Component 3

1. Introduction

The objective is to accurately classify the handwritten digits in the MNIST dataset utilizing Convolutional Neural Network (CNN) architecture. The MNIST dataset can be accessed through the following URL: http://yann.lecun.com/exdb/mnis

2. Methodology

After pre-processing the MNIST dataset, a base CNN model was developed. Figure 1 and Table 1 present the architecture and key parameters of the base model. Different optimizers were tested to determine the best one for the analysis. Various CNN models were then trained and tested using different regularization methods (e.g., L1 and L2 regularization). The number of convolution blocks in the CNN architecture was also varied to examine the quantitative effect of these changes. The impact of different learning rates on the CNN algorithm's performance was also investigated. Finally, we checked for overfitting by comparing the model's performance on the training and test data.

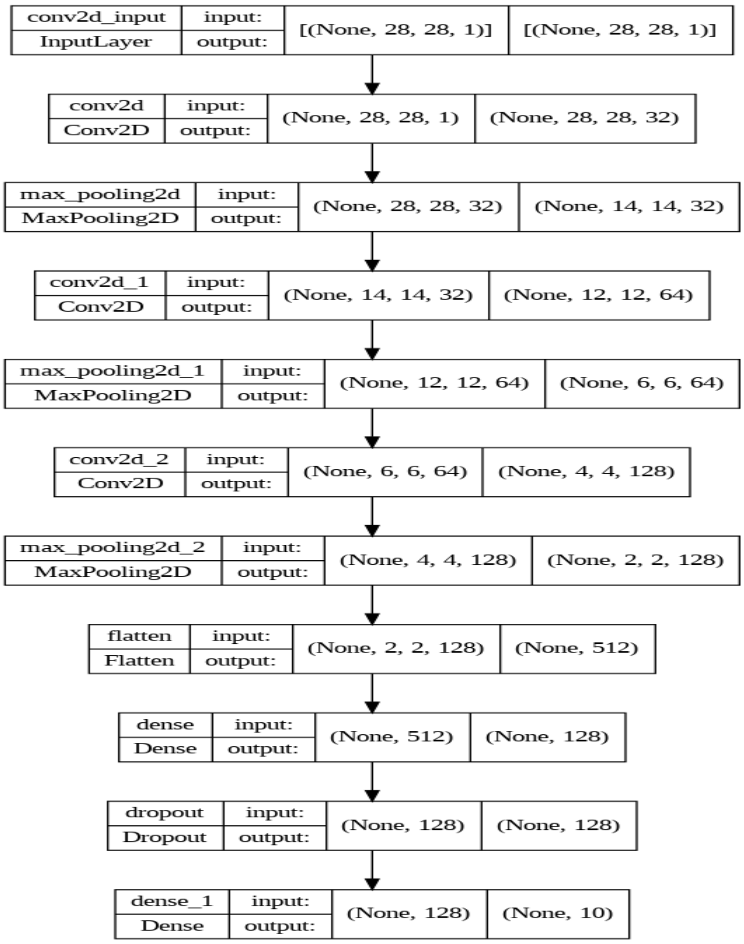

Fig.1 Architecture of the base model

Table 1: Parameters of the base model

Convolution layer	padding	Learning rate	Input layer activation	Output layer activation	Drop out	Optimizer
3	Same	0.001	Relu	Softmax	0.5	Adam

3. Result and discussion

3.1 Optimizer selection

An initial analysis was conducted to determine the optimal optimizer for the analysis. Based on the assessment of precision, recall, and F1-score metrics, as well as overall accuracy it was found that the Adam optimizer outperformed the RMSprop and SGD optimizers as shown in Fig1. The highest accuracy rate (98%) was achieved by the Adam model, with the RMSprop and SGD models achieving 95% and 92%, Thus, the Adam optimizer proved to be the most effective in performing the classification task.

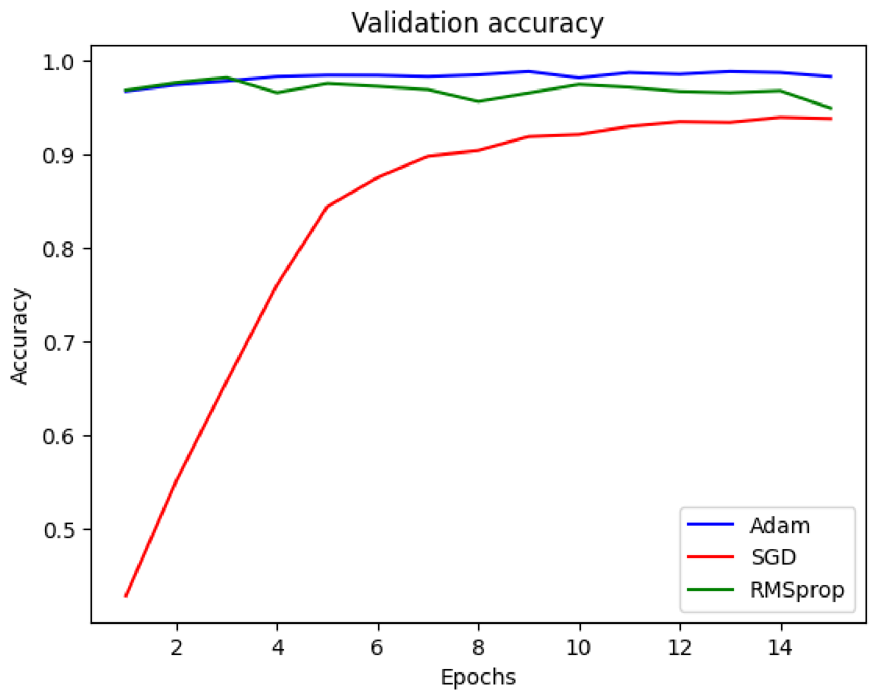

Fig.2 validation Accuracy of Adam SDG and RMSprop

Table 2. The effect of different regularization method on the performance of our CNN model

Regularization	Training		Validation	
	Loss	Accuracy	Loss	Accuracy
Drop out = 0.5	0.080	0.976	0.047	0.984
Drop out = 0.3	0.071	0.978	0.044	0.987
Drop out = 0.1	0.061	0.981	0.046	0.986
L1 regularization	0.475	0.928	0.352	0.962
L2 regularization	0.146	0.972	0.105	0.983
Batch normalization	0.059	0.982	0.042	0.988
Early stopping				
Patience = 7	0.053	0.983	0.046	0.986
Patience = 5	0.056	0.982	0.036	0.988
Patience = 3	0.048	0.984	0.034	0.989

Base model

As indicated in Table 2, the model's performance was enhanced by applying regularization techniques, except for L1 and L2 regularization, which marginally decreased training and validation accuracy compared to the base model. By increasing the dropout rate, the CNN model achieved better results, reducing from 0.5 to 0.3, leading to a slightly improved validation accuracy. However, further decreasing the rate to 0.1 did not significantly impact accuracy. Early stopping analysis revealed that patience values of 5 or 3 yielded the lowest validation loss and highest validation accuracy. Overall, the results suggest incorporating regularization or normalization methods into the model can enhance its performance, as shown in Fig 3.

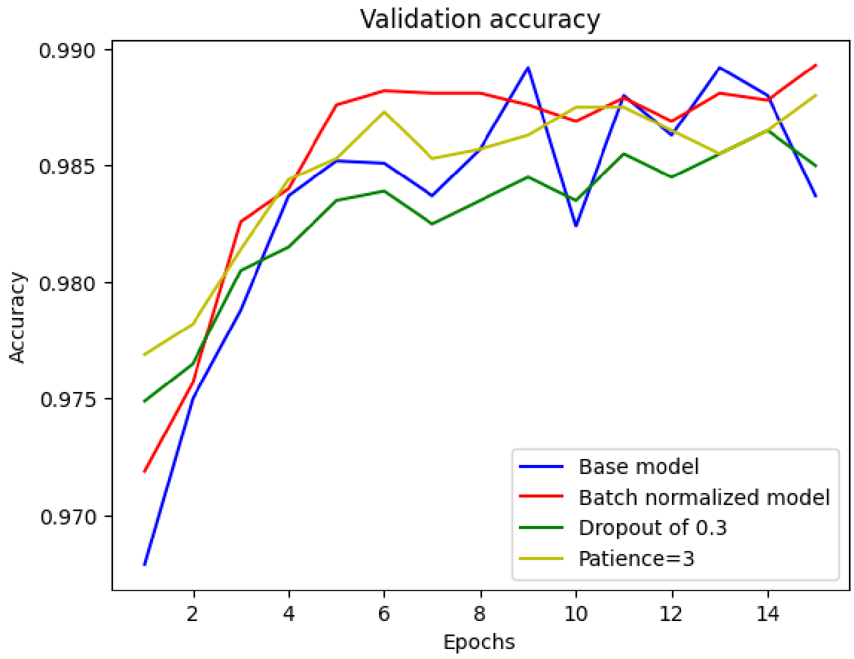

Fig 3. Effect of regularization on the performance on the base model

3.2 Effect of number convolutional layer on the CNN model

Compared to the three-layer model used as the base model, the four-layer model exhibited a significant improvement in both training and validation losses and training and validation accuracies. Specifically, the four-layer model demonstrated a 7.8% reduction in training loss and a 6.4% reduction in validation loss, along with a 0.2% increase in training accuracy and a 0.3% increase in validation accuracy. These improvements in model performance suggest that increasing the convolutional layers can lead to a more effective model for this particular task. However, it is worth noting that even the three-layer model achieved high accuracy and demonstrated satisfactory performance, indicating that it may be a viable option for certain use cases.

Table 3 Effect of the number of convolutional layers on the performance of the CNN model

Number of convolution block	Training		Validation	
	Loss	Accuracy	Loss	Accuracy
Four layers	0.071	0.978	0.044	0.987
Three layers	0.080	0.976	0.047	0.984
Two layers	0.121	0.964	0.044	0.984

Base model

3.3 Effect of learning rate

Table 4 shows that the base model with a learning rate of 0.001 achieved the highest accuracy and lowest loss among the models tested. On the other hand, models with a higher learning rate of 0.01 performed worse, possibly due to overshooting the optimal value. Similarly, models with a lower learning rate of 0.0001 demonstrated poorer performance, likely because they required more time to reach the optimal state. As shown in Fig,4 the model with a shorter learning rate converges faster as reach high accuracy relatively quickly.

Table 4 Effect of learning rate on the performance of the CNN model

Learning rate	Training		Validation	
	Loss	Accuracy	Loss	Accuracy
0.01	0.464	0.871	0.206	0.945
0.001	0.080	0.976	0.047	0.984
0.0001	0.543	0.913	0.232	0.935

Base model

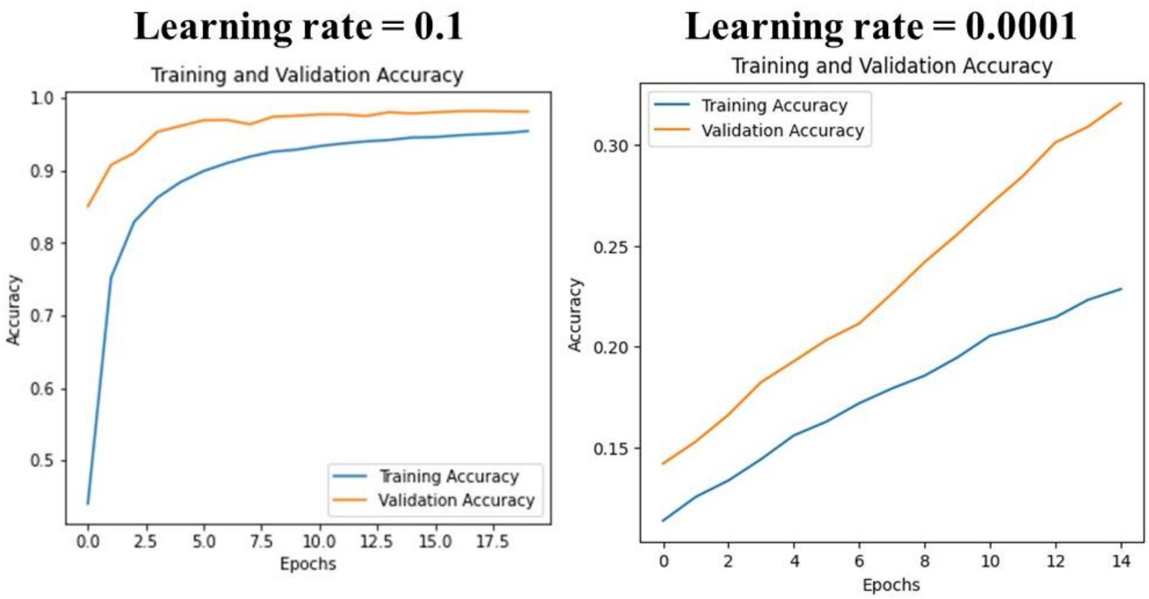

Fig.4 Model performance at different learning rate

3.4 Case of Overfitting

During the batch normalization process, some instances of overfitting were observed, where the training accuracy exceeded that of the validation dataset. This can be seen in the Fig 5, particularly at Epochs 2/15, 5/15, and 9/15, where the training loss was 0.1325, validation loss was 0.1165, training loss was 0.0848 and validation loss was 0.0901. Training loss was 0.0684, and validation loss was 0.0794, respectively. The final validation accuracy was still very high, meaning the model could generalize well despite the few instance of overfitting.

Fig 5. overfitting during batch normalization

4.Conclusion

Based on the results obtained, it was found that the CNN model demonstrated superior performance when the Adam optimizer was employed. The model's accuracy was further enhanced using batch normalization, dropout, and a patience value of 0.3. Additionally, the optimal number of convolution layers was found to be 3, and a learning rate of 0.001 was determined to be the most effective.