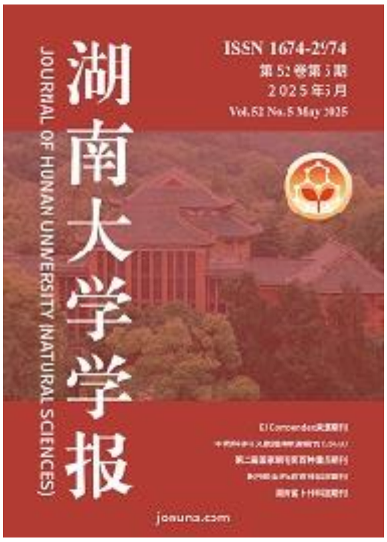

Journal of Hunan University

(Natural Sciences)

Vol. 52 No. 5

May 2025

Available online at

https://jonuns.com

Open Access Article

https://doi.org/10.55463/issn.1674-2974.52.5.19

Hyperparameter Analysis of Adam-Optimized Deep Transfer Learning for Indonesian Banknote-Denomination Recognition

Yuni Yamasari[1*], Bagas Ahmad Sadewa[1], Anita Qoiriah[1], Ervin Yohannes[1], Ricky Eka Putra[1], and Tohari Ahmad[2]

[1]Department of Informatics, Universitas Negeri Surabaya, Surabaya, Indonesia

[2]Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

* Corresponding author: yuniyamasari@unesa.ac.id

Article History:

Received: *April 30, 2025*
Revised: *May 29, 2025*
Accepted: *June 8, 2025*
Published online: *June 30, 2025*

Abstract: Accurately distinguishing Indonesian banknote denominations remains challenging for older adults and people with visual impairments, especially after the recent release of a redesigned currency. To address this gap, we propose an intelligent recognition system based on deep transfer learning with systematic hyperparameter optimization. Our approach fine-tunes a pre-trained ResNet-50 backbone while simultaneously calibrating the learning rate, batch size, and training epochs via an am-optimized grid search. Extensive experiments across multiple convolutional neural-network architectures confirmed that the ResNet-50 model, trained with a learning rate of 0.0001, batch sizes 20 and 15 epochs, achieved state-of-the-art performance: 99.29 % accuracy, 99.32 % precision, 99.29 % recall, and 99.29 % F1-score. These findings demonstrate that carefully tuned transfer-learning pipelines can deliver near-perfect classification of the new Indonesian banknote series while retaining strong generalization to previously unseen images, thereby offering a practical assistive tool for the visually impaired and elderly.

Keywords: Deep transfer learning; ResNet-50; Convolutional neural networks; Indonesian banknote recognition; Hyper-parameter optimization; Assistive computer vision.

基于 Adam 优化的深度迁移学习在印度尼西亚纸币面额识别中的超参数分析

摘要

对于老年人和视障人士而言，精准区分印度尼西亚纸币面额依然颇具挑战，尤其在新版纸币发行后这一问题更加突出。为弥补研究空缺，本文提出一种整合深度迁移学习与系统化超参数优化的智能识别系统。我们以预训练 ResNet-50 为骨干网络实施微调，并借助 Adam 优化的网格搜索同时校准学习率、批量大小和训练轮数。对多种卷积神经网络架构展开的大量实验表明，当学习率设定为 0.0001、批量大小为 20、训练 15 轮时，ResNet-50 模型取得了业界领先的性能：准确率 99.29%，精确率 99.32%，召回率 99.29%，F1 分数 99.29%。结果显示，经过精细调参的迁移学习流程不仅能对新版印度尼西亚纸币实现近乎完美的分类，还能在未见样本上保持优异的泛化能力，从而为视障人士与老年人提供实用的辅助工具。

关键词

深度迁移学习；ResNet-50；卷积神经网络；印度尼西亚纸币识别；超参数优化；辅助计算机视觉

1. Introduction

Daily cash transactions constitute a fundamental component of both individual and national economic activity. As Lubis [1] notes, routine exchanges of goods and services rely on the smooth use of currency as a medium of exchange. While handling money is effortless for sighted people, older adults and those with visual impairments often struggle to distinguish banknote denominations because of age-related vision decline or a total loss of sight. This difficulty can erode confidence in social interactions and expose vulnerable users to fraud and financial exploitation. The recent introduction of redesigned Indonesian banknotes has further complicated the task of quick and reliable banknote identification.

Over the past decade, computer vision systems powered by deep learning have revolutionized object recognition in images (Chen et al. [2]; Chai et al. [3]). Convolutional neural networks (CNNs) can automatically learn hierarchical visual features, enabling robust classification of complex image data.

Pribhdas [4] and Górriz et al. [5] highlighted how refinements in network depth and architecture, coupled with advances in explainable artificial intelligence (Ahmed et al. [6]), have expanded the scope of assistive technologies.

Banknote recognition has directly benefited from these advances. Sadyk et al. [7] reviewed CNN-based models tailored to currency identification, whereas Pham et al. [8] demonstrated that reflection-image analysis can grade banknote fitness with high precision. Alzubaidi et al. [9] provided a broader survey of CNN architectures and noted their suitability for fine-grained visual-classification tasks. Specific to the Indonesian rupiah, Ramadhan et al. [10] applied fuzzy k-nearest-neighbor methods, whereas Pham et al. [11] and Kumara et al. [12] showed that smartphone-captured images and lightweight YOLOv8 detectors, respectively, could facilitate denomination recognition for the visually impaired.

Transfer-learning strategies further boost performance, particularly when labelled currency datasets are limited. Mallesh et al. [13] emphasized that knowledge transfer improves the diagnostic accuracy in biomedical image classification, a principle that is equally applicable to currency. Meshram, Patil, and Meshram [14] evaluated multiple pretrained CNNs on a banknote dataset and identified ResNet-based models as top performers. Similarly, Linkon et al. [15] combined a lightweight CNN backbone with transfer learning to recognize Bangladeshi banknotes and achieved state-of-the-art results with a modest computational overhead.

Building on this literature, the present study investigates how systematic hyper-parameter tuning, specifically learning rate, batch size, and epoch count, affects the performance of an Adam-optimized ResNet-50-based transfer-learning pipeline for recognizing the latest series of Indonesian banknotes. By focusing on end-user needs and leveraging recent advances in deep learning, we aim to deliver an accessible, high-accuracy assistive tool that restores financial autonomy to the elderly and visually impaired individuals.

This combination makes deep learning and CNN powerful technologies in various image processing applications, including recognizing Indonesian banknotes.

Figure 1. Sample of Indonesian banknotes issued in 2022
Source: Developed by the authors

Previous research related to the application of technology for recognizing paper currency includes the application of the Fuzzy K-nearest neighbor method to identify paper currency [10]. These methods often rely on manually provided feature representations, which may not be robust enough to capture the richness of features present in banknote denomination images. Other studies have identified banknote denominations by applying deep learning methods [11]. Furthermore, other researchers detected banknotes for blind communities using the Yolo8 algorithm. The results of this previous research have helped the blind because and even some research results have achieved a very high accuracy of 99% [12]. However, the applied algorithm has weaknesses in overcoming the problem of large variations, such as banknote denominations under different conditions.

Related to deep learning, this method also has several weaknesses, including a lack of knowledge transfer capability [13], limitations in adapting to new tasks, and a lack of robustness to limited data. Next, we applied the deep transfer learning method to overcome this weakness. These previous studies applied deep transfer learning with a different focus; for example, applying this method to the classification of Indian currency so that the process is faster and accuracy increases [14], for a lightweight model so that the model can be developed into an IoT-based system for detecting banknotes [15], detecting counterfeit money [16], detecting banknotes, and then converting them to audio forms aimed at the blind [17], [18]. These studies applied the deep transfer learning method to banknote identification owing to the various advantages of deep transfer learning. For example, it can leverage existing knowledge from previous models, reduce reliance on large training datasets, and increase a model's adaptability to new tasks or domains. With the learned features, the learning results can be transferred to a new task by utilizing a smaller dataset, thereby accelerating the training process. This occurs because of the use of a network that has previously been trained and can extract a variety of features [19], so that it has a better ability to recognize special features.

Therefore, our research also applied deep transfer learning to identify nominal banknotes. To develop the pre-trained model, we explored several architectures: VGG-16, ResNet50, and EfficientNetV2M. Next, training and fine-tuning are performed on the pre-trained model with the Adam Optimizer. This was intended to produce the most optimal pre-trained model for recognizing Indonesian banknote denominations. Our research also carried out further testing and analysis of existing parameters to produce a model that could identify banknote denominations with the best performance.

2. MATERIAL AND METHOD

This chapter explains the data collection process and the techniques used in this research. This describes how the data were systematically gathered to ensure relevance. In addition, this chapter outlines the specific methods employed to support the research objectives.

This study used a quantitative experimental research method that focused on the systematic testing and evaluation of deep transfer learning models. This process involved structured experiments using three pre-trained CNN architectures (VGG-16, ResNet50, and EfficientNetV2M), followed by fine-tuning using the Adam optimizer. Key hyperparameters, including the learning rate, batch size, and number of epochs, vary across controlled trials. Model performance was measured using standard metrics such as accuracy, precision, recall, F1 score, and AUC to ensure rigorous analysis. This scientific method allows for reproducibility and supports empirical conclusions regarding the most effective model configuration for Indonesian banknote classifications.

2.1. Data Collection

The primary data used in this research were obtained by taking pictures directly using a cell phone camera. The image data used in this research were obtained from images of Indonesian banknotes for the year 2022. Image data were taken from banknotes in good condition and in conditions with signs of folding or damage. The research population included all images of Indonesian banknotes for the 2022 issue of seven nominal values, and the sample data consisted of 840 images.

The sampling process involves photographing images of banknotes under various conditions, such as upside-down, folded, and with poor lighting. An example of the sample image data is shown in Figure 1.

The objective of this study, namely the 2022 edition of Indonesian banknotes, was chosen because of its relevance and urgency in the real world. The newly launched banknotes pose significant challenges for the elderly and the blind in identifying denominations owing to changes in color, layout, and security features. By addressing this issue, this study makes a practical contribution, especially to the development of assistive technology, such as mobile applications or embedded systems that help users identify banknotes accurately and independently in daily financial transactions.

2.2. Proposed Method

The method proposed in this study consists of several steps, as shown in Figure 2. After the dataset containing image data of Indonesian banknote denominations is formed, the next step is data pre-processing. Preprocessing is the process of carrying out steps on data before it is used for training or further analysis, and aims to prepare the image so that it is ready for processing at the next stage. Good preprocessing will have a positive effect on the model performance. The desired features in an image are expected to become more prominent [17]. At this stage, the images are resized to 224 × 224 pixels in RGB color format and categorized into appropriate class labels. Next, the image data are saved in pickle format, and the class labels are saved in a CSV file for the analysis and training of banknote nominal recognition models.

Deep transfer learning involves leveraging models trained on large and diverse datasets and fine-tuning them for a specific task. In this study, we focus on three popular pre-trained models, VGG-16, ResNet50, and EfficientNetV2M, highlighting their architecture and implementation steps in the context of Indonesian banknote identification.

- VGG-16

VGG-16 is a convolutional neural network model proposed by the Visual Geometry Group at the University of Oxford. It is known for its simplicity and effectiveness and is characterized by its deep architecture with 16 layers. This architecture requires an input size of 224 × 224 × 3, and consists of 13 convolutional layers, five max-pooling layers, and three Fully Connected layers. The output of the convolutional layer can be formulated using Eq. (1).

$$O = \left(\frac{W-F+2P}{S} + 1\right) \times \left(\frac{H-F+2P}{S} + 1\right) \times K \qquad (1)$$

where W, H is the Width and Height of the input, respectively, F is the Filter size, P is the Padding, S is the Stride, and K is the Number of filters. The output of the pooling layer is formulated as in Equation (2).

$$O = \left(\frac{W-F}{S} + 1\right) \times \left(\frac{H-F}{S} + 1\right) \times D \qquad (2)$$

Where D is the Depth of input (number of channels)

- ResNet50

ResNet50 (Residual Network) is a deep neural network that introduces residual learning to solve the vanishing gradient problem in very deep networks. The architecture consists of 50 layers. It consists of residual blocks that utilize identity blocks and convolution with skip connections to bypass the nonlinearity. Its convolution layers included 49 convolution layers, followed by fully connected layers. Batch normalization

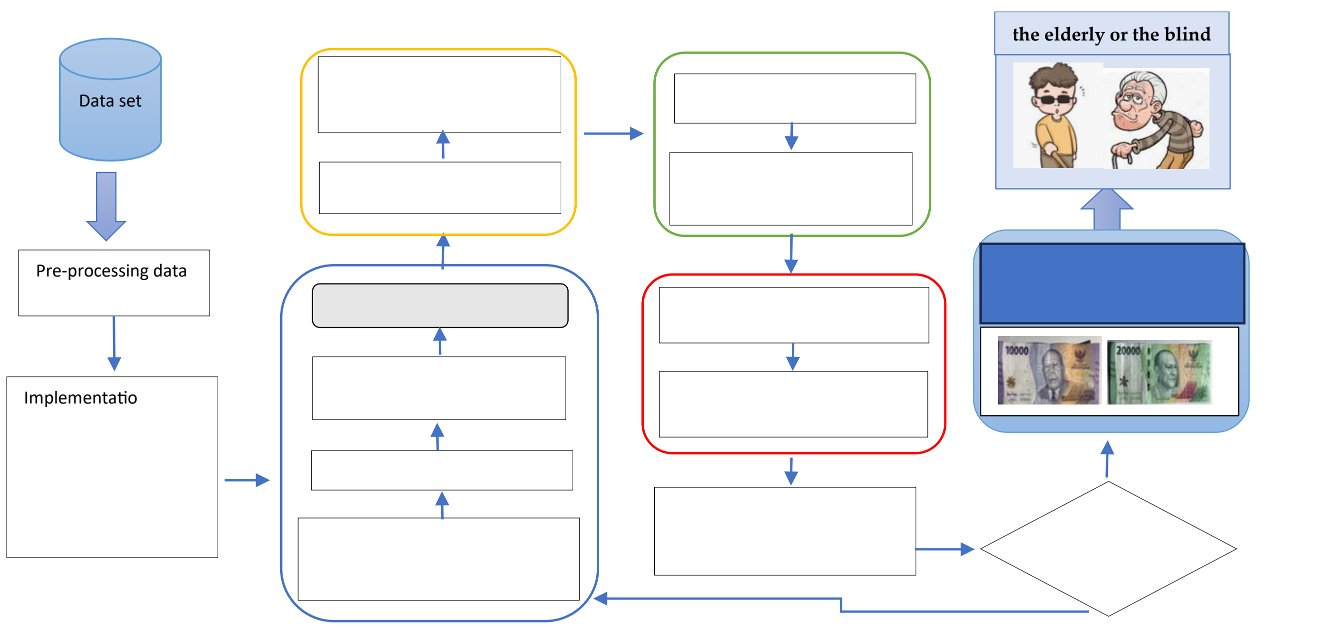

Figure 2. The proposed method to identify the Indonesian banknote nominal
Source: Developed by the authors

was applied after each convolution to normalize the activations, and the activation function used was the ReLU activation function [20]. The output of the residual block is given by Equation (3):

$$O = ReLU(F(x) + x) \quad (3)$$

Where F(x) is the convolutional operations, and x is the input tensor.

- EfficientNetV2M

EfficientNetV2M is an efficient neural network architecture with a 224 × 224 × 3 input size that utilizes compound scaling for adaptive depth, width, and resolution adjustments. It employs MBConv and Fused-MBConv layers, optimized for efficient feature extraction, and achieves high performance across diverse computational resources. The formula for the MBConv block output is given by Equation (4):

$$O = Conv2D\left(DepthwiseConv2D\left(Conv2D(x)\right)\right) \quad (4)$$

Where Conv2D is the standard convolution, and DepthwiseConv2D is Depthwise separable convolution. The formula for compound scaling is defined by Equation (5).

$$EfficientNet\ Scaling: \{d = \alpha^2, w = \beta^2, r = \gamma^2\} \quad (5)$$

where d is the network depth, w the network width, and r the input resolution.

Training and fine-tuning a pre-trained model with the Adam optimizer involves leveraging transfer learning capabilities to adapt a model's parameters to our dataset. Adam, an adaptive learning rate optimization algorithm, adjusts the learning rates for each parameter individually, resulting in efficient optimization and faster convergence. By initializing the model with pre-trained weights learned from ImageNet and fine-tuning it on a target dataset using Adam, our model can effectively learn specific features while benefiting from the general knowledge captured during pre-training.

The next step, testing and analyzing the performance of a pre-trained model, involves a systematic evaluation to determine how well the model generalizes to new, unseen data. This process started by using the pre-trained model to make predictions on a separate test dataset that was not used during training. Performance was quantified using metrics such as accuracy, precision, recall, F1-score, AUC, and computation time. These steps generate the best pre-trained model.

Related to hyperparameters, this research involves testing and analyzing the performance of different learning rates, batch sizes, and the number of epochs. First, testing and analysis were performed to determine the impact of various learning rate settings on the training process and model performance. This process begins by training the model multiple times using different learning rates. After each training session, key metrics such as accuracy, loss, and convergence speed were recorded. A lower learning rate might result in slow convergence, whereas a higher learning rate might cause the model to overshoot the optimal point or result in unstable training. By comparing these results, we can determine the most effective learning rate that balances fast convergence with stable and accurate training outcomes, ultimately enhancing the performance of the model on the validation or test dataset.

Second, testing and analyzing the performance of different batch sizes involves evaluating how various batch sizes affect the training efficiency and model performance. This process includes training the model multiple times using different batch sizes and recording metrics such as training time, model performance, and loss. Smaller batch sizes may lead to more stable updates and better generalization; however, they can be slower and more computationally intensive. Larger batch sizes can speed up training; however, they might result in less accurate gradient estimates and poorer generalization. By comparing these results, the optimal batch size can be determined by balancing the training speed, resource utilization, and model performance on the validation or test dataset.

Third, testing and analyzing the performance of the number of epochs involves evaluating how varying the number of epochs affects the model training and performance. This process includes training the model multiple times with different epoch counts and tracking key metrics such as training loss, validation loss, and accuracy. Fewer epochs might result in underfitting, where the model has not learned enough from the data, whereas too many epochs can lead to overfitting, where the model performs well on training data but poorly on unseen data. By examining these metrics across different epoch settings, the optimal number of epochs can be identified, ensuring that the model learns sufficiently without overfitting, thereby maximizing performance on the test dataset.

The next step was to evaluate the performance of the classification to assess the model's performance and effectiveness in correctly identifying different denominations. This process includes the use of a test dataset of banknote images that the model had not seen before. The performance metrics used were accuracy, precision, recall, F1-score, and AUC, which were calculated to determine how well the model distinguished between various denominations. Confusion matrices were analyzed to identify specific misclassifications. This comprehensive evaluation ensures that the model is reliable and accurate for practical applications in the identification of Indonesian banknotes.

Then, if the optimal model is not generated, the training and fine-tuning of the pre-trained model using the Adam Optimizer are repeated. If the optimal model has been generated, it can be used to identify Indonesian banknote denominations that help the elderly and blind.

3. EXPERIMENTAL RESULT

This section describes the data mined for this research. It also presents test results and evaluates the performance of the model. Finally, this chapter provides an analysis of the results.

3.1. Data Description

This research uses a total of 840 Indonesian banknote images for the 2022 issue year with nominal values of IDR 1,000.00, IDR 2,000.00, IDR 5,000.00, IDR 10,000.00, IDR 20,000.00, IDR 50,000.00, and IDR 100,000.00. The data consist of seven classes based on nominal currency, and each class has 120 images. This research uses balanced data because dataset imbalance is a crucial aspect of model training that needs to be addressed, as in previous studies [21], [22], [23], [24]. In addition, with a balanced dataset, the model can better generalize and learn the features that represent each class, ultimately improving model performance [25], [26].

3.2. Implementation of Deep Transfer Learning

The first step that must be taken is importing all libraries needed to build the model. Next, the dataset saved in Google Drive in pickle format is loaded. The dataset containing the images was converted into an array using the Numpy library. Image data are stored in the data array and label data are stored in the label data array. Next, One-hot encoding is performed, which is important in classification because it converts categorical variables into a format that the model can understand and allows it to process categorical data effectively to produce more accurate results [27], [28], [29], [30], [31]. Next, the dataset was divided into 80% training data and 20% testing data, which were used for model training. Once the data are ready to be processed, pre-training model initialization is performed. In this study, VGG-16, ResNet50, and EfficientNetV2M were used. The top layer of the model was then removed. The previous weights used the ImageNet dataset, and the input size expected by the model was 224 × 224 pixels with three RGB color channels. The base model is flattened, and custom layers are added, consisting of Dense, Batch Normalization, and Dropout. It ends with the Softmax activation function to perform classification with seven nominal currency classes. After the new model was built, the next step was to determine the optimizer for training the model. In this study, we used the Adam Optimizer by changing the learning rate values to 0.001, 0.0001, and 0.00001. The model is then compiled with metric accuracy and categorical cross-entropy for multiclass classification. Next, the model was trained for 5, 10, and 15 epochs. The batch size was varied between 10, 20, and 30. After the training process is complete, the next step is to calculate the resulting evaluation metric values.This research uses a total of 840 images of Indonesian banknotes for the 2022 issue year with a nominal value.

3.3. Testing and Analyzing the Performance of the Pre-trained Model

This study built a model using several pre-trained models. The next step is to test these models to find the most suitable one for identifying the nominal value of Indonesian banknotes. The pre-trained models used in this study were VGG-16, ResNet-50, and EfficientNetV2M. In addition to using pre-trained models, fine-tuning is required by adding additional layers. Therefore, this process also involves testing the effects of fine-tuning to determine the influence of the base pre-trained model and the pre-trained model with fine-tuning.

The testing of the pre-trained models was conducted using seven scenarios: the pre-trained models from the three architectures without tuning, the pre-trained models from the three architectures with tuning, and the pre-trained models with tuning.

After seven combination trials of pre-trained models with and without tuning were performed, evaluation results for accuracy, precision, recall, f1-score, and AUC metrics were obtained. The evaluation metric results obtained after several experiments are listed in Table 1. Based on the test results that have been carried out, the performance of the three pre-trained models is measured using evaluation metrics, namely accuracy, precision, recall, f1 score, and AUC. Based on the combination of testing the pre-trained model and parameters without tuning, the performance of pre-trained model ResNet50 had the highest performance, producing accuracy values of 98.57%, precision of 98.64%, recall of 98.57%, f1-score of 98.56%, and AUC of 99.99%.

Based on this performance comparison, it is known that not all pre-trained models are suitable for Indonesian banknote classification and the 2022 emission Indonesian currency dataset. Next, the ResNet50 pre-trained model was chosen for implementation in building the Indonesian banknote classification model.

To improve the model performance, fine-tuning is

Table 1. The result of Pre-trained Model Testing
Source: Developed by the authors

No	Pre-trained Model	Parameter	Metrics Evaluation					Time (second)
			Accuracy	Precision	Recall	F1-Score	AUC	
1	VGG16	without tuning	0.9429	0.9501	0.9429	0.9426	0.9984	167.941
2	ResNet50	without tuning	0.9857	0.9864	0.9857	0.9856	0.9999	163.46
3	EfficientNetV2	without tuning	0.9643	0.9659	0.9643	0.9645	0.9994	396.841
4	VGG16	tuning	0.8929	0.9263	0.8929	0.8906	0.9999	206.178
5	**ResNet50**	**tuning**	**0.9929**	**0.9931**	**0.9929**	**0.9928**	**0.9999**	**137.702**
6	EfficientNetV2	tuning	0.9857	0.9865	0.9857	0.9858	0.9999	383.815
7	without	tuning	0.8714	0.8746	0.8714	0.8716	0.9672	28.622

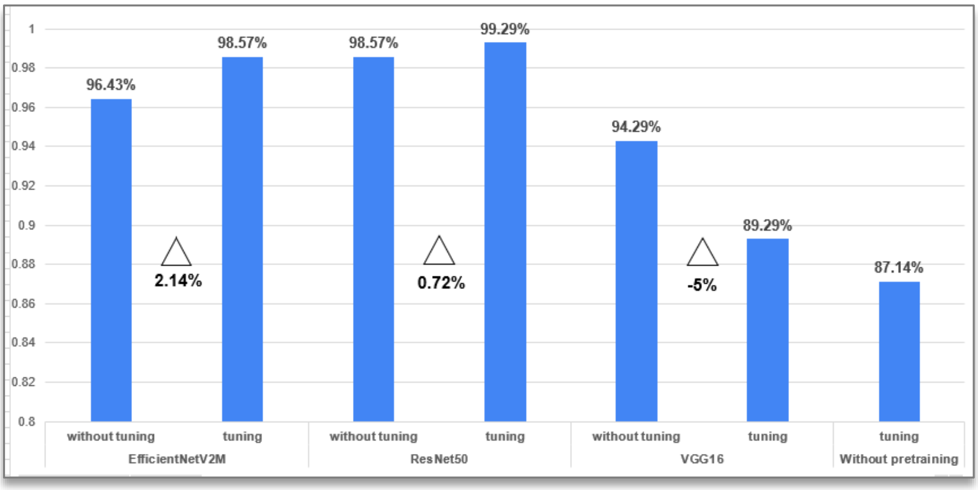

Figure 3. Comparison of the difference in accuracy levels of all pre-trained models after tuning
Source: Developed by the authors

performed by adding several layers and adjusting the hyperparameters. In this research, fine-tuning is performed by adding dense layers with values of 200, 100, and 50 by the RELU activation function. A layer is also added to prevent overfitting, namely batch normalization, which normalizes the output of the convolution such that it can increase the stability and generalization of the model. To reduce the number of neurons randomly, A dropout value of 30% was required. A visualization of the results from comparing the accuracy of the three pre-trained models with fine-tuning is shown in Figure 3.

From this figure, it can be seen that fine-tuning can improve the model's accuracy and performance. The accuracy of The EfficientNetV2M base model had an accuracy of 96.43%, after fine-tuning, it has increased by 2.14%, resulting in an accuracy of 98.57%. The ResNet50 base model had an accuracy of 98.57%, after fine-tuning, the model's performance increased by 0.72%, resulting in an accuracy of 99.29%. However, the accuracy decreases when the VGG16 base model is fine-tuned. The VGG16 base model had an accuracy of 94.29%; after fine-tuning, it became 89.29%. The performance of the VGG16 model decreased by -5%. Next, the experiment was performed without using a pretraining model. Experiments without a pre-trained model produced lower performance, with an accuracy of 87.14%.

Based on the results of the various experiments above, the ResNet50 model pre-trained with fine-tuning has the most optimal performance with a faster computation time than the other models pre-trained, namely 137,703 s. Implementing a suitable pre-trained model architecture and correct dataset can increase the accuracy of the resulting classification model. It is also known that the performance of the pre-trained model will be more optimal if fine-tuning is performed. It is important to adapt the previously trained model architecture to a new case or task, namely, classification of Indonesian banknotes. Adding a fine-tuning layer must be precise because not all pre-trained models are trained with the same dataset beforehand. This requires adjustment by changing the parameters. Meanwhile, if we do not use a pre-trained model, the performance will be lower because the model has not learned sufficient patterns with a small dataset. Therefore, transfer learning is required using a pre-trained model, which can improve model performance.

Table 2. The Scenario of Learning Rate Testing
Source: Developed by the authors

Pre-trained Model	Hyperparameter		
	Learning Rate (LR)	Batch Size	The number of Epochs
ResNet50	Without LR	20	15
ResNet50	0.001	20	15
ResNet50	0.0001	20	15
ResNet50	0.00001	20	15

Table 3. The Result of Learning Rate Testing
Source: Developed by the authors

Learning Rate	Accuracy	Precision	Recall	F1-Score	AUC	Time (second)
-	0.6071	0.7035	0.6071	0.6245	0.7967	163.2
0.001	0.7214	0.7699	0.7214	0.6822	0.9510	164.09
0.0001	**0.9929**	**0.9931**	**0.9929**	**0.9928**	**0.9999**	**137.7**
0.00001	0.9929	0.9931	0.9929	0.9928	0.9989	138.16

3.4. Testing and Analyzing the Performance of Learning Rate

After the optimal pre-training model is obtained, the next step is to test various learning rate values. When using the Adam optimizer, it is necessary to adjust the learning rate hyperparameter to achieve optimal performance. In this research, experiments with four variations were carried out, namely, without learning rate, using learning rates of 0.001, 0.0001, and 0.00001, which are presented in Table 2.

The results of the learning rate testing that has been carried out are presented in Table 3. The model performance results were measured using six measures: accuracy, precision, recall, f1-score, AUC, and computation time. The learning rate controls the extent to which a model learns from the training data. The learning rate is very influential on model performance, if you don't use the learning rate the ResNet50 model produces poor performance, namely only being able to produce accuracy values of 61.71%, precision of 70.35%, recall of 60.71%, f1-score 62.45%, and AUC 79.67% with a computation time of 163,195 seconds. Meanwhile, a learning rate of 0.001 produced better performance with a computation time difference of only 0.891 s, namely 164,086 s, which can produce accuracy values of 72.14%, precision 76.99%, recall 72.14%, f1-score 68.22%, and AUC 95.10%.

The learning rate test results of 0.0001 and 0.00001 had the same performance, namely producing accuracy values of 99.29%, precision 99.31%, recall 99.29%, f1-score 99.28%, and AUC 99.99%. Although it produces the same performance, the computation time required is different. A learning rate of 0.00001 requires a longer computation time, namely 138,162 s, or a difference of 1 s with a learning rate of 0.0001. Based on the learning rate test results described above, we can see that choosing the correct learning rate is crucial in model training. The more optimal the chosen learning rate value, the better the model's performance in predicting the test data. Moreover, if the learning rate is too low, the model learns slowly. Meanwhile, if it is too large, divergence can occur, and the model cannot converge to a good solution. Therefore, a learning rate of 0.0001 was chosen for application in building a classification system for Indonesian banknotes because this learning rate provides a good balance between performance and computation time.

3.5. Testing and Analyzing the Performance of Batch Size

This subsection discusses the batch size testing of deep transfer learning and continues with the analysis. The batch size in deep transfer learning is the number of data samples processed simultaneously in each model training iteration. An appropriate batch size is important because it can speed up training and prevent overfitting. This study tested various batch sizes to determine the batch size that produced the most optimal performance. Testing started with no batch size and then continued with batch sizes of 10, 20, and 30. Each configuration was analyzed to understand its impact on the system performance by focusing on factors such as accuracy, precision, recall, f1-score, AUC, and required computation time. It can be seen that the use of a certain batch size can increase the efficiency and effectiveness of training the Indonesian banknote classification model. The batch size test scenario is presented in Table 4.

In connection with the results of this batch size testing, the model performance results were also measured using six metrics, namely, accuracy, precision, recall, f1-score, AUC, and computation time,

Table 4. The Scenario of Batch Size Testing
Source: Developed by the authors

Pre-trained Model	Hyperparameter		
	Learning Rate (LR)	Batch Size	The number of Epochs
ResNet50	0.0001	-	15
ResNet50	0.0001	10	15
ResNet50	0.0001	20	15
ResNet50	0.0001	30	15

Table 5. The result of Batch Size Testing
Source: Developed by the authors

Batch Size	Accuracy	Precision	Recall	F1-Score	AUC	Time (second)
-	0.9786	0.9814	0.9786	0.9790	0.9999	163.2
10	0.9857	0.9870	0.9857	0.9859	0.9999	158
20	**0.9929**	**0.9931**	**0.9929**	**0.9928**	**0.9999**	**137**
30	0.9857	0.9870	0.9857	0.9858	0.9999	171.1

which are presented in Table 5. The batch size refers to the number of data samples processed in one iteration of a machine learning algorithm. In model training, data are divided into batches so that they can be processed efficiently using a computer. The batch size can vary depending on the requirements. Based on the test results in Table 5, using a smaller batch size allows the model to reach the optimal values more quickly than using a larger batch size. A large batch size was chosen because it can increase computing speed. This can be seen in tests using a batch size of 30, which requires a computation time of 171.1 seconds, while a small batch size of 10 requires a faster computation time of 158 s. Batch sizes 10 and 30 had the same performance, namely 97.86% accuracy, 98.70% precision, 98.57% recall, 97.90% f1-score, and 98.59% AUC.

However, when model training was carried out without using a batch size, the model performance decreased and only achieved an accuracy of 97.86%, precision of 98.14%, recall of 97.86%, f1-score of 97.90%, and AUC of 99.99%. If a batch size is not used, the computation time is not significantly different from testing using a batch size, that is it takes 163.2 seconds. The ideal batch size configuration used in this research was a batch size of 20. This produces a higher performance than batch sizes of 10 and 30. Batch size 20 obtained an accuracy value of 99.29%, precision 99.31%, recall 99.29%, f1-score 99.28%, and AUC 99.99%. Requires fast computation time, namely 137.7 seconds. These results also show that the more optimal the batch size, the higher is the model performance.

3.6. Analyzing the Performance of the Number of Epochs

This subsection explains the testing of the number of epochs on the model performance and continues with the analysis. An epoch is an iteration cycle carried out by the system to learn the training data. If the ability to understand the training data increases, the system's performance in predicting test data will also improve [6]. In this research, various variations of epochs were compared, namely, the number of epochs 5, 10, 15, and without using epochs whose test scenarios are presented in Table 6.

The results of testing the number of epochs were measured using the following measures: accuracy, precision, recall, f1-score, AUC, and computation time, as presented in Table 7. Based on the results in the table, the ideal epoch value with the best performance is 15 epochs. The average accuracy values achieved were 99.29%, precision 99.31%, recall 99.29%, f1-score 99.28%, and AUC 99.99%. Whereas. The computation time required was not significantly different from the other variations of epochs, namely 137,702 s. A difference of 16 s from epoch 10 required a computation time of 121,105 seconds. Epochs 10 produced lower performance, an accuracy 98.70%, precision 98.70%, recall 98.57%, f1-score 99.28%, and AUC 99.99%. Meanwhile, the lowest epoch value, namely 5, produced poor performance with an accuracy of 97.86%, precision of 98.04%, recall of 97.86, f1-score 97.84%, and AUC of 99.99% with a required computation time of 104,324 s. This is because the system was not sufficient for learning the training data. As the number of epochs increases, the model has more opportunities to learn from the training data and fit more complex patterns, which can ultimately improve the model performance.

3.7. Evaluating of Model Performance

Based on the results of the experiments carried out in the previous sub-section, optimal model performance results can be achieved for the pre-trained ResNet50

Table 6. The Scenario of the Number of Epochs Testing
Source: Developed by the authors

Pre-trained Model	Hyperparameter		
	Learning Rate (LR)	Batch Size	The number of Epochs
ResNet50	0.0001	20	Without Epochs
ResNet50	0.0001	20	5
ResNet50	0.0001	20	10
ResNet50	0.0001	20	15

Table 7. The result of the Number of Epochs Testing
Source: Developed by the authors

The number of Epochs	Accuracy	Precision	Recall	F1-Score	AUC	Time (second)
Without Epochs	0.3643	0.6379	0.3643	0.3397	0.8809	60.914
5	0.9786	0.9804	0.9786	0.9784	0.9999	104.324
10	0.9857	0.9870	0.9857	0.9858	0.9999	121.105
15	**0.9929**	**0.9931**	**0.9929**	**0.9928**	**0.9999**	**137.702**

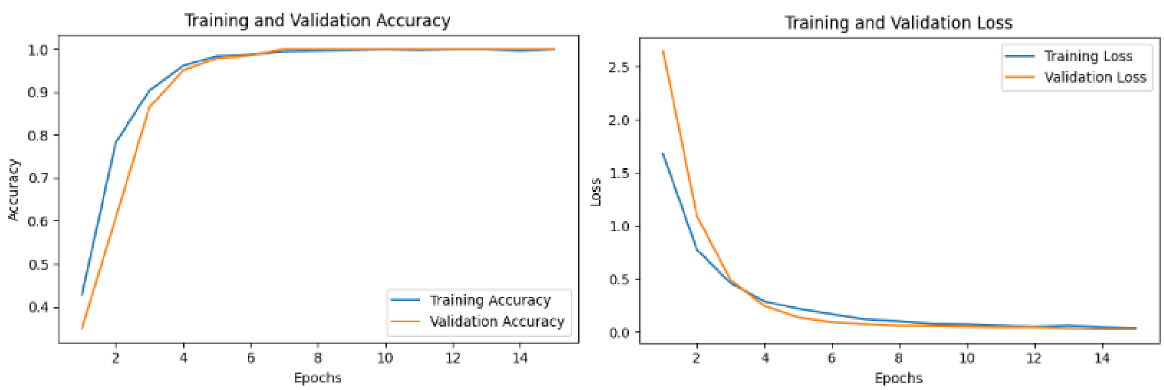

Figure 4. The Graph of (a). Accuracy and Validation, (b). Loss During Training
Source: Developed by the authors

model, with a learning rate of 0.0001, batch size 20, and number of epochs 15. The ResNet50 architecture adds additional layers in the form of dense batch normalization, and dropout with a value of 30%. Then, training with Adam optimization was carried out, resulting in a diagram of the evaluation results during training, as shown in Figure 4. (a).

Based on this graph, the accuracy and validation graph gradually increased consistently and approached or even reached the desired maximum value of 1. This indicated that the model performance was good. Moreover, the graph shows a consistent and steady upward trend, indicating that the model learned well from the training dataset. Additionally, the difference between the accuracy and validation scores decreased over time, indicating that the model not only remembered the training data but was also able to generalize well to new, never-before-seen data. This illustrates the reliability and effectiveness of the model for making accurate predictions. Furthermore, the graph shows a consistently improved performance, and both the accuracy and validation scores are close to the desired maximum values. This indicates that the model did not experience overfitting. Furthermore, to gain a more complete understanding of model performance, training and validation loss graphs are discussed, as shown in Figure 4. (b). The training and validation loss graphs depict the changes in loss values during the model training process. Both loss curves show a stable decreasing trend as the number of epochs increases, and the curves slowly approach 0. Then, there is a meeting point between the training and validation loss curves, where the difference between the two decreases with time. This indicates that the model can generalize well to new data that has never been seen before and shows that there is no overfitting.

The next step was to carry out test data testing on the 140 images. Testing was performed to measure how well the model could classify the images. The total number of test data was 140, with 20 images for each

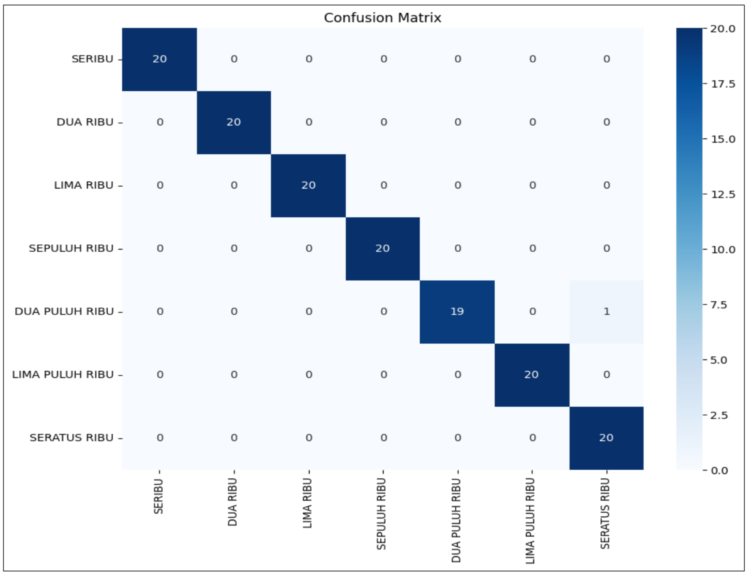

Figure 5. Confusion Matrix Classification Results
Source: Developed by the authors

Table 8. The result of the evaluation metrics for each Nominal Indonesian Banknote
Source: Developed by the authors

Nominal	Accuracy	Precision	Recall	F1-Score	AUC
SERIBU (IDR 1,000)	0.9929	1.0000	1.0000	1.0000	1.0000
DUA RIBU (IDR 2,000)	0.9929	1.0000	1.0000	1.0000	1.0000
LIMA RIBU (IDR 5,000)	0.9929	1.0000	1.0000	1.0000	1.0000
SEPULUH RIBU (IDR 10,000)	0.9929	1.0000	1.0000	1.0000	1.0000
DUA PULUH RIBU (IDR 20,000)	0.9929	1.0000	0.9500	0.9744	1.0000
LIMA PULUH RIBU (IDR 50,000)	0.9929	1.0000	1.0000	1.0000	1.0000

currency nominal class. Testing was performed to measure how well the model could classify images. The classification results display the confidence scores of the prediction results. The average confidence scores from the nominal predictions of SERIBU, DUA RIBU, LIMA RIBU, SEPULUH RIBU, DUA PULUH RIBU, LIMA PULUH RIBU, and SERATUS RIBU are 0.9932, 0.9944, 0.9703, 0.9939, 0.9760, 0.9945, and 0.9850, respectively. This indicates that the built model is effective at predicting Indonesian banknotes. Of the total 140 image data tested, there was one image that was mispredicted, namely a nominal value of IDR 20.000, which was incorrectly predicted at IDR 100.000. Other images can be correctly predicted under various image conditions, such as folded banknotes, low lighting, and shots from various angles.

Next, the classification results were visualized using a confusion matrix, as shown in Figure 5. After obtaining the confusion matrix, the evaluation metrics can be calculated, namely, accuracy, precision, recall, f1-score, and AUC. This calculation was performed for each nominal class of Indonesian currency. The above steps are carried out so that the evaluation metric values can be obtained for all currency classes. The evaluation metric results for each nominal banknote based on the results of the image data testing are presented in Table 8.

Based on this table, the results of the evaluation metrics when testing using test data show that the model can predict the image of rupiah banknotes for the 2022 emission year well. The model prediction results produced an average accuracy value of 99.29%, precision of 99.32%, recall of 99.29%, f1-score of 99.29%, and AUC 100%. The evaluation results demonstrate that the model performed exceptionally well. In detail, The average accuracy of the model was 99.29%, indicating that the model could accurately classify images of banknotes. The Precision, recall, and F1-score also showed nearly perfect performance, with an average precision of 99.32%, recall of 99.29%, and F1-score of 99.29%. This indicates that the model is accurate and consistent in recognizing different classes of banknotes. The AUC value for all classes was 1.0000, which indicates that the model has the perfect ability to distinguish between different classes. Further, the model was evaluated for each banknote denomination (1,000, 2,000, 5,000, 10,000, 20,000, 50,000, and 100,000). The evaluation results show that the model has an almost perfect performance for each class, although there is a slight decrease in recall for the 20.000 class (0.9500) and precision for the 100.000 class (0.9524). Overall, the model is highly effective in recognizing and classifying Indonesian banknote images with high accuracy and reliability. Thus, this model can help elderly and blind people identify Indonesian banknote denominations.

Furthermore, the implications of these findings are important for real-world applications such as financial automation systems and counterfeit money detection. By leveraging transfer learning, organizations can develop more effective solutions even with limited datasets. However, this study also highlights the need for experimentation with larger datasets to ensure the generalizability of the results.

4. DISCUSSION

This study adds new insights into banknote image classification using deep transfer learning, in line with previous research trends that emphasize the effectiveness of pre-trained models for domain-specific classification tasks. In comparison, Prajwal and Ilavarasi (2023) [32] showed that ResNet and its variants have demonstrated excellent results in numerous image-processing tasks, including classification and segmentation. However, this study extends these findings by showing that the fine-tuning performance can vary based on the specific dataset used.

In addition, our results regarding the influence of hyperparameters support those of Smith and Topin (2019), which highlights the importance of choosing the learning rate and batch size to achieve optimal accuracy [33]. In the context of our study, a lower learning rate allows for more stable learning, especially for very complex models, such as EfficientNetV2M. However, this study also adds that the optimal combination of hyperparameters can vary depending on the characteristics of the dataset, such as the lighting variations and shooting angles.

This study also faced common limitations in small datasets, such as overfitting, as highlighted by Goodfellow et al. (2016) [34]. However, the transfer learning implementation successfully overcomes this limitation by leveraging knowledge from large datasets, such as ImageNet. This allowed the model to learn more representative features, even though the training data used were relatively small. However, this study recognizes that the developed model still needs to be tested on more diverse datasets to improve its generalization and external validity.

5. CONCLUSION

This study demonstrates that, among several deep transfer learning architectures, the fine-tuned ResNet50 model with optimal hyperparameters yields the best performance in classifying Indonesian banknotes. The application of transfer learning significantly improves the accuracy and generalization, particularly when dealing with relatively small datasets. Specifically, the ResNet50 model, enhanced with a learning rate of 0.0001, batch size of 20, and 15 training epochs, achieved the highest classification accuracy of 99.29%. The addition of custom layers, such as dense, batch normalization, and dropout, further improves the model's ability to robustly distinguish between different banknote denominations.

This study makes a meaningful academic contribution by demonstrating how fine-tuned deep transfer learning, particularly using ResNet50, can effectively classify complex visual characteristics in Indonesian banknotes. The novelty lies in the structured analysis of hyperparameters, which has not been extensively explored in previous studies of local currency recognition. The findings not only enhance model performance but also enrich the literature on inclusive financial technology, particularly for visually impaired users.

Based on the results, it is recommended that future work focus on deploying the proposed model in real-time applications, such as mobile or embedded systems, to assist visually impaired individuals in financial transactions. Furthermore, future research could benefit from using more varied and real-world datasets, including worn or damaged banknotes, and exploring lightweight architectures suitable for resource-limited environments, such as IoT-based devices.

The outcomes of this study are expected to contribute to the development of intelligent recognition systems with broad implications for assistive technologies, automation, and financial security.

Declarations

Author Contributions

Conceptualization, Y.Y.; methodology, Y.Y. and A.Q.; data collection, B.A. and E.Y.; software, B.A. and R.E.; validation, Y.Y. and R.E.; formal analysis, Y.Y. and A.Q.; investigation, E.Y.; resources, B.A. and R.E.; data curation, B.A. and E.Y.; writing—original draft preparation, Y.Y. and B.A; writing—review and editing, Y.Y. and T.A.; visualization, B.A. and R.E..; supervision, Y.Y., and T.A.; project administration, Y.Y., and T.A.; funding acquisition, Y.Y., and T.A. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement

The data used in this study can be accessed by contacting the corresponding author upon request.

Funding

This research was funded by the Directorate of Research, Technology, and Community Service (Direktorat Riset, Teknologi, dan Pengabdian kepada Masyarakat, DRTPM) of the Indonesian Government under Fundamental Research Grant.

Acknowledgements

We would like to express our deepest thanks to all parties who have contributed to this research. Special thanks go to the 2020 Informatics Engineering students who contributed to this research.

Institutional Review Board Statement

Not Applicable

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this manuscript. In addition, ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies, have been completely observed by the authors.

References

[1] M. I. Lubis, "Online buying and selling transactions under international private law," *J. Priv. Commer. Law*, vol. 2, no. 1, 2018, doi: 10.15294/jpcl.v2i1.14499. https://doi.org/10.15294/jpcl.v2i1.14499

[2] Y. Chen, S. Wang, L. Lin, Z. Cui, and Y. Zong, "Computer Vision and Deep Learning Transforming Image Recognition and Beyond," *Int. J. Comput. Sci. Inf. Technol.*, vol. 2, no. 1, 2024, doi: 10.62051/ijcsit.v2n1.06. https://doi.org/10.62051/ijcsit.v2n1.06

[3] J. Chai, H. Zeng, A. Li, and E. W. T. Ngai, "Deep learning in computer vision: A critical review of emerging techniques and application scenarios," *Mach. Learn. with Appl.*, vol. 6, 2021, doi: 10.1016/j.mlwa.2021.100134. https://doi.org/10.1016/j.mlwa.2021.100134

[4] Golani Hema Pribhdas, "Deep Learning for Convolution Neural Networks and Recurrent Neural Networks: A Review," *Recent trends Manag. Commer.*, vol. 2, no. 4, 2021, doi: 10.46632/rmc/2/4/2. https://doi.org/10.46632/rmc/2/4/2

[5] J. M. Górriz *et al.*, "Computational approaches to Explainable Artificial Intelligence: Advances in theory, applications and trends," *Inf. Fusion*, vol. 100, p. 101945, Dec. 2023, doi: 10.1016/J.INFFUS.2023.101945. https://doi.org/10.1016/j.inffus.2023.101945

[6] S. F. Ahmed *et al.*, "Deep learning modelling techniques: current progress, applications, advantages, and challenges," *Artif. Intell. Rev.*, vol. 56, no. 11, 2023, doi: 10.1007/s10462-023-10466-8. https://doi.org/10.1007/s10462-023-10466-8

[7] U. Sadyk, C. Turan, and R. Baimukashev, "Overview of Deep Learning Models for Banknote Recognition," 2023, doi: 10.1109/ICECCO58239.2023.10147142. https://doi.org/10.1109/icecco58239.2023.10147142

[8] T. D. Pham, D. T. Nguyen, W. Kim, S. H. Park, and K. R. Park, "Deep learning-based banknote fitness classification using the reflection images by a visible-light one-dimensional line image sensor," *Sensors (Switzerland)*, vol. 18, no. 2, 2018, doi: 10.3390/s18020472. https://doi.org/10.3390/s18020472

[9] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J. Big Data 2021 81*, vol. 8, no. 1, pp. 1–74, Mar. 2021, doi: 10.1186/S40537-021-00444-8. https://doi.org/10.1186/s40537-021-00444-8

[10] R. Ramadhan, J. Y. Sari, and I. P. Ningrum, "Identification of Authenticity and Nominal Value of Indonesia Banknotes Using Fuzzy KNearest Neighbor Method," *IJNMT (International J. New Media Technol.*, vol. 6, no. 1, pp. 32–37, Aug. 2019, doi: 10.31937/ijnmt.v6i1.989. https://doi.org/10.31937/ijnmt.v6i1.989

[11] T. D. Pham, C. Park, D. T. Nguyen, G. Batchuluun, and K. R. Park, "Deep Learning-Based Fake-Banknote Detection for the Visually Impaired People Using Visible-Light Images Captured by Smartphone Cameras," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2984019. https://doi.org/10.1109/access.2020.2984019

[12] I. M. S. Kumara, G. P. R. S. Jati, and N. P. W. Yuniari, "Integrate Yolov8 Algorithm For Rupiah Denomination Detection In All-In-One Smart Cane For Visually Impaired," *Techno.Com*, vol. 23, no. 1, pp. 176–186, Feb. 2024, doi: 10.62411/tc.v23i1.9734. https://doi.org/10.62411/tc.v23i1.9734

[13] N. Mallesh *et al.*, "Knowledge transfer to enhance the performance of deep learning models for automated classification of B cell neoplasms," *Patterns*, vol. 2, no. 10, 2021, doi: 10.1016/j.patter.2021.100351. https://doi.org/10.1016/j.patter.2021.100351

[14] V. Meshram, K. Patil, and V. Meshram, "Evaluation of Top Pretrained Models Using Transfer Learning on Banknote Dataset with Quality Parameter," *Ing. des Syst. d'Information*, vol. 28, no. 3, 2023, doi: 10.18280/isi.280319. https://doi.org/10.18280/isi.280319

[15] A. H. M. Linkon, M. M. Labib, F. H. Bappy, S. Sarker, M. E. Jannat, and M. S. Islam, "Deep learning approach combining lightweight CNN architecture with transfer learning: An automatic approach for the detection and recognition of bangladeshi banknotes," 2020, doi: 10.1109/ICECE51571.2020.9393113. https://doi.org/10.18280/isi.280319

[16] A. Yildiz, A. A. Almisreb, Š. Dzakmic, and N. M. Tahir, "Banknotes Counterfeit Detection Using Deep Transfer Learning Approach," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 5, 2020, doi: 10.30534/ijatcse/2020/172952020. https://doi.org/10.30534/ijatcse/2020/172952020

[17] A. Khalil *et al.*, "Mobile Deep Classification of UAE Banknotes for the Visually Challenged," 2022, doi: 10.1109/FiCloud57274.2022.00053. https://doi.org/10.1109/ficloud57274.2022.00053

[18] M. B. Alejo, J. L. D. Villanueva, M. P. E. Garchitorena, S. C. Reyes, J. M. B. Delos Reyes, and Q. A. L. Marasigan, "Philippine Banknote Counterfeit Detection through Domain Adaptive Deep Learning Model of the Convolutional Neural Network," *Int. J. Comput. Digit. Syst.*, vol. 13, no. 1, 2023, doi: 10.12785/ijcds/130103. https://doi.org/10.12785/ijcds/130103

[19] T. Wagner and S. Sommer, "Feature Based Bearing Fault Detection With Phase Current Sensor Signals Under Different Operating Conditions," *PHM Soc. Eur.*

Conf., vol. 6, no. 1, 2021, doi: 10.36001/phme.2021.v6i1.2852. https://doi.org/10.36001/phme.2021.v6i1.2852

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-Decem, doi: 10.1109/CVPR.2016.90. https://doi.org/10.1109/cvpr.2016.90

[21] E. Rendón, R. Alejo, C. Castorena, F. J. Isidro-Ortega, and E. E. Granda-Gutiérrez, "Data sampling methods to dealwith the big data multi-class imbalance problem," *Appl. Sci.*, vol. 10, no. 4, 2020, doi: 10.3390/app10041276. https://doi.org/10.3390/app10041276

[22] A. K. Azlim Khan and N. H. Ahamed Hassain Malim, "Comparative Studies on Resampling Techniques in Machine Learning and Deep Learning Models for Drug-Target Interaction Prediction," *Molecules*, vol. 28, no. 4. 2023, doi: 10.3390/molecules28041663. https://doi.org/10.3390/molecules28041663

[23] S. Li, Q. Li, and M. Li, "A Method for Network Intrusion Detection Based on GAN-CNN-BiLSTM," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 5, 2023, doi: 10.14569/IJACSA.2023.0140554. https://doi.org/10.14569/ijacsa.2023.0140554

[24] T. Sulistyowati, P. Purwanto, F. Alzami, and R. A. Pramunendar, "VGG16 Deep Learning Architecture Using Imbalance Data Methods For The Detection Of Apple Leaf Diseases," *Monet. J. Keuang. dan Perbank.*, vol. 11, no. 1, 2023, doi: 10.32832/moneter.v11i1.57. https://doi.org/10.32832/moneter.v11i1.57

[25] T. M. Alam *et al.*, "An Efficient Deep Learning-Based Skin Cancer Classifier for an Imbalanced Dataset," *Diagnostics*, vol. 12, no. 9, 2022, doi: 10.3390/diagnostics12092115. https://doi.org/10.3390/diagnostics12092115

[26] M. O. Ojo and A. Zahid, "Improving Deep Learning Classifiers Performance via Preprocessing and Class Imbalance Approaches in a Plant Disease Detection Pipeline," *Agronomy*, vol. 13, no. 3, 2023, doi: 10.3390/agronomy13030887. https://doi.org/10.3390/agronomy13030887

[27] A. Y. Hussein, P. Falcarin, and A. T. Sadiq, "Enhancement performance of random forest algorithm via one hot encoding for IoT IDS," *Period. Eng. Nat. Sci.*, vol. 9, no. 3, 2021, doi: 10.21533/pen.v9i3.2204. https://doi.org/10.21533/pen.v9i3.2204

[28] B. Gu and Y. Sung, "Enhanced reinforcement learning method combining one-hot encoding-based vectors for cnn-based alternative high-level decisions," *Appl. Sci.*, vol. 11, no. 3, 2021, doi: 10.3390/app11031291. https://doi.org/10.3390/app11031291

[29] S. Bagui, D. Nandi, S. Bagui, and R. J. White, "Machine Learning and Deep Learning for Phishing Email Classification using One-Hot Encoding," *J. Comput. Sci.*, vol. 17, no. 7, 2021, doi: 10.3844/jcssp.2021.610.623. https://doi.org/10.3844/jcssp.2021.610.623

[30] M. Susanty, R. Hertadi, A. Purwarianti, and T. L. E. Rajab, "Low Complexity Classification of Thermophilic Protein using One Hot Encoding as Protein Representation," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 12, 2022, doi: 10.14569/IJACSA.2022.0131212. https://doi.org/10.14569/ijacsa.2022.0131212

[31] L. Yu, R. Zhou, R. Chen, and K. K. Lai, "Missing Data Preprocessing in Credit Classification: One-Hot Encoding or Imputation?," *Emerg. Mark. Financ. Trade*, vol. 58, no. 2, 2022, doi: 10.1080/1540496X.2020.1825935. https://doi.org/10.1080/1540496x.2020.1825935

[32] T. S. Prajwal and A. K. Ilavarasi, "A Comparative Study of RESNET-Pretrained Models for Computer Vision," *ACM Int. Conf. Proceeding Ser.*, pp. 419–425, Aug. 2023, doi: 10.1145/3607947.3608042. https://doi.org/10.1145/3607947.3608042

[33] L. N. Smith and N. Topin, "Super-convergence: very fast training of neural networks using large learning rates," 2019, doi: 10.1117/12.2520589. https://doi.org/10.1117/12.2520589

[34] J. Heaton, "Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning The MIT Press, 2016, 800 pp, ISBN: 0262035618," *Genet. Program. Evolvable Mach.*, vol. 19, no. 1–2, 2018.

.

參考

[1] M. I. Lubis, "國際私法下的網上買賣交易", J. Priv.商業。法律, 卷2, 沒有。 2018 年 1 月 1 日, doi：10.15294/jpcl.v2i1.14499。 https://doi.org/10.15294/jpcl.v2i1.14499

[2] Y. Chen、S. Wang、L. Lin、Z. Cui 和 Y. Zong, "計算機視覺和深度學習改變圖像識別及其他領域", Int. J. Comput.科學信息技術, 卷2, 沒有。 2024 年 1 月 1 日, doi：10.62051/ijcsit.v2n1.06。 https://doi.org/10.62051/ijcsit.v2n1.06

[3] J. Chai、H. Zeng、A. Li 和 E. W. T. Ngai, "計算機視覺中的深度學習：對新興技術和應用場景的批判性回顧", Mach。學習。與 Appl., 卷2021 年 6 月, DOI：10.1016/j.mlwa.2021.100134。 https://doi.org/10.1016/j.mlwa.2021.100134

[4] Golani Hema Pribhdas, "卷積神經網路和循環神經網路的深度學習：回顧", 近期趨勢管理。商業, 卷 2, 沒有。 2021 年 4 月 4 日, doi：10.46632/rmc/2/4/2。 https://doi.org/10.46632/rmc/2/4/2

[5] J. M. Górriz 等人, "可解釋人工智慧的計算方法：理論、應用和趨勢的進展", Inf。融合, 卷100, 頁 101945, 2023 年 12 月, doi：10.1016/J.INFFUS.2023.101945。 https://doi.org/10.1016/j.inffus.2023.101945

[6] S. F. Ahmed 等人, "深度學習建模技術：當前進展、應用、優勢和挑戰", Artif。智力。修訂版, 卷56, 無。 2023 年 11 月 11 日, doi：10.1007/s10462-023-10466-8。 https://doi.org/10.1007/s10462-023-10466-8

[7] U. Sadyk、C. Turan 和 R. Baimukashev, "紙幣識別深度學習模型概述", 2023 年, doi：10.1109/ICECCO58239.2023.10147142。 https://doi.org/10.1109/icecco58239.2023.10147142

[8] T. D. Pham、D. T. Nguyen、W. Kim、S. H. Park 和 K. R. Park, "使用可見光一維線圖像感測器的反射圖像進行基於深度學習的紙幣適合度分類", 《感測器》

（瑞士），卷。 18，沒有。 2, 2018, doi: 10.3390/s18020472。 https://doi.org/10.3390/s18020472
[9] L. Alzubaidi 等人，"深度學習回顧：概念、CNN 架構、挑戰、應用、未來方向"，J. Big Data 2021 81，vol. 8，沒有。 1，第 1-74 頁，2021 年 3 月，doi：10.1186/S40537-021-00444-8。 https://doi.org/10.1186/s40537-021-00444-8
[10] R. Ramadhan、J. Y. Sari 和 I. P. Ningrum，"使用模糊最近鄰方法識別印度尼西亞紙幣的真實性和麵值"，IJNMT（國際新媒體技術雜誌），第 6 卷，第 1 期，第 32-37 頁，2019 年 8 月，doi：10.31937/ijnv.37/ijn。 https://doi.org/10.31937/ijnmt.v6i1.989
[11] T. D. Pham、C. Park、D. T. Nguyen、G. Batchuluun 和 K. R. Park，"基於深度學習、使用智慧型手機相機捕捉的可見光影像為視障人士進行假鈔檢測"，IEEE Access，卷。 8，2020，doi：10.1109/ACCESS.2020.2984019。 https://doi.org/10.1109/access.2020.2984019
[12] I. M. S. Kumara、G. P. R. S. Jati 和 N. P. W. Yuniari，"在視障人士一體式智能手杖中集成 Yolov8 算法進行印尼盾面額檢測"，Techno.Com，卷。 23，沒有。 1，第 176-186 頁，2024 年 2 月，doi：10.62411/tc.v23i1.9734。 https://doi.org/10.62411/tc.v23i1.9734
[13] N. Mallesh 等人，"知識轉移以增強深度學習模型對 B 細胞腫瘤自動分類的性能"，Patterns，vol. 2，沒有。 2021 年 10 月 10 日，doi：10.1016/j.patter.2021.100351。 https://doi.org/10.1016/j.patter.2021.100351
[14] V. Meshram、K. Patil 和 V. Meshram，"使用遷移學習對具有質量參數的紙幣數據集進行頂級預訓練模型評估"，Ing.系統設計d'Information，卷28，沒有。 2023 年 3 月 3 日，doi：10.18280/isi.280319。 https://doi.org/10.18280/isi.280319
[15] A.H. M. Linkon、M. M. Labib、F. H. Bappy、S. Sarker、M. E. Jannat 和 M. S. Islam，"將輕量級 CNN 架構與遷移學習相結合的深度學習方法：一種自動檢測和識別孟加拉紙幣的方法"，2020 年，doi：10.1091315135135135135135131313135135135313931313531393135333333時，https://doi.org/10.18280/isi.280319
[16] A. Yildiz，A. A. Almisreb，Š。 Dzakmic 和 N. M. Tahir，"使用深度遷移學習方法檢測紙幣偽造"，Int. J. Adv.趨勢計算。科學英語，卷9，不。 2020 年 12 月 5 日，doi：10.30534/ijatcse/2020/172952020。 https://doi.org/10.30534/ijatcse/2020/172952020
[17] A. Khalil 等人，"面向視障人士的阿聯酋紙幣移動深度分類"，2022 年，doi：10.1109/FiCloud57274.2022.00053。 https://doi.org/10.1109/ficloud57274.2022.00053
[18] M. B. Alejo、J. L. D. Villanueva、M. P. E. Garchitorena、S. C. Reyes、J. M. B. Delos Reyes 和 Q. A. L. Marasigan，"通過卷積神經網絡的域自適應深度學習模型進行菲律賓鈔票防偽檢測"，Int. J. Comput.數字。系統，卷13，沒有。 2023 年 1 月 1 日，doi：10.12785/ijcds/130103。 https://doi.org/10.12785/ijcds/130103
[19] T. Wagner 和 S. Sommer，"基於特徵的不同運行條件下的相電流感測器訊號軸承故障檢測"，PHM Soc. 歐元Conf.，卷6，沒有。 2021 年 1 月 1 日，DOI：10.36001/phme.2021.v6i1.2852。 https://doi.org/10.36001/phme.2021.v6i1.2852
[20] K. He、X. Zhang、S. Ren 和 J. Sun，"深度殘差學習用於圖像識別"，載於《IEEE 計算機學會計算機視覺與模式識別會議論文集》，2016 年，第 15 卷。 2016 年 12 月，doi：10.1109/CVPR.2016.90。 https://doi.org/10.1109/cvpr.2016.90
[21] E. Rendón、R. Alejo、C. Castorena、F. J. Isidro-Ortega 和 E. E. Granda-Gutiérrez，"處理大數據多類不平衡問題的數據採樣方法"，Appl。科學，卷10，不。 4，2020，doi：10.3390 / app10041276。 https://doi.org/10.3390/app10041276
[22] A.K. Azlim Khan 和 N.H. Ahamed Hassain Malim，"機器學習和深度學習模型中重採樣技術在藥物-靶標相互作用預測中的比較研究"，Molecules，vol. 28，沒有。 4. 2023 年，doi：10.3390/分子28041663。 https://doi.org/10.3390/molecules28041663
[23] S. Li、Q. Li 和 M. Li，"一種基於 GAN-CNN-BiLSTM 的網路入侵檢測方法"，Int. J. Adv.計算。科學應用，卷14，沒有。 2023 年 5 月 5 日，doi：10.14569/IJACSA.2023.0140554。 https://doi.org/10.14569/ijacsa.2023.0140554
[24] T. Sulistyowati、P. Purwanto、F. Alzami 和 R. A. Pramunendar，"使用不平衡數據方法檢測蘋果葉片疾病的 VGG16 深度學習架構"，Monet。 J. Keuang。 dan Perbank.，卷11，無。 2023 年 1 月 1 日，doi：10.32832/moneter.v11i1.57。 https://doi.org/10.32832/moneter.v11i1.57
[25] T. M. Alam 等人，"基於深度學習的不平衡數據集皮膚癌分類器"，《診斷學》，第。 12，沒有。 9，2022，doi：10.3390 / diagnostics12092115。 https://doi.org/10.3390/diagnostics12092115
[26] M. O. Ojo 和 A. Zahid，"通過預處理和類別不平衡方法在植物病害檢測流程中提高深度學習分類器的性能"，《農學》，卷。 13，沒有。 3，2023，doi：10.3390 / agronomy13030887。 https://doi.org/10.3390/agronomy13030887
[27] A.Y. Hussein、P. Falcarin 和 A.T. Sadiq，"透過獨熱編碼增強物聯網入侵檢測系統中隨機森林演算法的性能"，期刊。英語納特。科學，卷9，不。 2021 年 3 月 3 日，doi：10.21533/pen.v9i3.2204。 https://doi.org/10.21533/pen.v9i3.2204
[28] B. Gu 和 Y. Sung，"結合基於獨熱編碼的向量的增強強化學習方法，用於基於 cnn 的替代高級決策"，應用。科學，卷11，沒有。 3，2021，doi：10.3390 / app11031291。 https://doi.org/10.3390/app11031291
[29] S. Bagui、D. Nandi、S. Bagui 和 R. J. White，"使用獨熱編碼進行網絡釣魚電子郵件分類的機器學習和深度學習"，J. Comput.科學，卷17，沒有。 2021 年 7 月，doi：10.3844/jcssp.2021.610.623。 https://doi.org/10.3844/jcssp.2021.610.623
[30] M. Susanty、R. Hertadi、A. Purwarianti 和 T. L. E. Rajab，"使用獨熱編碼作為蛋白質表示對嗜熱蛋白

進行低複雜度分類”, Int. J. Adv.計算。科學應用, 卷13, 沒有。 2022 年 12 月 12 日, doi : 10.14569/IJACSA.2022.0131212。 https://doi.org/10.14569/ijacsa.2022.0131212

[31] L. Yu, R. Zhou, R. Chen 和 K. K. Lai, 「信用分類中的缺失資料預處理:獨熱編碼還是插補?」Emerg.標記。金融。貿易, 卷58, 沒有。 2, 2022 年, doi : 10.1080/1540496X.2020.1825935。 https://doi.org/10.1080/1540496x.2020.1825935

[32] T.S. Prajwal 和 A.K. Ilavarasi, “計算機視覺 RESNET 預訓練模型的比較研究”, ACM Int.會議。Proceeding Ser., 第 419–425 頁, 2023 年 8 月, doi : 10.1145/3607947.3608042。 https://doi.org/10.1145/3607947.3608042

[33] L. N. Smith 和 N. Topin, “超收斂：使用較大學習率快速訓練神經網路”, 2019 年, doi : 10.1117/12.2520589。 https://doi.org/10.1117/12.2520589

[34] J. Heaton, “Ian Goodfellow、Yoshua Bengio 和 Aaron Courville：深度學習”, 麻省理工學院出版社, 2016 年, 800 頁, ISBN：0262035618”, Genet。程序。可進化機器, 第卷19, 沒有。 1–2, 2018年。

Appendix A

The appendix is an optional section that can contain details and data supplemental to the main text, for example, explanations of experimental details that would disrupt the flow of the main text, but nonetheless remain crucial to understanding and reproducing the research shown; figures of replicates for experiments of which representative data is shown in the main text can be added here if brief, or as supplementary data. Mathematical proofs of the results not central to the paper can be added as an appendix.

Word count: 7,199 words, excluding references.

Peer-Review Record
• Fast-track status: Not fast-tracked
• First-round reviews received: 3 reports
• Revision cycles completed: 3 rounds
• Final version submitted: 8 June 2025