Jurnal Teknik Informatika (JUTIF) Vol. 6, No. 2, April 2025, Page. 521-538
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4169

Enhanced Identity Recognition Through the Development of a Convolutional Neural Network Using Indonesian Palmprints

Diah Mitha Aprilla*[1], Ario Yudo Husodo[2], I Gede Pasek Suta Wijaya[3]

[1,2,3]Information Technology, Universitas Mataram, Indonesia

Email: [1]diahmithaaprilla@gmail.com

Received : Dec 4, 2024; Revised : Dec 23, 2024; Accepted : Dec 27, 2024; Published : Apr 26, 2025

Abstract

The use of palmprint as an identification system has gained significant attention due to its potential in biometric authentication. However, existing models often face challenges related to computational complexity and the ability to scale with larger datasets. This research aims to develop an efficient Convolutional Neural Network (CNN) model for palmprint identity recognition, specifically tailored to address these challenges. A novel contribution of this study is the creation of an original palmprint dataset consisting of 700 images from 50 Indonesian college students, which serves as a foundation for future research in Southeast Asia. The dataset includes different scenarios with varying input sizes (32x32, 64x64, 96x96 pixels) and the number of classes (30, 40, 50) to assess the model's scalability and performance. Three CNN architectures were designed with varying layers, activation functions, and dropout strategies to capture the unique features of palmprints and improve model generalization. The results show that the best-performing model, Model 3, which incorporates dropout layers, achieved 95% accuracy, 96% precision, 95% recall, and 95% F1-score on 50 classes with 1.2 million parameters. Model 1 achieved 98% accuracy, 99% precision, 98% recall, and 98% F1-score on 40 classes with 1.7 million parameters. These findings demonstrate that the proposed CNN models not only achieve high accuracy but also maintain computational efficiency, offering promising solutions for real-time palmprint authentication systems. This research contributes to the advancement of biometric authentication systems, with significant implications for real-world applications in Southeast Asia.

Keywords : *Batch normalization, Biometric authentication, Convolutional Neural Network, Dropout, Palmprint.*

1. INTRODUCTION

The palmprint is a distinctive feature of the human body, exhibiting a unique pattern of lines and a distinctive overall shape of the hand surface. The development of palmprint as an identification system is widely developed because palm features, such as shape, contour, and global texture patterns, provide more information than fingerprints [1], [2]. Even between the right and left palms have different unique characteristics. With these unique characteristics can be used as verification of a person's identity because the existing characteristics are stable and fixed and not easy to duplicate or fake [3].

Several studies related to palms for the recognition of a person's identity have been conducted previously. The methods used in model building also vary. Research conducted by Ata, et al [4]. for palm image recognition by comparing 7 machine learning algorithms, namely Neural Network, SVM, Naïve Bayes, KNN, Random Forest, Tree, and Adaptive Boosting. From the study, it was found that the Neural Network algorithm has the best performance compared to other algorithms. Furthermore, research conducted by Pin, et al [5]. to get the performance comparison results of traditional machine learning algorithms, namely SVM and deep learning algorithms, namely Convolutional Neural

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4169

Network (CNN). From the study, it was obtained that CNN's performance was much better than SVM in image classification. This is because deep learning has better image recognition accuracy on large datasets and this algorithm has very complex and abstract features that allow CNN to capture deeper patterns [4], [5], [6]. On the other hand, traditional machine learning is able to provide better solutions for small datasets and limited features so that traditional machine learning is less able to capture very deep patterns [5], [7].

Based on related research that explains the effectiveness of each model for image recognition, it is found that the Convolutional Neural Network (CNN) algorithm is the most suitable model to use in palm image recognition. With supervised learning of the palm image, the CNN model can be used to learn deep patterns in the image [8], [9]. This is related to the features of the palm of the hand that are very complex so that it requires a model that is able to capture existing patterns where this ability is owned by the CNN model [10], [11].

There are several architectural variations on the Convolutional Neural Network model. In general, CNN is composed of convolution layer, pooling layer, and fully connected layer [12], [13]. These layers can be customized and developed according to the complexity of the task and the specific needs of the problem at hand. In addition, there are architectural models that apply transfer learning techniques [14], [15]. This technique uses learning or knowledge from previously trained models called pre-trained models. Using this technique can save training time because training is done from previously learned patterns [14], [16]. However, transfer learning also has the disadvantage that this technique is not always effectively used when the pre-trained model is irrelevant or not representative enough for the target task thus allowing no significant performance improvement from the pre-trained model used [17], [18]. There are various pre-trained models available, including: ResNet50, Inception, Xception, VGG16, MobileNet, DenseNet, EfficientNet and so on where each model has a different layer design [14], [16].

A number of studies have been conducted to develop palm recognition as an identification system with methods such as Neural Network and Convolutional Neural Network (CNN) [19], [20]. However, most of these studies use datasets that are not public or have limited access. Furthermore, no study has specifically developed a comprehensive palm image dataset of the Indonesian population that is publicly accessible. This creates a gap in the literature as biometric image recognition can be affected by geographical or race-specific factors [21]. Therefore, this research focuses not only on developing a better CNN architecture but also on introducing a new dataset collected directly from the local population that makes a significant contribution to the literature related to identity recognition based on palm image.

The contributions of this research include two main aspects. First, the collection of an original dataset of palm images from the Indonesian population, which has not been available as a publicly accessible dataset until now. This provides a strong foundation for future research, especially involving populations in the Southeast Asian region. Secondly, the authors propose the development of a CNN architecture that is specifically designed to capture complex and specific patterns in palm images, especially to overcome the limitations often found in pre-trained models. While pre-trained models perform well in general image recognition, they are not always effective in palm recognition tasks because their architecture is not optimized to capture the unique patterns present in palms, such as shape and variations between individuals.

The CNN architecture proposed in this study is designed to optimize the network's ability to recognize holistic features, such as the unique shapes and contours present in palm images, including palm and finger proportions, as well as the overall contours that distinguish each individual [22]. This development was carried out by composing several convolution layers that were specifically tailored to improve the palm shape extraction capability, and pooling layers that were optimized to retain

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4169

important geometric details of the palm image, despite the resize process. In addition, the model is also equipped with dropout and batch normalization mechanisms to ensure the model can handle the data better and reduce the risk of overfitting [23], [24].

One of the main advantages of the proposed architecture is its flexibility in customizing the convolution and pooling layers based on the dataset characteristics. This architecture allows for more efficient adjustment of hyperparameters, such as kernel size, number of filters, and regularization techniques to capture deeper and more complex patterns present in palm images [25]. Compared to pre-trained models that rely on general features, this architecture is more specific to the task of palm image recognition and thus provides better performance especially in the case of images that have variations such as those found in this new dataset.

This study aims to develop a CNN architecture optimized for palmprint recognition using a novel Indonesian dataset, addressing the limitations of pre-trained models. With this contribution, the research conducted not only expands the scope of using CNN models in palm image recognition but also provides a new architecture design that is more efficient and accurate in capturing the shape and overall features of the palm. Furthermore, coupled with a new dataset reflecting the Indonesian population, it has the potential to open new research avenues in palm biometrics.

2. METHOD

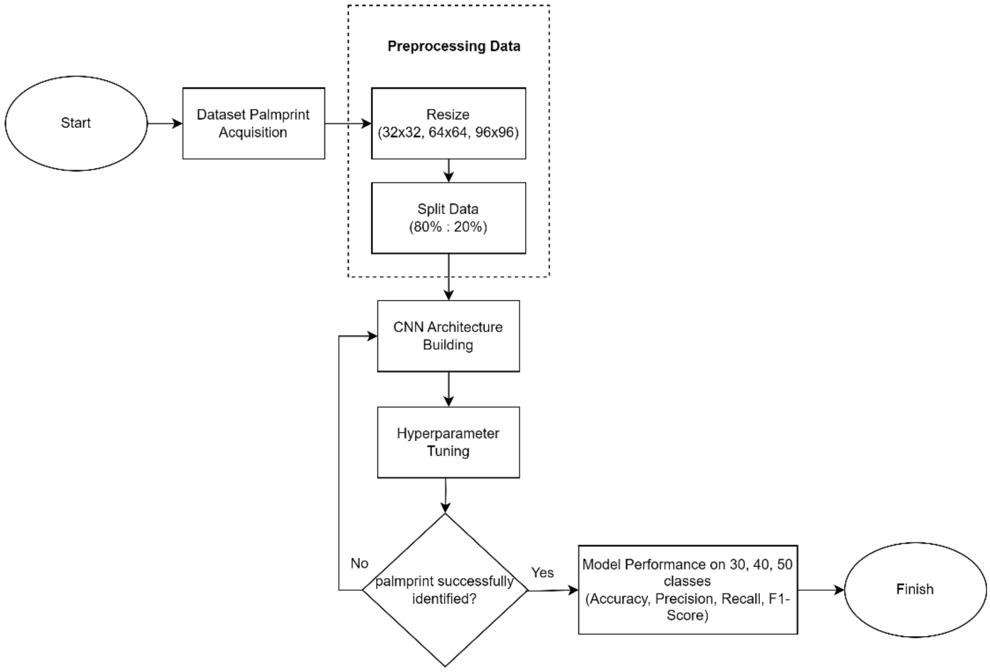

Figure 1. Research Flowchart

Figure 1 is a research flowchart that starts with a collecting palmprint data through an acquisition process that involves taking pictures of palms from Mataram University students which includes variations in image capture so that the model can learn the characteristics of the palm effectively. Next, split the data into two, namely train and test data. After that, the construction of the CNN model with the arrangement of layers through trial and error also justifyies the hyperparameter. Finally, the model is evaluated the model performance using test data. If the image is well recognized with an accuracy above 85%, it will continue to display results in the form of evaluation metrics including accuracy, precision, recall, and f1-score. Accuracy measures the extent to which the model

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4169

can correctly recognize the palm image, while precision and recall measure the quality of the prediction results.

2.1. Dataset Palmprint Acquisition

The dataset in this research was obtained through taking images of the right and left palms of 50 students of the Faculty of Engineering, Mataram University, consisting of male and female students. Each subject was captured 14 times, including 7 images of the right palm and 7 images of the left palm. In total, 700 images were collected with a variety of shooting rotations to capture natural variations in hand positioning, simulating real-world conditions. The images were taken at different times of the day to introduce variations in lighting conditions, thus improving the generalization ability of the model. All images were captured inside a white square box with dimensions of 30 cm, using a 50 MP resolution camera, and illuminated by a 3-watt LED light with 270 lumens. The dataset acquisition process is shown in Figure 2.

Figure 2. Dataset Acquisition Process

Each palmprint image is labeled with ID 01 to 50 based on the identity of the participants, ensuring consistency and facilitating easy tracking for evaluation purposes. These labeled images encompass a range of angles to increase the robustness of the recognition system. Moreover, the images include various accessories worn by participants, such as rings, bracelets, watches, or long-sleeved clothing, which were not removed during the image acquisition process. This adds a layer of variability to the dataset, reflecting real-world scenarios.

The dataset was then used for several experiments, with varying numbers of classes: 30 classes, 40 classes, and 50 classes, to evaluate the performance of the proposed models. To access the dataset used in this study, it can be downloaded via the following link: Palmprint Dataset. Samples of the palmprint images can be seen in Figure 3.

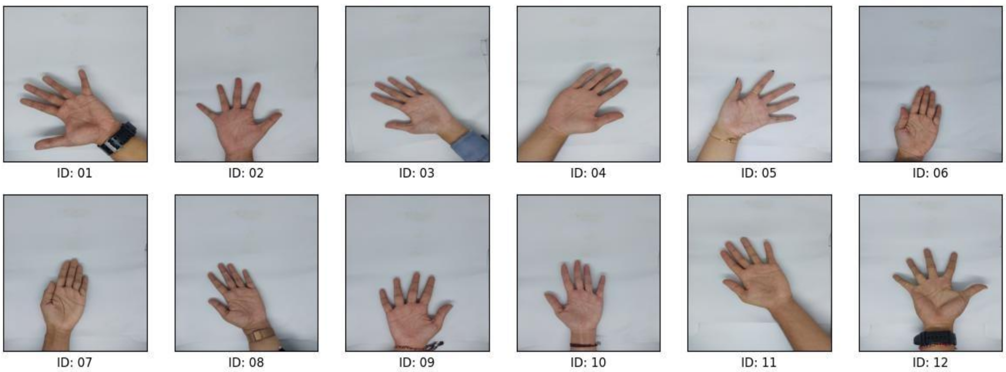

Figure 3. Palmprint Sample

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4169

2.2. Pre-processing Data

The pre-processing technique used in this study is that the image is converted from RGB to grayscale format to reduce image complexity and focus on shapes that are more relevant in palm identification by removing color information [26]. The pre-processing steps, including resizing and grayscale conversion, were implemented using TensorFlow, a widely used deep learning framework that provides efficient tools for image processing and neural network training [27].

2.2.1. Resize

The image is resized into different scenarios. The original size of the whole image, which is 1224 x 1632 pixels, then resizes the whole image with 3 different scenarios, including: 32 x 32 pixels, 64 x 64 pixels, and 96 x 96 pixels. This is done to determine the best model performance at different image sizes.

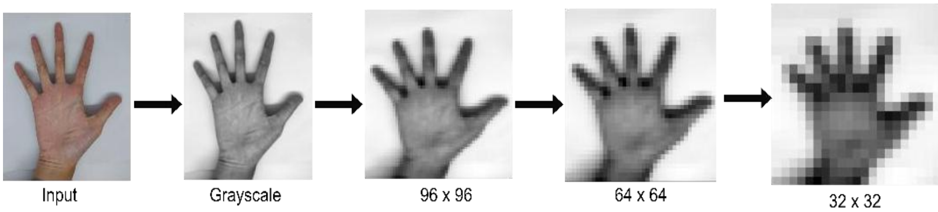

Figure 4. Image Resize

Figure 4 shows the pre-processing steps applied in this study. First, the image, which was originally in RGB format, was converted to grayscale in order to reduce information complexity and emphasize the structural shape of the palm. After conversion to grayscale, the image is then resized to three different size scenarios: 96 x 96 pixels, 64 x 64 pixels, and 32 x 32 pixels. The selection of each resize was driven by a specific reasoning. The 32 x 32 pixel size was chosen to assess the model's capacity for generalization from minimal data, thereby providing insights into the network's performance under extreme compression. The 64 x 64 pixel size was chosen as a balanced option, offering enough detail to capture key features while keeping computational requirements manageable. Meanwhile, 96 x 96 pixels allows for capturing finer structural details, ensuring that the model can leverage higher resolution data without incurring excessive computational costs.

The pixel sizes that have been selected are consistent with the study's emphasis on identifying holistic characteristics, such as the distinctive shapes and contours present in palmprint, including palm and finger proportions, as well as the overall contours that differentiate each individual [22]. Given the emphasis on these structural features rather than specific palm lines, higher pixel resolutions are deemed unnecessary, enabling the utilisation of smaller image sizes that are computationally efficient.

2.2.2. Split Data

The dataset utilized in this study was partitioned into a training set and a testing set, with a proportion of 80 : 20. This approach ensured that the model had sufficient samples from each class to accurately identify the distinctive patterns and characteristics of palmprint. The dataset contains 14 images per class, leading to the utilization of 80% of the images for training, equivalent to 11 images per class. This approach ensures sufficient variety to develop a robust model. The test data, constituting 20% of the dataset, involves the use of 3 images per class for evaluation. This is sufficient to assess the model's performance on data not encountered during the training process. The utilization of an equal proportion of data ensures that the evaluation outcomes are more indicative of the model's

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4169

generalizability when confronted with novel data variations. By employing a substantial portion of the data for training, the model is more likely to discern patterns rather than to memorize data, thereby mitigating the risk of overfitting [28].

2.3. CNN Architecture

Three Convolutional Neural Network (CNN) architecture models are proposed in this study. These three models are designed with variations in the number of layers and activation functions to explore their effect on model performance. Each model has significant differences in the arrangement of convolutional layers, the use of Batch Normalization, activation functions (ReLU and Leaky ReLU), and the application of dropout in the dense layer.

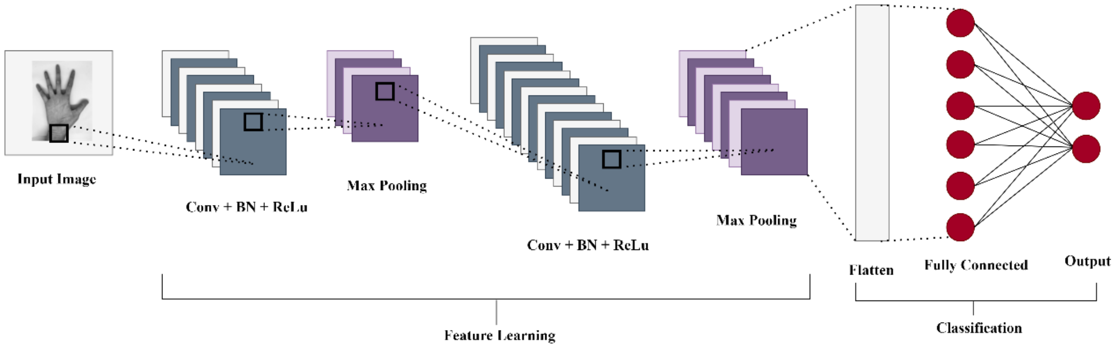

Figure 5. Visualization of Proposed CNN Architecture

Figure 5 shows the proposed CNN architecture for palmprint recognition starts with a grayscale input of the palm image, which is then passed through several stages of feature extraction. The model first applies a series of convolutional layers, each followed by batch normalization to stabilize learning and improve convergence. ReLU activation is used to introduce non-linearity, and max pooling is applied to reduce spatial dimensions and retain the most important features. These operations are repeated in multiple stages to progressively extract hierarchical features from the palmprint. After the feature learning stages, the model transitions to a flattening layer that converts the 2D feature maps into a 1D vector. This is followed by a fully connected layer, where the design varies across the proposed models. Some models use Leaky ReLU activation to allow small negative values, others incorporate Dropout to prevent overfitting, while some do not include either of these techniques. However, all three proposed models include regularizers to further enhance model generalization and prevent overfitting during training. This architecture aims to improve the accuracy and robustness of palmprint recognition.

Table 1. CNN Proposed Model Architecture 1

Block	Layer
Conv Block 1	Conv2D (filters: 32)
	Batch Normalization
	Activation ReLu
	MaxPooling2D (pool size: 2x2)
Conv Block 2	Conv2D (filters: 64)
	Batch Normalization
	Activation ReLu
	MaxPooling2D (pool size: 2x2)
Conv Block 3	Conv2D (filters: 96)

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4169

Block	Layer
	Batch Normalization
	Activation ReLu
	MaxPooling2D (pool size: 2x2)
Fully Connected	Flatten
	Dense 256 (ReLU, **regularizers 0.01**)
	Dense 256 (ReLU, **regularizers 0.01**)
	Dense
Output Layer	Activation Softmax

Table 1 shows the architecture of the first CNN model proposed in the study using the standard approach in CNN architecture with three convolution layers (Conv2D) followed by Batch Normalization, ReLU activation function, and MaxPooling2D layers. The addition of Batch Normalization after each convolution layer aims to stabilize the input distribution of each layer which can accelerate convergence and improve model performance [29]. The dense layer in this model uses twice the ReLU activation function with regularization to prevent overfitting. The model ends with a dense layer with Softmax activation function used for multi-class classification.

Table 2. CNN Proposed Model Architecture 2

Block	Layer
Conv Block 1	Conv2D (filters: 16)
	Batch Normalization
	Activation ReLu
	MaxPooling2D (pool size: 2x2)
Conv Block 2	Conv2D (filters: 32)
	Batch Normalization
	Activation ReLu
	MaxPooling2D (pool size: 2x2)
Conv Block 3	Conv2D (filters: 64)
	Batch Normalization
	Activation ReLu
	MaxPooling2D (pool size: 2x2)
Fully Connected	Flatten
	Dense 128 (ReLU, **regularizers 0.01**)
	Leaky ReLu (alpha: 0.1)
	Dense 256 (ReLU, **regularizers 0.01**)
	Leaky ReLu (alpha: 0.1)
Output Layer	Dense (Activation Softmax)

Table 2 shows the architecture of the second CNN model proposed in this research, which has the same basic structure as the first model in the convolution layer, but differs in the dense layer after the flattening process. In this model, Leaky ReLU activation function is used in some dense layers. Leaky ReLU was introduced to overcome the problem of "dying ReLU," where output units die during training [30]. The use of Leaky ReLU provides the ability for the model to capture more feature variations and improve the representation of more complex data. Furthermore, Softmax activation function is used in the output layer for classification.

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4169

Table 3. CNN Proposed Model Architecture 3

Block	Layer
Conv Block 1	Conv2D (filters: 16) Activation ReLu MaxPooling2D (pool size: 2x2)
Conv Block 2	Conv2D (filters: 32) Activation ReLu MaxPooling2D (pool size: 2x2)
Conv Block 3	Conv2D (filters: 64) Activation ReLu MaxPooling2D (pool size: 2x2)
Fully Connected	Flatten Dense 256 (Activation ReLU) **Dropout 0.4** Dense 256 (Activation ReLU) **Dropout 0.4** Dense 256 (Activation ReLU) **Dropout 0.5**
Output Layer	Dense (Activation Softmax)

Table 3 shows the architecture of the third CNN model proposed in the study which introduces significant changes by removing the Batch Normalization layer and adding multiple Dropout layers after the dense layer. The absence of Batch Normalization aims to reduce computation time, while the addition of Dropout after the dense layer is designed to prevent overfitting by partially randomizing the connections between neurons during training [29]. The model uses three convolution layers with ReLU activation function, followed by MaxPooling2D to shrink the feature size. After the Flattening process, three dense layers with ReLU activation functions accompanied by Dropout help the model become more robust and generalize to new data. The final layer uses Softmax Activation function for classification.

2.4. Hyperparameter Tuning

Hyperparameter tuning is conducted to optimize the model's performance. Several hyperparameters, such as optimizer, learning rate, number of epochs, and batch size were adjusted to optimize the model results. The selection of the Adam optimizer was based on its ability to adaptively adjust the learning rate, while a learning rate value of 0.001 was chosen to ensure stable convergence during the training process [31]. The number of 150 epochs used was set to achieve a balance between training time and model accuracy. The relatively small batch size used allows for more frequent updates of the model parameters within a single epoch thus helping the model to converge faster [32]. This size also considers the memory capacity of the GPU used to ensure the training process can run smoothly without running out of memory. The hyperparameter tuning used in the experiments are shown in Table 4.

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4169

Table 4. Hyperparameter Settings

Hyperparameter	Value
Optimizer	Adam
Learning Rate	0.001
Epochs	150
Batch Size	32

2.5. Evaluation Metrics

The evaluation metrics provide a quantitative basis for the assessment of the model's capability to process and analyse palmprint data with precision. By employing these metrics, the performance of the model can be measured in terms of its predictive accuracy, reliability, and consistency across a range of test scenarios. The evaluation of the CNN model is focused on key metrics such as accuracy, precision, recall, and f1-score, which provide a robust and comprehensive assessment of the model's performance [33].

2.5.1. Accuracy

Accuracy is the most commonly used metric for evaluating a model's performance. It calculates the proportion of correct predictions to the total number of predictions made by the model. The formula is as follows [33]:

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} \tag{1}$$

True Positive (TP): The number of positive samples correctly predicted as positive
True Negative (TN): The number of negative samples correctly predicted as negative
False Positive (FP): The number of negative samples incorrectly predicted as positive
False Negative (FN): The number of positive samples incorrectly predicted as negative

2.5.2. Precision

Precision measures the accuracy of positive predictions made by the model. This metric is highly relevant in scenarios where false positive predictions have significant consequences, such as in security systems. Precision helps evaluate how well the model avoids false positives, ensuring that positive predictions are highly reliable. The formula is [33]:

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \tag{2}$$

2.5.3. Recall

Recall or sensitivity, measures the extent to which the model can identify all positive samples in the dataset. Recall is crucial in applications where failing to detect positive cases (false negatives) has serious implications. The formula is [33] :

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \tag{3}$$

2.5.4. F1-Score

F1-Score is the harmonic mean of Precision and Recall. This metric balances the trade-off between the two, especially in situations where there is an imbalance between the number of positive and negative classes. F1-Score provides a comprehensive view of the model's ability to make accurate and relevant predictions for the positive class. The formula is [33]:

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4169

$$F1 - Score = 2\frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

3. RESULT

The performance of the proposed architecture model in identifying a person's identity based on palm image through experiments was conducted by dividing the data into 80% training data and 20% testing data. The experiment using palmprint images with different resize scenarios (32 x 32, 64 x 64, and 96 x 96 pixels) and varying numbers of classes (30, 40, and 50 classes). The test results are evaluated based on several metrics, namely precision, recall, f1-score, and accuracy. In addition, there are parameters that show the computational complexity of each model. The performance results of the architecture model proposed in this study are shown below.

3.1. Model Performance on 30 Classes

The first experiment was conducted by testing each proposed model with an experiment of 30 classes and scenario resize of 32 x 32, 64 x 64, and 96 x 96 pixels. The number of classes used in the first experiment was done to compare the performance of the proposed model with the research conducted in [34] using palm images from 30 people. After the training process for 150 epochs, an evaluation was carried out on 84 testing data and the best accuracy was obtained by Proposed Model 3 with an input size of 32 x 32 pixels, achieving an accuracy of 94%.

Although this is slightly lower than the 97% accuracy achieved by MobileNet in [34], Proposed Model 3 demonstrated superior performance in terms of precision, achieving 95%, which is 1% higher than the baseline model. The inclusion of Dropout layers in proposed model effectively reduces overfitting, making the model more robust for real-world applications. Moreover, the computational complexity of Proposed Model 3 was significantly reduced, as evidenced by the parameter count of 425 thousand, compared to 3.5 million parameters in MobileNet [34]. This reduction in parameter size directly contributes to faster computation times and improved efficiency. These results highlight the importance of carefully balancing architectural complexity and performance. While larger models like MobileNet can achieve slightly higher accuracy, they come at the cost of increased computational requirements.

Table 5. Result Evaluation 30 Classes

Architecture	Resize	Precision	Recall	F1-Score	Accuracy	Parameters
Proposed Model 1	32	0.93	0.91	0.90	0.90	541,950
	64	0.94	0.92	0.91	0.92	1,721,598
	96	0.92	0.90	0.89	0.89	3,687,678
Proposed Model 2	32	0.95	0.93	0.93	0.93	195,678
	64	0.95	0.93	0.92	0.93	588,894
	96	0.93	0.91	0.89	0.90	1,244,254
Proposed Model 3	**32**	**0.95**	**0.94**	**0.94**	**0.94**	**424,990**
	64	0.93	0.89	0.89	0.89	1,211,422
	96	0.95	0.92	0.91	0.92	2,522,142

Table 5 shows that Proposed Model 3 has the best overall balance between performance and computational efficiency for the 30 class scenario. For the 32 x 32 resize, Proposed Model 3 achieves the highest accuracy of 94%, with precision, recall and F1 scores of 95%, 94% and 94% respectively. These results are superior to the other models for the same resize scenario. Furthermore, the number of parameters for Proposed Model 3 is only 425 thousand, which is significantly less than Proposed Model 1, which requires 541 thousand parameters for slightly lower accuracy and metrics. The result

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4169

of all evaluation metrics is also higher than the other models and the smaller parameter size makes it more suitable for real-time applications where computational efficiency is crucial.

3.2. Model Performance on 40 Classes

The second experiment was conducted by increasing the number of classes used to 40 classes. This was done to determine the performance of the proposed model when the number of datasets increases. Testing was performed on each proposed model using input resize scenarios of 32 x 32, 64 x 64, and 96 x 96 pixels. From the test results, the best performance was obtained in Proposed Model 1 with an input size of 64 x 64 with an accuracy of 98%, precision 99%, recall 98%, and f1-score 98%.

These results show a significant improvement compared to previous studies, where MobileNet [34] and CNN+LBP [35] achieved accuracies of 97% and 91%, respectively, on datasets with fewer classes (30 and 20 classes). The results indicate that the proposed model significantly improves performance when scaling up the number of classes. In terms of computational complexity, Proposed Model 1 remained efficient, generating only 1.7 million parameters, which is significantly lower than the 3.5 million parameters in previous studies. This reduction demonstrates the model's capability to handle increased class numbers without a substantial increase in computational requirements.

Table 6. Result Evaluation 40 Classes

Architecture	Resize	Precision	Recall	F1-Score	Accuracy	Parameters
Proposed Model 1	32	0.93	0.91	0.91	0.90	544,520
	64	**0.99**	**0.98**	**0.98**	**0.98**	**1,724,168**
	96	0.99	0.98	0.98	0.98	3,690,248
Proposed Model 2	32	0.92	0.90	0.90	0.89	198,248
	64	0.97	0.96	0.96	0.96	591,464
	96	0.98	0.97	0.97	0.97	1,246,824
Proposed Model 3	32	0.92	0.92	0.91	0.91	427,560
	64	0.95	0.93	0.92	0.92	1,213,992
	96	0.91	0.88	0.87	0.88	2,524,712

Table 6 show that Proposed Model 1 outperforms the other models in all evaluation metrics. In terms of resize scenarios, both 64 x 64 and 96 x 96 pixel sizes yield similar performance. However, it can be observed that the number of parameters generated for the 64 x 64 pixel resize is much lower, approximately half, at around 1.7 million compared to 3.6 million for the 96 x 96 pixel. This indicates that the smaller resize in Proposed Model 1 leads to better efficiency with a significantly reduced number of parameters. Additionally, the reduced parameter count ensures that the model remains computationally efficient, even with the increased number of classes.

3.3. Model Performance on 50 Classes

The last experiment was conducted by increasing the number of classes used to 50 classes. This test was conducted on each proposed model with a resize scenario of 32 x 32, 64 x 64, and 96 x 96 pixels. From the test results, the best performance was obtained at input size 64 x 64 with accuracy of 95%, precision 96%, recall 95% and f1-score 95%. The absence of Batch Normalization reduces computational overhead, while the addition of Dropout layers in the dense layers effectively mitigates overfitting, particularly in high-class scenarios. This design allows Proposed Model 3 to generalize well across a larger number of classes.

Compared to previous studies, the results of this experiment show a clear advantage of the proposed model. MobileNet [34], which achieved a slightly higher accuracy of 97% on 30 classes, exhibits a steep increase in the number of parameters, making it less suitable for scenarios requiring resource efficiency. Similarly, the CNN+LBP [35] performed significantly worse, with an accuracy of only 91% on 20 classes, indicating its limited scalability to larger datasets. In the Proposed Model 3, the number of parameters is 3 times less, which is 1.2 million when compared to MobileNet [34] with the number of parameters reaching 3.5 million. This indicates that even with a larger number of classes, the proposed model can achieve better performance in terms of computational efficiency as indicated by fewer parameters.

Table 7. Result Evaluation 50 Classes

Architecture	Resize	Precision	Recall	F1-Score	Accuracy	Parameters
Proposed Model 1	32	0.95	0.93	0.93	0.93	547,090
	64	0.93	0.93	0.91	0.92	1,726,738
	96	0.97	0.96	0.95	0.96	3,692,818
Proposed Model 2	32	0.94	0.89	0.88	0.89	200,818
	64	0.96	0.95	0.94	0.94	594,034
	96	0.95	0.94	0.93	0.94	1,249,394
Proposed Model 3	32	0.94	0.90	0.89	0.90	430,130
	64	**0.96**	**0.95**	**0.95**	**0.95**	**1,216,562**
	96	0.92	0.89	0.88	0.89	2,527,282

Table 7 show that Proposed Model 3 in resize 64 x 64 pixel generally delivers the best performance while the accuracy is slightly lower by 1% compared to Proposed Model 1. Proposed Model 3 excels in terms of computational efficiency. It generates fewer parameters, which indicates better efficiency. This trend continues across all resize scenarios, where Proposed Model 3 consistently maintains a lower parameter count, resulting in reduced computational demands. Thus, despite a slight decrease in accuracy, the overall performance of Proposed Model 3, including its efficiency with fewer parameters, makes it a more optimal choice for practical applications, especially in scenarios where model complexity and computational resources are concerns.

4. DISCUSSIONS

In comparative analysis, the proposed model demonstrates enhanced performance than the research of MobileNet [34] and CNN+LBP [35] measured by the metrics of precision, recall, f1-score, and accuracy. In addition, the parameters generated in the proposed model are also much less which results in a faster and more efficient computational process during training but still able to achieve the best performance. This indicates that a larger number of classes can create a model with fewer parameters. In addition to the architectural design, the number of parameters resulting from the model is also significantly less affected by the resize utilized. The smaller the input image size, the faster and more efficient the training computation time. In other words, even though the resize is done with quite small input sizes, namely 32 x 32, 64 x 64, and 96 x 96 pixels, the model is still able to learn the existing images without losing the unique features possessed by the palmprint image.

Table 8. Comparison with Previous Studies

Algorithm	Dataset	NClasses	Accuracy	Parameters (million)
Proposed Model 1	**Collected Palm Dataset**	**40**	**98%**	**1.7**
		50	96%	3.6
Proposed Model 2	Collected Palm Dataset	40	97%	1.2
		50	94%	1.2
Proposed Model 3	**Collected Palm Dataset**	40	92%	1.2
		50	**95%**	**1.2**
MobileNet [34]	BMPD	30	97%	3.5
CNN + LBP [35]	BMPD	20	91%	NA

Jurnal Teknik Informatika (JUTIF) Vol. 6, No. 2, April 2025, Page. 521-538
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4169

Table 8 shows a comparison of the model proposed by the researcher with previous studies MobileNet [34] and CNN+LBP [35]. It can be seen that the Proposed Models 1, 2, and 3 are able to identify palm images with a larger number of subjects. Proposed Model 1 achieved the highest accuracy of 98% on the Collected Palm Dataset with 40 classes, but with a relatively large number of parameters, namely 1.7 million. In contrast, Proposed Model 2 also uses the same dataset and number of classes, but has a smaller number of parameters, namely 1.2 million and slightly decreases the accuracy to 97%. This shows that with a smaller number of parameters, the model can still maintain almost equal accuracy despite a slight decrease.

In the 50 class scenario, Model 1 achieved the highest accuracy, reaching 96%. However, it has 3.6 million parameters, which makes it quite large in terms of model complexity. In comparison, Proposed Model 3, while achieving a slightly lower accuracy of 95%, offers a better balance between accuracy and model complexity. Despite the minor reduction in accuracy, Proposed Model 3 is considered more effective due to its more efficient use of parameters with only 1.2 million parameters, which makes it more suitable for classification tasks involving a larger number of classes.

On the other hand, MobileNet achieved 97% accuracy on the BMPD dataset with 30 classes, but with a much larger number of parameters of 3.5 million. This shows that although MobileNet can achieve high accuracy, it requires more parameters which can lead to higher computational resource usage. Meanwhile, the CNN + LBP model with 20 classes on the same dataset achieved 91% accuracy, but the number of parameters is not available so it cannot be compared directly regarding parameter efficiency.

From this analysis, it can be seen that using a smaller number of parameters, such as in Proposed Model 3, can achieve accuracy that is competitive with more complex models such as MobileNet, but with better resource efficiency. This is important when limited computing resources are a major consideration.

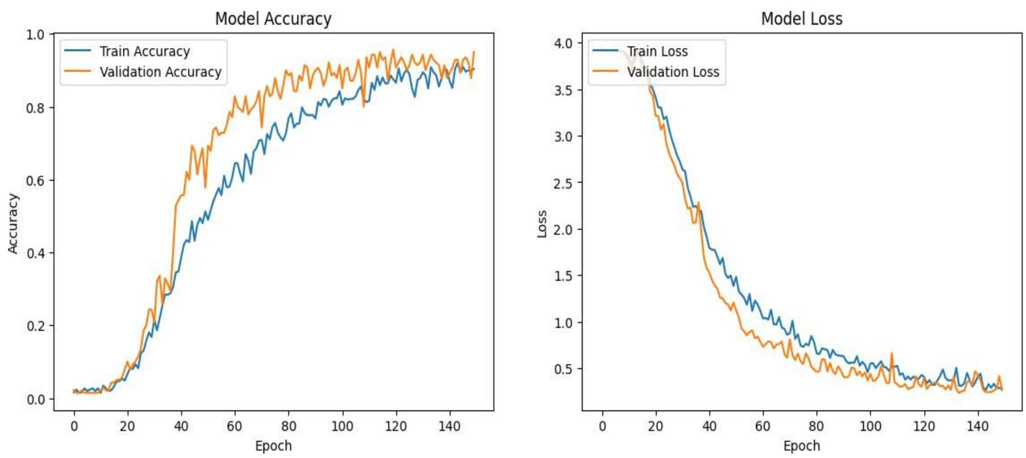

Figure 6. Plot Accuracy and Loss Proposed Model 3

Figure 5 shows the best accuracy and loss plots generated in this study for Proposed Model 3. The accuracy graph shows a steady increase as the number of epochs increases until it reaches 150 epochs. This shows that the model can continue to learn from the training data well, without experiencing a drop in performance on the testing data. In contrast, the loss graph shows a consistent decrease in the error value as the number of epochs increases for both training and testing data. The accuracy increases and remains stable without significantly decreasing, and the loss continues to decrease without signs of increase, indicating that the model is not overfitting. The model performed well with optimal generalization ability on new data.

P-ISSN: 2723-3863
E-ISSN: 2723-3871
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4169

The use of Dropouts in the Proposed Model 3 also contributes to the stability of the graph. Dropout is a regularization technique that randomly disables a number of neurons during training which helps prevent the model from overfitting the training data. In this graph, the effect of dropout can be seen as the accuracy of the testing data remains stable without large fluctuations and the loss continues to decrease as the epochs increase. In other words, dropout helps the model to learn more general and relevant features, rather than memorizing the training data. This explains proposed model 3 not only shows a consistent increase in accuracy, but also a continuous decrease in loss that reflects the model's ability to generalize well on data it has never seen.

The study's limitations are due to the dataset, which is limited to palmprint images from Indonesia. This means the findings may not fully represent palmprint diversity in Southeast Asia. A larger dataset from other Southeast Asian populations would provide a more comprehensive evaluation of the model's robustness and generalizability in this region. The proposed models in this study present significant contributions to the field of palmprint recognition by demonstrating scalability and efficiency. With fewer parameters compared to previous studies such as MobileNet [34], the proposed approaches achieve competitive accuracy while maintaining computational efficiency, making them highly suitable for resource-constrained environments. These findings have significant implications for real-world applications, particularly in designing real-time palmprint authentication systems for mobile devices or embedded systems. By optimizing parameter counts, these models can be deployed on devices with limited memory and processing power without compromising accuracy, making them suitable for diverse populations and environments. By optimizing the balance between performance and efficiency, this research contributes to the advancement of methodologies presented in previous studies.

5. CONCLUSION

This study shows that the proposed model is able to achieve better performance compared to previous studies in terms of accuracy, precision, recall, and f1-score, while reducing the number of parameters generated. Experiments used palm data with the number of classes varied from 30, 40, and 50. As a result, Proposed Model 1 showed the highest accuracy in 40 classes with 98% accuracy, 99% precision, 98% recall, and 98% f1-score with 1.7 million parameters. In the dataset with 50 classes, Model 1 once again showed the highest accuracy of 96%, but with a much larger parameter size of 3.6 million. In contrast, Proposed Model 3, which achieved a slightly lower accuracy of 95%, 96% precision, 95% recall, and 95% f1-score, managed to optimize the balance between accuracy and model complexity, utilizing only 1.2 million parameters. The efficiency of Proposed Model 3 makes it more suitable for handling larger classification tasks without a substantial increase in model complexity. Therefore, despite a slight decrease in accuracy, the efficiency and stability of Proposed Model 3 in managing classification with an increased number of classes makes it the optimal choice in this study. This research contributes to the development of efficient CNN models for palmprint recognition, reducing model complexity while maintaining high accuracy. It has practical implications in biometric applications and image-based identity recognition. To further enhance the model's generalization, future research could explore the inclusion of palmprint datasets from other Southeast Asian countries. This could provide a more diverse representation of hand characteristics and improve the robustness of the model across different populations. Additionally, implementing the model into real-world systems, such as mobile or embedded devices, could help evaluate its performance in practical scenarios. Testing the model on larger and more varied datasets will also help reduce bias and increase its applicability.

Vol. 6, No. 2, April 2025, Page. 521-538
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4169

CONFLICT OF INTEREST

The authors affirm that there are no conflicts of interest associated with the publication of this paper. All contributors have played an equal role in conducting the research and preparing the manuscript.

ACKNOWLEDGEMENT

The author would like to express gratitude to the Intelligent System Laboratory, Master of Information Technology, Universitas Mataram for supporting this research.

REFERENCES

[1] P. Pratyasha, S. Bharati, and A. P. Padhy, "A Comparative Analysis of Different Feature Extraction Techniques for Palm-print Images," *bioRxiv (Cold Spring Harbor Laboratory)*, pp. 01–12, May 2020, doi: 10.1101/2020.05.04.076240.

[2] N. B. K. Jaisawal, N. Dr. Y. Perwej, N. S. K. Singh, N. S. Kumar, N. J. P. Dixit, and N. N. K. Singh, "An Empirical Investigation of Human Identity Verification Methods," *Int J Sci Res Sci Eng Technol*, vol. 10, no. 1, pp. 16–38, Jan. 2023, doi: 10.32628/ijsrset2310012.

[3] S. Balakrishnan, K. Venkatesan, and M. S. S. Hameed, "An embarking user friendly palmprint biometric recognition system with topnotch security," in *Proceedings - 5th International Conference on Intelligent Computing and Control Systems, ICICCS 2021*, Institute of Electrical and Electronics Engineers Inc., May 2021, pp. 1028–1032. doi: 10.1109/ICICCS51141.2021.9432230.

[4] M. M. Ata, K. M. Elgamily, and M. A. Mohamed, "Toward Palmprint Recognition Methodology Based Machine Learning Techniques," *European Journal of Electrical Engineering and Computer Science*, vol. 4, no. 4, pp. 1–10, Jul. 2020, doi: 10.24018/ejece.2020.4.4.225.

[5] P. Wang, E. Fan, and P. Wang, "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning," *Pattern Recognit Lett*, vol. 141, pp. 61–67, Jan. 2021, doi: 10.1016/j.patrec.2020.07.042.

[6] I. Intan, S. Suryani, S. A. D. Ghani, Moh. Rifkan, and S. Bahri, "Implementation of CNN of Mobile-based COVID-19 Chest X-Ray Images," *CogITo Smart Journal*, vol. 10, no. 1, pp. 204–220, Jun. 2024, doi: 10.31154/cogito.v10i1.640.625-641.

[7] S. Y. Chaganti, I. Nanda, K. R. Pandi, T. G. N. R. S. N. Prudhvith, and N. Kumar, "Image Classification using SVM and CNN," *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, pp. 1–5, Mar. 2020, doi: 10.1109/ICCSEA49143.2020.9132851.

[8] K. Lee, S. Yi, S. Hyun, and C. Kim, "Review on the Recent Welding Research with Application of CNN-Based Deep Learning Part II: Model Evaluation and Visualizations," *Journal of Welding and Joining*, vol. 39, no. 1, pp. 20–26, Feb. 2021, doi: 10.5781/jwj.2021.39.1.2.

[9] D. Bhatt *et al.*, "CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope," *Electronics (Switzerland)*, vol. 10, no. 20, pp. 1–28, Oct. 2021, doi: 10.3390/electronics10202470.

[10] I. K. G. D. Putra, D. Witarsyah, M. Saputra, and P. Jhonarendra, "Palmprint Recognition Based on Edge Detection Features and Convolutional Neural Network," *Int J Adv Sci Eng Inf Technol*, vol. 11, no. 1, pp. 380–387, Feb. 2021, doi: 10.18517/ijaseit.11.1.11664.

[11] S. Chen, Z. Guo, J. Feng, and J. Zhou, "An Improved Contact-Based High-Resolution Palmprint Image Acquisition System," *IEEE Trans Instrum Meas*, vol. 69, no. 9, pp. 6816–6827, Sep. 2020, doi: 10.1109/TIM.2020.2976081.

[12] Y. D. Zhang, S. C. Satapathy, S. Liu, and G. R. Li, "A five-layer deep convolutional neural network with stochastic pooling for chest CT-based COVID-19 diagnosis," in *Machine Vision and Applications*, Springer Science and Business Media Deutschland GmbH, Feb. 2021. doi: 10.1007/s00138-020-01128-8.

[13] A. Zafar *et al.*, "A Comparison of Pooling Methods for Convolutional Neural Networks," *Applied Sciences*, vol. 12, no. 17, p. 8643, Sep. 2022, doi: 10.3390/app12178643.
[14] G. Vrbančič and V. Podgorelec, "Transfer learning with adaptive fine-tuning," *IEEE Access*, vol. 8, pp. 196197–196211, 2020, doi: 10.1109/ACCESS.2020.3034343.
[15] M. A. H. Akhand, S. Roy, N. Siddique, M. A. S. Kamal, and T. Shimamura, "Facial emotion recognition using transfer learning in the deep CNN," *Electronics (Switzerland)*, vol. 10, no. 9, May 2021, doi: 10.3390/electronics10091036.
[16] A. W. Salehi *et al.*, "A Study of CNN and Transfer Learning in Medical Imaging: Advantages, Challenges, Future Scope," *Sustainability (Switzerland)*, vol. 15, no. 7, pp. 1–28, Apr. 2023, doi: 10.3390/su15075930.
[17] Z. Zhao, L. Alzubaidi, J. Zhang, Y. Duan, and Y. Gu, "A comparison review of transfer learning and self-supervised learning: Definitions, applications, advantages and limitations," May 15, 2024, *Elsevier Ltd*. doi: 10.1016/j.eswa.2023.122807.
[18] A. Safonova, G. Ghazaryan, S. Stiller, M. Main-Knorn, C. Nendel, and M. Ryo, "Ten deep learning techniques to address small data problems with remote sensing," Dec. 01, 2023, *Elsevier B.V.* doi: 10.1016/j.jag.2023.103569.
[19] J. Prakash Veigas and S. Kumari, "Deep learning approach for Touchless Palmprint Recognition based on Alexnet and Fuzzy Support Vector Machine 551 Original Scientific Paper," *International Journal of Electrical and Computer Engineering Systems*, vol. 13, no. 7, pp. 551–559, Sep. 2022, doi: https://doi.org/10.32985/ijeces.13.7.
[20] S. Trabelsi, D. Samai, F. Dornaika, A. Benlamoudi, K. Bensid, and A. Taleb-Ahmed, "Efficient palmprint biometric identification systems using deep learning and feature selection methods," *Neural Comput Appl*, vol. 34, no. 14, pp. 12119–12141, Jul. 2022, doi: 10.1007/s00521-022-07098-4.
[21] W. M. Matkowski, T. Chai, and A. W. K. Kong, "Palmprint Recognition in Uncontrolled and Uncooperative Environment," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1601–1615, 2020, doi: 10.1109/TIFS.2019.2945183.
[22] L. Fei, B. Zhang, L. Zhang, W. Jia, J. Wen, and J. Wu, "Learning Compact Multifeature Codes for Palmprint Recognition from a Single Training Image per Palm," *IEEE Trans Multimedia*, vol. 23, pp. 2930–2942, 2021, doi: 10.1109/TMM.2020.3019701.
[23] C. F. G. Dos Santos and J. P. Papa, "Avoiding Overfitting: A Survey on Regularization Methods for Convolutional Neural Networks," *ACM Comput Surv*, vol. 54, no. 10, Jan. 2022, doi: 10.1145/3510413.
[24] B. Jabir and N. Falih, "Dropout, a basic and effective regularization method for a deep learning model: A case study," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 2, pp. 1009–1016, Nov. 2021, doi: 10.11591/ijeecs.v24.i2.pp1009-1016.
[25] A. Habring and M. Holler, "Neural-network-based regularization methods for inverse problems in imaging," *GAMM Mitteilungen*, vol. 47, no. 4, pp. 1–35, Nov. 2024, doi: 10.1002/gamm.202470004.
[26] I. Zeger, S. Grgic, J. Vukovic, and G. Sisul, "Grayscale Image Colorization Methods: Overview and Evaluation," *IEEE Access*, vol. 9, pp. 113326–113346, 2021, doi: 10.1109/ACCESS.2021.3104515.
[27] O. C. Novac *et al.*, "Analysis of the Application Efficiency of TensorFlow and PyTorch in Convolutional Neural Network," *Sensors*, vol. 22, no. 22, p. 8872, Nov. 2022, doi: 10.3390/s22228872.
[28] T. Rahman *et al.*, "Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images," *Comput Biol Med*, vol. 132, p. 104319, May 2021, doi: 10.1016/j.compbiomed.2021.104319.
[29] C. Garbin, X. Zhu, and O. Marques, "Dropout vs. batch normalization: an empirical study of their impact to deep learning," *Multimed Tools Appl*, vol. 79, no. 19–20, pp. 12777–12815, May 2020, doi: 10.1007/s11042-019-08453-9.
[30] Y. Bai, "RELU-Function and Derived Function Review," *SHS Web of Conferences*, vol. 144, p. 02006, 2022, doi: 10.1051/shsconf/202214402006.

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4169

[31] Y. Li, X. Ren, F. Zhao, and S. Yang, "A zeroth-order adaptive learning rate method to reduce cost of hyperparameter tuning for deep learning," *Applied Sciences (Switzerland)*, vol. 11, no. 21, p. 10184, Nov. 2021, doi: 10.3390/app112110184.
[32] H. Yong, J. Huang, D. Meng, X. Hua, and L. Zhang, "Momentum Batch Normalization for Deep Learning with Small Batch Size," in *Lecture notes in computer science*, 2020, pp. 224–240. doi: 10.1007/978-3-030-58610-2_14.
[33] Ž. Vujović, "Classification Model Evaluation Metrics," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, pp. 599–606, 2021, doi: 10.14569/IJACSA.2021.0120670.
[34] D. M. Aprilla, F. Bimantoro, and I. G. P. S. Wijaya, "The Palmprint Recognition Using Xception, VGG16, ResNet50, MobileNet, and EfficientNetB0 Architecture," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 8, no. 2, p. 1065, Apr. 2024, doi: 10.30865/mib.v8i2.7577.
[35] T. Hidayat, N. Khasanah, D. Uki, E. Saputri, U. Khultsum, and R. L. Pratiwi, "Klasifikasi Gambar Palmprint Berbasis Multi-Kelas Menggunakan Convolutional Neural Network," *Jurnal Sistem Informasi*, vol. 11, no. 1, pp. 01–06, Feb. 2022, doi: 10.51998/jsi.v11i1.474.

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4169