



Siamese Model-Based Face Verification Using CNN and MobileNetV2

Abd Rahman¹, Agus Mohamad Soleh², Erfiani³

^{1,2,3}Statistics and Data Science, School of Data Science Mathematics and Informatics, IPB University, Bogor, Indonesia

*rahmanabedee@gmail.com

Abstract

Face verification plays an important role in computer vision, especially in mobile and embedded systems with limited computational capacity. This study proposes a face verification system based on the Siamese Neural Network (SNN) architecture by integrating six embedding models. These models consist of a standard CNN, an L2-normalized CNN, a baseline MobileNetV2, a structurally adjusted MobileNetV2, a pre-trained MobileNetV2, and a fine-tuned MobileNetV2. The dataset includes facial images captured from three webcams and additional samples obtained from the Labeled Faces in the Wild and ImageNet datasets. The experimental procedure includes image preprocessing, construction of balanced positive and negative image pairs, model training, and evaluation using accuracy, precision, recall, F1-score, and AUC. The results show that the pre-trained MobileNetV2 and the standard CNN achieve the highest verification accuracy, reaching 100 percent and 99.998 percent, respectively. Among all models, the structurally adjusted MobileNetV2 presents the best trade-off by combining high accuracy, computational efficiency, and training stability while successfully avoiding overfitting. The real-time implementation involves only the structurally adjusted MobileNetV2 model due to its lightweight structure and consistent performance. This model produces low embedding distances, low latency, and high throughput during CPU-based inference. The performance outperforms GPU execution in one-by-one image processing. The proposed system offers a practical and efficient face verification solution for deployment in identity authentication applications on resource-constrained platforms. These findings support the development of scalable and adaptive biometric security systems that rely on deep learning.

Keywords: convolutional neural network; face verification; MobileNetV2; real-time; siamese neural network

How to Cite: Abd Rahman, Agus Mohamad Soleh, and Erfiani, "Siamese Model-Based Face Verification Using CNN and MobileNetV2", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 10, no. 2, pp. 234 - 246, Mar. 2026.

Permalink/DOI: <https://doi.org/10.29207/resti.v10i2.6996>

Received: July 23, 2025

Accepted: February 6, 2026

Available Online: March 12, 2026

This is an open-access article under the CC BY 4.0 License

Published by Ikatan Ahli Informatika Indonesia

1. Introduction

Face verification is a fundamental component of biometric authentication that supports advancements in digital identity systems, contactless security, and smart devices. Face verification models must be capable of determining whether two facial images belong to the same individual, even under challenging conditions such as variations in lighting, facial expressions, and occlusion [1].

Convolutional Neural Networks (CNNs) have demonstrated strong performance in extracting spatial features from facial images and are widely adopted as the backbone in modern face verification systems [2]. Siamese Neural Networks (SNNs), commonly used among CNN-based methods, compare paired inputs in a shared embedding space. These networks utilize contrastive or triplet loss functions to learn similarity-based representations, making them effective for verification tasks [3].

Conventional CNN-based SNNs achieve high accuracy but often require substantial computational resources, which limits their deployment in real-time or embedded environments [4]. Recent studies have explored lightweight architectures such as MobileNetV2 to overcome this challenge. This architecture employs depthwise separable convolutions and inverted residual blocks to reduce computational complexity, making it suitable for resource-constrained devices [5], [6].

MobileNetV2 has achieved promising results in facial analysis tasks, particularly when deployed on edge devices such as Raspberry Pi. It provides high accuracy with minimal computational overhead [7]. Limited research has directly compared the performance of MobileNetV2 and traditional CNNs when integrated as embedding backbones within Siamese architectures for face verification, despite its promising standalone results.

Researchers have developed scalable face verification systems using model compression, fine-tuning, and transfer learning [8], [9]. Optimization applied to CNN and MobileNetV2 has produced lightweight models with strong performance [10], [11]. Prior studies show that SNNs with lightweight backbones can outperform conventional models in mobile environments while maintaining high verification accuracy [12]. However, no comprehensive evaluation has compared CNN and MobileNetV2 embeddings under the same training pipeline in a unified Siamese setup.

This study aims to evaluate and compare CNN and MobileNetV2 as embedding backbones within a Siamese network framework. All models are trained under identical configurations involving normalization, fine-tuning, and transfer learning. The evaluation focuses on verification accuracy and computational efficiency in real-time scenarios. By providing a benchmark comparison based on standardized metrics and real-world testing, this research supports the development of optimal face verification systems for mobile and embedded platforms [13], [14]. In response to the growing demand for fast, lightweight, and reliable AI-based authentication, this study offers timely insights for designing efficient verification solutions [15]. The findings are expected to guide future efforts in scalable and adaptable biometric system design.

The novelty of this study lies in the integration of MobileNetV2 as a lightweight embedding backbone within a Siamese Neural Network framework, which has traditionally relied on conventional CNN architectures. This integration is intended to enhance computational efficiency while preserving verification accuracy, thereby enabling a fair comparison under uniform training conditions.

2. Methods

This study employed a Siamese Neural Network (SNN) architecture for face verification using six distinct embedding models. The embedding models consisted of a standard Convolutional Neural Network (CNN), an L2-normalized CNN, a baseline MobileNetV2, a structurally adjusted MobileNetV2, a pre-trained MobileNetV2, and a fine-tuned MobileNetV2 [16], [7]. All models were trained to convert facial images into low-dimensional feature vectors, which were then compared using the Euclidean distance metric [17].

The SNN architecture utilized the contrastive loss function with a fixed margin of 1.0 to optimize the distance between positive and negative image pairs during training [18]. Model training was performed using the Adam optimizer with an exponentially decaying learning rate, further supported by the ReduceLROnPlateau callback to ensure convergence stability and to mitigate overfitting [19], [20]. All six models were trained and evaluated under identical experimental conditions to ensure consistency and fairness in comparative analysis.

The performance evaluation involved several metrics, including accuracy, precision, recall, area under the curve (AUC), confusion matrix, and inference latency under real-time conditions. This experimental setup aimed to analyze the trade-offs between predictive accuracy and computational efficiency within a unified Siamese network configuration [21], [22].

This framework ensures consistent training and evaluation across models, providing a reliable basis for performance comparison. It also enables objective analysis of how different embedding strategies affect accuracy and efficiency. The procedure is summarized in Figure 1.

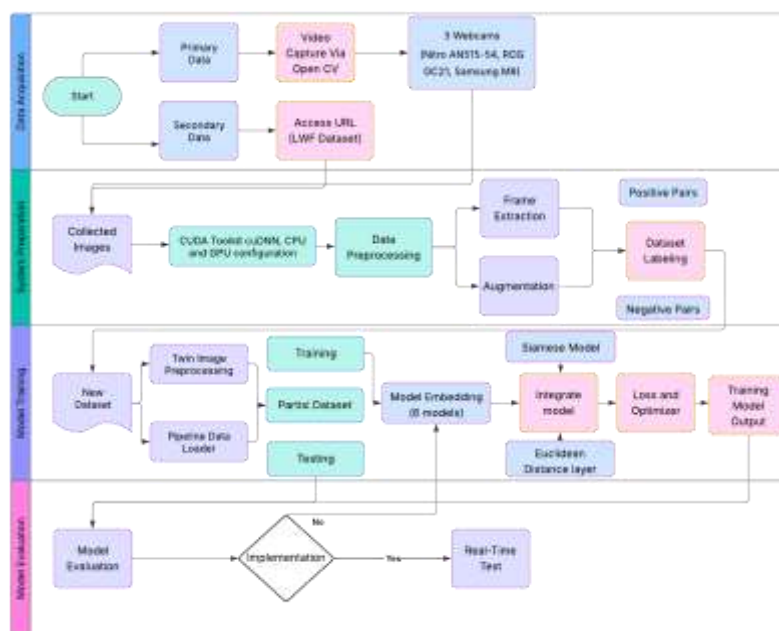


Figure 1. Flowchart of the face verification process using a Siamese Neural Network

2.1 Dataset

This study utilized two types of datasets: primary and secondary. The primary dataset consisted of 2,520 facial images collected from eight individuals (four males and four females) using a custom Python-based image acquisition application. Three webcams were used to ensure variation in image quality and viewing angles: an Acer Nitro AN515-54 (720p, 30 fps), an Asus ROG GC21 (1080p, 60 fps, with WDR and autofocus), and a Samsung Smart Monitor M8 (1080p, with auto-framing functionality). Each participant contributed 315 images, with 105 images per device, capturing various facial orientations. The image acquisition process is illustrated in Figure 2. The secondary dataset included facial images obtained from the Labeled Faces in the Wild (LFW) dataset. These images were used to generate negative pairs for training and evaluation purposes [23]. Additionally, pre-trained weights from the ImageNet dataset were used to initialize selected MobileNetV2 models. This approach supported transfer learning, enabling the models to adapt more effectively to the characteristics of the primary dataset [24].

This acquisition setup ensured adequate variation in illumination, orientation, and device characteristics, thereby providing a comprehensive dataset for robust face verification.

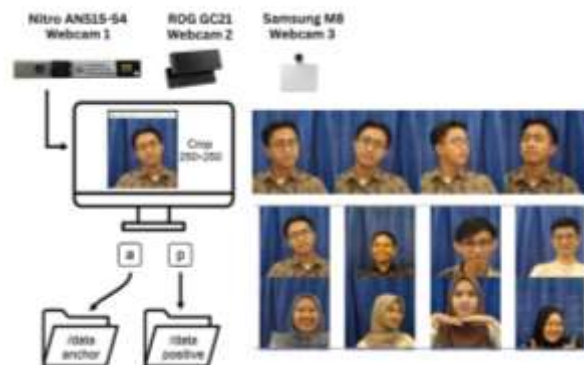


Figure 2. Image sequence capture for primary data

2.2 Preprocessing

The preprocessing stage aimed to standardize, augment, and prepare facial images for training within the Siamese Neural Network (SNN) framework. Each image from the anchor and positive sets was first resized to 250×250 pixels to ensure frame consistency, and then scaled to 160×160 pixels to match the input dimensions required by the embedding models. All pixel values were normalized to the range of $[0, 1]$ by dividing each value by 255.

Augmentation techniques were applied to both anchor and positive images to increase data diversity and improve generalization. Anchor images were augmented using six methods, namely horizontal flipping, rotation, brightness adjustment, Gaussian blur, noise addition, and sharpening. This process produced

630 images per individual ($105 \text{ originals} \times 6 \text{ augmentations}$). Positive images underwent additional transformations, including zoom, motion blur, color jitter, perspective transformation, cropping, occlusion, sharpening, and enhanced noise injection. The selection of these augmentation techniques was based on previous research demonstrating that geometric and photometric transformations such as rotation, brightness adjustment, and noise addition effectively enhance the robustness and generalization of face recognition models [25]. This process yielded approximately 2,730 positive images per individual, generated from two webcam sources. Figure 3 presents examples of the augmented anchor and positive images, demonstrating the visual variations introduced by these techniques. Augmentation strategies of this kind have been shown to improve model robustness in face verification tasks [26].

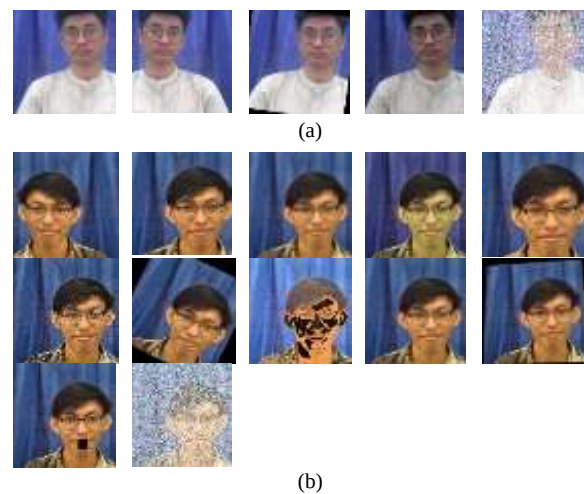


Figure 3. Anchor (a) and Positive (b) image augmentations

Training pairs were constructed using 40,113 images, which were categorized into three labeled sets: 13,233 negative samples from the LFW dataset, 5,040 anchor images, and 21,840 positive images from the primary dataset. Each anchor image was linked with four to five positive samples of the same identity, resulting in 21,840 anchor–positive pairs. Likewise, anchor–negative pairs were generated by associating each anchor image with four to five unrelated images from the negative set, resulting in 21,840 anchor–negative pairs. All image pairs were labeled as 1 (positive) or 0 (negative) to support supervised learning with the contrastive loss function [3].

The steps of resizing, normalization, augmentation, and balanced pair construction followed standard practices in facial verification preprocessing workflows using Siamese networks [2]. These procedures ensured consistent input representation and improved the model's generalization across diverse facial variations.

2.3. Model Architecture

The Siamese Neural Network (SNN) architecture implemented in this study consists of two identical

branches with shared weights. Each branch receives an input image, either as an anchor or as a comparison, and processes it through the same embedding model. The embedding process generates two fixed-length feature vectors. The similarity between these vectors is measured using the Euclidean distance, which serves as the primary similarity metric in this architecture.

Six embedding models were integrated into the SNN framework for comparative evaluation. These models include a standard Convolutional Neural Network (CNN), an L2-normalized CNN, a baseline MobileNetV2, a structurally adjusted lightweight MobileNetV2, a MobileNetV2 initialized with ImageNet weights, and a fine-tuned version of MobileNetV2. All models were embedded into the SNN framework without modifying the core verification pipeline, thereby maintaining a consistent architecture for fair performance comparison.

Table 1 presents the architectural details of the embedding models. The table includes trainable parameters and the dimensionality of the output vectors.

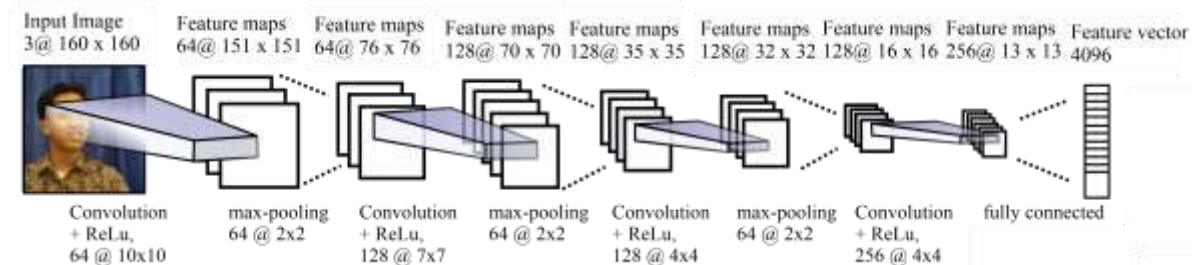
Table 1. Architectural Complexity of Embedding Models

Embedding	Trainable Parameters	Dimension
CNN	178,421,056	4096
CNN L2 Normalization	23,360,448	512
MobileNetV2	12,193,536	512
MobileNetV2 Adjusted	3,833,472	128
MobileNetV2 Pre-trained	9,969,664	512
MobileNetV2 Fine-Tuned	11,831,104	512

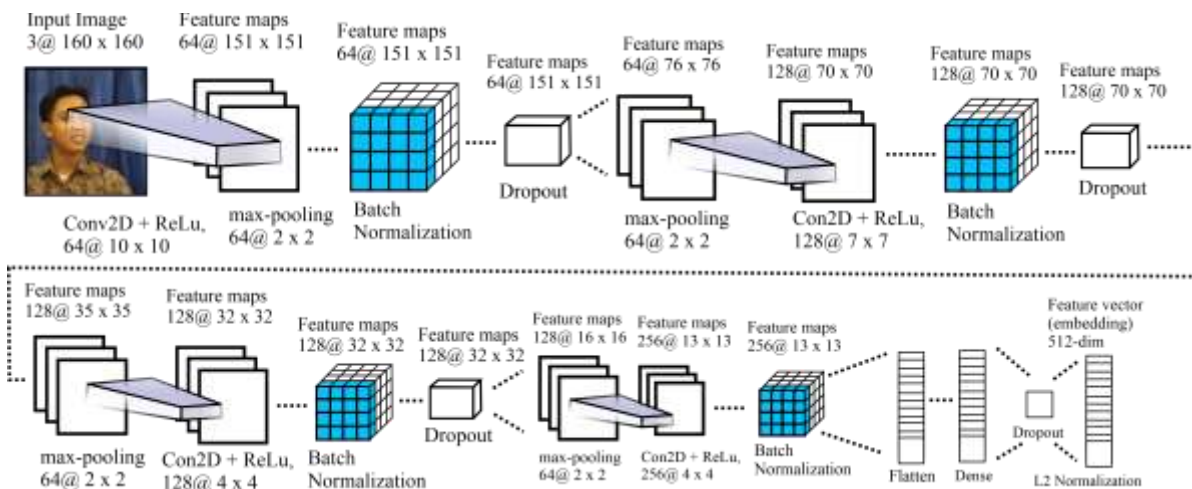
The visual structure and internal composition of each embedding model, including convolutional layers, normalization mechanisms, and fully connected components, are illustrated in Figure 4. The six architectural variants were designed based on the depth and configuration of core layers, the number of trainable parameters, and the trade-off between computational efficiency and discriminative feature extraction. This architectural diversity provides a foundation for fair performance comparison within the unified Siamese framework in face verification tasks. Each model highlights a balance of complexity and performance, supporting consistent and meaningful comparison across architectures.

Figure 4 illustrates six embedding models integrated into the Siamese framework [1], [3], [5], [15], [16]. The CNN model employs convolutional blocks and a fully connected layer as the baseline, while the L2-normalized CNN incorporates batch normalization and dropout to improve feature stability. MobileNetV2 variants include baseline, pre-trained, fine-tuned, and adjusted configurations, each differing in depth and parameter tuning. The adjusted model, proposed in this study, reduces model depth and integrates internal data augmentation to enhance robustness and efficiency [25].

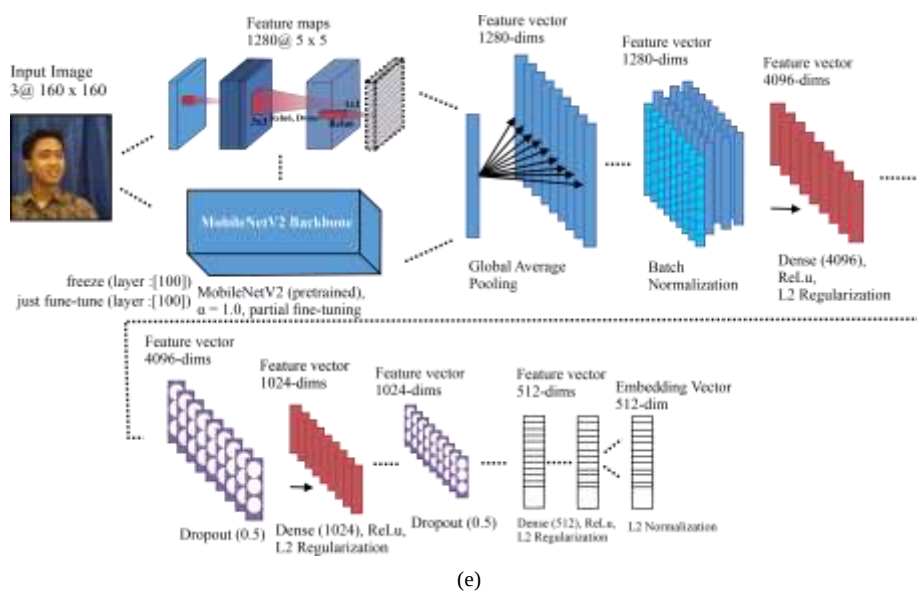
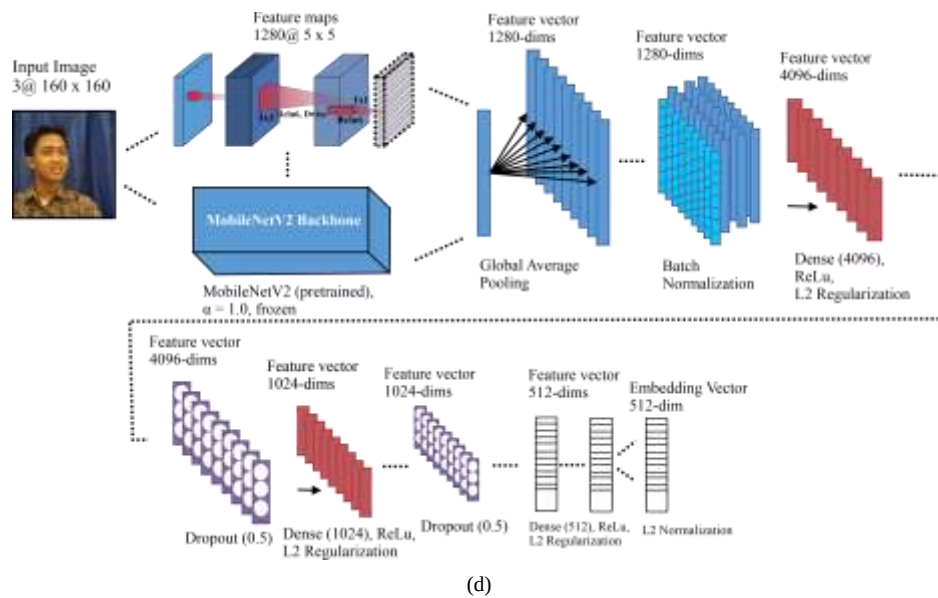
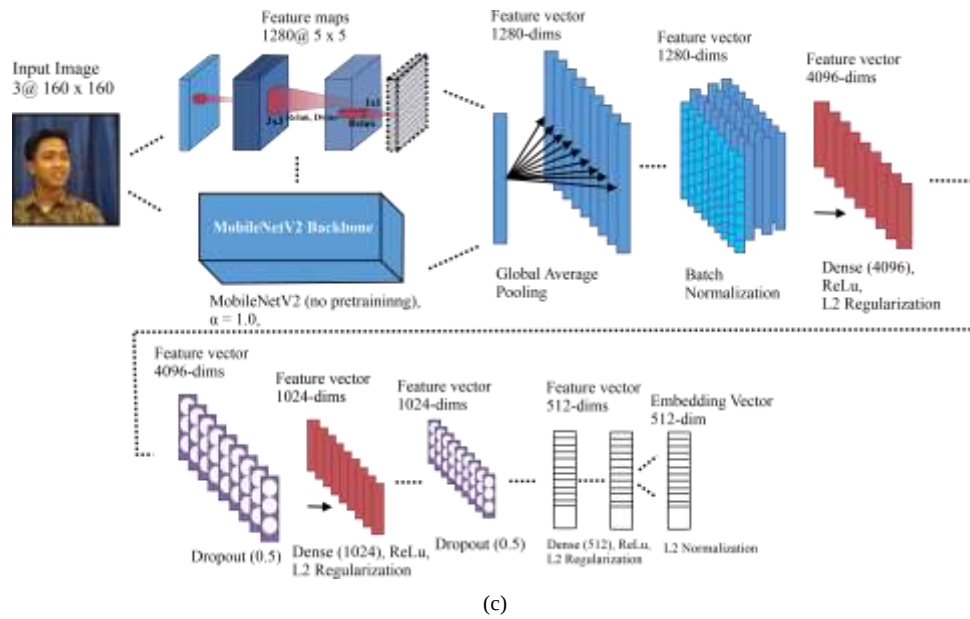
All models function as modular feature extractors with varying complexities, enabling a fair and consistent evaluation within a unified Siamese architecture.



(a)



(b)



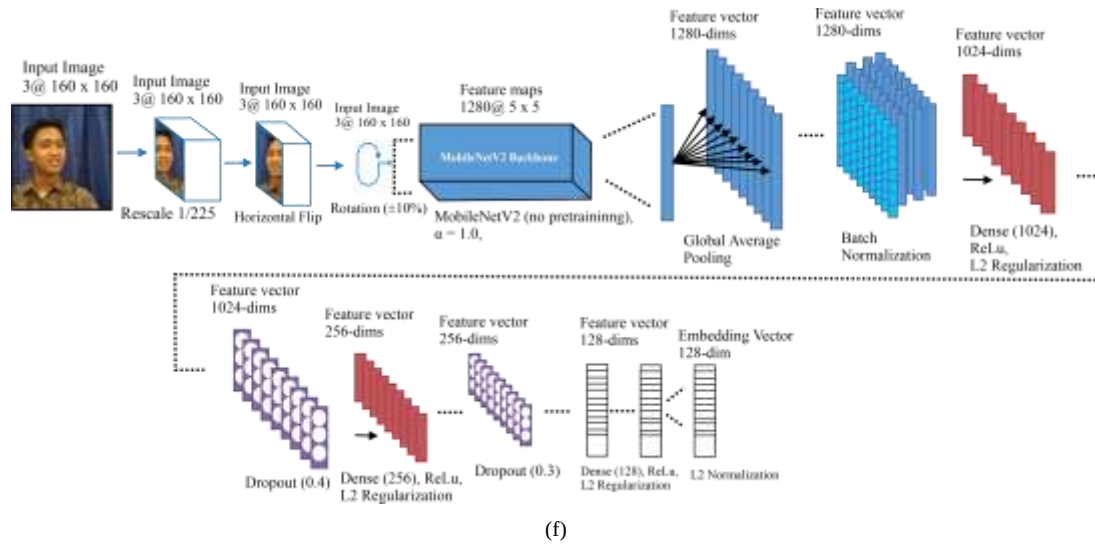


Figure 4. Architectural design of each embedding model integrated into the Siamese framework (a) CNN, (b) CNN L2 Normalization, (c) MobileNetV2, (d) MobileNetV2 Pre-trained, (e) MobileNetV2 Fine-Tuned, (f) MobileNetV2 Adjusted

The final layer computes the Euclidean distance between the two feature vectors using Equation 1.

$$D(x_1, x_2) = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2} \quad (1)$$

The Siamese Neural Network consists of two identical subnetworks (anchor and comparison branches) with shared weights to ensure consistent feature extraction [16], [18]. Each subnetwork processes an image input using the same embedding model, producing fixed-length feature vectors in a shared latent space. The Euclidean distance measures their similarity, where lower values indicate higher resemblance [17]. The network is trained using contrastive loss to minimize distances between positive pairs and maximize those between negative pairs [18], [21]. This twin-branch configuration enables relational similarity learning and enhances generalization across unseen facial variations [16], [18].

The output is a scalar value representing the similarity between two input images. A lower value indicates higher similarity, while a higher value reflects greater dissimilarity [27].

The use of modular embedding models in this architecture enables flexible experimentation while preserving the verification process. The evaluation results and analysis for each configuration are discussed in a subsequent section.

The training process was optimized using early stopping by monitoring the validation loss. Training stopped when the loss did not improve beyond a defined threshold δ for p consecutive epochs, as Equation 2.

$$|L_{current} - L_{best}| \leq \delta \text{ for } p \text{ epochs} \quad (2)$$

Here, $L_{current}$ is the validation loss at the current epoch, and L_{best} is the lowest recorded value. This method

helps prevent overfitting and ensures efficient convergence [28], [29].

3. Results and Discussions

This section presents the experimental results and analysis of the face verification system built on the Siamese Neural Network (SNN) framework using six distinct embedding models. The evaluation includes training outcomes, overfitting behavior, validation accuracy, architectural complexity, and performance under real-time conditions. Each model is analyzed to measure its effectiveness and efficiency within identical testing environments. The goal of this analysis is to determine the most suitable embedding configuration for deployment on platforms with limited computational capacity.

3.1 Overfitting Analysis

This section evaluates how each embedding model generalizes from training to testing data. Overfitting is indicated by noticeable gaps between training and validation accuracy or loss, as well as irregular curve patterns. Accuracy and loss are plotted per epoch to reveal overfitting or underfitting. Figures 5 to 10 show training curves from five trials to assess stability and generalization.

Figure 5 shows that the CNN model demonstrates excellent training performance, with high accuracy and consistently low loss values on both training and validation data. Minor fluctuations in the validation loss, particularly in one of the runs, do not indicate significant signs of overfitting. Overall, the model exhibits stable generalization capability.

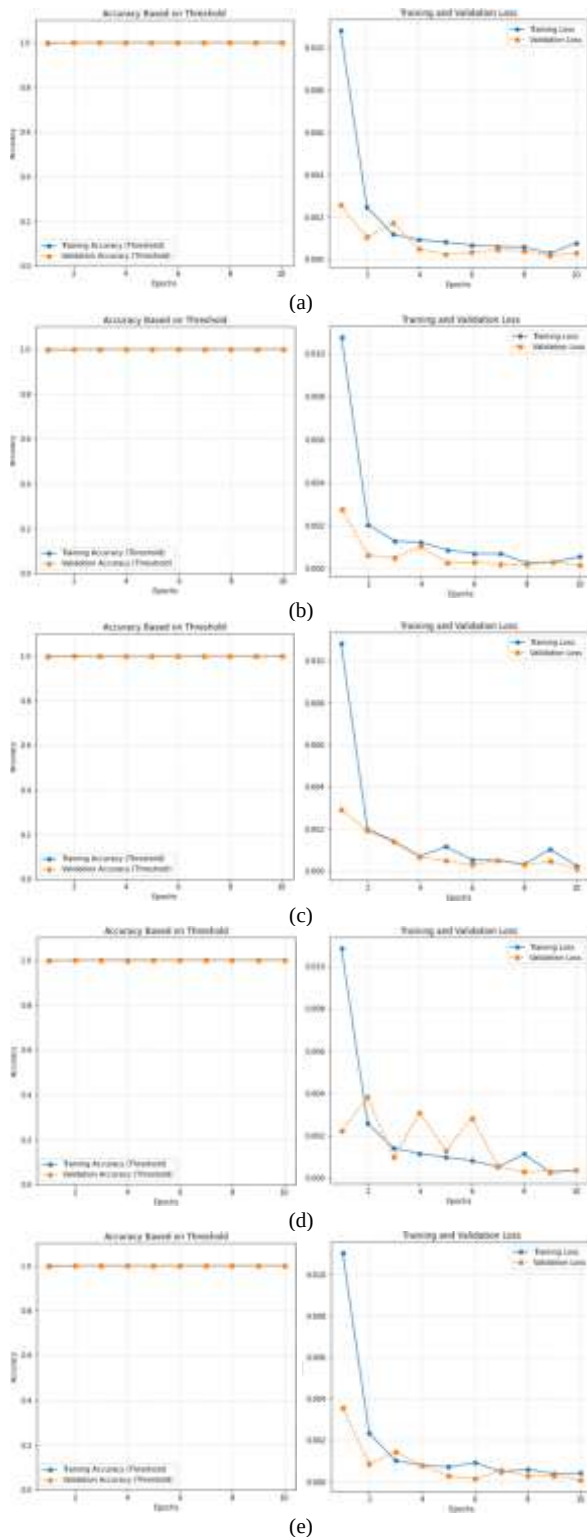


Figure 5. loss and accuracy curve CNN

The L2 Normalization CNN model as shown in Figure 6 showed fluctuations in accuracy and loss, with frequent changes in validation loss across epochs. This indicates a potential risk of overfitting, although validation accuracy stayed within 85 to 91 percent. The model learned reasonably well but lacked consistent validation performance, suggesting instability in feature generalization.

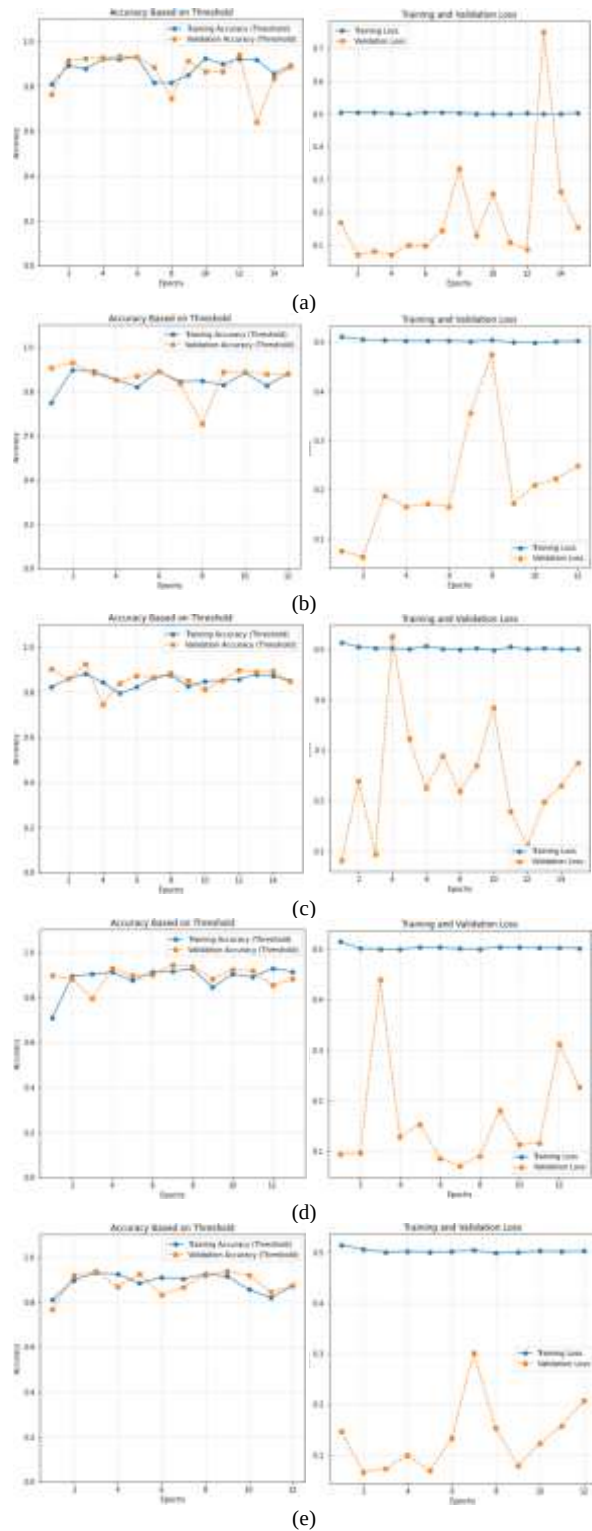


Figure 6. loss and accuracy curve CNN L2 Normalization

The MobileNetV2 model as shown in Figure 7 exhibited an unstable training pattern with significant fluctuations in validation loss, despite achieving high accuracy in several runs. The notable gap between training and validation loss suggests a potential risk of overfitting. This instability also indicates limited generalization capability on validation data.

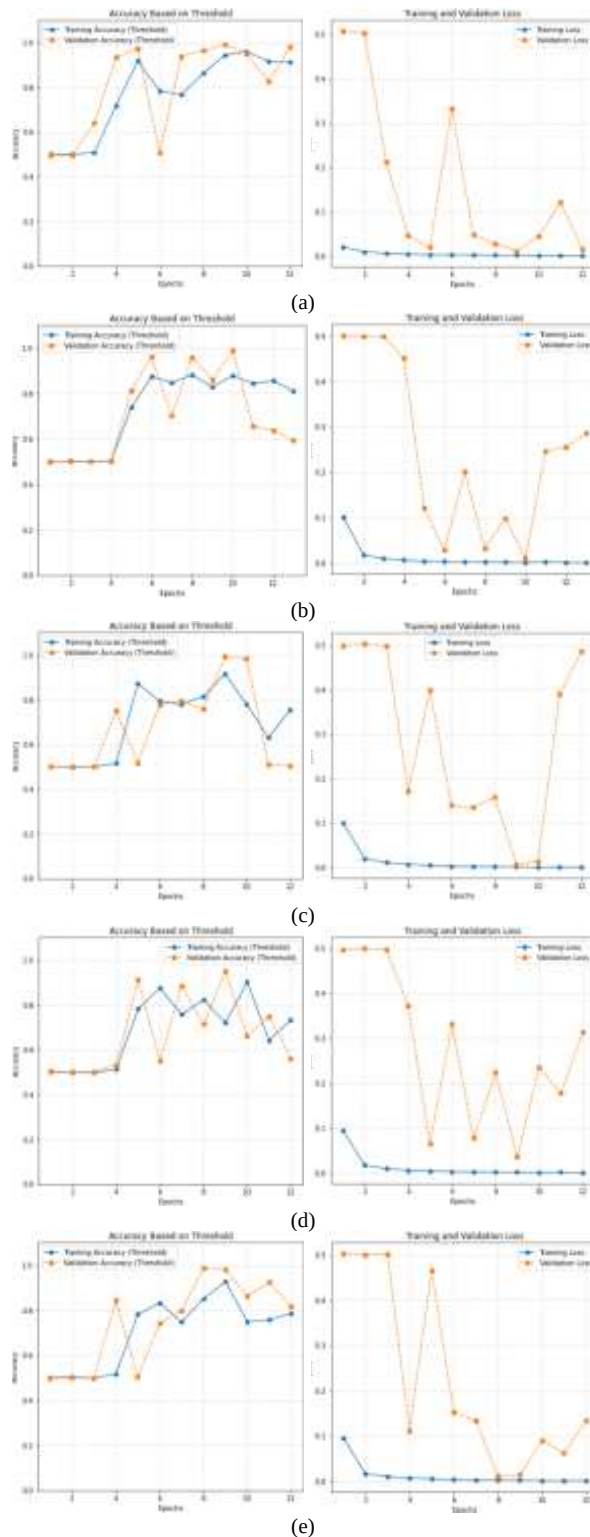


Figure 7. loss and accuracy curve MobileNetV2

The MobileNetV2 Adjusted model as shown in Figure 8 demonstrated stable training with consistent accuracy and loss across training and validation data. Smooth curves with minimal fluctuations reflect strong generalization and low overfitting risk. This stability indicates that the architectural adjustments effectively balanced performance and efficiency.

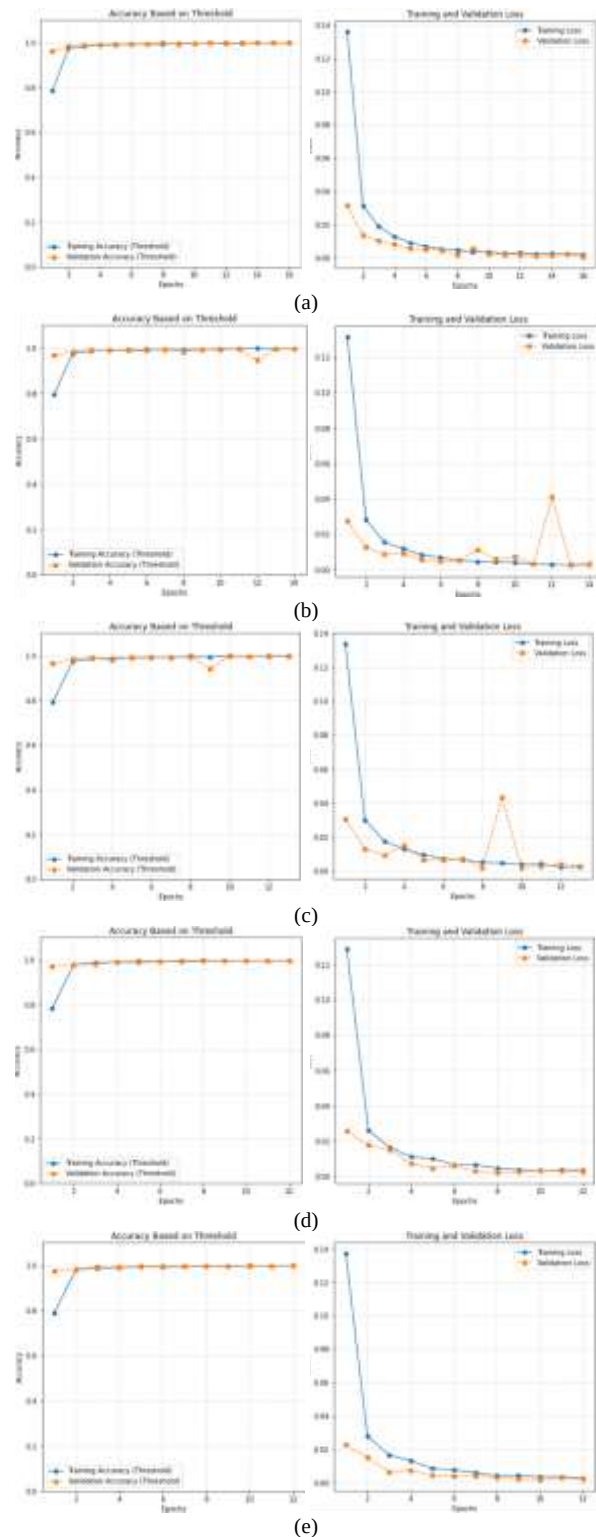


Figure 8. loss and accuracy curve MobileNetV2 Adjusted

The pre-trained MobileNetV2 model as shown in Figure 9 demonstrated stable performance, with consistent accuracy and loss curves between training and validation. The absence of major fluctuations indicates effective training without overfitting. ImageNet-based initialization accelerated convergence and enhanced generalization, making the model reliable

and efficient for face verification, especially in resource-constrained environments.

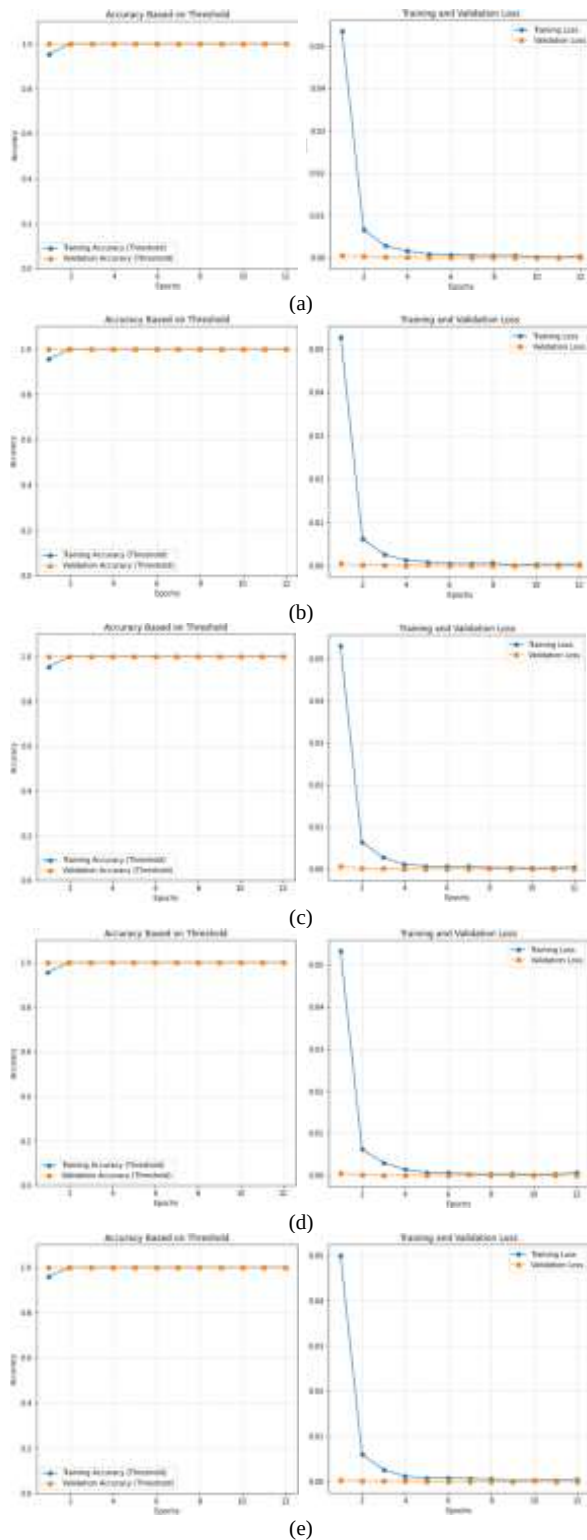


Figure 9. loss and accuracy curve MobileNetV2 Pre-trained

The fine-tuned MobileNetV2 model as shown in Figure 10 achieved high training and validation accuracy, but showed signs of overfitting in several runs, as indicated by unstable validation loss curves. While most epochs exhibited good generalization, occasional performance

drops suggest that further tuning is needed to ensure consistent results across all training iterations.

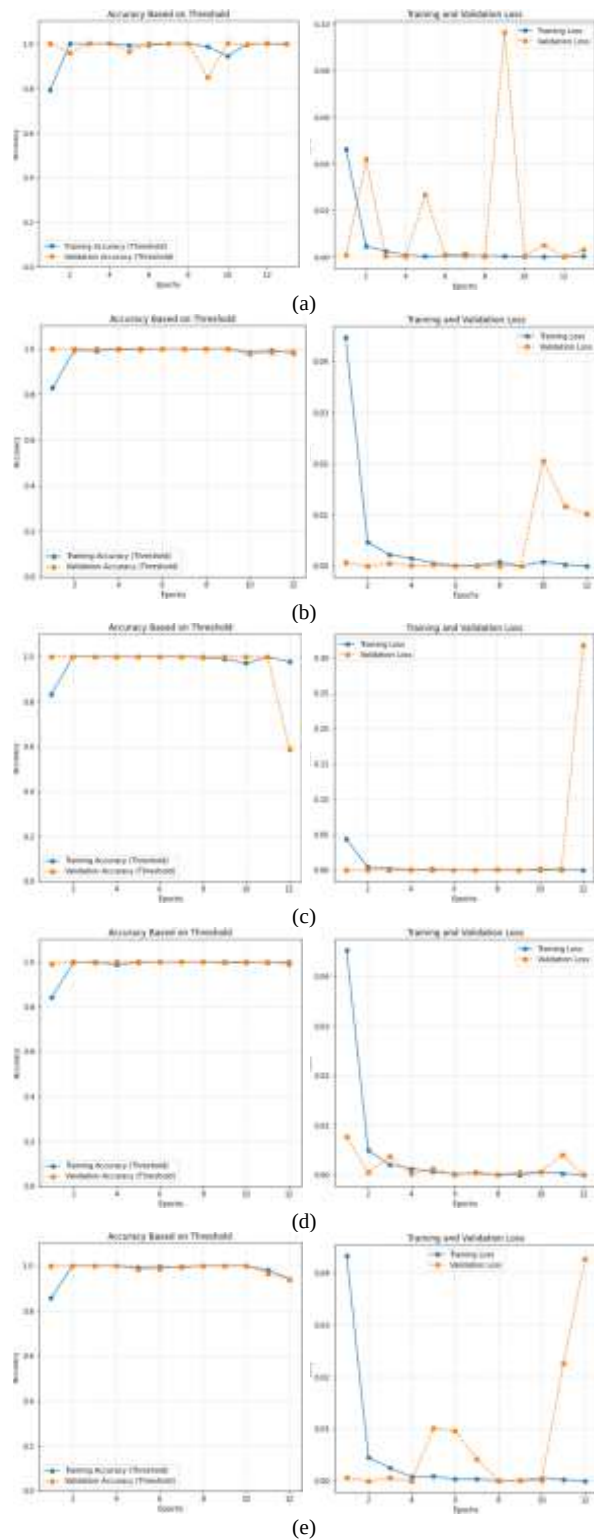


Figure 10. loss and accuracy curve MobileNetV2 Fine-Tuning

3.2 Performance Evaluation of Models

This subsection compares six embedding models using accuracy, AUC, precision, recall, and F1-score on the

validation dataset. Tables 2 and 3 summarize the results for all evaluation metrics.

Table 2. Accuracy and AUC of Embedding Models

Model	Accuracy	AUC
CNN	0.99998	1.000
CNN L2 Normalization	0.88020	0.94380
MobileNetV2	0.69144	0.85500
MobileNetV2 Adjusted	0.99696	0.99986
MobileNetV2 Pre-trained	1.000	1.000
MobileNetV2 Fine-Tuned	0.90010	0.99970

Table 3. F1 Score, Precision, and Recall of Embedding Models

Embedding	F1 Score	Precision	Recall
CNN	0.99998	1.000	0.99996
CNN L2 Normalization	0.87470	0.88020	0.87040
MobileNetV2	0.74370	0.69122	0.85300
MobileNetV2 Adjusted	0.99694	0.99456	0.99908
MobileNetV2 Pre-trained	1.000	1.000	1.000
MobileNetV2 Fine-Tuned	0.92550	0.87880	0.99980

The pre-trained MobileNetV2 achieves the highest validation performance across all metrics on the evaluated dataset. The CNN and the MobileNetV2 Adjusted model achieve near-perfect scores and maintain strong verification capability. The baseline MobileNetV2 and the L2-normalized CNN show lower performance, with the largest reductions appearing in F1-score and recall. The fine-tuned MobileNetV2 achieves high recall but produces lower precision than the best-performing models. The results indicate that transfer learning and architectural adjustment improve validation performance and stability compared with the baseline configuration.

3.3 Interpretation of Performance and Efficiency

This subsection relates validation performance to architectural complexity using the parameter counts in Table 1. The pre-trained MobileNetV2 achieves the highest validation performance with moderate complexity (approximately 10 million parameters). The CNN achieves near-perfect performance but requires substantially higher capacity (approximately 178 million parameters), which limits practicality for real-time deployment.

The MobileNetV2 Adjusted model provides the most balanced trade-off by maintaining high validation performance with low complexity (approximately 3.8 million parameters), which supports embedded deployment. The baseline MobileNetV2, the L2-normalized CNN, and the fine-tuned MobileNetV2 show less consistent behavior under the evaluated setting. The results show that embedding selection should prioritize both discriminative performance and computational efficiency for real-time verification on constrained hardware.

3.4 Implementation in Real-Time

The MobileNetV2 Adjusted model was selected for real-time deployment because it provides high validation performance with low complexity (3.8 million trainable parameters). The implementation uses

a Kivy-based application that includes user registration and identity verification. The system stores 50 facial images per user during registration and performs verification by computing the average Euclidean distance (AD) between a query image and stored samples. The system accepts an identity when AD is less than or equal to 0.3 as shown in Figure 11.

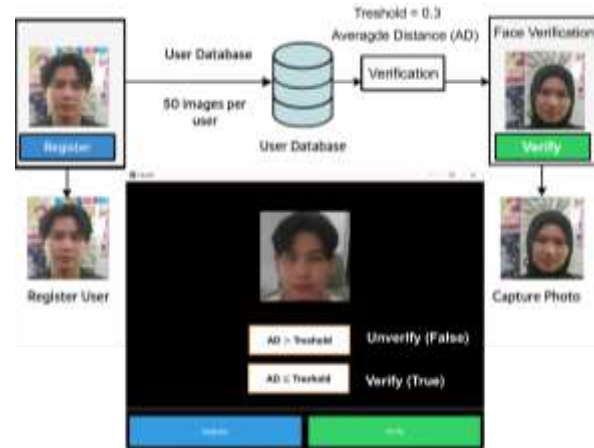


Figure 11. Real-Time Face Verification Interface Using Kivy

Table 4 reports real-time verification results using AD, inference latency, and throughput.

Table 4. Real-Time Verification Results

Citra Webcam	Registered User	AD	Latency (ms)	Throughput (img/sec)	Verified
		0.0204	234.09	4.19	True
		0.0536	241.79	4.05	True
		0.4162	310.43	3.15	False
		0.4262	299.49	3.27	False

The results show that registered users produce AD values below the threshold, whereas unregistered users produce AD values above the threshold, which supports effective separation between genuine and impostor attempts. Figure 12 shows that CPU-based inference achieves lower latency and higher throughput than GPU execution in single-image processing scenarios. GPU execution introduces overhead from kernel launching and data transfer, which reduces efficiency for low-complexity sequential workloads [30], [31]. CPU-based one-by-one inference supports practical real-time

deployment on resource-constrained systems without reliance on batch-level parallelism [32].

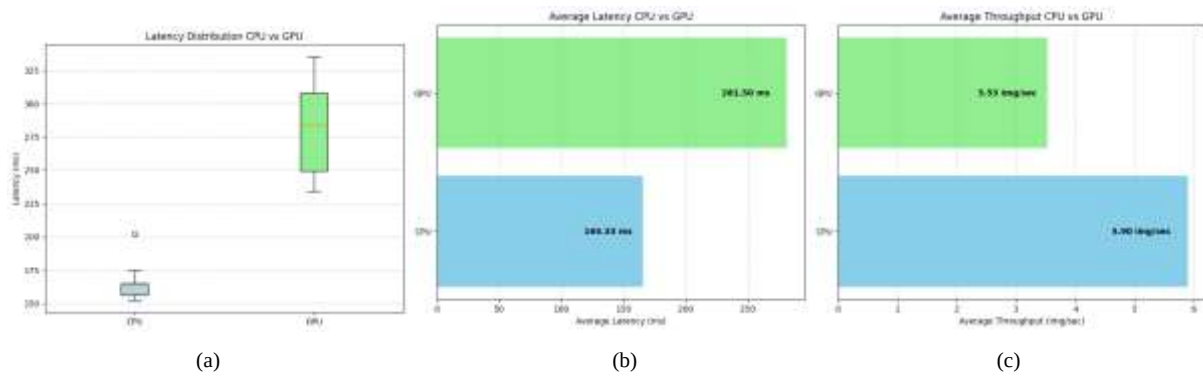


Figure 12. CPU vs GPU Latency and Throughput

3.5 Statistical and Efficiency Analysis

Further analysis statistically validated performance differences among evaluated models. The evaluation focused on three architectures: CNN, MobileNetV2 Adjusted, and Pre-trained MobileNetV2. A MANOVA test was applied, followed by one-way ANOVA and Tukey HSD post-hoc tests, to compare accuracy, precision, recall, F1-score, and loss. Table 5 reports ANOVA results, showing that all metrics differed significantly among the three models ($p < 0.05$). These findings confirm statistically significant variations in classification performance. The Tukey HSD test shows that MobileNetV2 Adjusted differs significantly from both CNN and the Pre-trained model, while the latter two do not differ statistically. This result reflects the distinct behavior of the adjusted model due to its simplified architecture as shown in Table 6.

Table 5. One-way ANOVA results for performance metrics

Metric	Sum Sq	Df	F Value	p-Value	Remark
Accuracy	0.000869	2	288.0863	7.224e-11	Sign.
Precision	0.00152	2	151.5481	3.051e-09	Sign.
Recall	0.000837	2	104.6095	2.548e-08	Sign.
F1-score	0.000828	2	224.4532	3.115e-10	Sign.
Loss	0.000527	2	369.4075	1.667e-11	Sign.

Table 6. Post-hoc Tukey HSD comparison for F1-score metric

Metric	Model 1	Model 2	Mean Diff	p-adj	Sign
F1-score	CNN	MNetV2 Adjusted	-0.0161	0.000	Yes
F1-score	CNN	Pre-trained MNetV2	0.0004	0.8925	No
F1-score	MNetV2 Adjusted	Pre-trained MNetV2	0.0165	0.000	Yes

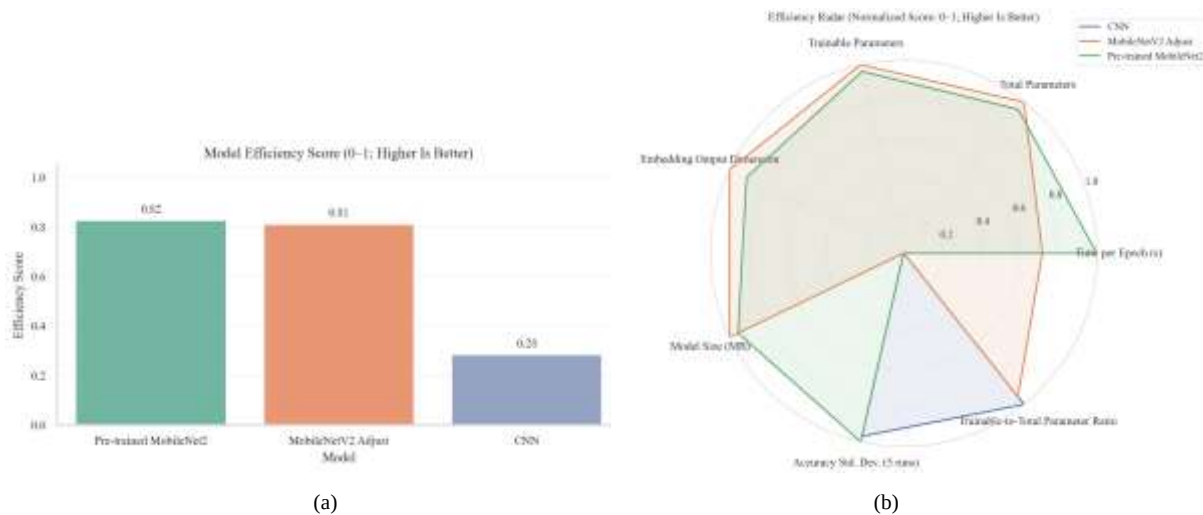


Figure 13. Model efficiency comparison: (a) Efficiency score, (b) Efficiency radar chart

Efficiency analysis compared six normalized factors reflecting model complexity and training behavior. Figure 13 reports efficiency scores of 0.82 for the pre-trained MobileNetV2, 0.81 for the MobileNetV2 Adjusted model, and 0.28 for CNN. The pre-trained model achieves the highest score, while the adjusted model maintains a comparable score with lower

complexity. The results support the MobileNetV2 Adjusted model for resource-constrained deployment.

4. Conclusions

This study developed a Siamese Neural Network-based face verification system that integrates six embedding backbones, including CNN and MobileNetV2 variants.

The pre-trained MobileNetV2 achieved the highest validation performance with moderate complexity, while the CNN reached near-perfect performance with higher computational cost. The structurally adjusted MobileNetV2 provided the most balanced trade-off between accuracy, efficiency, and training stability. This study implemented real-time deployment using the MobileNetV2 Adjusted model in a Kivy-based application and achieved low-latency CPU inference for single-image verification with stable throughput. The results demonstrate that lightweight embedding architectures support biometric authentication on resource-constrained platforms. Future work will investigate quantization, compression, and hardware acceleration to improve scalability and efficiency.

References

- [1] D. Zeng, R. Veldhuis, L. Spreeuwers, and R. Arendsen, "Occlusion-invariant face recognition using simultaneous segmentation," *IET Biometrics*, vol. 10, pp. 679–691, 2021, doi: 10.1049/BME2.12036.
- [2] Q. Zhou, T. Lu, Y. Zhang, Z. Xiong, H. Chen, and Y. Wu, "Context-aware based discriminative Siamese Neural Network for face verification," in *Proceedings of the International Conference on Pattern Recognition*, pp. 113–124, 2020, doi: 10.1007/978-3-030-56178-9_9.
- [3] T. Lu, Q. Zhou, W. Fang, and Y. Zhang, "Discriminative metric learning for face verification using enhanced Siamese neural network," *Multimedia Tools and Applications*, vol. 80, pp. 8563–8580, 2020, doi: 10.1007/s11042-020-09784-8.
- [4] R. Zong, "On privileged information driven robust face verification: A Siamese convolutional neural network approach," in *Proceedings of the 2020 IEEE International Conference on Big Data (Big Data)*, pp. 5874–5878, 2020, doi: 10.1109/BigData50022.2020.9378074.
- [5] M. Liu, "Masked Face Recognition Using MobileNetV2," in *Proceedings of the 2022 International Conference on Computing and Data Science*, 2022, doi: 10.1145/3561613.3561650.
- [6] A. S. Mohammad, T. G. Jarullah, M. T. S. Al-Kaltakchi, J. Alshehbi Al-Ani, and S. Dey, "IoT-MFaceNet: Internet-of-Things-Based Face Recognition Using MobileNetV2 and FaceNet Deep-Learning Implementations on a Raspberry Pi-400," *Journal of Low Power Electronics and Applications*, vol. 14, no. 3, Art. no. 46, 2024, doi: 10.3390/jlpea14030046.
- [7] A. A. A. Aboluhom and I. Kandilli, "Face recognition using deep learning on Raspberry Pi," *The Computer Journal*, vol. 67, no. 10, pp. 3020–3030, 2024, doi: 10.1093/comjnl/bxae066.
- [8] F. Boutros, P. Siebke, M. Klemm, N. Damer, F. Kirchbuchner, and A. Kuijper, "PocketNet: Extreme lightweight face recognition network using neural architecture search and multistep knowledge distillation," *IEEE Access*, vol. 10, pp. 46823–46833, 2022, doi: 10.1109/ACCESS.2022.3170561.
- [9] A. Tsai, Y.-Y. Ou, W.-C. Wu, and J.-F. Wang, "Integrated SSD and pre-trained CNN for partially occluded face recognition," *IEEE Access*, vol. 9, pp. 164148–164158, 2021, doi: 10.1109/ACCESS.2021.3133446.
- [10] Q. Guo, "Occluded face recognition based on deep learning," *Frontiers in Computer Intelligence System*, vol. 5, no. 2, 2023, doi: 10.54097/fcis.v5i2.13134.
- [11] M. Rodrigo, C. Cuevas, and N. García, "Comprehensive comparison between vision transformers and convolutional neural networks for face recognition tasks," *Scientific Reports*, vol. 14, 2024, doi: 10.1038/s41598-024-72254-w.
- [12] F. Zhao, P. Zhang, R. Zhang, and M. Li, "DGFaceNet: Lightweight and efficient face recognition," *Engineering Applications of Artificial Intelligence*, vol. 124, Art. no. 106513, 2023, doi: 10.1016/j.engappai.2023.106513.
- [13] M. Rouhsedaghat, Y. Wang, S. Hu, S. You, and C.-C. J. Kuo, "Low-Resolution Face Recognition in Resource-Constrained Environments," *Pattern Recognition Letters*, vol. 149, pp. 193–199, 2021, doi: 10.1016/j.patrec.2021.05.009.
- [14] F. Perez-Montes, J. Olivares-Mercado, G. Sanchez-Perez, G. Benitez-Garcia, L. Prudente-Tixteco, and O. Lopez-Garcia, "Analysis of Real-Time Face-Verification Methods for Surveillance Applications," *Journal of Imaging*, vol. 9, no. 2, Art. no. 21, 2023, doi: 10.3390/jimaging9020021.
- [15] J. Zhumagalieva and D. Kumalakov, "Face recognition with Siamese Neural Networks," *Journal of Problems in Computer Science and Information Technology*, vol. 3, 2025, doi: 10.26577/jpcsit20253102.
- [16] Miftakhurrokhmat, R. A. Rajagade, and R. Rahmadi, "Classroom Attendance Based on Smiling Face Patterns and Nearby Wifi with Deep Learning," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 1, pp. 31–38, 2021, doi: 10.29207/resti.v5i1.2575.
- [17] D. Liu, M. Liang, and Y. Guo, "Face recognition using CNN based on Euclidean distance," *Journal of Physics: Conference Series*, vol. 1820, no. 1, Art. no. 012087, 2021, doi: 10.1088/1742-6596/1820/1/012087.
- [18] A. Zhalgas, B. Amirgaliyev, and A. Sovet, "Robust Face Recognition Under Challenging Conditions: A Comprehensive Review of Deep Learning Methods and Challenges," *Applied Sciences*, vol. 15, no. 17, Art. no. 9390, 2025, doi: 10.3390/app15179390.
- [19] Z. Rasheed, H. Maqsood, K. S. Al-Maqroun, S. Ditta, M. Manzoor, and I. Ashraf, "Automated classification of brain tumors from magnetic resonance imaging using deep learning," *Brain Sciences*, vol. 13, no. 4, Art. no. 602, 2023, doi: 10.3390/brainsci13040602.
- [20] S. Seenath and M. Dharmaraj, "Conformer-Based Human Activity Recognition Using Inertial Measurement Units," *Sensors*, vol. 23, no. 17, Art. no. 7357, 2023, doi: 10.3390/s23177357.
- [21] S. Tehsin, A. Hassan, F. Riaz, I. M. Nasir, N. L. Fitriyani, and M. Syafrudin, "Enhancing Signature Verification Using Triplet Siamese Similarity Networks in Digital Documents," *Mathematics*, vol. 12, no. 17, Art. no. 2757, 2024, doi: 10.3390/math12172757.
- [22] N. El Fadel, "Facial Recognition Algorithms: A Systematic Literature Review," *Journal of Imaging*, vol. 11, no. 2, Art. no. 58, 2025, doi: 10.3390/jimaging11020058.
- [23] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," *Technical Report 07-49, University of Massachusetts, Amherst*, Oct. 2007. [Online]. Available: <http://vis-www.cs.umass.edu/lfw/>.
- [24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.
- [25] H. Kawai, K. Ito, H.-T. Chen, and T. Aoki, "FSerasing: Improving face recognition with data augmentation using face parsing," *IET Biometrics*, 2024, doi: 10.1049/2024/6663315.
- [26] B. Desai and S. Satapathy, "Facial recognition using Siamese Neural Network and data augmentation techniques," in *Proceedings of the 2024 2nd World Conference on Communication & Computing (WCONF)*, IEEE, 2024, pp. 1–6, doi: 10.1109/WCONF61366.2024.10692075.
- [27] S. Jain, A. Pundir, S. Singh, and G. Saxena, "Exploring face recognition: A comparative study using loss-based methodologies," in *Proceedings of the 2023 IEEE 2nd International Conference on Data, Decision and Systems (ICDDS)*, IEEE, 2023, pp. 1–5, doi: 10.1109/ICDDS59137.2023.10434764.
- [28] S. Soleymani, B. Chaudhary, A. Dabouei, J. Dawson, and N. M. Nasrabadi, "Differential morphed face detection using deep Siamese networks," in *Lecture Notes in Computer Science (LNCS)*, vol. 12535, Springer, 2020, pp. 495–506, doi: 10.1007/978-3-030-68780-9_44.
- [29] B. Ciapas and P. Treigs, "Centre-loss: A preferred class verification approach over sample-to-sample in self-checkout

- products datasets,” *IET Computer Vision*, vol. 18, no. 1, pp. 1–12, 2024, doi: [10.1049/cvi2.12302](https://doi.org/10.1049/cvi2.12302).
- [30] H. Prashanth and M. Rao, “Roofline performance analysis of DNN architectures on CPU and GPU systems,” in *Proceedings of the 2024 25th International Symposium on Quality Electronic Design (ISQED)*, IEEE, Mar. 2024, pp. 1–8, doi: [10.1109/ISQED60706.2024.10528752](https://doi.org/10.1109/ISQED60706.2024.10528752).
- [31] Z. Lyu, D. Zhang, and J. Luo, “A GPU-free real-time object detection method for apron surveillance video based on quantized MobileNet-SSD,” *IET Image Processing*, vol. 16, no. 8, pp. 2196–2209, 2022, doi: [10.1049/ipr2.12483](https://doi.org/10.1049/ipr2.12483).
- [32] C. Song, C. Ye, Y. Sim, and D. S. Jeong, “Hardware for Deep Learning Acceleration,” *Advanced Intelligent Systems*, vol. 6, no. 10, 2024, doi: [10.1002/aisy.202300762](https://doi.org/10.1002/aisy.202300762).