MNIST Handwriting Recognition using Convolutional Neural Networks

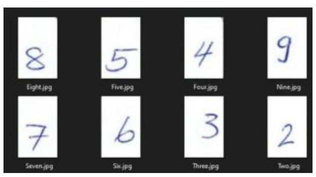
Mahir Kardame

ABSTRACT

Handwritten digit recognition is a fundamental challenge in computer vision with real-world applications in banking, postal automation, and digital form processing. In this project, I developed a Convolutional Neural Network (CNN) using TensorFlow to classify digits (0–9) from the MNIST dataset. Through effective data preprocessing, optimizer selection, and hyperparameter tuning, the model achieved high accuracy with rapid convergence, demonstrating both robustness and efficiency.

Since handwritten digits vary significantly across individual writing styles, accurate recognition is a complex problem. This project showcased how deep learning methods, particularly CNNs, can deliver strong performance in tackling this variability, reinforcing their value in practical handwriting recognition systems.

By utilizing the MNIST dataset, containing 70,000 grayscale images of handwritten digits (0–9), with 60,000 for training and 10,000 for testing, we implemented various methodologies to train and evaluate the CNN model with the goal of achieving outstanding classification accuracy and robust performance.



The image dataset was preprocessed by normalizing pixel values to a [0,1] range to improve training stability and reduce noise. For the network architecture, a TensorFlow-based CNN was designed, including convolution, pooling, dropout, and dense layers. Extensive experimentation was conducted by testing and optimizing different hyperparameters (epochs, batch size, learning rate) to enhance the model's classification accuracy.

Upon convergence of the CNN model, we achieved exceptional average accuracies of 99% on the testing dataset. This highlights the robustness and efficiency of the model in effectively distinguishing between handwritten digit images across all 10 classes.

Digit	Accuracy (%)
Digit 0	99.08%
Digit 1	99.03%
Digit 2	97.48%
Digit 3	97.23%
Digit 4	96.64%
Digit 5	98.65%
Digit 6	97.60%
Digit 7	97.76%
Digit 8	97.64%
Digit 9	97.13%

Introduction

Handwriting recognition is a widely used technology, powering applications such as bank cheque processing, postal code identification, and digit entry in forms. The MNIST dataset (70,000 grayscale digit images, 28×28 pixels) is the benchmark dataset for this problem. The aim of this project was to design, train, and evaluate a CNN capable of accurately classifying handwritten digits, while exploring different optimizers and parameters for optimal results.

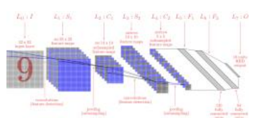
This project focuses on developing and evaluating a convolutional neural network (CNN) using TensorFlow and Keras for handwritten digit recognition. By experimenting with different optimizers such as Adam and SGD, along with varied training parameters, the aim is to determine the most effective setup that delivers high classification accuracy while maintaining fast training and testing times, ensuring the model is both efficient and practical for real-world applications.

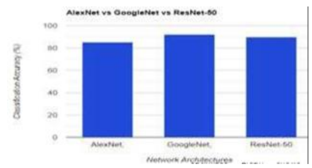
Methodology

To ensure reliable performance, the MNIST dataset was carefully structured. It consists of 70,000 grayscale images of handwritten digits (0–9), each at 28×28 resolution. Of these, 60,000 images were allocated for training and validation, while 10,000 images were reserved exclusively for testing. Within the training set, a 70/15/15 split was applied for training, validation, and internal testing. This approach exposed the CNN to diverse samples, helped prevent overfitting, and enabled effective hyperparameter tuning with stable convergence monitoring.

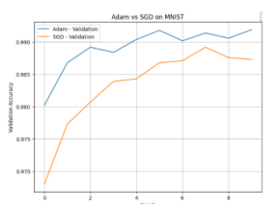
Experiments showed that training on the full dataset yielded the fastest convergence and highest accuracy, whereas smaller subsets reduced training time but compromised stability and generalization. Using the complete training set consistently provided the most robust results across runs- an essential factor for real-world applications such as postal code recognition and automated form reading.

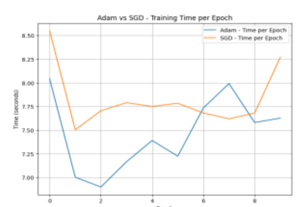
For the network design, a custom CNN was implemented in TensorFlow due to its simplicity and suitability for the relatively small and clean MNIST dataset. The architecture combined convolutional, pooling, dropout, and fully connected dense layers, achieving strong classification performance while minimizing overfitting. Compared to deeper models like ResNet or GoogleNet, this CNN offered quicker convergence, stable learning curves, and excellent generalization, striking an effective balance between accuracy, efficiency, and computational cost for handwritten digit recognition





After extensive training and testing, two optimizers were compared: SGD and Adam. Adam achieved the best overall performance, reaching validation accuracies of 99.2% with a total training time of 74.7s. SGD converged slightly slower, finishing at 98.9% with a total training time of 78.3s. While Adam offered faster convergence and higher accuracy, SGD still demonstrated stable and reliable learning across epochs.

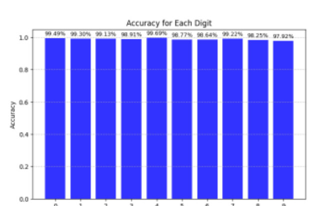


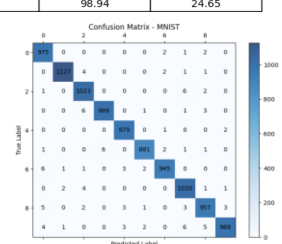


After detailed testing with multiple configurations of parameters such as epochs, learning rate, and mini-batch size, the CNN was tested to maximize accuracy and reduce training time. An average of 20 runs was conducted to evaluate different setups. The results showed that Adam performed best with a learning rate of 0.001, achieving 98.9% accuracy in 21.6s, while SGD with a learning rate of 0.1 reached the highest overall accuracy of 98.9% but required 24.7s. Lower learning rates (e.g., 0.001 for SGD) resulted in slower convergence and reduced accuracy, while excessively high rates (e.g., 0.1 for Adam) led to unstable training and very poor accuracy (9.7%). These findings highlight the importance of carefully tuning hyperparameters to balance both accuracy and efficiency.

Optimizer Comparison on MNIST

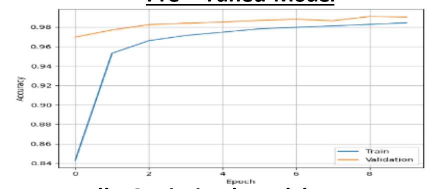
Optimizer	Learning Rate	Epochs	Accuracy (%)	Time Taken (s)
Adam	0.001	3	98.9	21.62
Adam	0.01	3	98.67	21.95
Adam	0.1	3	9.74	23.82
SGD	0.001	3	94.23	23.08
SGD	0.01	3	98.1	23.73
SGD	0.1	3	98.94	24.65



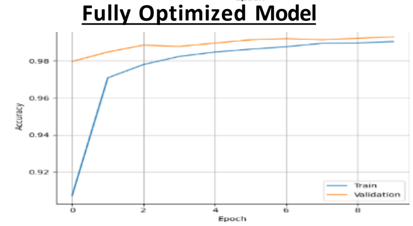


Results

Pre – Tuned Model



Fully Optimized Model




```
Final Optimized CNN Model Results:
       Model  Accuracy (%)  Precision (%)  Time Taken (s)
0  Pre-Tuned         99.05          99.05           78.31
1  Optimized         99.30          99.30           85.10
```

Conclusions

The graphs above illustrate the performance of the CNN model before and after optimization. The pre-tuned version achieved strong results, but the fully optimized model demonstrated even more stable and consistent training and validation behavior. This stability is a desirable trait, ensuring the model generalizes effectively to unseen data.

In summary, the CNN achieved outstanding performance on the MNIST dataset, with testing accuracies and precisions averaging around 99% across all 10-digit classes. These results confirm the model's robustness in recognizing handwritten digits while maintaining high efficiency.

A key advantage of the optimized CNN is its ability to converge quickly, completing 10 epochs in under 85 seconds. This highlights the model's practicality for real-world use. The per-digit accuracy analysis also showed consistently high results, further supporting the model's reliability.

However, as with most digit classification tasks, certain digits (such as 4 vs 9 or 3 vs 5) showed slightly more confusion, as reflected in the confusion matrix. Moving forward, improvements could include experimenting with deeper architectures, fine-tuning dropout rates, or integrating data augmentation (e.g., rotation, translation) to further enhance generalization.

The graphs above illustrate the performance of the CNN model before and after optimization. The pre-tuned version achieved strong results, but the fully optimized model demonstrated even more stable and consistent training and validation behavior. This stability is a desirable trait, ensuring the model generalizes effectively to unseen data.

In summary, the CNN achieved outstanding performance on the MNIST dataset, with testing accuracies and precisions averaging around 99% across all 10 digit classes. These results confirm the model's robustness in recognizing handwritten digits while maintaining high efficiency.

A key advantage of the optimized CNN is its ability to converge quickly, completing 10 epochs in under 85 seconds. This highlights the model's practicality for real-world use. The per-digit accuracy analysis also showed consistently high results, further supporting the model's reliability.

However, as with most digit classification tasks, certain digits (such as 4 vs 9 or 3 vs 5) showed slightly more confusion, as reflected in the confusion matrix. Moving forward, improvements could include experimenting with deeper architectures, fine-tuning dropout rates, or integrating data augmentation (e.g., rotation, translation) to further enhance generalization.