CNN ALGORITHM IMPLEMENTATION ON COINS CLASSIFICATION

Disusun sebagai salah satu syarat menyelesaikan Program Studi Strata I pada Program Studi Teknik Informatika Fakultas Komunikasi dan Informatika

Oleh:

DIMAS RISWANDA PRADANA PUTRA

L200194011

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA

2023

HALAMAN PERSETUJUAN

CNN ALGORITHM IMPLEMENTATION ON COINS CLASSIFICATION

PUBLIKASI ILMIAH

oleh:

DIMAS RISWANDA PRADANA PUTRA
L200194011

Telah diperiksa dan disetujui untuk diuji oleh:
Dosen Pembimbing

Helmi Imaduddin, S.Kom., M.Eng.

NIDN. 0607069301

HALAMAN PENGESAHAN

CNN ALGORITHM IMPLEMENTATION ON COINS CLASSIFICATION

OLEH

DIMAS RISWANDA PRADANA PUTRA

L200194011

Telah dipertahankan di depan Dewan Penguji
Fakultas Komunikasi dan Informatika
Universitas Muhammadiyah Surakarta
Pada hari Senin, 6 Februari 2023
dan dinyatakan telah memenuhi syarat

Dewan Penguji:

1. **Helmi Imaduddin, S.Kom., M.Eng.** (..................)

   (Ketua Dewan Penguji)

2. **Nurgiyatna, S.T., M.Sc., Ph.D.** (..................)

   (Anggota I Dewan Penguji)

3. **Husni Thamrin, M.T., Ph.D.** (..................)

   (Anggota II Dewan Penguji)

Dekan
Fakultas Komunikasi dan Informatika

Nurgiyatna S.T. M.Sc. Ph.D.
NIK. 881

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam naskah publikasi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam naskah dan disebutkan dalam daftar pustaka.

Apabila kelak terbukti ada ketidakbenaran dalam pernyataan saya di atas, maka akan saya pertanggungjawabkan sepenuhnya.

Surakarta, 20 Februari 2023

Penulis

DIMAS RISWANDA PRADANA PUTRA

L200194011

CNN ALGORITHM IMPLEMENTATION ON COINS CLASSIFICATION

Abstrak

Klasifikasi koin yang akurat adalah tugas penting dengan berbagai aplikasi dunia nyata. Metode yang ada untuk klasifikasi koin memiliki berbagai masalah dalam efisiensi dan akurasi. Studi ini menyajikan pendekatan untuk klasifikasi koin menggunakan *convolutional neural network* (CNN) dengan konfigurasi yang disesuaikan dan menguji kinerjanya dengan menggunakan kumpulan data gambar yang diambil melalui teknik *crawling* dari lima jenis nominal uang koin yang berbeda. Kemudian tingkat performa dievaluasi lebih lanjut menggunakan grafik *training-validation*, *confusion matrix*, dan laporan klasifikasi. Grafik menunjukan tingkat akurasi 88% pada set validasi terakhir dengan sedikitnya terjadi *overfitting* dan total akurasi pada *f1-score* yang mencapai 91%. Penelitian ini menjanjikan CNN dalam klasifikasi koin dapat membuka peluang baru untuk eksplorasi lebih lanjut di bidang ini. Algoritma yang diusulkan memiliki potensi untuk meningkatkan efisiensi dan akurasi sistem klasifikasi koin di masa depan.

Keywords: *Digital Image Processing, Machine Learning*, CNN, *Computer Vision*.

Abstract

Accurate coin classification is an important task with many real-world applications. Existing methods for coin classification suffer from various problems in efficiency and accuracy. This study presents an approach for coin classification using a convolutional neural network (CNN) with customized configurations and tests its performance using a data set of images taken by crawling technique of five different coin denominations. Then the level of performance is further evaluated using training-validation graphs, confusion matrix, and classification reports. The graph shows an accuracy rate of 88% in the last validation set with minimal overfitting and total accuracy on the f1-score which reaches 91%. This research is promising CNN in coin classification can open up new opportunities for further exploration in this field. The proposed algorithm has the potential to improve the efficiency and accuracy of future coin classification systems.

Keywords: Digital Image Processing, Machine Learning, CNN, Computer Vision.

1. INTRODUCTION

1.1 Description

Coin classification is a crucial task in the field of automated coin handling systems, such as vending machines, coin counting machines, and coin-operated devices. The goal of coin classification is to accurately identify the denomination of a coin based on its image. However, this task can be challenging due to similarities in the shape and size of different types of coins, as well as variations in lighting and coin orientation in the images. There are three types of coin recognition systems available in the market, based on different methods: mechanical (Journal, n.d.), electromagnetic, and image processing (Shelgikar & Lobo, 2014).

Mechanical method-based systems use various physical parameters such as diameter, radius, thickness, weight, and magnetism to distinguish between coins. However, these parameters are not always sufficient to differentiate between coins made of different materials, making them susceptible to counterfeit coins. Electromagnetic method-based systems use a combination of factors such as diameter, thickness, weight, magnetism, and changes in amplitude and direction of frequency caused by different materials to differentiate between coins. This improves the security compared to mechanical method-based systems, but they can still be deceived by certain types of game coins.

Previous coin classification methods have relied on image processing techniques, such as thresholding, edge detection, and shape matching. These methods have limitations in terms of accuracy and efficiency, particularly when dealing with a large number of coins and variations in lighting and orientation.

One of image processing techniques is by implementing deep learning. Deep Learning is a branch of machine learning based on Artificial Neural Networks (ANN) or can be said to be a development of ANN. In deep learning, a computer learns to classify directly from images or sounds (Schmidhuber, 2015). Convolutional Neural Network (CNN/ConvNet) is a deep learning algorithm which is a development of Multilayer Perceptron (MPL) which is designed to process data in two dimensions, for example images or sound. CNN can learn directly from the image thereby reducing the burden of programming (Eka Putra, 2016).

Recently, deep learning techniques, particularly Convolutional Neural Networks (CNNs) have shown remarkable performance in image classification tasks. CNNs are able to learn complex features from images, and can be trained using large datasets. In this paper, we propose a CNN-based algorithm for coin classification and evaluate its performance using a dataset of images of five different types of coins. We also evaluate its performance with traditional methods and evaluate it with different evaluation metrics.

This study aims to address the limitations of traditional coin classification methods by proposing a CNN-based algorithm and evaluating its performance on a dataset of images of five different types of coins. Our results show that the proposed CNN algorithm achieves high accuracy and precision, and it opens up new opportunities for further research in this field. The proposed algorithm has the potential to improve the efficiency and accuracy of coin sorting systems in the future, making them more reliable and secure.

Additionally, the proposed algorithm can also be applied to other object classification tasks and other types of images. The study also evaluates the model's performance using a training-validation graph, a confusion matrix, and a classification report. It is important to note that coin classification is not just a binary or multi-class classification task, but a real-world problem that requires other factors to be considered, such as the speed, robustness, and cost-effectiveness of the algorithm. The proposed CNN algorithm shows great promise in addressing these challenges and has a wide range of applications in the industry, such as in banks, supermarkets, grocery stores, vending machines, and other places.

1.2 Literature Survey

A developed system for detecting pattern and implemented into 500 Japanese Yen coin and a 500 Korean Won coin in 1992 by using a fixed invariance network and trainable multi-layered network. The result of their neural network approach is work well in various rotation pattern (Fukumi et al., 1992).

A new approach proposed in 2005 by focusing only on the numerals rather than the use of other images presented in the front and rear side of the coin with 72 images of 3 different India Rupee coins with edge detection based on neural pattern. The result of their proposed approach produced an overall accuracy of 92.43% and for the edge detection and localization process provided an accuracy of 85% (Bremananth et al., 2005).

A continuation of the previous approach proposed in 2011 by using Robert’s edge detection method, Laplacian of Gaussion edge detection method, canny edge detection method and Multi-Level Counter Propagation Neural Network (ML-CPNN) on several denominations of Indian coins. The results are work well with Robert’s edge detection method gives 93% of accuracy, Laplacian of Gaussion method 95% of the result, Canny edge detection method yields 97.25% result and the ML-CPNN approach yields 99.47% of recognition rate (VeluC et al., 2011).

An approach proposed in 2015 by using multi-layered backpropagation neural network by combining different approach like image segmentation, edge detection and image transform methods to detect fake coin and real coin. The result of their approach is work well with recognition rate around 82% (Roomi & Rajee, 2015).

On the latest paper, an advanced method proposed by applying the faster region-based CNN, geometric constraints, and the residual network (ResNet) on the denomination of Jordian Dinar and Korean Won with more than 6,400 images as database. The results are pretty impressive with a better detection performance than the state-of-the-art methods based on handcrafted features and deep features (Park et al., 2020).

Based on the literature survey above, we can see that coin classifications have been improved from year to year. This paper has a decent positioning between simplicity and complexity due to the proposed method with one processing method.

2. METHODS

2.1 System Analysis

Figure 1. System workflow

In Figure 1, the proposed system flow begins with a data acquisition process called dataset retrieval. The dataset is taken from the image results from the crawling technique. Each coin image then enters the Pre-Processing stage, at this stage the image will experience resizing. The resized images will be used to train the network model on the CNN algorithm. The result of this training is the accuracy value of the classification of each class in the dataset. After the image is classified, then evaluation process is carried out to identify the coins.

2.2 Image Data Retrieval

The data used in this study are all nominal Rupiah coins that have been circulating in Indonesia, they are 50 Rupiah, 100 Rupiah, 200 Rupiah, 500 Rupiah, and 1000 Rupiah. The training dataset is obtained from the results of crawling techniques from the Kaggle website (*Kaggle*, n.d.; *World Coins*, n.d.).

2.3 Pre-processing Data

The training dataset consists of 180 images, each class has 36 images and is divided into 5 validations. Each validation has a training and testing process where the dataset is taken at random to be used as data training. At this stage the image data will be resized into 250x250 pixels using image resizing. Image resizing is a technique of changing the image size to be larger or smaller than the image, then the image is saved in the .npy (numpy array) file format.

2.4 CNN Algorithm (Convolutional Neural Network)

Convolutional Neural Networks is a machine learning method developed from the Multi-Layer Perceptron (MLP) designed to process two-dimensional data. CNN is included in the type of Deep Neural Network because of the level of depth of the network and is widely implemented in image data. The linear operation used is the convulsion operation and the weights used are in the form of 4 dimensions which are a collection of convulsive kernels (Eka Putra, 2016). On CNN there is a layer that functions for feature extraction and feature classification.

2.5 CNN Architecture

The CNN architecture used in this study consists of 2 convolution layers + ReLU, 1 subsampling layer and 2 fully connected layers.

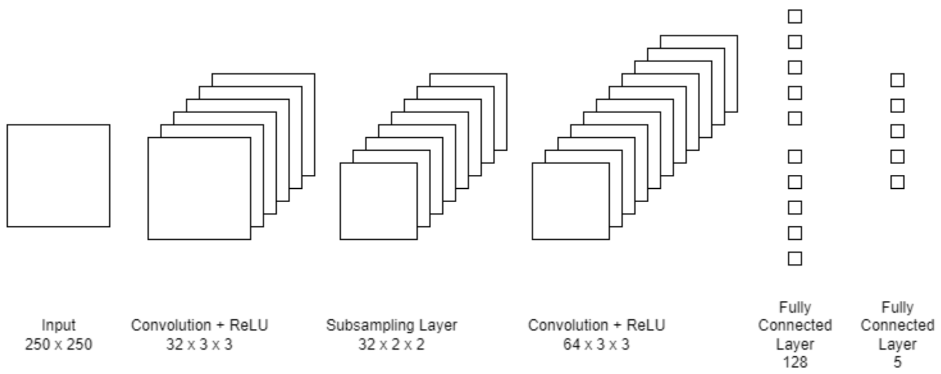

Figure 2. CNN Architecture

As we can see in Figure 2, the 250 x 250 images will be inserted. Then this input will be processed on different stages. First, feature extraction will be implemented start with 3 x 3 convolution layer with 32 filters with ReLU activation. Followed by a 2 x 2 subsampling layer with the same values as the previous layer and last 3 x 3 convolution layer with 64 filters. Second, the data will be flattened into one dimensional data followed with 128 dense fully connected layer for classification. The model will using cross entropy loss to calculate the performance of the model.

2.6 Feature Extraction

In the feature extraction stage of the image, there are two main layers, the convolution layer and the subsampling/pooling layer. There is an activation function used at this stage called ReLU. The number of layers to produce good accuracy is to do a lot of experiments. The order of layers is not always the same. The following is an explanation of each layer and its activation function:

A. Convolution Layers

The convolution process makes use of what is known as a filter. Like images, filters have a specific height, width, and thickness. This filter is initialized with a certain value, and this filter value is the parameter that will be updated in the learning process (Bhardwaj et al., 2018). The convolution layer is the result of multiplying the input filter and image. The following is the formula for the convolution function:

$$Q_1 = f\left(\sum_{i=1}^{N} l_{i,j} * k_{i,j}\right) \tag{1}$$

In simpler terms, the convolution function takes a small window of the input data and applies the kernel (filter) to it element-wise, producing a single output value. This process is then repeated for every possible window in the input data to produce the output feature map. The output feature map encodes the presence of different features (patterns) in the input data, as learned by the kernel.

B. Subsampling / Pooling Layers

The pooling or subsampling layer often follows the convolution layer directly on the CNN. Its role is to sample the output from the convolution layer along the spatial dimensions of height and width (Bhardwaj et al., 2018). The image will be divided into several parts according to the specified layer size. The method used in the subsampling layer is max pooling, by selecting the largest value in the image matrix.

C. ReLU (Rectified Linear Unit)

ReLu or rectified linear unit is one of the activation functions. The ReLu function is to remove negative values in the image. The way the ReLu activation function works is by replacing negative values in images or feature maps with a value of 0 (Agarap, 2019). The following is the formula for the ReLU function:

$$f(x) = \begin{cases} x(x \geq 0) \\ 0(x < 0) \end{cases} \tag{2}$$

Where x is the input value and f(x) is the output value. One of the main advantages of the ReLU function is that it is simple to compute and has a derivative that is easy to compute. This makes it computationally efficient and easy to optimize. The ReLU function is often used as an activation function in the hidden layers of a neural network, although it can also be used in the output layer for certain tasks.

2.7 Classification of Features

At this stage there is a fully connected layer and an activation function called softmax. The following is an explanation of the layers and activation functions at the feature classification stage:

A. Flatten

Flatten is the process of reshaping a feature (reshape feature map) into a vector so that it can be used as input from a fully connected layer.

B. Fully Connected Layer

This is a layer that is usually used in MLP applications and aims to transform data dimensions so that data can be classified linearly. Each neuron in the convolution layer needs to be converted to one-dimensional data before it can be included in the fully connected layer. Because this causes the data to lose its spatial information and cannot be reversed, a fully connected layer can only be implemented at the end (Eka Putra, 2016).

C. Softmax Activation

Softmax is a function that takes a real number vector input of K, and normalizes it to a probability distribution consisting of the probabilities of K. Before applying softmax, any component of a vector could be negative, or greater than one; and may not add up to 1, but after applying softmax, each component will be in the interval (0 - 1), and the components will add up to 1, so it can be interpreted as a probability. Furthermore, a larger input component will correspond to a greater probability. Softmax is often used in neural networks, Softmax is used to determine the appropriate output (Sharma et al., 2020). The following is the softmax activation function formula:

$$p(x) = \frac{e^x}{\sum_{k=1}^{k} e^x} \qquad (3)$$

the softmax function converts the input vector into a probability distribution by applying the exponential function to each element of the input vector and normalizing the result. The resulting output vector has values between 0 and 1 that sum to 1, representing the probabilities of the input vector belonging to each class. For example, given an input vector [1, 2, 3], the softmax function would output will be [0.09, 0.24, 0.67]. One of the main advantages of the softmax function is that it produces well-calibrated probability estimates, which are useful for classification tasks. However, it can also be sensitive to the scale of the input values, and it is often used in conjunction with other techniques such as temperature scaling to improve its performance.

D. Loss Function (Cross Entropy Loss)

Cross entropy loss is a function used to calculate the performance of a model, namely by calculating the error resulting from the model (Martinez & Stiefelhagen, 2019). Usually, this cross-entropy loss is used after the softmax function.

$$H(p, q) = -\sum_{i=1}^{N} p_i \log q_i \tag{4}$$

the cross entropy loss function measures the difference between the true and predicted probability distributions by computing the negative log-likelihood of the true distribution under the predicted distribution. The log-likelihood is calculated by summing the logarithm of the predicted probability for each true class. The cross entropy loss is a popular choice for classification tasks because it is easy to compute and has a nice probabilistic interpretation. However, it can be sensitive to the scale of the input values and may require the use of techniques such as weight scaling to work well in practice.

2.8 Evaluation

Training validation graphs and confusion matrices are commonly used to evaluate the performance of a machine learning model, while classification reports provide a summary of the precision, recall, and f1-score for each class.

A. Training Validation Graph

Training Validation Graphs are used to visualize the training and validation performance of a model over a series of epochs. These graphs are typically used to identify overfitting, which occurs when a model performs well on the training data but poorly on the validation data. Overfitting can be mitigated by reducing the model's capacity or increasing the amount of training data.

B. Confusion Matrix

A confusion matrix is a table that is used to evaluate the performance of a classification model. It shows the number of correct and incorrect predictions for each class. From a confusion matrix, it is possible to calculate a variety of evaluation metrics, such as precision, recall, and f1-score.

C. Classification Report

A classification report is a summary of the precision, recall, and f1-score for each class. It is a useful tool for comparing the performance of a classifier across different classes. Precision is the number of true positive predictions divided by the total number of positive predictions. It is a measure of the accuracy of the classifier when it predicts the positive class. Recall is the number of true positive predictions divided by the total number of actual positive instances. It is a measure of the classifier's ability to identify all instances of the positive class. The f1-score is the harmonic mean of precision and recall, with a higher score indicating better performance.

3. RESULT AND DISCUSSION

3.1 Training Validation Graph

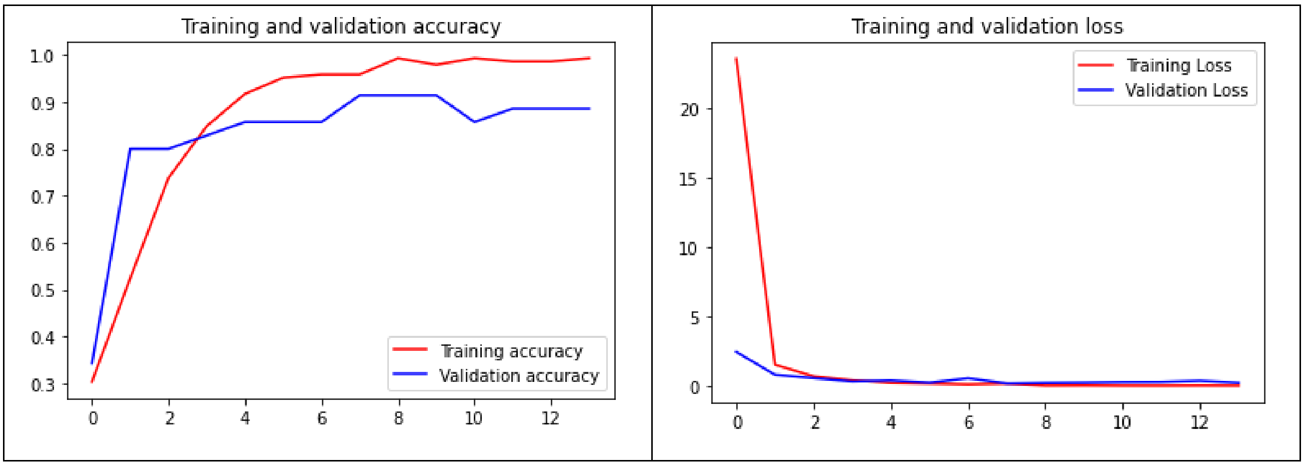

Figure 3. Accuracy and Loss graphs from Training and Validation

On the left side of Figure 3, the accuracy graph shows that the model's accuracy on the training set increases with each training epoch, reaching a peak of 99% at the final epoch. The model's accuracy on the validation set also increases with each epoch, reaching a peak of 88% at the final epoch. The gap between the training and validation accuracy is small, indicating that the model is not overfitting to the training data.

While on the right side of Figure 3, the loss graph shows that the model's loss on the training set decreases with each training epoch, reaching a minimum of 0.04 at the final epoch. The model's loss on the validation set also decreases with each epoch, reaching a minimum of 0.24 at the final epoch. The gap between the training and validation loss is also small, indicating that the model is not overfitting to the training data.

It is important to keep in mind that, on a Loss Graph, a lower value means that the model predictions are closer to the true values. So, the best model is one that has low loss on both training and validation dataset and low gap between them. While on accuracy graph, A higher value means that the model is making more correct predictions, So, the best model is one that has high accuracy on both training and validation dataset and low gap between them.

It can be observed that the model is not overfitting because the gap between the performance on the training and validation set is small for both accuracy and loss. This means that the model is generalizing well to new, unseen data.

3.2 Confusion Matrix

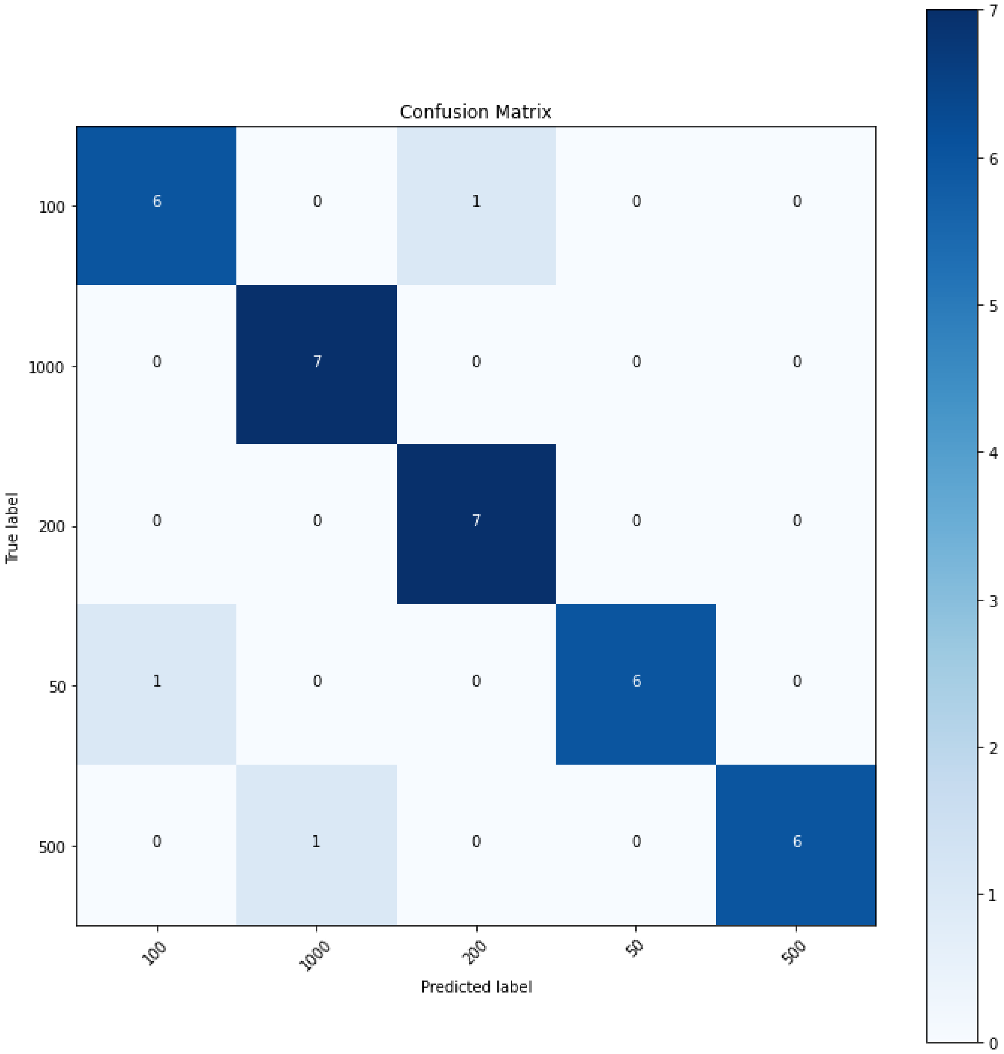

Figure 4. Confusion Matrix

Here in Figure 4, each row represents a predicted class, and each column represents an actual class. The diagonal cells contain the correct classifications made by the model, while the off-diagonal cells contain the misclassifications. For example, in this matrix, all 200 Rupiah are correctly classified, a 50 Rupiah are classified as 100 Rupiah, and a 500 Rupiah are classified as 1000 Rupiah. And so on.

From this confusion matrix, we can calculate various evaluation metrics such as accuracy, precision, recall, F1 score, and so on. Also, from this matrix it is quite clear that how well the model is classifying the different coins, in which it's performing well and in which not, also which class is being misclassified more often. It is important to keep in mind that, the high number of misclassification rate for a particular class, implies low performance for that class.

3.3 Classification Report

Table 1. Classification Report

Class	**Precision**	**Recall**	**F1-score**
50 IDR	100%	86%	92%
100 IDR	86%	86%	86%
200 IDR	88%	100%	93%
500 IDR	100%	86%	92%
1000 IDR	88%	100%	93%
Total			
Accuracy			91%
Macro Average	92%	91%	91%
Weighted Average	92%	91%	91%

In the table above, the classification report shows the performance of the coin classification model on four different classes: 50 IDR, 100 IDR, 200 IDR, 500 IDR, and 1000 IDR. For each class, it shows the precision, recall, and f1-score. The classification report also shows that the model is performing well on all classes.

- Precision: is high for all classes which means that when the model predicted a sample to be in a class, it was most likely to be in that class.

- Recall: is also high for all classes which means that when there was a sample in a class, the model was most likely to detect it.
- F1-score: is also high for all classes, this is a harmonic mean of precision and recall, this high value indicates that both precision and recall are high.

It's clear that the model is performing well in this scenario because the precision, recall, f1-score and support are all high for all classes. This indicates that the model is making accurate predictions for all classes and not suffering from any bias in one class.

It is important to keep in mind that, to get good results on the classification report, it's important to have a well-balanced dataset, well-tuned model and a good understanding of the trade-off between precision and recall. It is also important to note that, achieving good results on the classification report alone does not guarantee that the model is the best possible solution, it's important to consider the use-case and requirements of the problem.

3.4 Underperformance and Limitation Analysis

Based on the previous 3 performance evaluation analysis. We can analyse the underperformance and limitations of the image processing based system in this paper is focused on several key aspects: the data used, the limited size of the dataset, the image rescaling, the limited scope of the classification task, the CNN configuration, and the limitation of research time.

The data used in this study were coin images obtained from the results of the crawling technique. Although this method was effective in acquiring a sufficient number of images, it may have also introduced some variability or inaccuracies into the data. Additionally, the dataset only consisted of 180 images, which may have limited the robustness and generalizability of the results.

The images were rescaled into 250x250 pixels, which may have reduced the amount of information available for the model to learn from. Furthermore, the image classification only included 5 nominals of Indonesian Rupiah, which limited the scope of the study and the potential applications of the proposed algorithm.

In terms of the CNN configuration, the model used in this study may have had some limitations that affected its performance. For example, the number of layers, the type of activation functions, and the optimization algorithm may have not been optimally configured for the task of coin classification.

Finally, the limitation of research time may have also impacted the results of this study. For example, more data or a larger and more diverse dataset could have been used to further validate the proposed algorithm. Additionally, the CNN configuration could have been further optimized, or alternative algorithms could have been considered and evaluated.

In conclusion, the underperformance and limitations of the image processing based system in this study highlight the importance of considering the quality and diversity of the data, the scope of the classification task, the configuration of the model, and the available resources when developing and evaluating such systems.

4. CLOSING

In this study, we proposed a CNN algorithm for coin classification and evaluated its performance using a dataset of images of five different types of coins. We first pre-processed the data by resizing and normalizing the images, and then trained the CNN using a combination of convolutional layers, pooling layers, and fully connected layers. Our results showed that the proposed CNN algorithm achieved an accuracy of 88% on the validation set, which is a significant improvement over traditional method.

We also evaluated the model's performance using a training-validation graph, a confusion matrix, and a classification report. The training-validation graph showed that the model's accuracy on the training set increases with each epoch, reaching a peak of 99%, while the model's accuracy on the validation set also increases with each epoch, reaching a peak of 88%. The gap between the training and validation accuracy is small, indicating that the model is not often overfitting to the training data. One possible reason for this low performance is the limited number of training examples for each class, which may have resulted in overfitting to the other classes. Additionally, the model may not have been able to extract relevant features for each class due to the inherent differences in the images of that class compared to the others. The confusion matrix showed that the model is performing well and classification report also confirms that.

In conclusion, our study demonstrates the effectiveness of CNNs in coin classification, and it opens up new opportunities for further research in this field. The proposed CNN algorithm can be used to improve the efficiency and accuracy of coin sorting systems in the future. However, further research is needed to improve the performance on several denominations and also to explore other architectures that can be used in coin classification. Additionally, this approach could also be applied to other object classification tasks and other types of images. It is also important to keep in mind that, coin classification is not just a binary or multi-class classification task, but it's also a real-world problem which requires other factors to be considered, such as the speed, robustness, and cost-effectiveness of the algorithm. Nonetheless, our proposed CNN algorithm shows great promise in addressing these challenges.

BIBLIOGRAPHY

Agarap, A. F. (2019). *Deep Learning using Rectified Linear Units (ReLU)* (arXiv:1803.08375). arXiv. http://arxiv.org/abs/1803.08375

Bhardwaj, A., Di, W., & Wei, J. (2018). *Deep Learning Essentials: Your hands-on guide to the fundamentals of deep learning and neural network modeling*. Packt Publishing Ltd.

Bremananth, R., Balaji, B., Sankari, M., & Chitra, A. (2005). A New Approach to Coin Recognition using Neural Pattern Analysis. *2005 Annual IEEE India Conference - Indicon*, 366–370. https://doi.org/10.1109/INDCON.2005.1590191

Eka Putra, W. S. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101. *Jurnal Teknik ITS*, *5*(1). https://doi.org/10.12962/j23373539.v5i1.15696

Fukumi, M., Omatu, S., Takeda, F., & Kosaka, T. (1992). Rotation-invariant neural pattern recognition system with application to coin recognition. *IEEE Transactions on Neural Networks*, *3*(2), 272–279. https://doi.org/10.1109/72.125868

Journal, I. (n.d.). *REVIEW ON THE COIN RECOGNITION SYSTEM WITH ROTATION INVARIANT*. Retrieved January 24, 2023, from https://www.academia.edu/8331064/REVIEW_ON_THE_COIN_RECOGNITION_SYSTEM_WITH_ROTATION_INVARIANT

Kaggle: Your Machine Learning and Data Science Community. (n.d.). Retrieved December 6, 2022, from https://www.kaggle.com/

Martinez, M., & Stiefelhagen, R. (2019). Taming the Cross Entropy Loss. In T. Brox, A. Bruhn, & M. Fritz (Eds.), *Pattern Recognition* (pp. 628–637). Springer International Publishing. https://doi.org/10.1007/978-3-030-12939-2_43

Park, C., Cho, S. W., Baek, N. R., Choi, J., & Park, K. R. (2020). Deep Feature-Based Three-Stage Detection of Banknotes and Coins for Assisting Visually Impaired People. *IEEE Access*, *8*, 184598–184613. https://doi.org/10.1109/ACCESS.2020.3029526

Roomi, S. M. M., & Rajee, R. B. J. (2015). Coin detection and recognition using neural networks. *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*, 1–6. https://doi.org/10.1109/ICCPCT.2015.7159434

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Sharma, S., Sharma, S., & Athaiya, A. (2020). ACTIVATION FUNCTIONS IN NEURAL NETWORKS. *International Journal of Engineering Applied Sciences and Technology*, *04*(12), 310–316. https://doi.org/10.33564/IJEAST.2020.v04i12.054

Shelgikar, N. N., & Lobo, P. L. M. R. J. (2014). *Indian Coin Recognition with Rotation Invariance using Radial Blur Technique*. https://www.semanticscholar.org/paper/Indian-Coin-Recognition-with-Rotation-Invariance-Shelgikar-Lobo/fa924915ada16e8850ac15e6c59a2f16794e40e7

VeluC, M., Vivekanadan, P., & KashwanK, R. (2011). *Indian Coin Recognition and Sum Counting System of Image Data Mining Using Artificial Neural Networks*. https://www.semanticscholar.org/paper/Indian-Coin-Recognition-and-Sum-Counting-System-of-VeluC-Vivekanadan/3a2a5430b354c3fff4bf9ea75de2372fcd364a17

World Coins. (n.d.). Retrieved January 9, 2023, from https://www.kaggle.com/datasets/wanderdust/coin-images