TRANSMISI: JURNAL ILMIAH TEKNIK ELEKTRO, 27, (1), JANUARI 2025
p-ISSN 1411-0814 e-ISSN 2407-6422

CNN ALGORITHM OPTIMIZATION FOR CLASSIFYING NUMBERS IN HANDWRITING

Dinda Septiani*), Mohammad Fadhli dan Sopian Soim

Department of Electrical Engineering, Politeknik Negeri Sriwijaya, Palembang, Indonesia
Department of Telecommunication Engineering, Politeknik Negeri Sriwijaya, Palembang, Indonesia

*) *E-mail: dindasept09@gmail.com*

Abstrak

Pengenalan angka tulisan tangan merupakan tantangan penting dalam pengolahan citra, dengan aplikasi luas dalam berbagai bidang seperti pengolahan dokumen dan otomatisasi data. Penelitian ini bertujuan untuk mengoptimalkan kinerja model Convolutional Neural Network (CNN) dalam mengklasifikasi angka tulisan tangan pada dataset MNIST. Dalam penelitian ini, dilakukan eksperimen dengan variasi jumlah lapisan CNN untuk mengevaluasi pengaruhnya terhadap akurasi model. Hasil penelitian menunjukkan bahwa model dengan 4 lapisan konvolusi mencapai akurasi tertinggi sebesar 92,41%, yang menandakan peningkatan signifikan dalam kemampuan model untuk mengekstraksi fitur-fitur penting dari gambar dibandingkan dengan model dengan jumlah lapisan lebih sedikit. Penelitian ini juga mengaplikasikan model terbaik ke dalam sebuah website yang memungkinkan pengguna untuk mengenali angka tulisan tangan secara real-time. Langkah ini memberikan manfaat praktis dalam pengembangan sistem pengenalan karakter otomatis dan menunjukkan bagaimana teknologi ini dapat diterapkan langsung di kehidupan sehari-hari.

Kata kunci: pengenalan karakter tulisan tangan, Convolutional Neural Network, deep learning, optimasi model.

Abstract

Handwritten numeral recognition is an important challenge in image processing, with wide applications in areas such as document processing and data automation. This research aims to optimize the performance of Convolutional Neural Network (CNN) model in classifying handwritten numerals on MNIST dataset. In this research, experiments were conducted with variations in the number of CNN layers to evaluate their effect on model accuracy. The results show that the model with 4 convolutional layers achieves the highest accuracy of 92.41%, which signifies a significant improvement in the model's ability to extract important features from the image compared to the model with fewer layers. This research also applied the best model to a website that allows users to recognize handwritten numerals in real-time. This provides practical benefits in the development of automatic character recognition systems and shows how this technology can be applied directly in everyday life.

Keywords: handwritten character recognition, Convolutional Neural Network, deep learning, model optimization.

1. Introduction

In the modern era, information technology has undergone rapid development that significantly impacts various fields of life. One important development is artificial intelligence in image processing, which opens up opportunities to automate multiple processes, including number recognition in handwriting [1]. With computers' increasing capacity and capability, humans can now retrieve more in-depth information from images, including handwritten number recognition [2].

Handwritten number recognition using machine learning provides many significant benefits. First, document and data processing automation increases efficiency and reduces human error, enabling the processing of large amounts of data quickly and accurately. Deep learning algorithms, such as Convolutional Neural Networks (CNN), can recognize handwriting patterns with high accuracy, producing consistent results and reducing variability caused by human factors. This technology also improves accessibility and user experience through mobile applications that recognize handwriting for data input or identity verification. Example applications include bank check processing, postal code reading, questionnaire form analysis, and signature-based authentication systems [3].

Numeric classification in handwriting is an essential challenge in image processing and machine learning. The variability in different individuals' writing styles causes difficulties in recognizing and classifying the numbers. One solution is to use pattern recognition technologies, including statistical techniques, artificial neural networks, and support vector machines. Convolutional Neural Network (CNN) is an effective method for image recognition [4].

CNN is a branch of machine learning that uses a specialized artificial neural network architecture for image recognition. CNNs have proven effective in a variety of image recognition tasks, such as face, object, and text recognition. CNNs have an architecture inspired by biological neural networks, consisting of convolutional layers, pooling layers, and fully-connected layers [5]. This structure is designed to capture and process information from image data, with CNNs showing advantages in various applications over other machine learning methods [6].

Previous research Hidden Markov Model (HMM) has a major drawback in handwritten numeral recognition because the algorithm is designed for sequential data, not static image data. HMM is not able to capture the spatial relationship between pixels in an image, which is an important element for complex pattern recognition such as numbers [7]. Next research addressed the development of handwritten text recognition using deep learning and computer vision. The Emnist dataset, which contains alphanumeric characters, was used to train CNN models. Pre-processing techniques were applied, and the model was trained using Keras and TensorFlow, achieving 87% accuracy [8]. Another study discussed using Particle Swarm Optimization (PSO) with CNN for handwritten character recognition, showing improved accuracy over conventional methods [9]. Research on CNN on facial expressions with the JAFFE dataset shows 87.5% accuracy even though it only uses 3 CNN layers [10]. Another study using Retina Image Bank and MESSIDOR datasets showed good accuracy in fundus classification with specific parameters [11].

CNN excels in pattern recognition because it captures spatial relationships between pixels through convolution and pooling layers. However, the CNN algorithm needs to be optimized to achieve the best performance. Proper optimization allows CNN to perform excellent number classification in handwriting [12]. The addition of matrix operations such as dropout and batch normalization is important to optimize the model. Dropout helps prevent overfitting by randomly disabling neurons during training, while batch normalization normalizes the input of each layer to speed up training and model stability [13]. Based on this, this research will examine the optimization of the CNN algorithm to classify numbers in handwriting.

2. Methods

In Figure 1, the methodology used in this research is depicted. This research starts with problem identification, data collection, CNN model building, CNN model training, model evaluation, and website planning.

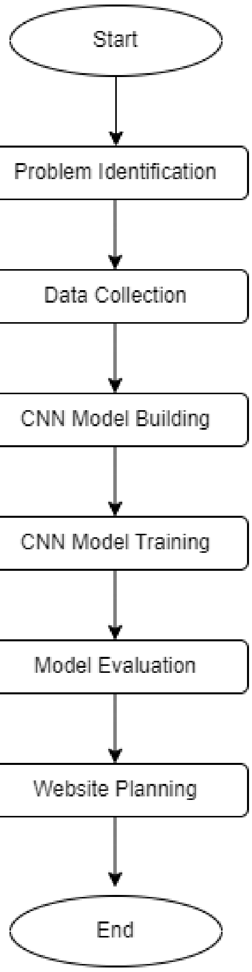


Figure 1. Research Framework

2.1. Problem Identification

This stage begins by determining the main problem faced in the optimization of handwritten number recognition systems; a model is needed that can identify and classify numbers with a high level of accuracy, even when the numbers are written in a variety of different styles and shapes. This research aims to optimize the Convolutional Neural Network (CNN) algorithm that can effectively classify numbers in handwriting to improve the accuracy and efficiency of the number character recognition process.

2.2. Data Collection

The data in this study was obtained from the Kaggle platform, an online community that provides a wide variety of datasets for machine learning and data science purposes. The dataset used in this research is named "**MNIST Handwritten Digit Dataset**" and contains about 14,843 handwritten digit images. This dataset will be divided into 11,879 training images and 2,964 testing images of handwritten digits (0-9), with an image size of 28x28 pixels.

2.3. CNN Model Building

At this stage, a Convolutional Neural Network (CNN) model will be designed to classify numbers in handwriting. The model architecture will include several convolution layers to extract important features from the image, a pooling layer to reduce dimensionality and prevent overfitting, and a fully connected layer to perform the final classification.

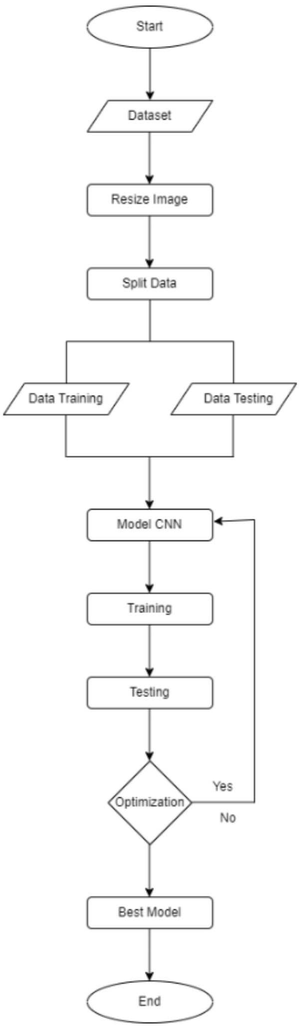


Figure 2. Flowchart of Model Generation

Figure 2 illustrates a CNN modeling flowchart with the Flowchart showing the process of developing a CNN model, from dataset processing, data division into training and testing data, to model training, testing, and optimization. The training data trains the model to recognize patterns, while the testing data evaluates the performance of the model. Although it appears to be done in parallel, the testing data is actually used only after the training is complete. This process is often repeated using techniques such as k-fold cross-validation to ensure the resulting model is effective and reliable.

2.4. CNN Model Training

At this stage, the designed CNN model will be trained using handwritten datasets, such as the MNIST dataset, which contains around 14,843 handwritten numeric images. This dataset will be divided into 11,879 training images and 2,964 testing images of handwritten numbers (0-9). The training process will involve optimizing training parameters such as learning rate, number of epochs, and batch size, which will be adjusted to achieve the best performance. Figure 3 illustrates the Convolutional Neural Network (CNN) training flowchart.

This model starts with an input layer that receives the image, and then passes through several Conv2D layers, which are responsible for recognizing important patterns such as edges or textures in the image. After that, each Conv2D layer is followed by MaxPool2D, which reduces the size of the data without losing important information, making the process more efficient. The processed data is then flattened into one dimension through the Flatten layer, so that it can enter the Dense or fully connected layer. The first Dense layer has 512 neurons in charge of processing the features, while the second Dense layer has 10 neurons each representing one output class (numbers 0-9 in number classification). The model also comes with a 25% dropout to prevent overfitting, which is when the model fits the training data so well that it cannot perform well on new data. Finally, there is a Softmax function that converts the output into probabilities to determine which class is most likely to be correct.

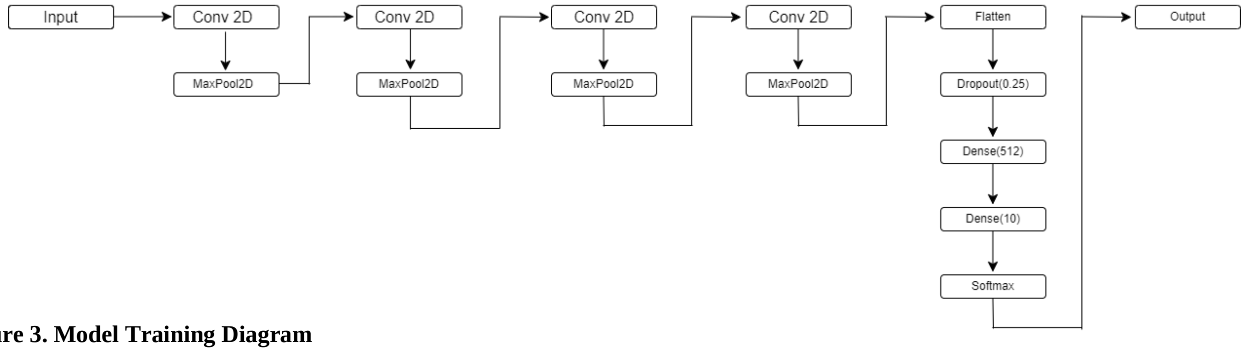


Figure 3. Model Training Diagram

2.5. Model Evaluation

The model testing process aims to obtain the results of handwritten number detection, specifically by counting the number of numbers detected from the test data. Accuracy evaluation is performed using Confusion Matrix, which includes metrics such as accuracy, precision, recall, and F1 score. Accuracy measures the model's ability to classify handwritten numerals as a whole. Precision indicates how often the model correctly classifies a particular number when predicting that number (detecting the number "5" without misidentifying other numbers as "5"). Recall measures the model's ability to detect all occurrences of a particular number, ensuring that the model does not miss many numbers present in the data. The F1 score provides a balance between precision and recall, allowing for a more comprehensive evaluation of the model. Equations (1) to (4) represent the formulas for calculating accuracy, precision, recall, and F1 score.

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

Precision is the ratio of all positive data considered positive to the total number of correct positive estimates. The following formula generates the Precision value.

$$Precision = \frac{TP}{TP + FP}$$

Recall is defined as the fraction of positive predictions to all correct positive data. The following formula generates the recall value.

$$Recall = \frac{TP}{TP + FN}$$

F1-Score is defined as the harmonic mean of recall and precision. The following formula generates the F1-Score value.

$$F1\ Score = 2 \times \frac{Recall \times Precission}{Recall + Precission}$$

Explanation:

1. True Positive (TP) represents image samples classified as positive and indeed positive.
2. True Negative (TN) represents image samples classified as negative and indeed negative.
3. False Positive (FP) represents image samples classified as negative but positive.
4. False Negative (FN) represents image samples classified as positive but negative [14-15].

2.6. Website Planning

This stage aims to assess the model's accuracy in identifying handwritten numbers. How well the CNN model detects and distinguishes various numbers is an important factor and a key component in evaluating its accuracy in handwritten number detection. Then, it is implemented on a website that allows users to upload handwritten images and get real-time detection results.

3. Results and Analysis

3.1. CNN Layer 1 Model Architecture Results

The first experiment used one convolution layer with 64 filters, 3x3 kernel size, ReLU activation function, and input shape (150, 150, 3). This was followed by Max pooling 2x2, Flatten() to convert the 2D matrix into a 1D vector, Dropout 25% to prevent overfitting, and a dense layer with 64 neurons and ReLU activation function. The last layer is output with ten neurons and a softmax activation function for multi-class classification.

Figure 4 shows a graph of model accuracy from the results of experiments carried out on the 1-layer convolutional neural network model. The model accuracy at epoch 40 was 0.4941, and the accuracy validation was 0.4312. It can be seen from the accuracy and validation of model accuracy using one layer that it is still very bad.

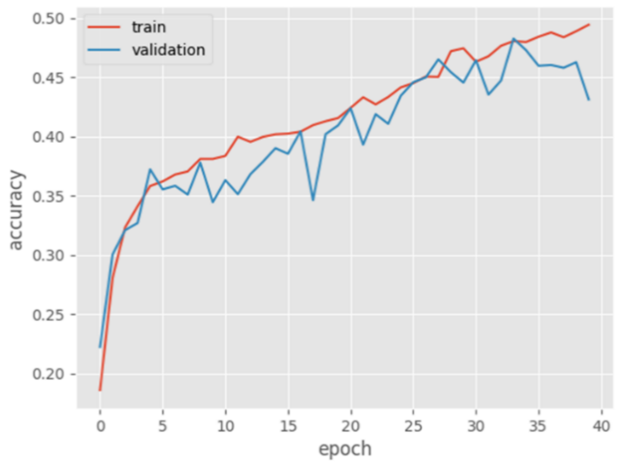


Figure 4. Accuracy Graph of CNN Layer 1 Model

In the Confusion Matrix, the number 7 was classified very well with 97 correct hits, while the number 0 was often mispredicted as 1, 4, or 9, with only 2 correct hits. In general, the model works quite well for certain numbers such as 7, but still struggles to distinguish numbers such as 0, 6, and 9. To improve performance, the model needs to be improved, for example by improving the training data or resetting the parameters.

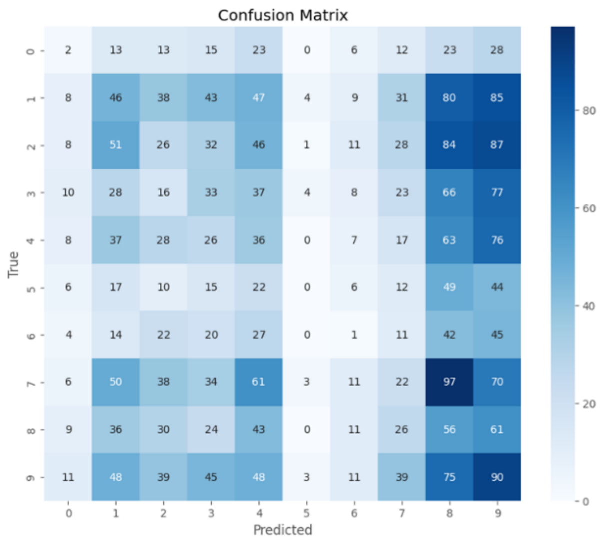


Figure 5. Confusion Matrix at 1 Layer

3.2. CNN Layer 2 Model Architecture Results

In the second experiment, two convolution layers were used. The first layer has 64 filters, a 3x3 kernel, ReLU activation, and an input shape (150, 150, 3) with Max pooling 2x2. The second layer uses 128 filters, a 3x3 kernel, and ReLU activation. After that, the Dense layer is applied with 128 neurons and ReLU activation.

Figure 6 shows a model accuracy graph from the results of experiments carried out on the 2-layer convolutional neural network model. The model accuracy at epoch 40 was 0.6777, and the accuracy validation was 0.6585. This can be seen from the accuracy and validation of model accuracy using two layers, which is better than experiments on one layer.

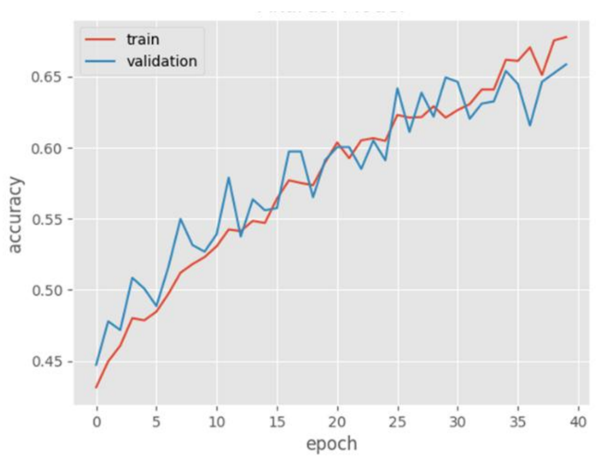


Figure 6. Accuracy Graph of CNN Layer 2 Model

On the Confusion Matrix, the number 3 has the most correct predictions with 139 data, indicating the model recognizes this number well. However, numbers such as 0 are often incorrectly predicted as 3 22 times, and the number 7 is also often incorrect as 3 22 times. This shows the model still has difficulty distinguishing certain numbers

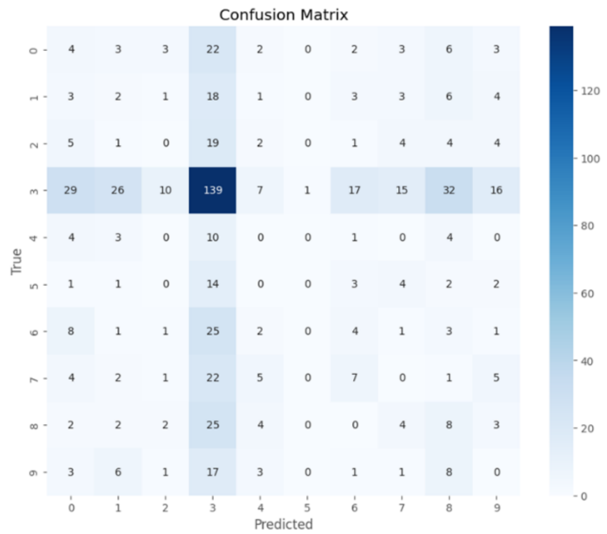


Figure 7. Confusion Matrix at 2 Layer

3.3. CNN Layer 3 Model Architecture Results

This experiment continues the first and second layer experiments by adding a third convolution layer. The third layer has 512 filters, a 3x3 kernel, and a ReLU activation function. In addition, a Dense layer with 512 neurons and ReLU activation is applied.

Figure 8 is a model accuracy graph from the results of experiments carried out on the 3-layer convolutional neural network model. The results of model accuracy at epoch 40 are 0.8903, and also a validation of accuracy of 0.8843. It can be seen that the accuracy and validation of model accuracy using three layers are better than experiments on one layer and on two layers.

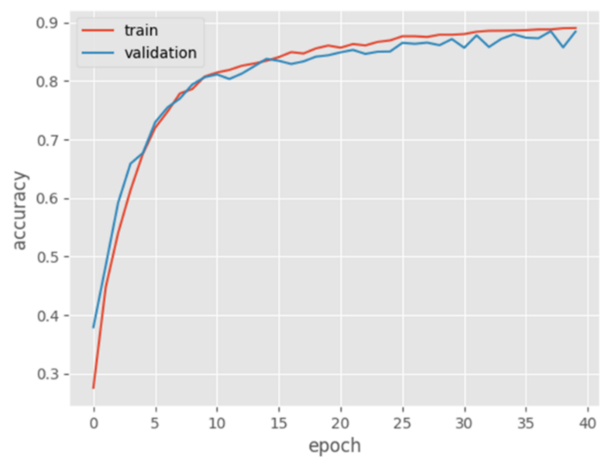


Figure 8. Accuracy Graph of CNN Layer 3 Model

In the Confusion Matrix, the number 2 had 60 correct predictions, while the number 9 was often incorrectly predicted as 7 54 times. The number 7 also has a high correct prediction of 66, but is often misrecognized as 1 or 9. This result shows that the model still has difficulty distinguishing similar numbers, such as 7 and 9.

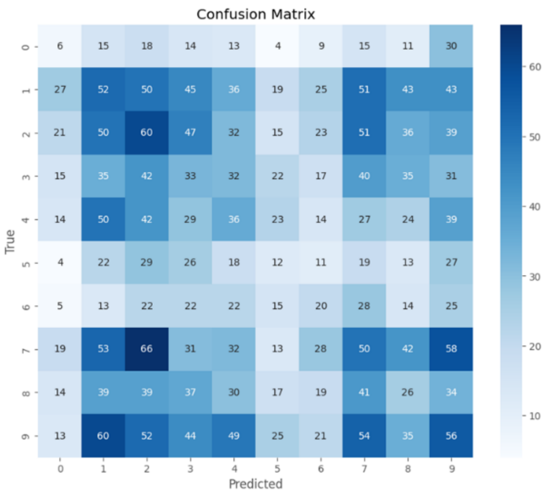


Figure 9. Confusion Matrix at 3 Layer

3.4. CNN Layer 4 Model Architecture Results

This experiment adds four convolution layers, with the number of filters 64, 128, 512, and 1024, 3x3 kernel size, and ReLU activation function, respectively. In addition, a Dense layer with 512 neurons and a ReLU activation function is applied.

Figure 10 shows a graph of model accuracy from the results of experiments carried out on the 4-layer convolutional neural network model, with the results of model accuracy at epoch 40 of 0.9241 and a validation accuracy of 0.9133. This can be seen from the accuracy and validation of the model using four layers, which is better than the previous experiment. This model of four layers can already be used to classify numbers in handwriting.

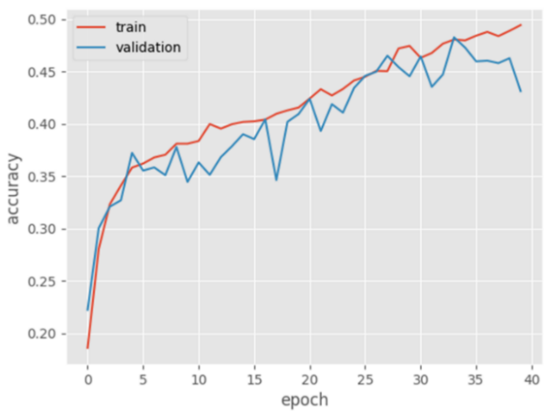


Figure 10. Accuracy Graph of CNN Layer 4 Model

Furthermore, the model with four layers has confusion matrix results of 9.35 precision, 9.35 recall, and 9.35 f1-score. Figure 11 shows a confusion matrix, which is the result of predicting the testing dataset that has been divided previously. It gets very good test results.

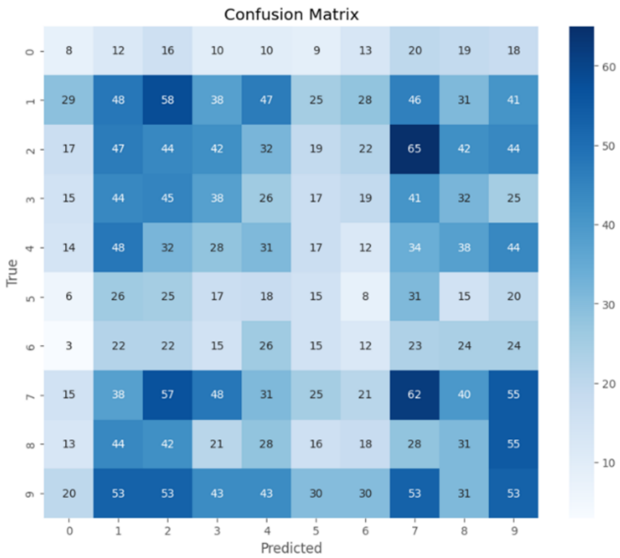
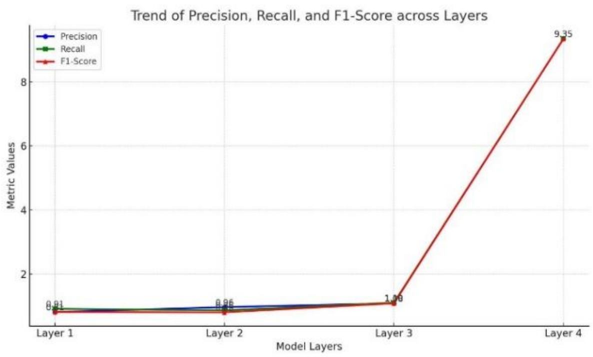


Figure 11. Confusion Matrix at 4 Layer

3.5. Analysis



Graph 1. Confusion Matrix Model Results

Graph 1 compares the performance of four different layers in detecting handwritten numerals.The improvement in metrics such as Precision, Recall, and F1-Score from Layer 1 to Layer 4 is due to the hyperparameter tuning process, such as setting the learning rate, number of filters, and batch size, which helps the model to learn more optimally. For example, Layer 1 has 64 filters, Layer 2 has 128 filters, Layer 3 has 512 filters, and Layer 4 has 1024 filters or kernels. Thus, it can be concluded that the hyperparameter tuning process plays a role in improving the Precision, Recall, and F1-Score metrics. This can be seen from the high accuracy, precision, recall, and f1-score values, which are much higher than those of other layers.

It can be concluded that the increase in the number of layers is directly proportional to the increase in the number of neurons in the filter, so the model becomes more specific in seeing data during training. This shows that the model with four layers can identify numbers with excellent accuracy and consistency, indicating that this model is very effective in classifying numbers in handwriting.

Then, the model with four layers that show the best performance is saved in the h5 model format to be implemented on the website.

3.6. Best Model Results on Website

In Figure 12, the improved model detected all tested number categories: 0 to 9. This shows that the modified model can accurately identify handwritten numbers from uploaded photos.

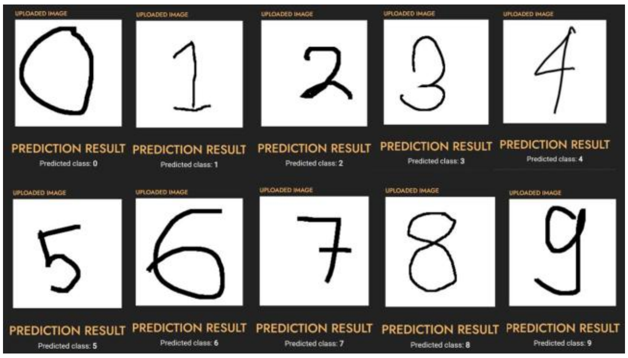


Figure 12. Number Prediction Results of the Best CNN Model

4. Conclusion

From this research, it is concluded that increasing the number of layers in the Convolutional Neural Network (CNN) model significantly improves the accuracy in classifying numbers in handwriting, especially for numbers 0-9. Experimental results show that models with more layers tend to provide higher accuracy. In particular, the model with four layers showed the best accuracy of 92.41%, with very high precision, recall, and f1-score performance for all digits 0-9. Adding layers can improve the model's ability to recognize and accurately classify numbers.

In addition, the CNN model implemented in this study showed excellent performance in detecting numbers in handwriting for digits 0-9. The 4-layer model, which was implemented into a website, was able to provide very accurate number predictions from uploaded handwriting images. The confusion matrix of the test results shows that the model has high precision, recall, and f1-score for all digits 0-9, indicating that the model is effective and reliable in handwritten number classification. Implementing this model on the website also shows that the system can be used practically and provide accurate real-time results.

References

[1]. Susim, T. & Cahyo D. (2021). Jurnal Syntax Admiration. Pengolahan Citra Untuk Pengenalan Wajah (Face Recognition) Menggunakan OpenCV, 2(3), 535.

[2]. Ratna, S. (2020). Jurnal Ilmiah Technologia. Pengolahan Citra Digital Dan Histogram Dengan Python Dan Text Editor PhyCharm, 11(3), 181.

[3]. Efrian, M. & Latifa, U. (2022). Jurnal POLEKTRO: Jurnal Power Elektronik. Image Recognition Berbasis Convolution Neural Network (CNN) Untuk Mendeteksi Penyakit Kulit Pada Manusia, 11(1), 278.

[4]. Purba et al. (2022). Jurnal Ilmiah Teknik Informatika. Perancangan Alat Pendeteksi Kematangan Buah Nanas Dengan Menggunakan Mikrokontroler Dengan Metode Convolutional Neural Network (CNN), 2(1), 13-21.

[5]. Mor et al. (2019). International Journal of Engineering and Advanced Technology (IJEAT). Handwritten Text Recognition: With Deep Learning and Android, 8(2S2), 172-178.

[6]. Kumar, S.A., Swarnalatha, S. & Babu, B.Shoban. (2021). International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET). Hand Written Character Recognition Using CNN and PSO Techniques, 10(2), 974-978.

[7]. Daniel & Irfan Maliki. (2020). Pengenalan Tulisan Tangan Menggunakan Metode Hidden Markov Model.

[8]. Safitri, K A. & Wulanningrum, R. (2020). Aplikasi Pengenalan Pola Tulisan Tangan Menggunakan Metode Support Vector Machine.

[9]. Viswanata et al. (2023). International Journal of Innovative Research in Computer and Communication Engineering. Handwritten Digit Recognition Using CNN, 11(1).

[10]. Alpaydin, E (2010), Introduction to Machine Learning, London, MIT Press.

[11]. Sidik, A.D. & Ansawarman, A. (2022). Formosa Journal of Multidisciplinary Research (FJMR). Prediksi Jumlah Kendaraan Bermotor Menggunakan Machine Learning, 1(3), 559-568.

[12]. Saputra, D. (2024). Jurnal JISSI. Upaya Pendidikan Menggunakan Machine Learning, 1(1), 57-62.

[13]. Demirkaya, K.G., & Cavusoglu, U. (2022). Academic Platform Journal of Engineering and Smart Systems (APJESS). Handwritten Digit Recognition With Machine Learning Algorithms, 10(1), 9-18.

[14]. Andika, L.A., Pratiwi, H & Handajani, S.S. (2019). Indonesian Journal of Statistics and Its Applications. Klasifikasi Penyakit Pneumonia Menggunakan Metode Convolutional Neural Network Dengan Optimasi Adaptive Momentum, 3(3), 331-340.

[15]. Prakosa, Andhika Bagas, Hendry & Radius Tanone. (2023). Jurnal Implementasi Model Deep Learning Convolutional Neural Network (CNN) Pada Citra Penyakit Daun Jagung Untuk Klasifikasi Penyakit Tanaman, 6(1), 2621-1467.