Learn. (ICML), 2020, pp. 11328–11339.

[4] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in Proc. 55th Annu. Meeting Assoc. Comput. Linguistics (ACL), 2017, pp. 1073–1083, doi: 10.18653/v1/P17-1099.

[5] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, "On faithfulness and factuality in abstractive summarization," in Proc. 58th Annu. Meeting Assoc. Comput. Linguistics (ACL), 2020, pp. 1906–1919, doi: 10.18653/v1/2020.acl-main.174.

[6] W. Kryściński, B. McCann, C. Xiong, and R. Socher, "Evaluating the factual consistency of abstractive text summarization," in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP/IJCNLP), 2019, pp. 933–938, doi: 10.18653/v1/D19-1092.

[7] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2019, pp. 3730–3740, doi: 10.18653/v1/D19-1382.

[8] S. Gehrmann, Y. Deng, and A. M. Rush, "Bottom-up abstractive summarization," in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2018, pp. 4098–4109, doi: 10.18653/v1/D18-1450.

# DESIGN AND EXPERIMENTAL STUDY OF A SEQUENTIAL BLINKING LED CIRCUIT USING HEART GEOMETRY

Shwe Sin Win[1], Chit Myo Naing[2], and Cherry Phyo[3]

[1,2,3] Department of Engineering Physics, Myanmar Aerospace Engineering University
*sshwesin75@gmail.com, chitmyonaing1992@gmail.com, cherryphyo2101995@gmail.com*

***ABSTRACT***

*A sequential blinking LED circuit based on heart geometry is designed and experimentally analyzed to achieve both functional performance and visual representation. The system utilizes a 555 timer IC operating in astable mode to generate uniform clock pulses, which are applied to a CD4017 decade counter for sequential control of light-emitting diodes. Red LEDs are arranged in a heart-shaped pattern to enhance aesthetic appearance and symbolic expression. With each incoming clock pulse, the counter advances its output, producing a smooth and continuous blinking sequence across the LED array. The circuit is characterized by simple construction, low cost, and minimal component requirements, making it suitable for educational experiments, hobbyist projects, and decorative electronic displays. Experimental results demonstrate stable timing behavior, reliable sequential operation, and consistent illumination. The study confirms that basic timing and counting integrated circuits can be effectively employed to develop creative and visually engaging electronic systems with dependable performance.*

***KEYWORDS***

*Sequential Blinking LED, 555 Timer IC, CD4017 Counter, LED Chaser Circuit, Heart Geometry.*

## 1. INTRODUCTION

Sequential blinking LED circuits play an important role in basic electronic system design because of their simple architecture, dependable operation, and low implementation cost. Such circuits are widely adopted in educational laboratories, training modules, and decorative lighting applications, where they serve as effective tools for demonstrating fundamental timing and counting concepts. By integrating timing and counting devices, orderly illumination sequences can be achieved without the need for complex control units or programmable hardware [1].

The 555 timer integrated circuit continues to be one of the most widely used components for clock signal generation. When operated in astable mode, it produces a stable and continuous square-wave output with adjustable frequency, which is well suited for sequential control applications [2]. To distribute these clock pulses, the CD4017 CMOS decade counter is commonly employed. This device offers ten decoded outputs that are activated sequentially in response to each input pulse, enabling smooth and systematic LED switching. Its low power consumption, ease of availability, and economical cost make it particularly suitable for LED chaser and display circuits [3].

Most existing implementations emphasize linear or basic arrangements of LEDs, with limited focus on geometric or symbolic patterns. Incorporating shape-based LED configurations can enhance visual appeal while retaining functional simplicity. In this context, the present work introduces a heart-shaped sequential blinking LED circuit that combines fundamental electronic principles with expressive and visually engaging design.

## 2. RELATED WORKS

Extensive research has been conducted on sequential LED circuits that employ discrete timing and counting components, particularly for educational, demonstration, and low-cost electronic applications. Early investigations

highlighted the effectiveness of using the 555 timer integrated circuit as a clock pulse generator. When configured in astable mode, the 555 timer produces a stable and continuous square-wave output, making it suitable for driving digital counters and sequential display systems. Due to its straightforward design, ease of configuration, and reliable performance, the 555 timer has remained a fundamental building block in introductory and applied electronic circuit designs [4].

In sequential control applications, the CD4017 CMOS decade counter has been widely studied and adopted. This integrated circuit provides ten decoded outputs that are activated sequentially in response to incoming clock pulses, allowing direct interfacing with light-emitting diodes without additional decoding circuitry. Several studies have demonstrated the successful implementation of the CD4017 in LED chaser circuits, running light displays, and basic automation systems. Researchers have emphasized its low power consumption, high noise immunity, and cost efficiency, which make it especially suitable for non-programmable and resource-constrained applications [5]. These advantages have led to its continued use in both academic experiments and practical electronic projects.

More recent research has shifted attention toward decorative and display-oriented LED systems that focus on visual appeal in addition to functional performance. Studies involving geometric and pattern-based LED arrangements suggest that shape-oriented designs can significantly improve aesthetic quality and user engagement in electronic displays [6]. Such configurations are

Table 1. Summary of Related Works on Sequential LED and Timing Circuits

| Author(s) | Method | Limitation |
|---|---|---|
| Boylestad and Nashelsky [1] | Presented fundamental electronic devices and circuit theory, including timer and counter-based sequential logic concepts used in LED applications. | Focused on theoretical analysis with limited emphasis on practical or decorative implementations. |
| Horowitz and Hill [2] | Discussed practical design techniques for timing circuits using the 555 timer and CMOS logic devices. | Designs are primarily generic and not optimized for visual or geometric LED displays. |
| Selvam [4] | Designed functional circuits using the 555 timer IC in astable and monostable configurations for timing and sequencing applications. | Concentrated on functional performance without addressing aesthetic or symbolic LED arrangements. |
| Liu *et al.* [6] | Developed LED lighting systems for decorative and landscape applications, emphasizing visual appearance and illumination effects. | Relied on complex control schemes, increasing system cost and design complexity. |
| Bilgic and Ozev [7] | Proposed low-cost circuit monitoring techniques for reliable analog and mixed-signal systems. | Focused on circuit reliability rather than visual display or LED sequencing patterns. |
| Franco and Malek [9] | Investigated low-power CMOS decade counters for sequential logic and control applications. | Did not explore LED-based visual output or geometric pattern representation. |
| Ahmed *et al.* [11] | Implemented low-cost LED display systems using digital control techniques for visual indication. | Mostly employed linear or matrix LED arrangements without symbolic geometry. |
| Proposed Work | Utilizes a 555 timer and CD4017 counter to implement a sequential blinking LED circuit arranged in heart geometry. | Limited to fixed geometric patterns and does not support programmable or adaptive sequencing. |

commonly employed in decorative lighting, artistic installations, and expressive electronic systems. However, many reported designs rely on microcontrollers or programmable logic devices to achieve complex patterns, which increases system complexity, power consumption, and overall cost.

To address these challenges, some conference-level works have proposed combining traditional timer and counter architectures with geometric LED layouts. These approaches aim to retain the simplicity and affordability of discrete components while achieving visually expressive lighting effects [7]. Building on these earlier contributions, the present work integrates heart-shaped LED geometry with a conventional 555 timer and CD4017-based sequential control circuit. This approach maintains low complexity and cost while providing experimental validation and enhanced visual presentation, thereby extending the scope of traditional sequential LED circuit design. A comparative summary of related works is presented in Table 1.

## 3. BACKGROUND THEORY

The sequential blinking LED circuit is developed using basic electronic components that support prototyping, timing control, sequential counting, and visual indication. A breadboard is used as the primary platform for circuit assembly, as it allows temporary connections without soldering and enables easy modification during testing. The internal conductive paths of the breadboard provide predefined electrical connections between inserted components, as shown in Figure 1 [8].

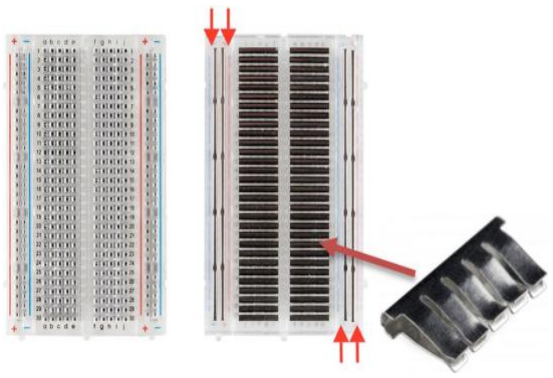


Figure 1. Internal connection diagram of a breadboard.

The CD4017 CMOS decade counter plays a key role in achieving sequential LED operation. It is a 16-pin integrated circuit that produces ten decoded outputs, which are activated one at a time for each incoming clock pulse. This feature allows LEDs to be driven directly in sequence without the need for additional decoding circuitry. Due to its low power consumption and simplicity, the CD4017 is widely used in LED chaser and display applications Figure 2 [9].

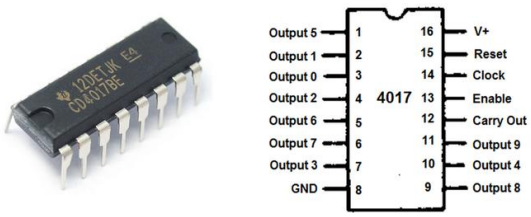



Figure 2. CD4017 CMOS decade counter IC and pin configuration.

The 555 timer integrated circuit is used as a clock pulse generator in the proposed system. When configured in astable mode, the 555

timer produces a continuous square-wave output whose frequency depends on the external resistor and capacitor values. These pulses are supplied to the clock input of the CD4017 to control the blinking speed of the LEDs, as illustrated in Figure 3 [10].

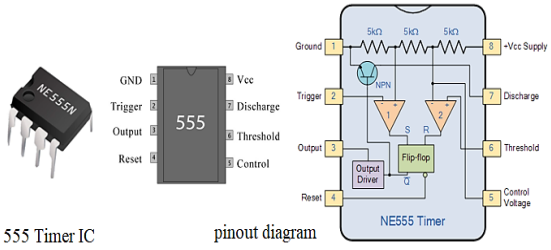



Figure 3. NE555 timer IC with pinout diagram.

Capacitors are used for timing and filtering purposes, while resistors limit current flow and protect circuit components. A variable resistor (trimpot) enables fine adjustment of the pulse frequency, allowing control over the LED blinking rate Figure 4,5 and 6. A 9 V battery

provides portable power to the circuit Figure7. Light-emitting diodes arranged in a heart-shaped pattern act as visual indicators, and jumper wires are used to interconnect components on the breadboard Figure 8 and 9 [11].

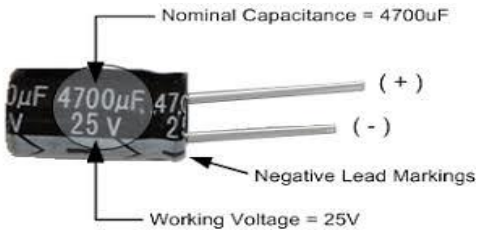



Figure 4. Radial electrolytic and polyester capacitors.

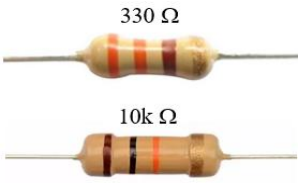



Figure 5. Fixed resistor used in the circuit.

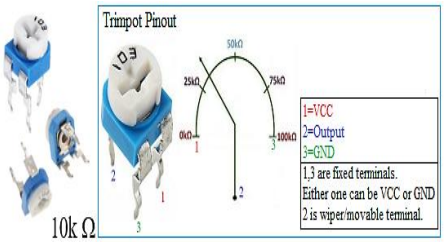



Figure 6. Preset potentiometer (Trimpot) and pin configuration.

Figure 7. 9 V battery and battery connector.

Figure 8. Red light-emitting diodes (LEDs).

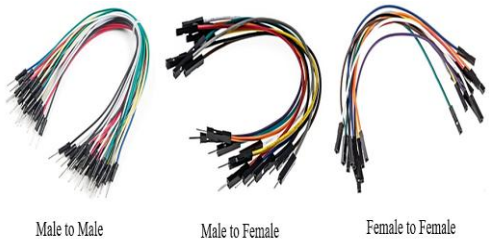



Figure 9. Types of breadboard jumper wires.

## 4. METHODOLOGY

The proposed sequential blinking LED circuit employs a straightforward timer–counter architecture that bridges fundamental electronic principles with practical visual output. This design builds upon established techniques in timing and sequential logic circuits while addressing the need for expressive LED arrangements. The system integrates a 555 timer IC configured in astable mode to generate continuous clock pulses and a CD4017 CMOS decade counter to manage the sequential activation of LEDs. This combination enables ordered illumination with minimal hardware complexity, eliminating the need for programmable components commonly used in more advanced display systems [12].

In this methodology, the 555 timer functions as a free-running oscillator. By selecting appropriate values for resistors and capacitors, the timer produces a stable square-wave with a frequency that dictates the switching rate of the sequential display. The astable operation of the 555 timer is well documented for its reliability and flexibility in timing circuits, allowing designers to adjust output frequency without additional control logic [13]. The clock pulses generated in this manner are routed directly to the clock input of the CD4017 decade counter, ensuring a synchronous sequence of output transitions.

The CD4017 decade counter interprets each incoming clock pulse and increments the active output in a ripple sequence from Q0 through Q9. Each counter output is connected to a specific LED positioned within a predefined geometric layout. In the current design, the LEDs are arranged to form a heart shape, with individual outputs driving corresponding diodes. As the counter

advances, only one LED at a time is energized, creating a perceivable pattern of sequential illumination that follows the heart geometry. Upon reaching the final output, the counter automatically resets and initiates the next illumination cycle without external intervention [14].

This methodology intentionally avoids the use of microcontrollers or programmable logic devices, thereby reducing system cost, power consumption, and design complexity. Instead, it leverages discrete analog and digital ICs, which are widely available and simple to implement. Experimental evaluation of the assembled circuit verified uniform LED switching, stable timing behavior, and consistent visual performance throughout extended operation. By integrating a geometric LED arrangement with established timer–counter circuitry, the proposed approach demonstrates a balanced integration of functional sequencing and aesthetic expression, aligning with the objectives of this study and reinforcing the practical applicability of traditional electronic design techniques [15].

## 5. System Design, Results and Discussion

### 5.1. System Design

The proposed system is designed using an NE555 timer IC and a CD4017 CMOS decade counter to achieve sequential LED operation with minimal circuit complexity. The NE555 timer is configured in astable mode and operates as a clock pulse generator. Its output produces a continuous square-wave signal, which is applied directly to the clock input (CLK) of the CD4017 decade counter. The complete circuit configuration is illustrated in Figure 10**.**

The oscillation frequency of the NE555 timer is determined by a fixed 10 kΩ resistor, a 10 kΩ trimmer potentiometer, and a 10 µF timing capacitor. This arrangement allows fine adjustment of the clock frequency, thereby controlling the speed of LED blinking. The supply voltage is applied to pin 8 (VCC), while pin 1 (GND) is connected to ground. To prevent unintended resets, the reset pin (pin 4) is tied directly to the positive supply. Since the control voltage pin (pin 5) is not actively used, a 0.01 µF capacitor is connected to ground to suppress high-frequency noise and ensure stable operation.

The CD4017 decade counter receives the clock pulses from the NE555 timer at pin 14. With each rising edge of the clock signal, the counter advances its active output sequentially. The clock enable pin (pin 13), which is active low, is connected to ground to allow continuous counting. Power is supplied to the IC through pin 16 (VDD), while pin 8 (VSS) is connected to ground. Selected decoded outputs are connected to three groups of LEDs arranged in a heart-shaped geometry. A 330 Ω resistor is used for current limiting, and since only one LED group is active at any given time, a single resistor is sufficient for safe operation.

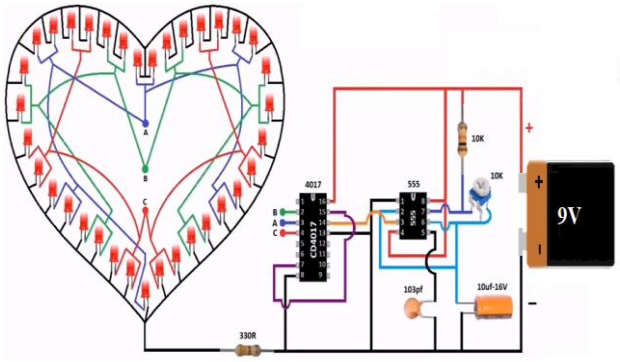



Figure 10. Circuit Diagram

### 5.2. Results and Discussion

After completing the circuit implementation, the system successfully demonstrates a sequential blinking pattern across the LED groups. As the clock pulses from the NE555 timer drive the CD4017 counter, the LEDs illuminate in a chasing sequence that follows the predefined heart-shaped layout. This visual output is shown in Figure 11, where the sequential activation creates the appearance of running lights within the heart geometry.

The experimental results confirm stable timing behavior and consistent LED switching without missed counts or overlapping outputs. The use of discrete logic components

eliminates the need for microcontrollers while maintaining reliable performance and low power consumption. The adjustable trimmer potentiometer allows smooth control of the blinking speed, enhancing the visual effect. Overall, the system achieves its design objective by combining simple timer–counter architecture with an aesthetic LED arrangement, demonstrating both functional correctness and effective visual presentation.

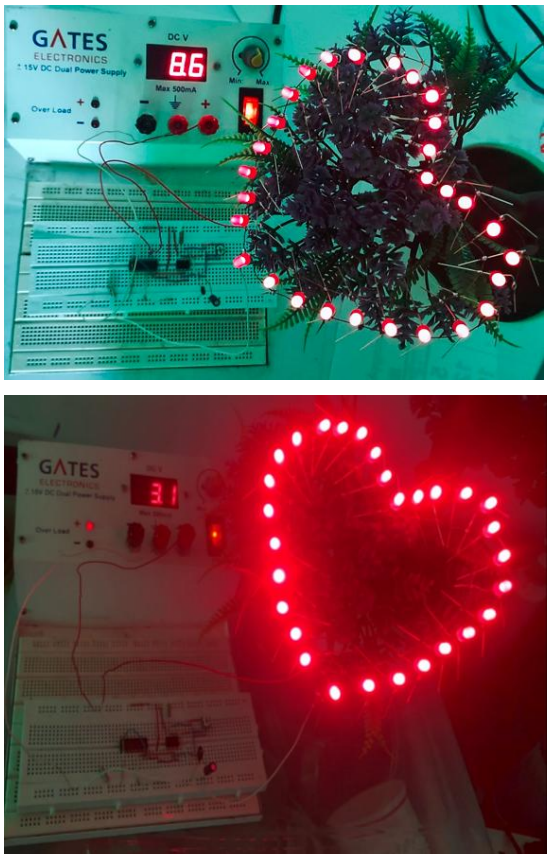



Figure 11. Sequential Blinking LED Circuit Using Heart Geometry

## 6. CONCLUSION

The sequential blinking LED circuit using heart geometry was successfully designed and tested. The 555 timer IC provided stable clock pulses, while the CD4017 decade counter enabled accurate sequential illumination of the LEDs. The experimental results confirmed reliable operation and visually smooth blinking performance. Due to its simplicity, low cost, and ease of implementation, the circuit is suitable for educational and decorative applications. This work demonstrates the effective use of fundamental electronic components in creating expressive and functional display systems.

## ACKNOWLEDGEMENTS

The authors would like to acknowledge the Myanmar Aerospace Engineering University's ERC committee for granting permission to submit this paper.

## REFERENCES

[1] Boylestad, R. L., & Nashelsky, L. (2011). *Electronic devices and circuit theory*. Pearson.

[2] P. Horowitz and W. Hill, *The Art of Electronics*, 3rd ed. Cambridge, U.K.: Cambridge University Press, 2015.

[3] Texas Instruments, "CD4017B CMOS Decade Counter," Datasheet, Dallas, TX, USA, 2019.

[4] Selvam, K. C. (2023). *Design of Function Circuits with 555 Timer Integrated Circuit*. CRC Press.

[5] Gray, P. R., Hurst, P. J., Lewis, S. H., & Meyer, R. G. (2024). *Analysis and design of analog integrated circuits*. John Wiley & Sons.

[6] Liu, H., Cheng, J., & Yusuf, A. H. (2020). Design of Light Emitting Diodes (LEDs) Lighting System and Its Application in Garden Landscape Decoration. *Journal of Nanoelectronics and Optoelectronics*, *15*(6), 734-742.

[7] Bilgic, B., & Ozev, S. (2023). Low-cost structural monitoring of analog circuits for secure and reliable operation. *IEEE Design & Test*, *40*(4), 5-16.

[8] Peppler, K. A., Sedas, R. M., & Thompson, N. (2023). Paper circuits vs. breadboards: Materializing learners' powerful ideas around circuitry and layout design. *Journal of Science Education and Technology*, *32*(4), 469-492.

[9] S. Franco and M. Malek, "Low-power CMOS decade counters for sequential logic applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 6, pp. 1765–1773, Jun. 2014.

[10] G. W. Roberts, "Design considerations for timer-based oscillator circuits," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 7, pp. 2472–2479, Jul. 2011.

[11] M. R. Ahmed, M. S. Islam, and T. Hasan, "Design and implementation of low-cost LED display systems," *IEEE Access*, vol. 8, pp. 145321–145330, 2020.

[12] Perumal, S. R., & Baharum, F. (2022). Design and simulation of a circadian lighting control system using fuzzy logic controller for LED lighting technology. *Journal of Daylighting*, *9*(1), 64-82.

[13] Esiri, A. M., Okayim, P. E., & Aloamaka, A. C. (2024). Design and implementation of an astable multivibrator.

[14] Hamid, S. S., Mariappan, S., Rajendran, J., Rawat, A. S., Rhaffor, N. A., Kumar, N., ... & Yarman, B. S. (2023). A state-of-the-art review on CMOS radio frequency power amplifiers for wireless communication systems. *Micromachines*, *14*(8), 1551.

[15] Ruo Roch, M., & Martina, M. (2022). VirtLAB: A low-cost platform for electronics lab experiments. *Sensors*, *22*(13), 4840.

**Authors**

**Biography**

**Shwe Sin Win** received the M.Sc. degree in Physics from Meikhtila University. She is currently serving as a Lecturer in the Department of Engineering Physics at Myanmar Aerospace Engineering University (MAEU).

**Chit Myo Naing** received the M.Sc. degree in Physics from Banmaw University and a Diploma in Computer Science (D.C.Sc.) from the University of Computer Studies, Banmaw. He later earned a Master of Business Administration (MBA) degree from Monywa University of Economics. He is also a Lecturer in the Department of Engineering Physics at Myanmar Aerospace Engineering University (MAEU).

**Cherry Phyo** received the M.Sc. degree in Physics from Magway University. She is also a Tutor in the Department of Engineering Physics at Myanmar Aerospace Engineering University (MAEU).

# DESIGN AND IMPLEMENTATION OF COLOR PICKER ROBOT USING INVERSE KINEMATICS METHOD

Hnin Thuzar Win[1], and Mie Mie Oo[2]

[1,2]Faculty of Computer Systems and Technologies, University of Computer Studies (Mandalay)

hninthuzarwin801@gmail.com, mimioo.mdy@gmail.com

## ABSTRACT

*The design of a Color Picker Robot using a 6DOF robotic arm managed by a Raspberry Pi 4B is presented in this paper. The approach overcomes common color-based sorting drawbacks like restricted dynamic performance, difficulties with multi-DOF control, and decreased accuracy under varied lighting. This system combines three methods to enhance accuracy and reliability: Inverse Kinematics (IK) for accurate joint angle calculation, Direct RGB Color Detection for real-time classification, and Computed Torque Control (CTC) for consistent trajectory tracking. When combined, these methods provide reliable, highly accurate colored object detection, classification, and manipulation. This paper provides a scalable solution for industrial automation tasks including packing, recycling, and quality control by increasing efficiency, lowering human error, and successfully adapting to changing circumstances.*

## KEYWORDS

*Color Picker Robot, 6DOF Robotic Arm, Raspberry Pi 4B, Inverse Kinematics, RGB Color Detection, Computed Torque Control, Industrial Automation*

## 1. INTRODUCTION

With its integration of computer vision, electronics, mechanical design, and sophisticated control techniques, robotics has emerged as one of the most dynamic and significant areas of contemporary engineering. Color picker robots are essential to packaging, recycling, and quality control procedures in industrial automation. In order to reduce human labor, minimize errors, and increase overall efficiency, these systems are made to precisely control items based on their color. Conventional color sorting systems are helpful, but they frequently have drawbacks like poor dynamic performance when handling objects at different speeds, decreased accuracy under changing lighting conditions, and challenges controlling robotic arms with multiple degrees of freedom (DOF).

This paper presents a Color Picker Robot that uses a 6DOF robotic arm managed by a Raspberry Pi 4B in order to address these issues. Three cutting-edge approaches are integrated into the system to provide dependable performance. First, the precise joint angles needed for the robotic arm's precise positioning are determined using Inverse Kinematics (IK). Second, a vision sensor's color readings are used to categorize objects in real time using Direct RGB Color Detection. Lastly, to guarantee dynamic stability and accurate trajectory tracking while performing manipulation tasks, Computed Torque Control (CTC) is employed[5][7].

The suggested robot can recognize, identify, and manipulate colored objects with high accuracy and resilience by integrating these methods. In addition to increasing the dependability of color-based item sorting,

this integration benefits industrial applications by increasing productivity, lowering human error, and successfully adjusting to changing conditions.

## 2. Literature Review

HSV segmentation and OpenCV contour detection were used in *AI-based Object Color Detection and Sorting Automation* (IJARIIE, Vol. 11, Issue 3, 2025) to categorize and sort objects based only on their dominant color attributes. The resilience of HSV-based color processing for real-time automation was confirmed by the system's 97% detection accuracy and 95% sorting accuracy under typical illumination circumstances. Another example of how HSV thresholding might facilitate autonomous decision-making without human intervention is the incorporation of a fallback mechanism for unknown hues.

A Bluetooth-controlled robotic arm driven by a PIC microcontroller is highlighted in *Design and Development of 7 Degrees of Freedom Robotic Arm for Industrial Application* (JETIR, Vol. 6, Issue 5, 2019), demonstrating how embedded systems may efficiently handle signal processing and motor actuation. When compared to 4-DOF or 5-DOF arms, the system's smoother and more complicated motions are achieved by expanding the design to 7 DOF, proving how embedded microcontroller-based control can improve autonomy and adaptability for real-time industrial applications.

To address uncertainties in robotic manipulators, adaptive control strategies have been developed. In *A Novel Formulation for Adaptive Computed Torque Control Enabling Low Feedback Gains in Highly Dynamical Tasks* (IEEE Access, 2025), a new approach is proposed that eliminates instability from mass matrix inversion while ensuring asymptotic stability in both joint and Cartesian spaces. This approach, validated by simulations and experiments on a 7-DOF robot, achieves exact trajectory tracking with low feedback gains, demonstrating its effectiveness in highly dynamic applications such as object throwing.

Ben-Ari and Mondada's *Elements of Robotics* provides practical insights into robot architecture, sensor integration, and motion planning, bridging theoretical modelling with real-world implementation. Their work is highly relevant for scalable robotic systems that require both mechanical design and vision-based control.

Successful robotic systems rely on kinematic modeling, feedback control, and color-based detection, according to a recurrent theme in the literature. Few studies provide a completely integrated framework with real-time performance, despite the fact that many validate these components independently. The innovative integration of inverse kinematics, direct RGB detection, and computed torque control in the proposed color picker robot is highlighted by the unresolved challenges of dynamic thresholding under uncontrolled lighting and adaptive control tuning.

## 3. Methodology

Robotic manipulators are mechanical systems composed of links and joints that define their degrees of freedom (DOF), enabling motion similar to human arms. A 6-DOF configuration provides full spatial mobility, consisting of three translational and three rotational movements. This structure offers a balance between mechanical complexity and computational efficiency, making it suitable for advanced tasks such as color-based sorting. In this study, a 6-DOF serial manipulator is employed to achieve high accuracy, flexibility, and efficiency in real-time industrial applications.

### 3.1. Kinematics and Inverse Kinematics for 6DOF Color Picker Robot

Kinematics defines the relationship between joint parameters and the position and orientation of the end-effector. Forward kinematics determines the end-effector pose

from given joint angles, whereas inverse kinematics (IK) computes the required joint angles for a desired target position. The Denavit–Hartenberg (DH) convention is used for systematic modelling of the manipulator. For precise color picking operations, IK is essential to ensure accurate positioning and orientation of the robot[4][6].

The wrist center position is calculated as:

$$P_w = P - d_6.R.\hat{z} \quad (1)$$

The first three joint angles are computed as follows:

$$Base\ Rotation = \tan^{-1}\left(\frac{y_w}{x_w}\right) \quad (2)$$

$$Shoulder = \tan^{-1}\left(\frac{z_w - d_1}{\sqrt{x_w^2 - y_w^2}}\right) - \cos^{-1}\left(\frac{a_2^2 + r^2 - a_3^2}{2a_2 r}\right) \quad (3)$$

$$Elbow = \pi - \cos^{-1}\left(\frac{a_2^2 + a_3^2 - r^2}{2a_2 a_3}\right) \quad (4)$$

The distance from the shoulder joint to the wrist center is:

$$distance = \sqrt{x_w^2 + y_w^2 + (z_w - d_1)^2} \quad (5)$$

The wrist orientation matrix $R_{wrist}$ (from frame 3 to frame 6) is used to compute the remaining joint angles:

$$wrist\ yaw = \tan^{-1}\left(\frac{R_{wrist}(2,3)}{R_{wrist}(1,3)}\right) \quad (6)$$

$$wrist\ pitch = \tan^{-1}\left(\frac{\sqrt{R_{wrist}(1,3)^2 + R_{wrist}(2,3)^2}}{R_{wrist}(3,3)}\right) \quad (7)$$

$$wrist\ roll = \tan^{-1}\left(\frac{R_{wrist}(3,2)}{-R_{wrist}(3,1)}\right) \quad (8)$$

Where:

- $d_1, d_6$ : Link offsets
- $R$ : End-effector rotation matrix
- $\hat{z}$ is unit vector along z-axis
- $a_2, a_3$ : link lengths
- $R_{wrist}$ : Wrist rotation matrix

## 3.4 Computed Torque Control and Applications

Computed Torque Control (CTC) is a nonlinear control strategy that utilizes the dynamic model of the robot to achieve precise trajectory tracking. It compensates for nonlinear effects such as inertia, Coriolis, centrifugal, and gravitational forces, effectively linearizing the system[4].

The control input torque is given by:

$$\tau = \tilde{M}(\theta)(\ddot{\theta}_d + K_p\theta_e + K_i\int \theta_e(t)dt + K_d\dot{\theta}_e) + \tilde{h}(\theta, \dot{\theta}) \quad (9)$$

This results in linearized closed-loop error dynamics:

$$\ddot{e} + K_d\dot{e} + K_p e = 0 \quad (10)$$

Where:

- $\theta, \dot{\theta}, \ddot{\theta}_d$ are desired joint position, velocity, acceleration
- $\int \theta_e(t)$ is the integral of the error
- $\tilde{M}(\theta)$ is estimated/modeled mass matrix
- $\tilde{h}(\theta, \dot{\theta})$ is estimated nonlinear dynamics
- $K_p$ is the proportional gain
- $K_i$ is the integral gain
- $K_d$ is the derivative gain

CTC is widely used in robotic systems requiring high precision, such as industrial automation tasks including assembly, welding, and material handling. In this work, it ensures smooth motion and accurate positioning for the 6-DOF color sorting robot.

## 3.5 Direct RGB color Detection

The direct RGB color detection method measures red, green, and blue components directly from an RGB sensor without converting to other color spaces (e.g., HSV). The sensor outputs raw RGB values based on reflected light intensity.

To reduce the effect of ambient lighting, normalization is applied, and color classification is performed using thresholding[8].

The thresholding operation is defined as:

$$I'(x,y) = \begin{cases} 255 & if\ T_{low} \le I(x,y) \le T_{high} \\ 0 & otherwise \end{cases} \quad (11)$$

Where:

- **I (x, y):** Original pixel intensity
- **I′ (x, y):** Thresholded output
- $T_{low}$ **and** $T_{high}$**:** Threshold limits

This method is computationally efficient and suitable for real-time robotic applications. Despite sensitivity to lighting conditions, it provides a practical balance between speed, simplicity, and accuracy for color-based sorting tasks.

## 4. System Overview

The proposed robotic system integrates a Raspberry Pi 4B, a camera module, a servo motor driver, multiple servo motors for the manipulator joints, and a DC motor for the conveyor belt. These components work together to enable real-time color detection, trajectory computation, and object manipulation.

The Raspberry Pi serves as the central processing unit (CPU), receiving visual input from the camera module via the CSI interface. It processes image data to perform color detection using RGB thresholding techniques. Upon identifying a target object, the Raspberry Pi computes the required joint angles using inverse kinematics (IK) and generates appropriate control signals.

The servo motor driver, interfaced through GPIO pins, controls the robotic arm joints, including the base, shoulder, elbow, wrist, and gripper. This allows precise positioning and manipulation of the end-effector. A DC motor drives the conveyor belt, transporting objects into the robot's workspace. To ensure smooth and stable motion, computed torque control (CTC) is applied to compensate for nonlinear dynamics such as inertia and external disturbances.

Power is distributed through the driver circuitry, supplying both the Raspberry Pi and the actuators, resulting in a unified and efficient system architecture. The integration of these subsystems enables reliable and accurate real-time color-based object sorting.

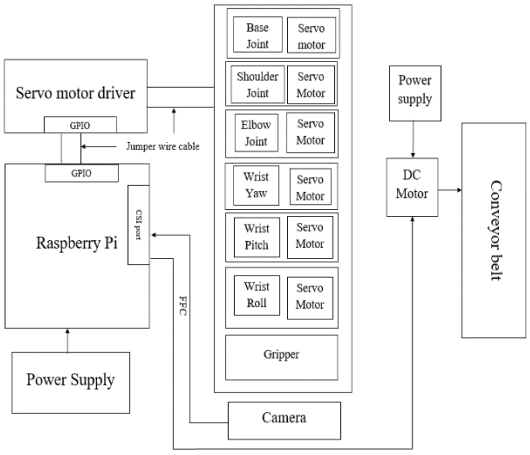



Figure 1. Block Diagram of a robot

During operation, the camera continuously monitors the conveyor belt and captures images of incoming objects. The Raspberry Pi processes these images and determines the object color. If the detected color matches the predefined target, control signals are generated to actuate the robotic arm. The arm then performs pick-and-place operations using coordinated joint motion and gripper control.

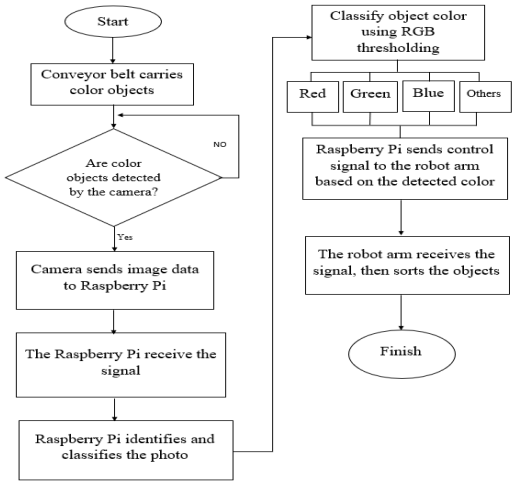



Figure 2. System Flow Chart

Figure 2 illustrates the workflow of the system. The process begins with objects being transported via the conveyor belt. The camera captures image data, which is analyzed by the Raspberry Pi for color classification. If no valid object is detected,

the system continues monitoring in a loop. Once a target object is identified, inverse kinematics is used to determine joint angles, and the robotic arm is actuated accordingly to grasp and sort the object.

This closed-loop operation demonstrates the integration of vision processing, kinematic modeling, and control algorithms. The modular architecture ensures scalability, while the real-time response enables efficient and autonomous color-based sorting.

## 4.1 Hardware Integration

The Raspberry Pi interfaces with the camera module via the CSI (Camera Serial Interface), enabling efficient image capture for real-time processing. Motor control is handled through GPIO pins connected to a dedicated HAT, which simplifies wiring and expands functionality. Power is supplied by a regulated battery pack, ensuring stable voltage for both the Pi and peripherals. The entire robotic arm system is mounted on a lightweight mechanical frame with an integrated gripper and sorting bins, designed for easy maneuverability and efficient field deployment.

Figure 3. Front view of Robot

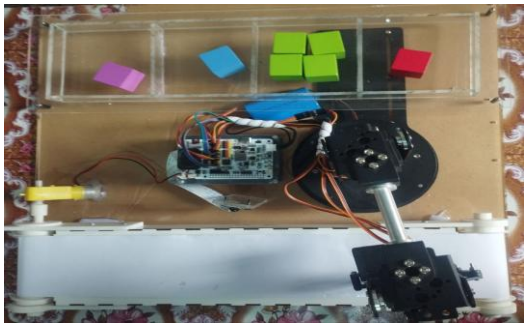


Figure 4. Upper View of Robot

## 5. Simulation and Results

This section presents the simulation studies and experimental results used to validate the performance of the color sorting robotic arm. The goal is to demonstrate the effectiveness of the integrated system including image processing, inverse kinematics, and servo motor control in achieving accurate and responsive sorting behavior under real-world conditions.

## 5.1 Inverse Kinematic Simulation

The robotic arm's motion was first simulated using MATLAB to verify the accuracy of the 6 DOF kinematic model. Given the link lengths $L_1$=0.10 m, $L_2$=0.12 m, $L_3$=0.08 m, $L_4$=0.06 m, $L_5$=0.05 m, $L_6$=0.04 m, the arm's joint angles (θ1…θ6) were computed based on target positions in the workspace. The simulation confirmed that when the inverse kinematics solver assigns equal target coordinates along the x-axis, the arm extends forward in a straight line. When the target position shifts laterally or vertically, the solver adjusts the relative joint angles, resulting in smooth rotational and lifting motions of the arm toward the designated sorting bin. These results validate the effectiveness of the 6 DOF kinematic model in achieving accurate and responsive color-based sorting tasks under real-world conditions. These findings provide a strong theoretical basis for subsequent hardware integration, ensuring that the color-based sorting robot achieves both efficiency and reliability in real-world applications.

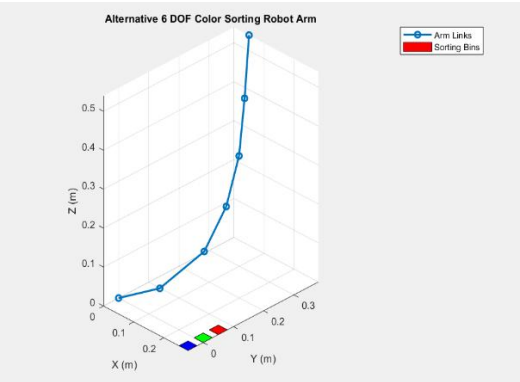



Figure 5. Matlab Simulation of Inverse Kinematic

## 5.2 CTC Controller Tuning

The Computed Torque Control (CTC) strategy was extended with PID terms to improve robustness. Gains were defined as diagonal matrices, but identical values were applied across joints for simplicity in evaluation. The optimal configuration was:

- $K_p$ = diag(100,100,80,50,50,30)
- $K_i$ = diag(10,10,8,5,5,3)
- $K_d$ = diag(20,20,15,10,10,5)

With this gain set, the robot arm tracked desired trajectories smoothly, stabilizing quickly without overshoot. Lower $K_p$ produced sluggish response, while higher $K_d$ introduced instability. The integral term reduced residual steady-state error, confirming PID-CTC as a robust solution for real-time color sorting[4].

## 5.3. Real-Time Behavior and Field Testing

Color picker robot was implemented using the inverse kinematics method to compute joint angles for the robotic arm. RGB thresholding was applied for color detection, and centroid analysis determined positional error relative to the workspace center. Experimental results indicated negligible detection latency, validating the system's real-time capability.

Field testing was conducted with a 1.5-foot conveyor carrying red, green, blue, and other blocks. The robot arm adjusted its joint angles through inverse kinematics and sorted each block into the correct bin. The PID-CTC controller refined servo outputs, ensuring smooth motion and preventing instability.

When no target color was present, the arm paused and resumed operation once a valid color reappeared. This pause-and-resume mechanism demonstrated reliable recovery and validated the robustness of the decision-making process..

Overall, the integration of inverse kinematics with real-time color detection and PID-CTC control produced accurate, stable, and responsive sorting. Field testing confirmed the robot's ability to operate continuously with high accuracy and minimal deviation.

## 5.4. Real-Time Test Results

To validate the robot's color-sorting behavior, live experiments were conducted in a controlled environment. The system detected colored objects, computed positional error, and executed pick-and-place operations using inverse kinematics. Results confirmed accurate recognition and stable sorting performance under pratical conditions.

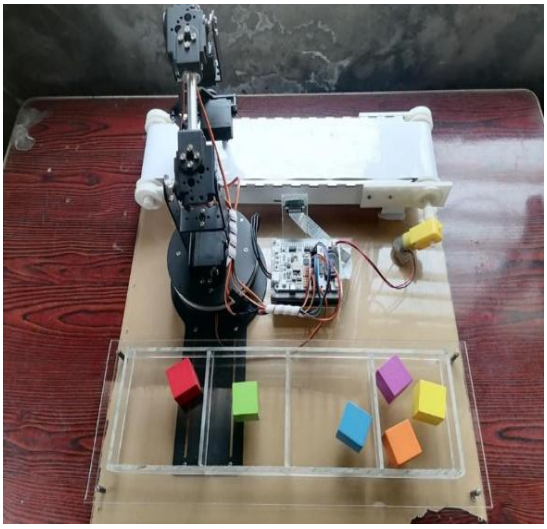


Figure 6. Testing With Color Objects

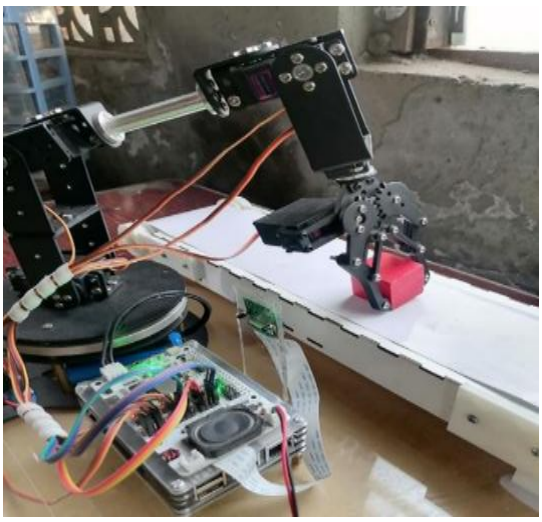


Figure 7. Testing With Color Object

| Run Time | | Run 1 | Run 2 | Run 3 | Run 4 | Run 5 | Average |
|---|---|---|---|---|---|---|---|
| Red | Total Picks | 20 | 20 | 20 | 20 | 20 | 93% |
| | Correct Picks | 18 | 16 | 20 | 19 | 20 | |
| | Accuracy | 90% | 80% | 100% | 95% | 100% | |
| Green | Total Picks | 20 | 20 | 20 | 20 | 20 | 92% |
| | Correct Picks | 17 | 18 | 19 | 20 | 18 | |
| | Accuracy | 85% | 90% | 95% | 100% | 90% | |
| Blue | Total Picks | 20 | 20 | 20 | 20 | 20 | 80% |
| | Correct Picks | 14 | 16 | 18 | 15 | 17 | |
| | Accuracy | 70% | 80% | 90% | 75% | 85% | |
| Others | Total Picks | 20 | 20 | 20 | 20 | 20 | 79% |
| | Correct Picks | 15 | 16 | 17 | 15 | 16 | |
| | Accuracy | 75% | 80% | 85% | 75% | 80% | |

Figure 8. Logged Data From Robot

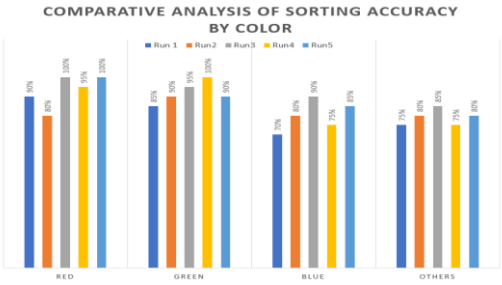



Figure 9. Logged Performance Metric form Robot

Figure 8 and 9 show the experimental results of the color sorting robot using inverse kinematics across five consecutive runs. Each run consisted of 20 pick-and-place attempts per color category. The recorded data include total picks, correct picks, and calculated accuracy. As shown, the robot achieved high accuracy in the primary color categories, with Red averaging **93%** and Green **92%**. Blue objects were sorted with an average accuracy of **80%**, while the "Others" category, which included mixed or secondary colors, reached **79%**. The lower accuracy in the "Others" category reflects the increased complexity of detecting non-primary colors under varying lighting conditions. Overall, the robot demonstrated consistent performance, validating the effectiveness of inverse kinematics in achieving stable and repeatable color-based sorting operations.

## 6. Conclusion

This paper details the developed 6DOF robotic arm, integrated with a conveyor belt and RGB vision system, provides a complete automation framework for color-based object manipulation. By applying the Denavit–Hartenberg (DH) convention, both forward and inverse kinematics were modeled to ensure accurate end-effector positioning. Real-time adaptability is achieved through RGB vision, which detects and classifies objects by color, while computed torque control compensates for nonlinear dynamics to deliver smooth and precise motion. The conveyor belt supplies continuous objects, and the robotic arm executes sorting and placement tasks efficiently. The design offers a scalable, low-cost solution for autonomous color-based manipulation.

## REFERENCES

[1] Kalidass M, Bhuvan Thejaswi T, Nandan D, Israr Raiz and Heyram S, "AI-based Object Color Detection and Sorting Automation", IJARIIE-ISSN(O)-2395-4396 Vol.11 Issue 3 2025

[2] Anand, S., Giridharan, K., Srinivasan, V. and Gopinath, B., "DESIGN AND DEVELOPMENT OF 7 DEGREES OF FREEDOM ROBOTIC ARM FOR INDUSTRIAL APPLICATION" Journal of Emerging Technologies and Innovative Research (JETIR) Volume 6, Issue 5, May 2019

[3] Giorgio Simonini, Marco Baracca, Tommaso V. Cavaliere, Antonio Bichhi, and Paolo Salaris, "A Novel Formulation for Adaptive Computed Torque Control Enabling Low Feedback Gains in Highly Dynamical Tasks", IEEE Access, received 10 March 2025, accepted 6 April 2025

[4] "Theory of Applied Robotics Kinematics, Dynamics, and Control", Third Edition 2022, by Reza N.Jazar

[5] Mordechai Ben-Ari Francesco Mondada (Element of Robotics)

[6] Xingyu Qiu, "Forward Kinematic and Inverse Kinematic Analysis of A 3-DOF RRR Manipulator", International Conference on Mechanics, Electronics Engineering and Automation (ICMEEA 2024), Advances in Engineering Research 240

[7] Nithin Reddy, B., Praveen, D., Deva Raj Yadav, J., Vishnu, K., Praveen, P. and Santosh Naik, P., "Mechanical Design and Analysis of Six-Degree-of Freedom (6-DOF)

SCARA Robot for Industrial Applications", International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653 Volume 12 Issue IV Apr 2024

[8] Rafael C. Gonzalez, and Richard E. Woods, "Digital Image Processing Fourth Edition:"

[9] DATASHEET Raspberry Pi 4 Model B Release 1.1 March 2024 Copyright 2024 Raspberry Pi (Trading) Ltd. All rights reserved.

**Authors Biography**

**Hnin ThuzarWin** received the Bachelor of Computer Technology (B.C.Tech.) degree in 2023 from the University of Computer Studies, Banmaw, Myanmar. She is currently pursuing her Master of Computer Technology (M.C.Tech.) degree at University of Computer Studies, Mandalay.