RMSProp Optimizer and KAN Method-Based CNN on Rupiah Banknote Classification for Visually Impaired

Manuscript received April 13, 2025; revised May 11, 2025

Dede Kurniadi*
Department of Computer Science
Institut Teknologi Garut
Garut, Indonesia
dede.kurniadi@itg.ac.id

Murni Lestari Rahmi
Department of Computer Science
Institut Teknologi Garut
line 4:Garut, Indonesia
2106035@itg.ac.id

Benedicto B. Balilo Jr
CS/IT Department, College of Science
Bicol University
Legazpi, Philippines
bjbbalilo@bicol-u.edu.ph

Hilmi Aulawi
Department of Industrial Engineering
Institut Teknologi Garut
Garut, Indonesia
hilmi_aulawi@itg.ac.id

Abstract*— The visually impaired refers to individuals who experience a loss of visual function. Approximately 4 million people, or about 1.5% of Indonesia's total population, are visually impaired. They rely on their sense of touch to recognize banknote denominations in financial transactions. However, damaged banknotes often hinder identification and increase the risk of fraud. Therefore, this study aims to develop a rupiah banknote denomination classification model to assist them in conducting independent transactions. The researchers developed a CNN-KAN model with the RMSProp optimizer using a private dataset comprising 800 images of Rupiah banknotes with denominations of IDR 1,000, IDR 2,000, IDR 5,000, IDR 10,000, IDR 20,000, IDR 50,000, IDR 75,000, and IDR 100,000 from the 2016, 2020, and 2022 emission years. The dataset encompasses variations in image perspectives, lighting conditions, and the physical state of banknotes, including both intact and damaged ones, with up to 30% of the samples comprising damaged banknotes. Data augmentation techniques were implemented to improve data diversity. The dataset was then utilized for training and testing with different split ratios: 50:50, 60:40, 70:30, 80:20, and 90:10. Performance evaluation was conducted using loss, accuracy, precision, recall, and AUC-ROC metrics. Experimental results indicate that the CNN-KAN model with the RMSProp optimizer achieved optimal performance. In the 90:10 data split scenario, the model achieved 100% accuracy, precision, and recall, with an AUC-ROC of 1 and a loss of 0.008. Therefore, the CNN-KAN model with the RMSProp optimizer has been proven effective for implementing Rupiah banknote denomination detection for the visually impaired in an automated system.

Keywords*— Classification, Convolutional Neural Network, KAN, RMSProp, Rupiah Banknotes.

I. INTRODUCTION

According to the Ministry of Health of the Republic of Indonesia, approximately 4 million people, or 1.5% of Indonesia's total population, are visually impaired [1]. Visual impairment refers to a condition characterized by dysfunction or loss of visual sensory function [2]. In their daily activities, visually impaired individuals rely on their other senses, such as touch. This includes recognizing banknote denominations, where they differentiate them by feeling the double lines on the right and left sides of the banknotes. However, visually impaired individuals still face challenges in identifying banknote denominations when the money is damaged, dirty, or poorly maintained [3]. This difficulty arises due to wear and tear on banknotes, which results from continuous circulation from one hand to another, as well as dirt that reduces tactile sensitivity. These challenges often lead to errors in financial transactions and even contribute to fraud cases. To assist visually impaired individuals in performing financial transactions in their daily lives, a system capable of identifying banknote denominations and providing voice output is proposed as a solution [4].

Several previous studies have been conducted on banknote denomination classification. The first study by Pratama et al. developed a classification system using the K-Nearest Neighbor (K-NN) algorithm to recognize IDR 2,000 and IDR 5,000 banknotes. Using a dataset of 40 images, the system achieved an accuracy of 93.7% [5]. Hermawan et al. designed a banknote denomination detection application using the YOLOv3 algorithm in the next study. With a dataset of 4,200 images, the system achieved 98% accuracy under optimal lighting conditions [4]. Prima et al. implemented a Single Shot Multibox Detector (SSD) with the MobileNetV3 architecture for banknote denomination detection. The dataset consisted of 7,035 images of 2016 and 2022 edition banknotes, and the model achieved an accuracy ranging from 80% to 95% under adequate lighting conditions [6]. Hamidah et al. developed a banknote denomination detection model based on Optical Character Recognition (OCR). The model attained an accuracy of 94%, using a dataset comprising 70 images of the 2016 edition banknotes [7]. Aprillia et al. compared Feedforward Neural Networks (FNN) and Convolutional Neural Networks (CNN) in the context of a rupiah banknote detection system tailored for vending machines. The dataset consisted of 1,120 images of the 2022 edition rupiah banknotes. The study demonstrated that the CNN model outperformed the FNN model, achieving 100% accuracy under bright lighting conditions, 96.43% under moderate lighting conditions, and 78.56% under low lighting conditions. In contrast, the FNN model consistently achieved only 53.57% accuracy across all lighting conditions [8].

Previous studies have implemented CNN-Kolmogorov Arnold Network (CNN-KAN) in image classification

 EISSN: 2961-872X ● PISSN: 2961-8924 ● DOI: 10.56741/IISTR.esl.00936

applications. One such study was conducted by Cheon, who applied CNN-KAN to classify satellite images in remote sensing. Using the EuroSAT dataset, which consists of 27,000 images across 10 different classes, the model achieved a maximum accuracy of 96% [9]. Another study by Igali & Shamoi utilized CNN-KAN with Fuzzy Pooling integration to enhance model accuracy and interpretability in classifying three different image datasets: CIFAR-10, MNIST, and FashionMNIST. Their approach yielded an accuracy of 67.06% on CIFAR-10, 98.91% on MNIST, and 89.88% on FashionMNIST [10].

Based on previous studies, there are gaps in dataset coverage and model accuracy. Prior research primarily utilized datasets containing images of banknotes under ideal lighting conditions, leading to a decline in model performance under dim lighting. Additionally, limited camera angles resulted in models that lacked robustness to variations in perspective. Furthermore, these studies only included certain banknote denominations and emissions, and the datasets consisted solely of well-preserved banknotes. Consequently, the models were not tested on worn-out, folded, dirty banknotes that naturally degrade due to everyday use. This study's novelty stems from the utilization of a comprehensive private dataset encompassing both intact and damaged rupiah banknotes across various denominations of IDR 1,000, IDR 2,000, IDR 5,000, IDR 10,000, IDR 20,000, IDR 50,000, IDR 75,000, and IDR 100,000 issued in 2016, 2020, and 2022. This dataset was collected under various bright and dim lighting conditions and from diverse camera angles to enhance model robustness. This study is the first to apply CNN-KAN to classify Rupiah banknote denominations. This model has not been previously used for this specific task, making it a novel approach expected to improve model accuracy and generalization. This study is a continuation of previous research published at the International Conference on ICT for Smart Society (ICISS) 2024 [11]. which evaluated the performance of a CNN AlexNet-VGGNet model in classifying Rupiah banknote denominations using the same dataset with the RMSProp optimizer. In that study, the model achieved a loss function of 1.28, 95.31% accuracy, 95.24% precision, 93.75%, and an AUC-ROC of 0.99. Therefore, this research aims to determine whether the CNN-KAN algorithm can deliver superior performance compared to CNN AlexNet-VGGNet when applied to the same dataset and optimizer.

II. Methodology

The workflow of this study follows the Machine Learning Life Cycle (MLLC) approach, beginning with data collection, data pre-processing, modeling, model evaluation, and result representation. Fig. 1 illustrates the research workflow.

The first stage of this research is data collection, which involves gathering the necessary image data to train the model, labeling, and manual augmentation. The image data is captured using a smartphone and labeled by classifying the images into eight categories based on their nominal values. Additionally, manual augmentation is performed by varying angles and lighting conditions to enhance dataset diversity.

The second stage is data pre-processing, which involves refining the collected data to improve its quality before training the model. This process includes automatic augmentation and data splitting. Automated augmentation techniques, including rotation, translation (shift), shear, and zoom, diversify the dataset and improve the model's capability in identifying various banknote conditions. The dataset is subsequently split into training and testing subsets, with five different ratio configurations examined to identify the most effective setup.

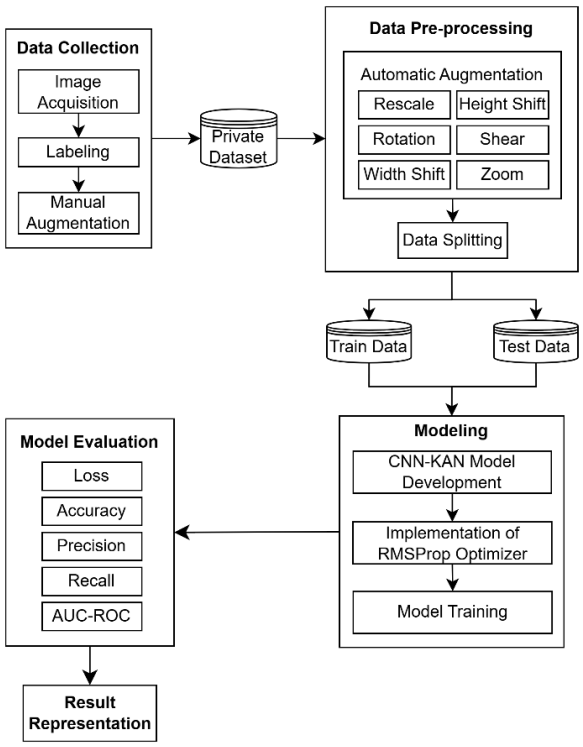


Fig. 1. Research workflow

The third stage is modeling, where the CNN-KAN model is developed and trained. CNN-KAN is a combination of traditional CNN architecture with the principles of the Kolmogorov-Arnold network, which emphasizes learning complex nonlinear functions through compositional layers. This approach replaces the conventional fully connected layers with spline-based functions that can be learned dynamically, thereby enhancing the model's ability to capture nonlinear patterns in visual data [12]. The CNN-KAN architecture consists of several multi-kernel convolutional blocks (1×1, 3×3, and 5×5), which are combined and normalized, followed by a pooling layer for spatial feature extraction. Subsequently, dense layers with SiLU activation are used to form complex nonlinear functions before classification into eight nominal classes using a fully connected layer with a softmax activation. For the training process, the RMSProp optimizer is used with standard parameter settings, namely learning rate=0.001, rho=0.9, and epsilon=1e-07. This optimizer is selected due to its ability to adaptively adjust the learning rate, allowing the model to address issues with large or small gradients that could hinder the training process. During this phase, the model learns to recognize visual patterns that distinguish each banknote denomination, such as differences in size, color, and other visual characteristics. The training process also involves testing various hyperparameters to find the best configuration. The main goal of this study is to guarantee that the model can accurately identify the banknote denominations based on their visual features.

Once the model is trained, the fourth stage is model evaluation, using testing data that has never been introduced during training. The evaluation is conducted by calculating various metrics, including accuracy, precision, recall, AUC-ROC, and loss, to measure the performance of each model. This stage assesses the model's ability to identify banknote denominations on previously unseen data. This process

 EISSN: 2961-872X ● PISSN: 2961-8924 ● DOI: 10.56741/IISTR.esl.00936

ensures that the model functions effectively when implemented in real-world conditions.

The final stage is result representation, where the performance evaluation of CNN-KAN is presented based on accuracy, precision, recall, AUC-ROC, and loss values. This result analysis aims to assess the model's reliability in detecting the denomination of Rupiah banknotes for the visually impaired.

III. Results and Discussion

A. Data Collection

The data collection process resulted in a private dataset comprising images of Indonesian rupiah banknotes with denominations of IDR 1,000, IDR 2,000, IDR 5,000, IDR 10,000, IDR 20,000, IDR 50,000, IDR 75,000, and IDR 100,000 from the 2016, 2020, and 2022 emission years. The constructed dataset consists of 96 images. Six images were captured under optimal lighting conditions using natural daylight outdoors during the daytime for each banknote denomination from each emission year. In comparison, another six images were taken indoors with minimal illumination under low-light conditions. Consequently, each banknote denomination is represented by 24 images, except for the IDR 75,000 denomination, which was issued in only one emission year and contains 12 images. All images were labeled according to their denominations, forming eight banknote classes. Manual data augmentation was performed using the Augmentor library in Python to enhance the dataset's diversity. This process involved applying transformation techniques such as rotation, brightness adjustment, zooming, and random distortion. Consequently, each class produced 100 augmented images, resulting in a total dataset of 800, as shown in Fig. 2.

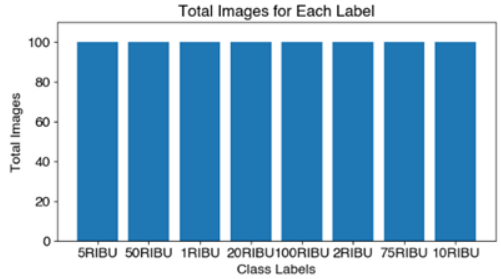
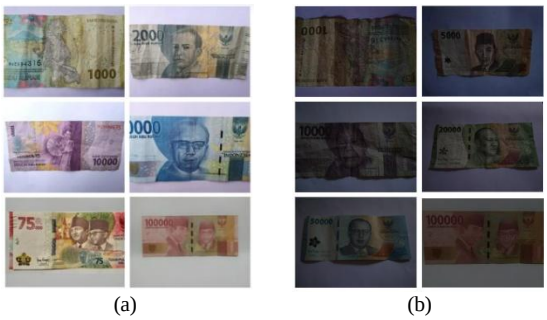


Fig. 2. Class distribution of the dataset

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.



(a) (b)

Fig. 3. Sample dataset, (a) adequate lighting, (b) dim lighting

Fig. 3 presents sample images from the dataset obtained during the data collection phase of this study. Each image represents a banknote of the Indonesian rupiah with various denominations, years of emission, physical conditions, viewing angles, and lighting variations. Specifically, Fig. 3(a) displays sample images captured under adequate lighting conditions, while Fig. 3(b) shows samples taken under low-light conditions.

B. Data Pre-processing

This stage takes place after the completion of the data collection process. This study involves two primary steps: automatic augmentation and data splitting for training and testing. Automatic data augmentation aims to enrich data variation by generating new images through various transformations, enabling the model to recognize objects under diverse conditions better. This augmentation process is performed using the *ImageDataGenerator* from the Keras library. The applied transformations include image reaching 20 degrees, shifts in both horizontal and vertical directions covering 20% of the image size, shear transformation reaching 20%, and zooming by a maximum of 20%. Moreover, transformations may produce empty pixels filled using the *nearest* method, which replaces empty areas with the nearest surrounding pixel values. All images are also normalized by rescaling pixel values from 0 to 1 to match the model's requirements.

After the automatic augmentation process separates the dataset into training and testing sets. This division is carried out in stages by exploring five proportion scenarios—50:50, 60:40, 70:30, 80:20, and 90:10 to analyze the impact of the training-to-testing data ratio on model performance. The dataset splitting process utilizes *ImageDataGenerator* with the *validation_split* parameter, where a portion of the data is allocated as testing data, while the remainder is used for training. The augmented data is then loaded using the *flow_from_directory* method, which allows images to be accessed directly from a predefined folder structure. At this stage, all images are resized to 200×200 pixels. The data is processed in 32 images per batch to optimize computational efficiency. The seed parameter is also set to 123 to maintain consistency in random processes during augmentation and data splitting.

C. Modeling

Convolutional Neural Network (CNN) is a sophisticated deep learning framework that has been extensively utilized in digital image processing and image recognition tasks [13]. This algorithm is uniquely designed to handle visual data, particularly images and their pixel compositions. CNN's ability to interpret visual information is claimed to resemble the way the human visual cortex recognizes objects [14]. The visual cortex itself plays a role in processing visual stimuli received by the eyes, making CNN the most superior method compared to other neural network types for solving image-based problems [15].

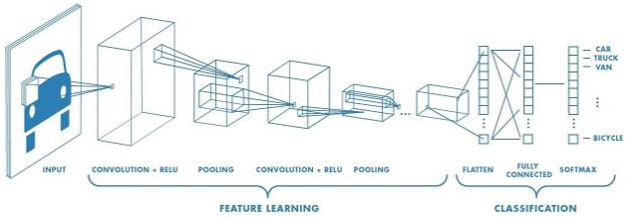


Fig. 4. Illustration of the CNN mechanism [8]

CNN is classified as a deep learning architecture known for its effectiveness in performing image classification and identification, as it efficiently extracts spatial features from

input images [8]. Fig. 4 illustrates the working mechanism of CNN. As presented in Fig. 4, the CNN architecture generally consists of two main stages, each serving a specific function. The first stage is *feature learning*, which involves a series of layers locally connected to specific regions of the previous layer. The first layer in this stage is the *convolutional layer*, which extracts crucial features from the image. The pooling layer subsequently refined these extracted features, which minimizes data dimensionality while preserving key features. These two layers are arranged alternately, with activation functions applied in between. This feature-learning process transforms the input image into a feature vector, which is then passed to the next stage [16]. The second stage, *classification*, categorizes the extracted features using several fully connected layers, beginning with a *flatten layer*. These layers process the feature vector obtained from the previous stage. A series of hidden layers transform the vector into the final output as class probabilities as the classification result [17].

To enhance the feature mapping capability of CNN, the Kolmogorov-Arnold Network (KAN) is a viable alternative method for implementation. KAN differs from Multi-Layer Perceptrons (MLPs) by utilizing learnable activation functions at each edge (weight), rather than fixed activation functions for each neuron, replacing linear weights with univariate spline-based functions [18]. One of the key characteristics of KAN is using the SiLU (Sigmoid Linear Unit) activation function, which is nonlinear and can improve the model's performance in capturing complex relationships within the data.

The researchers designed the model architecture by integrating CNN and KAN approaches. Feature extraction is performed using CNN through convolutional and pooling layers, after which the extracted features are processed by KAN to map complex nonlinear relationships before producing the final classification output. This approach combines CNN's strength in extracting visual features with KAN's capability to represent flexible mathematical patterns, enhancing performance.

The CNN-KAN model architecture employed in this study utilizes a Functional approach, enabling the formation of a branched architecture. In the first stage, known as feature learning, feature extraction is conducted using a CNN approach, which is modified through the application of the Kolmogorov-Arnold Block (KAB). Each KAB comprises three convolutional layers that run in parallel, each utilizing distinct kernel sizes of 1×1, 3×3, and 5×5, enabling the model to extract feature patterns across multiple spatial scales, ranging from small to large. Each convolution output is then processed through the ReLU activation function, which converts negative values to zero while preserving positive values. This function maps input values into a range spanning $(0, \infty)$ [11], as defined by the following equation:

$$y = ReLU(x) = max(0, x) \tag{1}$$

In Equation (1), the expression $y = ReLU$ denotes the result of applying the ReLU activation function to the input x, where the function returns 0 if $x < 0$ and retains the value of x itself when $x \geq 0$.

The outputs from the three convolutional pathways are then normalized using batch normalization to accelerate the training process and enhance stability. These three convolutional outputs are summed through an Add layer using element-wise addition, integrating feature information from multiple scales. Next, a 2×2 max pooling layer is applied to shrink the spatial dimensions while maintaining the crucial extracted features. The KAN blocks are applied sequentially, with increasing filters at each stage. The initial block utilizes 32 filters, the second block increases to 64 filters, and the third block further expands to 128 filters. After extracting features, the flattened layer transforms the last feature representation into a single-dimensional vector, which is mathematically expressed as:Type equation here.

$$z_{flatten} = Flatten(z_{pooling}) \tag{2}$$

In Equation (2), $z_{flatten}$ denotes the result of the flattening process, which converts the previous pooling layer's output from a tensor format into a one-dimensional vector.

The resulting feature vector is then processed in the classification stage, which employs two KANLinear layers. These layers function as dense layers utilizing the SiLU activation function, also known as the Swish Function [19], which is defined by the equation:

$$SiLU(x) = x \cdot \sigma(x) \tag{3}$$

In Equation (3), $\sigma(x)$ represents the logistic sigmoid function. The first KANLinear layer consists of 128 neuron units, followed by the second layer with 64 neuron units.

Finally, the output layer employs the softmax activation function with eight neurons to map the probability distribution of each rupiah banknote denomination class, as expressed in the following equation:

$$y_{output} = SoftMax\left(W'_{y_{dense}} + b'\right) \tag{4}$$

In Equation (4), y_{output} represents the probability of each class, with W' and b' denoting the weights and biases in the output layer, respectively.

The CNN-KAN model is optimized using the RMSProp algorithm in the second process. By adjusting the learning rate adaptively based on the moving average of squared gradients [20], as defined in (5) and (6):

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{v_t + \epsilon}} \frac{dL}{dw_t} \tag{5}$$

where

$$v_t = \beta v_{t+1} + (1 - \beta) \left[\frac{dL}{dw_t}\right] 2 \tag{6}$$

In Equation (5), w_{t+1} represents the updated weight, while v_t Equation (6) combines the previous and current gradients with a decay factor β. The initial value of v is set to 0 before the weight update process begins. This technique ensures more stable and efficient parameter updates.

In the third process, which involves training the model, the CNN-KAN model is trained with a batch size of 8 and a total of 50 epochs. The training continues until the model converges or completes the maximum epochs.

D. Model Evaluation

At this stage, a series of processes are conducted to evaluate and measure the model's performance. To ensure that the performance comparison between the CNN AlexNet-VGGNet model and the CNN-KAN model remains objective and comparable, this study employs the same evaluation metrics used in previous research, including accuracy, precision, recall, AUC-ROC, and loss function [11]. All these metrics are computed based on the Confusion Matrix, which compares the model's predicted outcomes with the actual labels.

 EISSN: 2961-872X ● PISSN: 2961-8924 ● DOI: 10.56741/IISTR.esl.00936

The Confusion Matrix is composed of four key elements: True Positive (TP) which denotes correctly classified positive class samples, True Negative (TN) referring to correctly identified negative class samples, False Positive (FP) indicating negative class samples that were incorrectly predicted as positive, and False Negative (FN) representing positive class samples that were mistakenly classified as negative.

The accuracy metric serves as a metric to evaluate the ratio of correctly classified instances to the total number of predictions made, as expressed in (7).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

Meanwhile, precision indicates the fraction of correctly identified positive cases relative to all instances classified as positive, as defined in (8).

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

The next metric, recall, also known as sensitivity, assesses the model's effectiveness in identifying all positive instances, formulated in (9).

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

Besides these metrics, classification performance is further evaluated using the Area Under the Curve-Receiver Operating Characteristic (AUC-ROC), which evaluates how well the model distinguishes between positive and negative classes. The AUC-ROC score spans from 0 to 1, with values approaching 1 indicating superior classification accuracy. This score is derived from the area under the ROC curve, as expressed in (10).

$$AUC\text{-}ROC = \int_0^1 TPR\left(FPR^{-1}(t)\right) dt \quad (10)$$

where the True Positive Rate (TPR) corresponds to recall, False Positive Rate (FPR) indicates the proportion of incorrectly classified negative samples, and $FPR^{-1}(t)$ is the inverse function of FPR concerning the threshold t.

Beyond metric-based performance evaluation, the model training process also incorporates a loss function to measure the discrepancy between predicted results and actual labels. Similar to previous studies, this study employs categorical cross-entropy [11] as the chosen loss function, making it well-suited for tasks involving multiple class classifications, as defined in (11).

$$L = -\sum_{i=1}^{N} t_i \log(p_i) \quad (11)$$

where N represents the number of classes, t_i denotes the actual label for class i, and p_i corresponds to the predicted probability for class i after applying the softmax function.

E. Result Representation

The performance evaluation results of the CNN-KAN model for classifying Rupiah banknote denominations for the visually impaired are presented in Table I. The evaluation was conducted using loss, accuracy, precision, recall, and AUC-ROC metrics by exploring five data split scenarios for training and testing: 50:50, 60:40, 70:30, 80:20, and 90:10.

Table I presents the performance of the CNN-KAN model across various training and testing data split scenarios. The results indicate that the model's performance improves as the proportion of training data increases. A significant enhancement is observed in the 90:10 split scenario, where the model achieves its best performance with a loss of 0.008, accuracy, precision, and recall of 100%, and a perfect AUC-ROC score of 1.

TABLE I. CNN-KAN MODEL PERFORMANCE

Split Data	Loss	Accuracy (%)	Precision (%)	Recall (%)	AUC-ROC
50:50	23.59	64.25	64.25	64.25	0.81
60:40	3.46	77.50	77.99	77.50	0.94
70:30	0.77	91.25	91.25	91.25	0.98
80:20	0.66	96.25	96.84	95.63	0.98
90:10	0.008	100	100	100	1

These findings suggest that more training data positively impacts the model's generalization ability in classifying Rupiah banknote denominations.

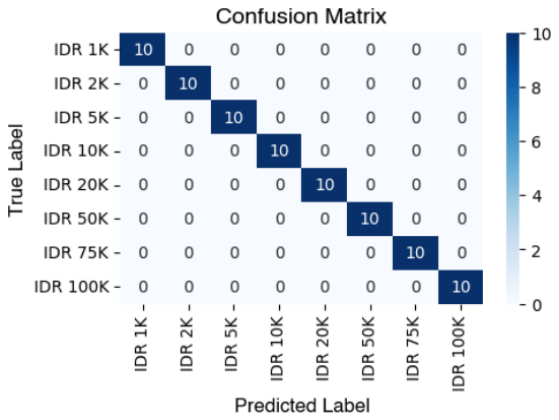


Fig. 5. Confusion matrix

Fig. 5 presents the confusion matrix from the evaluation of the CNN-KAN model on the test dataset, obtained through program output. This matrix illustrates the model's classification performance across eight classes of Rupiah banknote denominations. The confusion matrix reveals that the model achieved perfect accuracy, with all test samples correctly classified. This is indicated by the main diagonal values, which show 10 correct predictions for each class, with no misclassifications in other classes. These results indicate that the CNN-KAN model effectively recognizes and differentiates each denomination class.

TABLE II. COMPARISON OF RESEARCH RESULTS

Study	Model	Loss	Accuracy (%)	Precision (%)	Recall (%)	AUC-ROC
This study	CNN-KAN	0.008	100	100	100	1
Kurniadi et al. [11]	CNN AlexNet-VGGNet	1.28	95.31	95.24	93.75	0.99

Table II compares the results of this study with the CNN AlexNet-VGGNet model from a previous study [11], which used the same dataset and RMSProp optimizer. CNN-KAN demonstrated superior performance, achieving a loss of 0.008, 100% accuracy, precision, and recall, and an AUC-ROC of 1, indicating perfect classification. In contrast, the previous model lost 1.28, 95.31% accuracy, 95.24% precision, 93.75%, and an AUC-ROC of 0.99. These findings indicate that CNN-KAN significantly outperforms CNN AlexNet-VGGNet from the prior study in accurately classifying Rupiah banknote denominations.

IV. Conclusion

The test results indicate that CNN-KAN outperforms CNN AlexNet-VGGNet from the previous research, which used the same dataset and RMSProp optimizer. The CNN-KAN model achieved 100% accuracy, precision, and recall, an AUC-ROC of 1.00, and a loss of 0.008 under the 90:10 data split scenario. In contrast, under the same data split scenario, the CNN AlexNet-VGGNet model only achieved 95.31% accuracy, 95.24% precision, 93.75% recall, and an AUC-ROC of 0.99, with a loss of 1.28. These results demonstrate that CNN-KAN is more reliable in recognizing and distinguishing rupiah banknote denominations based on their visual characteristics. The superior performance of CNN-KAN in this study suggests that the model holds great potential for implementation in a rupiah banknote denomination detection system for visually impaired individuals. This study primarily focuses on model development; therefore, future research is expected to explore the application of CNN-KAN in software- or hardware-based systems to assist visually impaired users in conducting daily transactions more independently and accurately.

References

[1] Kementrian Kesehatan [Health Ministry], "Situasi Disabilitas," Pusat Data Dan Informasi Kementrian Kesehatan RI.

[2] M. Siahaan, C. H. Jasa, K. Anderson, M. V. Rosiana, S. Lim, and W. Yudianto, "Penerapan Artificial Intelligence (AI) Terhadap Seorang Penyandang Disabilitas Tunanetra," *J. Inf. Syst. Technol.*, vol. 1, no. 2, pp. 186–193, 2020, [Online]. Available: https://journal.uib.ac.id/index.php/joint/article/view/4322

[3] S. R. Argiandini, "Tunanetra Masih Kesulitan Kenali Uang, Latih Kepekaan via Lomba Pilih Koin," radarsolo. [Online]. Available: https://radarsolo.jawapos.com/pendidikan/842890785/tunanetra-masih-kesulitan-kenali-uanglatih-kepekaan-via-lomba-pilih-koin

[4] A. Hermawan, L. Lianata, Junaedi, and A. R. K. Maranto, "Implementasi Machine Learning Sebagai Pengenal Nominal Uang Rupiah dengan Metode YOLOv3," *SATIN - Sains dan Teknol. Inf.*, vol. 8, no. 1, pp. 12–22, Jun. 2022, doi: 10.33372/stn.v8i1.816.

[5] A. R. Pratama, M. Mustajib, and A. Nugroho, "Deteksi Citra Uang Kertas dengan Fitur RGB Menggunakan K-Nearest Neighbor," *J. Eksplora Inform.*, vol. 9, no. 2, pp. 163–172, Mar. 2020, doi: 10.30864/eksplora.v9i2.336.

[6] A. Prima, D. B. Santoso, and L. Nurpulaela, "Deteksi Otomatis Nominal Uang Kertas Rupiah Untuk Tunanetra Menggunakan Algoritma Arsitektur SSD MobiilenetV3," *TEKNOKOM*, vol. 6, no. 2, pp. 151–159, Aug. 2022, doi: 10.31943/teknokom.v6i2.166.

[7] W. Hamidah, T. S. B. Irawan, N. A. P. Hasbullah, and A. B. Kaswar, "Deteksi Nominal Uang Kertas Menggunakan OCR (Optical Character Recognition)," *Techno Xplore J. Ilmu Komput. dan Teknol. Inf.*, vol. 7, no. 2, pp. 72–76, 2022, doi: 10.36805/technoxplore.v7i2.2123.

[8] D. Aprillia, T. Rohana, T. Al Mudzakir, and D. Wahiddin, "KLIK: Kajian Ilmiah Informatika dan Komputer Deteksi Nominal Mata Uang Rupiah Menggunakan Metode Convolutional Neural Network dan Feedforward Neural Network," *Media Online*, vol. 4, no. 4, 2024, doi: 10.30865/klik.v4i4.1711.

[9] M. Cheon, "Kolmogorov-Arnold Network for Satellite Image Classification in Remote Sensing," 2024, doi: https://doi.org/10.48550/arXiv.2406.00600.

[10] A. Igali and P. Shamoi, "Image Classification using Fuzzy Pooling in Convolutional Kolmogorov-Arnold Networks," *Electron. Commer. Res.*, vol. 23, no. 1, pp. 43–73, Jul. 2024, doi: https://doi.org/10.48550/arXiv.2407.16268.

[11] D. Kurniadi, M. L. Rahmi, A. Mulyani, M. R. Nashrulloh, Y. Septiana, and I. Tri Julianto, "Classification of Rupiah Banknote for the Visually Impaired using Convolutional Neural Network with Custom Architecture and RMSProp Optimizer," in *2024 International Conference on ICT for Smart Society (ICISS)*, IEEE, Sep. 2024, pp. 1–6. doi: 10.1109/ICISS62896.2024.10751198.

[12] M. A. A. Al-qaness and S. Ni, "TCNN-KAN: Optimized CNN by Kolmogorov-Arnold Network and Pruning Techniques for sEMG Gesture Recognition," *IEEE J. Biomed. Heal. Informatics*, vol. 29, no. 1, pp. 188–197, Jan. 2025, doi: 10.1109/JBHI.2024.3467065.

[13] P. A. Nugroho, I. Fenriana, and R. Arijanto, "Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia," *Algor*, vol. 2, no. 1, pp. 12–21, 2020.

[14] Tri Wahyu Qur'ana, "Implementasi Metode Convolutional Neural Network (CNN) untuk Klasifikasi Motif pada Citra Sasirangan," *Media Inform.*, vol. 7, no. 2, p. 10, 2023, [Online]. Available: http://jurnal.big.go.id/index.php/GM/article/view/810

[15] N. A. Batubara and R. M. Awangga, *Tutorial Object Detection Plate Number with Convolution Neural Network (CNN)*. Bandung: Kreatif Industri Nusantara.

[16] M. R. Alwanda, R. P. K. Ramadhan, and D. Alamsyah, "Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle," *J. Algoritm.*, vol. 1, no. 1, pp. 45–56, 2020, doi: 10.35957/algoritme.v1i1.434.

[17] F. F. Maulana and N. Rochmawati, "Klasifikasi Citra Buah Menggunakan Convolutional Neural Network," *J. Informatics Comput. Sci.*, vol. 1, no. 02, pp. 104–108, 2020, doi: 10.26740/jinacs.v1n02.p104-108.

[18] Z. Liu *et al.*, "KAN: Kolmogorov-Arnold Networks," Apr. 2024, doi: 10.48550/arXiv.2404.19756.

[19] PyTorch Contributor, "SiLU." [Online]. Available: https://pytorch.org/docs/stable/generated/torch.nn.SiLU.html

[20] M. Uppal *et al.*, "Enhancing accuracy in brain stroke detection: Multi-layer perceptron with Adadelta, RMSProp and AdaMax optimizers," *Front. Bioeng. Biotechnol.*, vol. 11, no. September, pp. 1–15, 2023, doi: 10.3389/fbioe.2023.1257591.

 EISSN: 2961-872X ● PISSN: 2961-8924 ● DOI: 10.56741/IISTR.esl.00936