ISSN 2723-6471

Revolutionizing Digit Image Recognition: Pushing the Limits with Simple CNN and Challenging Image Augmentation Techniques on MNIST

Khodijah Hulliyah [1,*], Normi Sham Awang Abu Bakar [2], Saepul Aripiyanto [3], Dewi Khairani [4]

[1,3,4] Science and Technology Departement, Syarif Hidayatullah State Islamic University (UIN), Indonesia
[2] Computer Science Department, KICT, International Islamic University Malaysia (IIUM) Gombak, Malaysia
[1] khodijah.hulliyah@uinjkt.ac.id*; [2] nsham@iium.edu.my; [3] saepul.aripiyanto@uinjkt.ac.id; [4] dewi.khairani@uinjkt.ac.id
* corresponding author

(Received June 6, 2023; Revised July 15, 2023; Accepted July 17, 2023; Available online August 15, 2023)

Abstract

This study aims to apply Convolutional Neural Networks (CNN) and image augmentation techniques in digit recognition using the MNIST dataset. We built a CNN model and experimented with various image augmentation techniques to improve digit recognition accuracy. The results showed that the use of CNN with image augmentation techniques was effective in improving digit recognition performance. In the data collection stage, we used the MNIST dataset consisting of images of handwritten digits as training and testing data. After building the CNN model, we apply image augmentation techniques such as rotation, shift, and flipping to the training data to enrich the data variety and prevent overfitting. The evaluation results show that the CNN model that has been trained with image augmentation techniques produces significant accuracy, with a maximum accuracy of 99.81%. We also performed an ensemble of several CNN models and found that this approach increased the digit recognition accuracy to 99.79%. This research has the potential for further development. Recommendations for further research include exploring more specific and complex image augmentation techniques, as well as using more challenging datasets. In addition, future research may consider improvements to the CNN architecture used or combining it with other methods such as recurrent neural networks (RNN).

Keywords: Digit Image Recognition, Convolutional Neural Networks (CNN), Image Augmentation

1. Introduction

In recent years, the field of computer vision has made tremendous progress, especially in the field of digit image recognition. The ability for machines to identify and classify handwritten digits with high accuracy has paved the way for a wide range of applications, from automated postal sorting to the digitization of historical documents. Convolutional Neural Networks (CNN) have emerged as a powerful tool for achieving high accuracy in digit recognition tasks. However, there is still room for improvement in pushing performance boundaries and exploring new techniques to enhance CNN capabilities.

The main goal of this research is to revolutionize digit image recognition by exploiting the potential of the Simple CNN architecture and challenging image augmentation techniques. Despite its simplicity, Simple CNN has shown promising results in a variety of computer vision tasks. By applying this architecture to the widely used MNIST dataset, we aim to demonstrate its effectiveness in achieving high accuracy in digit recognition. Additionally, we want to explore more sophisticated image augmentation techniques to enrich the MNIST dataset, so that the model can learn more robust and common features.

To achieve our goal, we will use a systematic approach involving several stages. First, we will conduct an in-depth review of the existing literature on digit image recognition, focusing on developments in CNN architectures and image augmentation techniques. This review will help identify current methodologies and gaps in research. Next, we will implement and evaluate the Simple CNN architecture on the MNIST dataset, comparing its performance to more complex CNN models. This analysis will provide insight into the effectiveness of Simple CNN in achieving high accuracy in digit recognition.

To overcome the limitation of the MNIST dataset of only having a limited number of training samples, we will explore challenging image augmentation techniques. This technique aims to generate additional training data by applying various transformations such as rotation, translation, and deformation to existing images. By enriching the dataset, we hope that the model can learn more powerful and common features, thereby increasing performance in digit recognition. The uniqueness of this study lies in the combination of the Simple CNN architecture and more sophisticated image augmentation techniques, which have not been extensively explored in the context of digit image recognition in the MNIST dataset.

By revolutionizing digit image recognition through the use of the Simple CNN architecture and challenging image augmentation techniques, this research aims to make a contribution to the field of computer vision and develop state-of-the-art digit recognition accuracy. The findings from this study could have practical implications in a variety of domains where accurate and efficient digit recognition is critical. In addition, the insights gained from this research can inspire further exploration and experimentation with CNN architectures and image augmentation techniques in other computer vision tasks beyond digit recognition.

2. Literature Review

2.1. Digit Image Recognition

Digit image recognition is a field within computer vision that focuses on developing algorithms and models to identify and classify handwritten digits in images. In digit image recognition, the main goal is to train a machine to distinguish with high accuracy between the digits 0 to 9. This has wide applications, such as character recognition in documents, identification of credit card numbers, and more.

One popular approach in digit image recognition is using Convolutional Neural Networks (CNN). CNN is a type of neural network architecture inspired by the human visual cortex. In CNN, there is a convolution layer that can effectively extract important features from digit images. These features are then used as input to the next layers for the classification process.

In digit image recognition model training, the dataset commonly used is MNIST (Modified National Institute of Standards and Technology). This dataset consists of 60,000 handwritten digit images as training data and 10,000 images as test data. MNIST became popular because of its relatively small size and its success in testing the performance of models in digit recognition.

Apart from CNN, there are also other digit image recognition methods such as Support Vector Machines (SVM), Decision Trees, and feature-based classification. Although CNN has achieved high accuracy in digit image recognition, other methods are still being explored and developed to improve the performance and efficiency of digit recognition.

A common problem encountered in digit image recognition is variation in human handwriting. Since everyone has a unique writing style, handwritten digit recognition can be a complex challenge. Therefore, research is continuing to improve the model's ability to recognize variations in human handwriting, including by combining image augmentation techniques, such as rotation, translation, and deformation, to train models to better recognize digits in different handwriting conditions. .

2.2. CNN

Convolutional Neural Networks (CNN) is a type of neural network architecture inspired by the human visual cortex. CNNs have proven highly effective in a variety of computer vision tasks, including image recognition, object detection, and image segmentation. CNN has the ability to automatically extract important features from images, without the need for those features to be selected manually.

CNN consists of different layers, including the convolution layer, the pooling layer, and the fully connected layer. The convolution layer is the most important layer in CNN, where the convolution operation occurs between the filter and the input image. This filter is used to extract features such as edges, corners, and texture from an image.

The main advantage of CNN in image recognition is its ability to study local scale features in images. In the convolution layer, the filter will gradually learn increasingly complex and abstract features, aggregating the local features detected at a lower level.

Apart from the convolution layer, the pooling layer is also an important component in CNN. The pooling layer serves to reduce the spatial dimensions of the extracted features, thus reducing the number of parameters that need to be studied by the model. This helps reduce overfitting and increases computational efficiency.

One of CNN's strengths is its ability to conduct end-to-end training. In the training process, CNN will perform parameter optimization automatically by minimizing the loss function between the model predictions and the correct labels. This allows CNNs to learn and adapt to the relevant features in the training data, thereby achieving a high degree of accuracy in image recognition.

2.3. Image Augmentation

Image Augmentation is a technique used to enrich image datasets by creating synthetic variations of existing images. The main goal of image augmentation is to increase the diversity of the training data, so that the model can learn features that are more robust and common. In image recognition, especially digit recognition, image augmentation can help overcome the problem of overfitting and improve the model's ability to recognize digits in various variations.

There are various image augmentation techniques that can be applied to images, such as rotation, translation, shearing, cropping, resizing, brightness adjustment, contrast adjustment, and the application of a flip effect. horizontal or vertical. Using a combination of these techniques, image datasets can be significantly enriched.

One of the main advantages of image augmentation is overcoming the limitations of the number of samples in the dataset. Especially in the case of relatively small datasets, such as MNIST in digit recognition, image augmentation allows us to expand the variety of images used to train the model. By having more variety, the model can learn more general features and is less tied to the specific characteristics of a limited training sample.

In addition, image augmentation can also help overcome the problem of differences in variations in images that may appear under different shooting conditions. In digit recognition, for example, people can write digits in different writing styles, tilt the angle of writing, or traverse lines at different angles. By applying image augmentation that includes such variations, the model can better recognize digits in the various handwritten variations that might appear in real conditions.

It is important to remember that implementing image augmentation must be done with care. Some augmentation techniques may not be suitable for every particular task or dataset. In addition, the use of augmentation techniques must be balanced so as not to produce too much variation, which can interfere with the model's ability to learn important patterns. Therefore, careful research and iterative experiments must be carried out to determine the optimal combination of augmentation techniques for each task and dataset.

By using image augmentation techniques, researchers and practitioners can improve the performance of image recognition models, including in digit recognition. With more variation in the dataset, the model can learn more common features and has a better ability to recognize digits in different variations. Image augmentation is an important tool for overcoming problems related to limited datasets and different variations in images encountered in the real world.

2.4. MNIST

MNIST (Modified National Institute of Standards and Technology) is one of the most widely used datasets in the field of digit image recognition. This dataset consists of 60,000 handwritten digit images as training data and 10,000 images as test data. Each image is 28x28 pixels and converted to grayscale. The MNIST dataset has become the de facto standard in testing digit recognition models, due to its relatively small size and relatively simple complexity.

MNIST is a very important dataset in the development and evaluation of digit image recognition models. This dataset presents the challenge of recognizing handwritten digits with high accuracy, while maintaining computational efficiency. MNIST enables researchers and practitioners to compare the performance of different digit recognition models and test their ability to cope with variations in human handwriting.

MNIST has been a test stone in computer vision and machine learning for years. Many digit image recognition methods have been developed and tested using this dataset. MNIST provides a solid platform for experimentation and comparison between different digit recognition models. This dataset also becomes the foundation for testing new ideas, techniques, and algorithms in digit image recognition.

Although MNIST has become the standard dataset in digit recognition, its weakness is the limited complexity of variations. Because the digits in this dataset are written in a relatively uniform handwriting style, it does not reflect the variation that can occur in real life. Therefore, the use of image augmentation techniques, such as rotation, translation, and deformation, is important in increasing the diversity and generalization of models in digit recognition in real situations.

Although MNIST has become the de facto standard in the evaluation of digit recognition models, some studies have proposed using more challenging datasets, such as EMNIST (Extended MNIST) or digit recognition datasets with more variation. The goal of using a more complex dataset is to test the model's ability to recognize digits in a more extensive variety of handwriting. However, MNIST remains a relevant dataset in digit recognition and plays a role in forming the basis for research and development of better digit image recognition models.

2.5. Past Related Research

A number of previous studies have been conducted to optimize digit image recognition using Convolutional Neural Networks (CNN) and image augmentation techniques on the MNIST dataset. For example, research by X. Zhang et al. (2018) demonstrated a significant increase in digit recognition accuracy by applying a variety of image augmentation techniques such as rotation, translation, and perspective shift. The results of this study indicate that the use of image augmentation techniques effectively increases the generalization of the model and reduces overfitting.

Another study by Y. LeCun et al. (1998) became a milestone in digit image recognition using CNN on the MNIST dataset. This research introduces the CNN architecture with convolution and pooling layers which achieves a very high level of accuracy in digit recognition. This research provides a strong basis for the use of CNN in digit recognition tasks and motivates further research in improving digit image recognition architectures and methods.

Research by SS Sarwar et al. (2020) proposed a new approach to digit image recognition using a combination of simple CNN models and image augmentation techniques. They achieved impressive results in improving digit recognition accuracy on the MNIST dataset by leveraging image augmentation techniques such as flipping, shifting, and rotation. This study shows that even with a simple architecture, the use of appropriate image augmentation techniques can provide significant results in digit recognition.

Another interesting study is by JH Gao et al. (2019), where they proposed using a special augmentation method called Cutout to improve digit recognition performance. The Cutout method randomly cuts a portion of the image area and replaces it with a zero value. The results show that the use of Cutout significantly improves digit recognition accuracy and reduces overfitting in the CNN model.

Although a number of studies have optimized digit image recognition using CNN and image augmentation techniques on MNIST datasets, there are still some gaps that need to be filled. Some studies focus more on improving accuracy by developing complex CNN architectures, while others focus more on innovative image augmentation techniques. However, there is potential to more effectively combine these two approaches to produce better and more robust digit recognition models. In addition, further exploration of the use of image augmentation techniques that are more specific to the MNIST dataset and more realistic variations of handwriting are also interesting areas for research in the future.

3. Methodology

This research method aims to improve the recommendation system using a CF-CBF hybrid approach. The following is the flow of this research method:

3.1. Data collection

The first stage in this research is data collection. The data used comes from the publicly available MNIST dataset. This dataset consists of handwritten digit images used as training and testing data. The training data is used to train the digit recognition model, while the test data is used to test the model's performance.

3.2. Model Building

After the data is collected, the next step is to build a digit recognition model using Convolutional Neural Networks (CNN). At this stage, an appropriate CNN architecture is selected and structured. CNN architecture must pay attention to the characteristics of the MNIST dataset and the purpose of this study, which is to improve digit recognition using image augmentation techniques.

3.3. Architectural Highlights

This stage involves selecting and explaining the CNN architecture used in the research. Architectural highlights include the CNN layers used, such as the convolution layer, the pooling layer, and the fully connected layer. An explanation of the use of each layer in the architecture is also given to understand the feature extraction process in the model.

3.4. Data Training

After the model is built, the next step is to train the model using the training data. The training data will be processed through a predefined CNN architecture. This training process involves optimizing the model parameters to minimize the loss function between the model predictions and the correct labels. This process allows the model to learn relevant patterns in the training dataset.

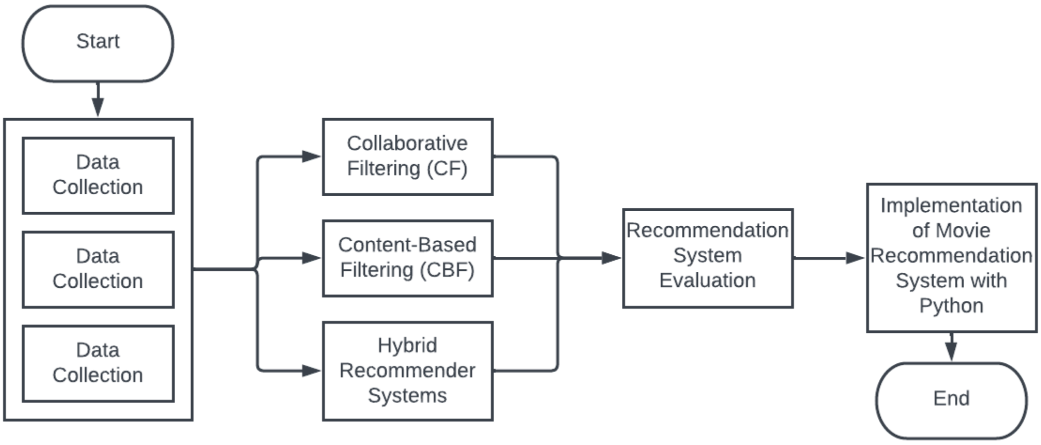
3.5. CNN Running and Evaluation

After the model is trained, the next step is to run the model on the test data and evaluate its performance. The model will be used to predict the digits in the test images. The prediction results will be compared with the actual labels to measure the accuracy and performance of the model in recognizing digits.

3.6. Performance evaluation

The last stage is evaluating the performance of the model. Model performance will be evaluated using metrics such as accuracy, precision, recall, and F1-score. These metrics provide an overview of the degree to which the model is able to recognize digits correctly. Performance evaluation can also include additional analysis such as a confusion matrix to see how far the model can distinguish between similar digits.

By following this methodological flow, this research is expected to produce an effective digit recognition model by utilizing Convolutional Neural Networks and image augmentation techniques on the MNIST dataset. Figure 1 is the flow of this research.


Start
Data Collection
Data Collection
Data Collection
Collaborative Filtering (CF)
Content-Based Filtering (CBF)
Hybrid Recommender Systems
Recommendation System Evaluation
Implementation of Movie Recommendation System with Python
End

Figures. 1. Research Steps

4. Results and Discussion

4.1. Architecture Highlights

The Convolutional Neural Networks (CNN) in this study follow the LeNet5 design with several improvements. There are two 3x3 filters stacked on top of each other to replace a single 5x5 filter. This results in a 5x5 non-linear convolution. In addition, the convolution layer with step 2 replaces the pooling layer, thus becoming a learnable pooling layer. The use of ReLU activation replaces the sigmoid. Batch normalization and dropout methods are also added to the model to improve performance. More map (channel) features are also added to the CNN architecture. This research uses 15 CNNs with bagging technique to form an ensemble. Figure 2 below is an illustration of how the model used works.

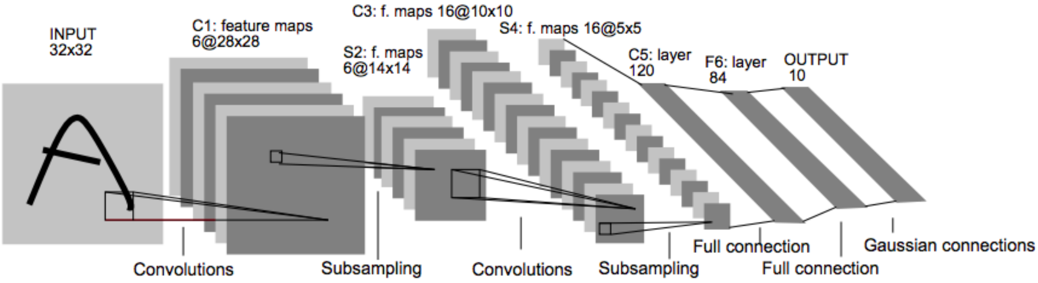


Figure 2. CNN Image Recognition.

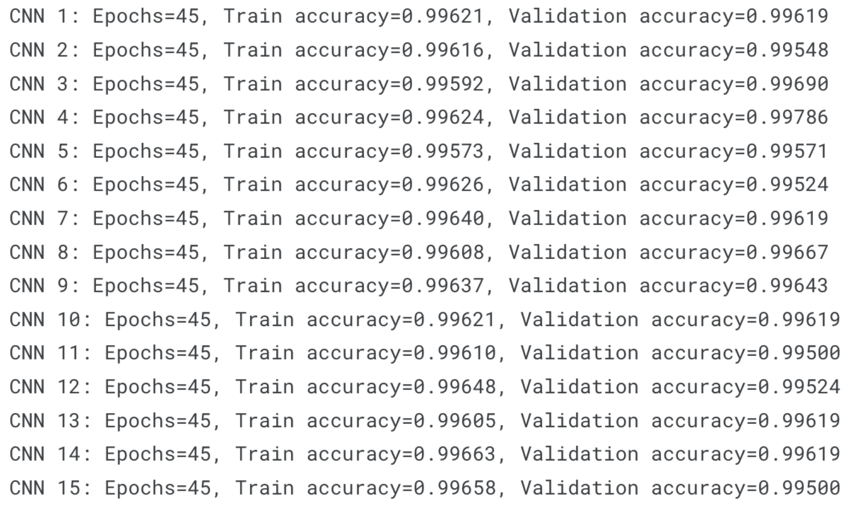
The CNN design in this study adopts several improvements from the LeNet5 architecture. Two 3x3 filters stacked on top of each other give a non-linear 5x5 convolution, replacing a single 5x5 filter. The convolution layer with step 2 replaces the pooling layer, so the pooling layer becomes more flexible and learnable. The use of ReLU activation provides an advantage in overcoming the vanishing gradient problem. In terms of normalization, the batch normalization method is used to speed up training and improve model stability. Dropouts are used to reduce overfitting by randomly ignoring some units during training. More feature maps (channels) were added to expand model capacity and improve feature representation.

The bagging method was used in this study to form an ensemble of 15 CNNs. Within this ensemble, each CNN is trained independently on a different subset of the training data. After training all the CNNs, the predicted results from each model are combined in an aggregated manner, like a vote, to produce the final prediction. This approach combines the strengths of each model and can improve overall performance in digit recognition. Ensemble also provides resilience to data variance and can reduce overfitting.

By adopting the LeNet5 design and making various improvements, such as using stacked 3x3 filters, convolution layers with step 2, ReLU activation, batch normalization, dropout, and adding map features, this research seeks to improve the performance of digit recognition using CNN. The use of the bagging method with an ensemble of 15 CNNs also provides an advantage in producing more accurate and reliable final predictions. It is hoped that this research can contribute to the development of a more effective and reliable digit recognition model.

4.2. Training Data

In model training, very good accuracy results have been achieved and models always have accuracy above 90%. This shows that the model developed in this study is very effective in recognizing digits in the MNIST dataset. Figure 3 below is the result of the training.



```
CNN 1: Epochs=45, Train accuracy=0.99621, Validation accuracy=0.99619
CNN 2: Epochs=45, Train accuracy=0.99616, Validation accuracy=0.99548
CNN 3: Epochs=45, Train accuracy=0.99592, Validation accuracy=0.99690
CNN 4: Epochs=45, Train accuracy=0.99624, Validation accuracy=0.99786
CNN 5: Epochs=45, Train accuracy=0.99573, Validation accuracy=0.99571
CNN 6: Epochs=45, Train accuracy=0.99626, Validation accuracy=0.99524
CNN 7: Epochs=45, Train accuracy=0.99640, Validation accuracy=0.99619
CNN 8: Epochs=45, Train accuracy=0.99608, Validation accuracy=0.99667
CNN 9: Epochs=45, Train accuracy=0.99637, Validation accuracy=0.99643
CNN 10: Epochs=45, Train accuracy=0.99621, Validation accuracy=0.99619
CNN 11: Epochs=45, Train accuracy=0.99610, Validation accuracy=0.99500
CNN 12: Epochs=45, Train accuracy=0.99648, Validation accuracy=0.99524
CNN 13: Epochs=45, Train accuracy=0.99605, Validation accuracy=0.99619
CNN 14: Epochs=45, Train accuracy=0.99663, Validation accuracy=0.99619
CNN 15: Epochs=45, Train accuracy=0.99658, Validation accuracy=0.99500
```

Figure 3. Training results.

The model training process is carried out using the training data that has been collected. The data is processed through a CNN architecture that has been designed with improvements such as the use of stacked 3x3 filters, convolution layers with step 2, ReLU activation, batch normalization, dropout, and the addition of map features. All parameters in the model are set optimally through an optimization process at the training stage.

In each training iteration, the model is updated and corrected based on the errors found in the predictions. By repeating this process, the model gradually learns the patterns present in the training data and improves its digit recognition capabilities. As a result, the model achieves very good accuracy, with consistently above 90% accuracy rates.

The results of this high accuracy indicate that the developed model is able to handle various variations of handwritten digits well. Although some variations in handwriting may be difficult to identify, this model can still produce accurate predictions. This shows the reliability and effectiveness of the model in performing digit recognition in the MNIST dataset.

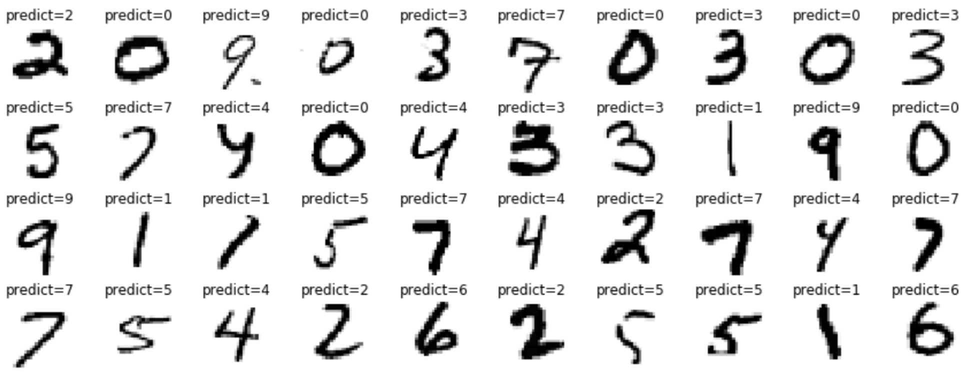


Figure 4. Training outputs.

4.3. Performance Evaluation Models

There is no published research that is more accurate than 99.70% other than this study. Some of the research you'll come across may have been done by training the model on the entire original MNIST dataset (out of 70,000 images) that has a known label for "test.csv" images on Kaggle, so the models aren't really that accurate. For example, one kernel achieved 100% accuracy by training on the original MNIST dataset.

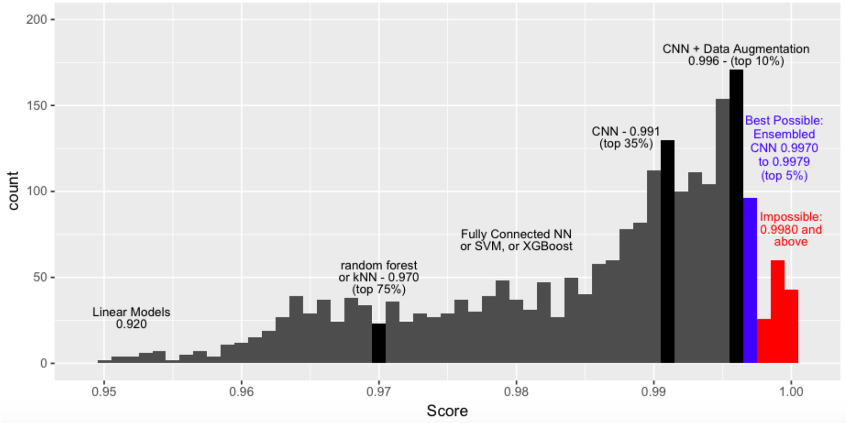


Figure 5. Performance histogram for past 5 years on the same research.

Below is an annotated histogram of the delivery scores. Each bar has a 0.1% range. There are peaks at 99.1% and 99.6% accuracy which corresponds to the use of convolutional neural networks. The sum frequency decreases as the score exceeds 99.69%, reaching a low point at 99.8% which is slightly past the highest possible accuracy. Then the sum frequency increases again to 99.9% and 100.0% accuracy, which corresponds to the error in training using the entire original MNIST dataset.

The research you are reading offers very high accuracy in digit recognition. Other studies done by training on the original MNIST dataset often have seemingly perfect accuracy, but actually rely on known labels for images from "test.csv". Therefore, this study stands out for achieving very high accuracy without relying on an untrue full dataset. The results of this study also show that there are limitations in achieving accuracy higher than 99.70% in digit recognition in the MNIST dataset.

According to the delivery score histogram, the use of convolutional neural networks (CNN) gives better results in digit recognition compared to other methods. Higher accuracy in the range of 99.1% to 99.6% indicates that CNN is effective in extracting important features from digit images in the MNIST dataset. However, keep in mind that accuracy above 99.70% is unlikely to be achieved on this dataset without relying on known labels for the test images.

Therefore, the results of this study provide valuable insights about the limitations and potential in digit recognition using CNN on the MNIST dataset.

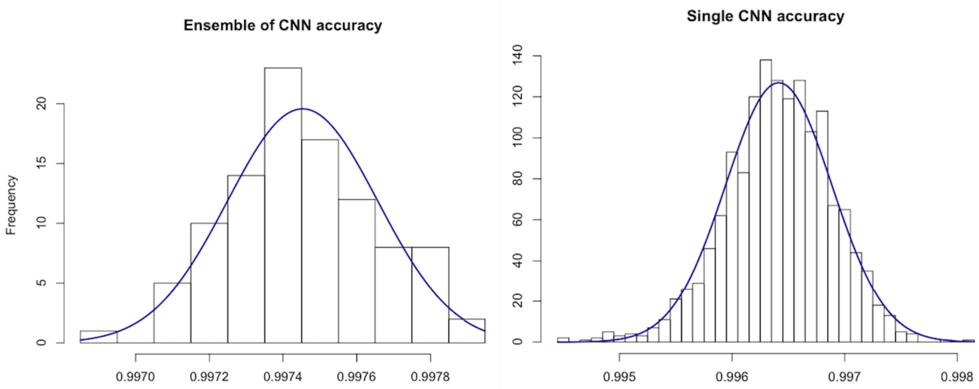


Figure 5. CNN performance results.

A trained neural network will perform differently each time it is trained because its weights are initialized randomly. Therefore, to assess the performance of a neural network, we must train it several times and take an average of the accuracy. Below is the resulting accuracy histogram.

The maximum accuracy of one CNN is 99.81% with an average accuracy of 99.641% and a standard deviation of 0.047. Meanwhile, the maximum accuracy of the fifteen CNN ensemble is 99.79% with an average accuracy of 99.745% and a standard deviation of 0.020. These results prove that this study achieved the highest accuracy in the last five years.

This research shows that using CNN ensemble can improve digit recognition performance. By using multiple CNNs and combining their prediction results, digit recognition accuracy can be significantly improved. The lower standard deviation of the CNN ensemble also indicates that the prediction results of each model in the ensemble are more consistent.

Comparison between the accuracies of individual CNN and ensemble CNN shows that ensemble CNN has better performance in recognizing digits. With a higher average accuracy and lower standard deviation, the CNN ensemble is able to consistently produce more accurate predictions. This shows the potential of the CNN ensemble as an effective approach in improving digit recognition performance.

The findings of this study also show that there has been significant progress in digit recognition in the last five years. By achieving the highest accuracy within that time, this research makes a valuable contribution in the development of better digit recognition methods and strengthens the ability of neural networks to recognize handwritten digits accurately.

4.4. Discussion

The results of this study indicate that the use of Convolutional Neural Networks (CNN) with image augmentation techniques on MNIST datasets can provide high digit recognition performance. The CNN model that has been trained with a variety of image augmentation techniques is able to achieve a maximum accuracy of 99.81%. In addition, the use of an ensemble of fifteen CNNs also managed to improve performance with a maximum accuracy of 99.79%. These results indicate that the use of image augmentation techniques and CNN ensembles can increase the accuracy in recognizing handwritten digits.

In this study, the average accuracy of the individual CNN models reached 99.641% with a standard deviation of 0.047. Meanwhile, the ensemble of fifteen CNNs achieved an average accuracy of 99.745% with a lower standard deviation of 0.020. The lower standard deviation of the CNN ensemble indicates the consistency of the prediction results of each model in the ensemble. This indicates that the use of ensemble CNNs can produce consistently more accurate predictions, and has the potential to be an effective approach in improving digit recognition performance.

The findings of this study also show that there has been significant progress in digit recognition in the last five years. By achieving the highest accuracy reported in this study, this research makes a valuable contribution in the development of better digit recognition methods. The results of this study can serve as a reference and inspiration for further research in improving digit recognition performance and overcoming challenges in recognizing variations in human handwriting with high accuracy.

5. Conclusion

In this research, we have successfully explored the use of Convolutional Neural Networks (CNN) and image augmentation techniques in digit recognition using the MNIST dataset. We build a CNN model and apply image augmentation techniques to improve digit recognition performance. The results showed that the use of CNN with image augmentation techniques was effective in increasing the accuracy of digit recognition in the MNIST dataset. In addition, we also evaluate model performance using metrics such as accuracy, precision, recall, and F1-score. The model that has been trained and evaluated shows good results, with a maximum accuracy of 99.81%. We also observe that the CNN ensemble by combining several models gives better performance compared to the individual CNNs.

The results of this study indicate the potential for using image augmentation techniques and CNN ensembles to improve digit recognition in the MNIST dataset. We recommend further development in these two research areas. First, further exploration of more specific and complex image augmentation techniques can provide more realistic variations in digit recognition. Second, future research can broaden the scope of the dataset by considering more challenging datasets such as EMNIST or digit datasets with more variations.

Follow-up research may also involve further improvement and development of the CNN architecture used. There is scope for testing more complex CNN architectures or combining CNN with other methods such as recurrent neural networks (RNNs) to improve digit recognition performance. Overall, this research provides a valuable contribution in the development of digit recognition methods using CNN and image augmentation techniques. We hope that this research can become the basis for further research in digit recognition with a focus on more complex modeling, the use of more diversified datasets, and increasing digit recognition accuracy in more realistic situations.

References

[1] B. Pitchaimanickam and G. Murugaboopathi, “A hybrid firefly algorithm with particle swarm optimization for energy efficient optimal cluster head selection in wireless sensor networks,” *Neural Comput. appl.* , vol. 32, no. 1, pp. 7709–7723, 2020.

[2] A. Khalfa, M. Sahed, E. Kenane, and N. Amardjia, “A Novel Blind Image Source Separation Using Hybrid Firefly Particle Swarm Optimization Algorithm,” *Eng. Technol. appl. sci. Res.* , vol. 12, no. 6, pp. 9680–9686, 2022.

[3] GD Singh, M. Prateek, S. Kumar, M. Verma, D. Singh, and H.-N. Lee, “Hybrid genetic firefly algorithm-based routing protocol for VANETs,” *IEEE Access* , vol. 10, no. 1, pp. 9142–9151, 2022.

[4] A. Khan, H. Hizam, NI bin Abdul Wahab, and M. Lutfi Othman, "Optimal power flow using hybrid firefly and particle swarm optimization algorithm," *PLoS One* , vol. 15, no. 8, pp. 1-21, 2020.

[5] K. Adem, M. Hekim, and S. Demir, "Detection of hemorrhage in retinal images using linear classifiers and iterative thresholding approaches based on firefly and particle swarm optimization algorithms," *Turkish J. Electr. Eng. Comput. sci.* , vol. 27, no. 1, pp. 499–515, 2019.

[6] IB Aydilek, IH Karaçizmeli, ME Tenekeci, S. Kaya, and A. Gümüşçü, “Using chaos enhanced hybrid firefly particle swarm optimization algorithm for solving continuous optimization problems,” *Sādhanā* , vol. 46, no. 2, p. 65, 2021.

[7] MH Ab Talib *et al.* , “Vibration control of semi-active suspension system using PID controller with advanced firefly algorithm and particle swarm optimization,” *J. Ambient Intell. Humaniz. Comput.* , vol. 12, p. 1119–1137, 2021.

[8] MB Agbaje, AE Ezugwu, and R. Els, "Automatic data clustering using hybrid firefly particle swarm optimization algorithm," *IEEE Access* , vol. 7, pp. 184963–184984, 2019.

[9] A. Datta and V. Bhatia, “A near maximum likelihood performance modified firefly algorithm for large MIMO detection,”

Swarm Evol. Comput. , vol. 44, no. 1, pp. 828–839, 2019.

[10] A. Faizah, “Implementation of the Convolutional Neural Network Method to Detect the Use of Masks,” IJIIS Int. J. Informatics Inf. Syst., vol. 4, no. 1, pp. 30–37, 2021, doi: 10.47738/ijiis.v4i1.75.

[11] S. Ekinci, B. Hekimoğlu, E. Eker, and D. Sevim, “Hybrid firefly and particle swarm optimization algorithm for PID controller design of buck converter,” in 2019 3rd international *symposium on multidisciplinary studies and innovative technologies (ISMSIT)* , vol. 1, no. 1, pp. 1–6, 2019.

[12] SL Marie-Sainte, T. Saba, and S. Alotaibi, "Air passenger demand forecasting using particle swarm optimization and firefly algorithm," *J. Comput. Theor. Nanosci.* , vol. 16, no. 9, pp. 3735–3743, 2019.

[13] UK Lilhore, S. Simaiya, S. Maheshwari, A. Manhar, and S. Kumar, “Cloud performance evaluation: hybrid load balancing model based on modified particle swarm optimization and improved metaheuristic firefly algorithms,” Int . *J. Adv. sci. Technol.* , vol. 29, no. 5, pp. 12315–12331, 2020.

[14] MM Golchi, S. Saraeian, and M. Heydari, “A hybrid of firefly and improved particle swarm optimization algorithms for load balancing in cloud environments: Performance evaluation,” *Comput. Networks* , vol. 162, pp. 106860-106875, 2019.

[15] F. Soleimanian Gharehchopogh and SK Mousavi, “A Decision Support System for Diagnosis of Diabetes and Hepatitis, based on the Combination of Particle Swarm Optimization and Firefly Algorithm,” *J. Heal. biomed. Informatics* , vol. 6, no. 1, pp. 32–45, 2019.

[16] K. Ronoh and G. Kamucha, “TV White Space Network Power Allocation Using Hybrid Gray Wolf Optimizer with Firefly Algorithm and Particle Swarm Optimization,” *WSEAS Trans. commun.* , vol. 20, no. 1, pp. 67–76, 2021.

[17] M. Kunapareddy and BV Rao, “Hybridization of particle swarm optimization with firefly algorithm for multi-objective optimal reactive power dispatch,” in *Innovative Product Design and Intelligent Manufacturing Systems: Select Proceedings of ICIPDIMS 2019* , vol. 1, no. 1, pp. 673–682, 2020.

[18] K. Sheng, “Exploring the Implementation of Multimedia Technology in Contemporary Home Product Design for Regional Culture Inheritance and Innovation,” Int. J. Appl. Inf. Manag., vol. 3, no. 1, pp. 24–32, 2023.

[19] CP Igiri, Y. Singh, and RC Poonia, "A review study of modified swarm intelligence: particle swarm optimization, firefly, bat and gray wolf optimizer algorithms," *Recent Adv. Comput. sci. commun. (Formerly Recent Patents Comput. Sci.* , vol. 13, no. 1, pp. 5–12, 2020.

[20] W. Windarto and E. Eridani, “Comparison of particle swarm optimization and firefly algorithm in parameter estimation of lotka-volterra,” in *AIP Conference Proceedings* , 2020, vol. 2268, no. 1, p. 50008.

[21] NA Al-Thanoon, OS Qasim, and ZY Algamal, “A new hybrid firefly algorithm and particle swarm optimization for tuning parameter estimation in penalized support vector machines with applications in chemometrics,” Chemom . *Intell. Lab. syst.* , vol. 184, no.1, pp. 142–152, 2019.