Dual CNNs for Real-Time Handwritten Digit Recognition via Webcam

By Syed Yasir Ahmed

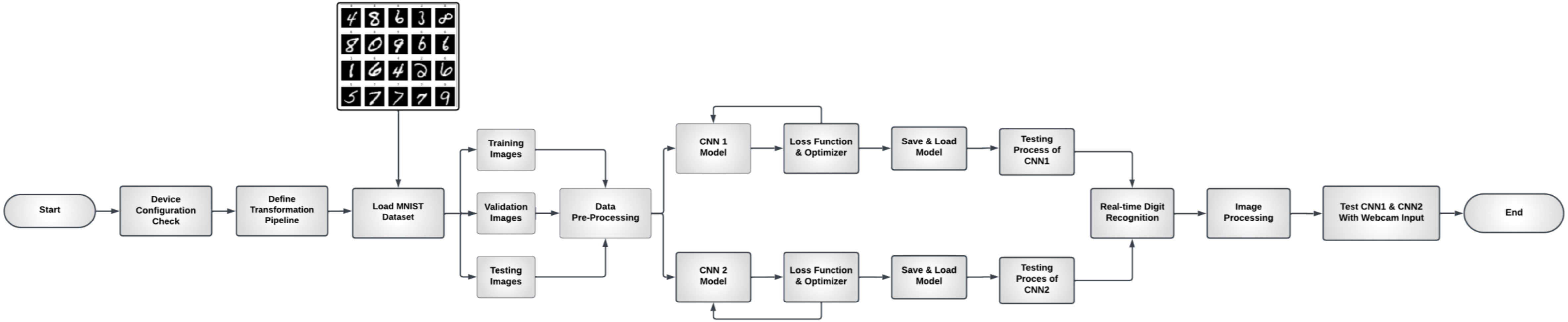
Introduction:

This project focuses on developing, training, and testing two CNN architectures, CNN1 and CNN2, to recognize handwritten digits from the MNIST dataset. It aims to compare the performance of these models. The dataset is augmented and normalized, then split into training, validation, and test sets with appropriate data loaders. CNN1 is a simpler model with two convolutional layers and max-pooling, while CNN2 is more complex with three convolutional layers, batch normalization, average pooling, and dropout. Both models are trained and validated using the Adam optimizer and Cross Entropy Loss. Additionally, image preprocessing functions enhance digit detection, and preprocessed images are fed into the trained models for prediction. For practical application, a webcam captures images, processes them, and highlights detected digits with their predicted values in real-time.

CNN-1 VS CNN-2

FEATURES	*CNN-1*	*CNN-2*
Number of Convolution Layers	Two	Three
First Convolution Layer	16 filters, kernel size 5x5, ReLU activation, Max Pooling	16 filters, kernel size 5x5, ReLU activation, Batch Normalization, Average Pooling
Second Convolution Layer	32 filters, kernel size 5x5, ReLU activation, Max Pooling	32 filters, kernel size 5x5, ReLU activation, Batch Normalization, Average Pooling
Third Convolution Layer	N/A	64 filters, kernel size 3x3, ReLU activation, Batch Normalization, Average Pooling
Activation Functions	ReLU	ReLU
Pooling Layers	Max Pooling	Average Pooling
Batch Normalization	No	Yes
Dropout	No	Yes (40%)
Fully Connected Layers	One fully connected layer	Two fully connected layers with dropout
Output Layer	One fully connected layer	One fully connected layer

FLOW CHART


Start
Device Configuration Check
Define Transformation Pipeline
Load MNIST Dataset
Training Images
Validation Images
Testing Images
Data Pre-Processing
CNN 1 Model
Loss Function & Optimizer
Save & Load Model
Testing Process of CNN1
CNN 2 Model
Loss Function & Optimizer
Save & Load Model
Testing Proces of CNN2
Real-time Digit Recognition
Image Processing
Test CNN1 & CNN2 With Webcam Input
End

Comparison Between CNN-1 & CNN-2 Learning Rate 0.01

CNN-1

Accuracy:

Training: (82.07% to 90.90%) & Validation: (86.75% to 91.16%)

Loss:

Training: (0.5581 to 0.3028) & Validation: (0.3998 to 0.2911)

CNN-2

Accuracy:

Training: (90.68% to 97.37%) & Validation: (95.35% to 95.75%)

Loss:

Training: (0.3079 to 0.0923) & Validation: (0.1668 to 0.1597)

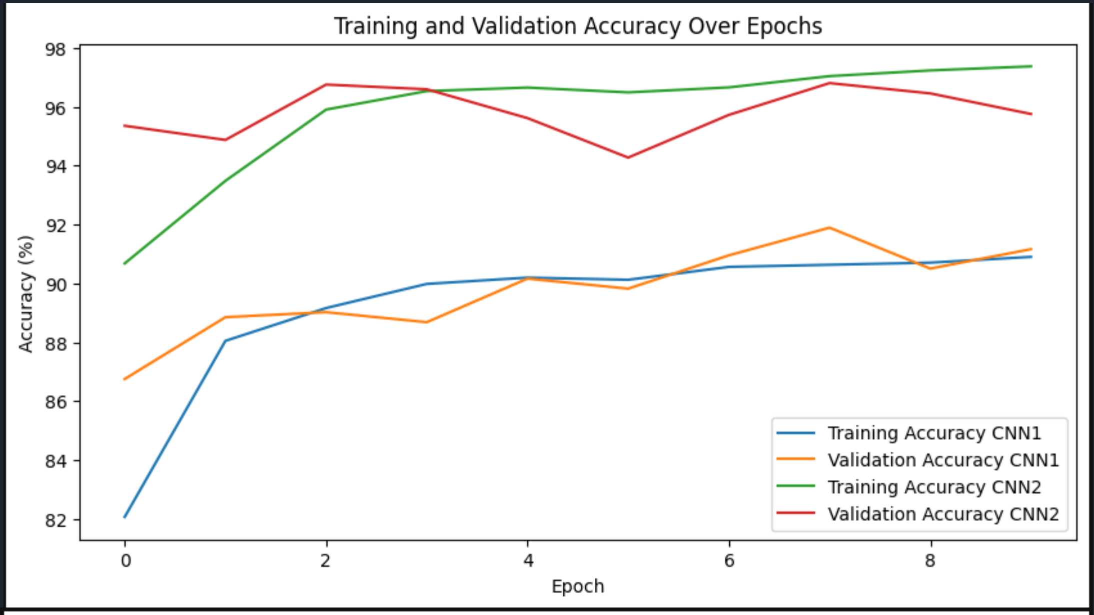
Graphs Overview

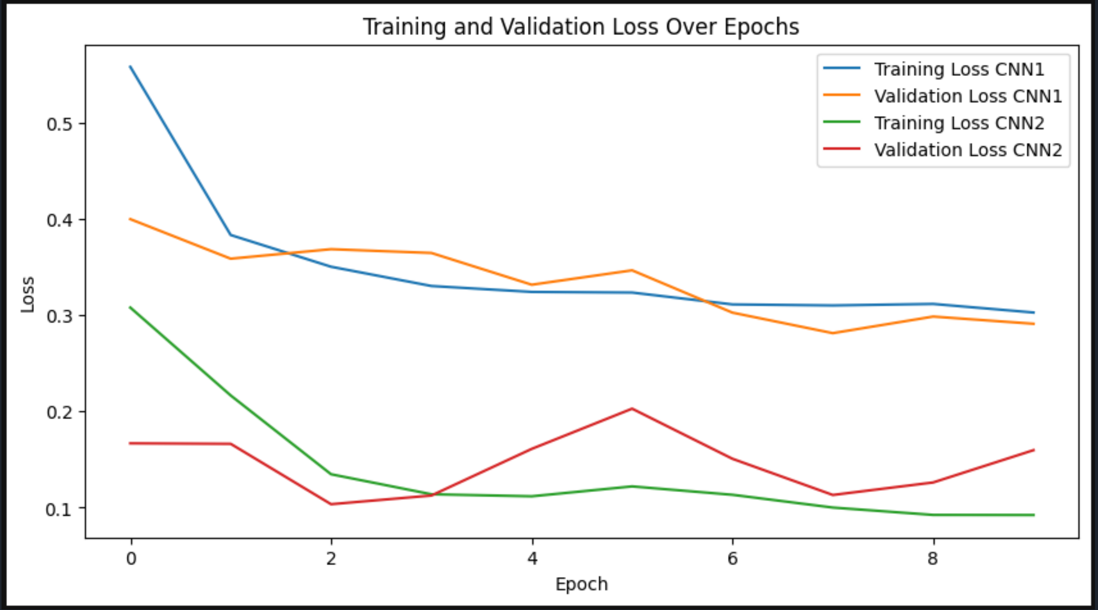
Accuracy Graph:

- CNN2 consistently outperforms CNN1 in both training and validation accuracy.
- CNN1 shows steady improvement but remains lower than CNN2.

Loss Graph:

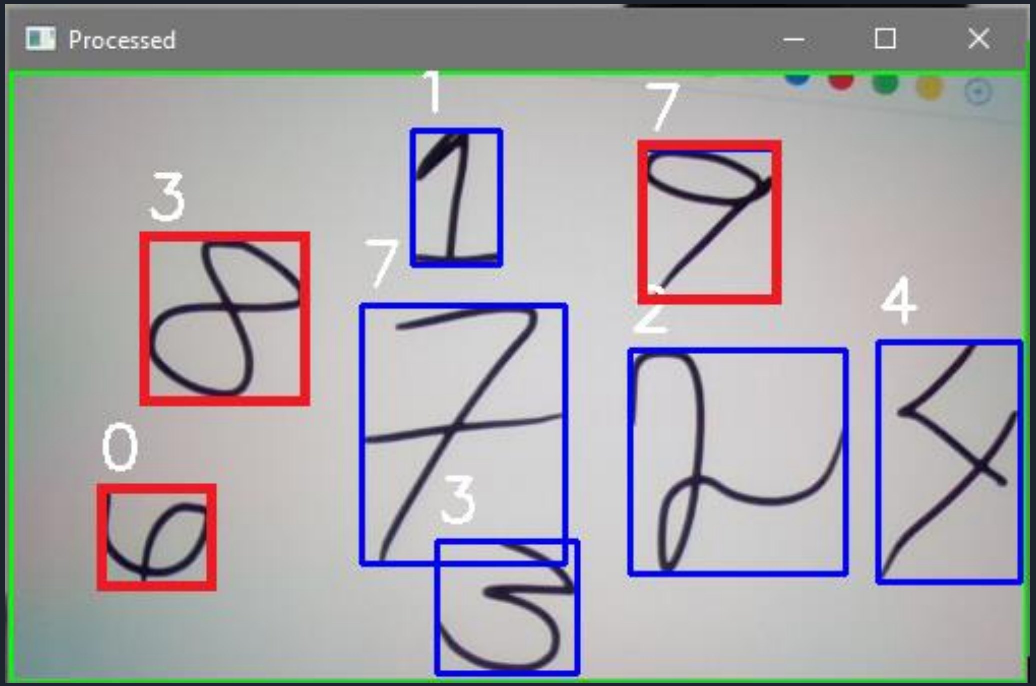
- CNN2 has significantly lower training and validation loss compared to CNN1.
- CNN1 shows a gradual decrease in loss, but it's higher overall than CNN2.



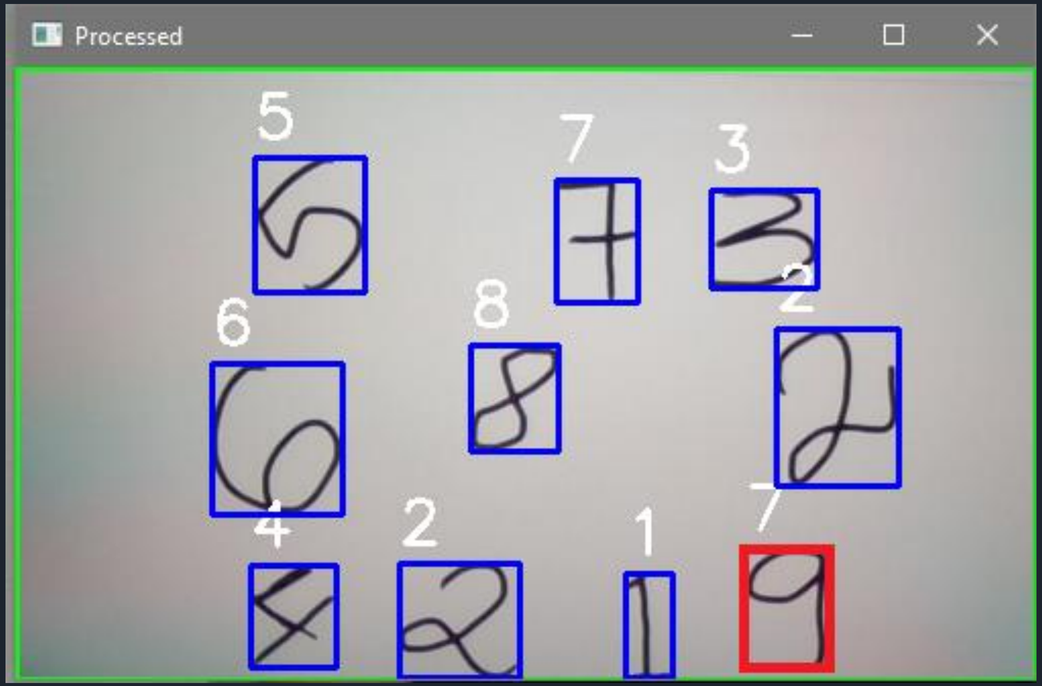


Results of CNN-1 & CNN-2
Learning Rate 0.01

RESULT OF CNN-1



RESULT OF CNN-2



CNN-2 is more effective in correctly classifying handwritten digits with fewer errors, demonstrating better generalization and robustness compared to CNN-1 at the learning rate of 0.01.

Comparison Between CNN-1 & CNN-2 Learning Rate 0.1

CNN-1

Accuracy:

Training: (10.41% to 10.66%)
Validation: (9.73% to 10.93%)

Loss:

Training: (2.8946 to 2.3101)
Validation: (2.3128 to 2.3151)

CNN-2

Accuracy:

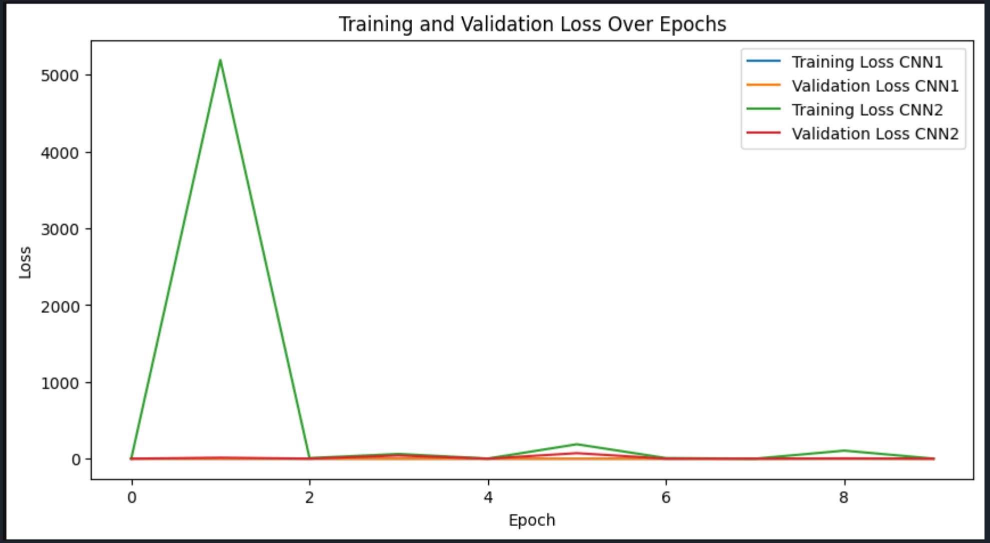
Training: (73.13% to 10.28%)
Validation: (71.42% to 10.60%)

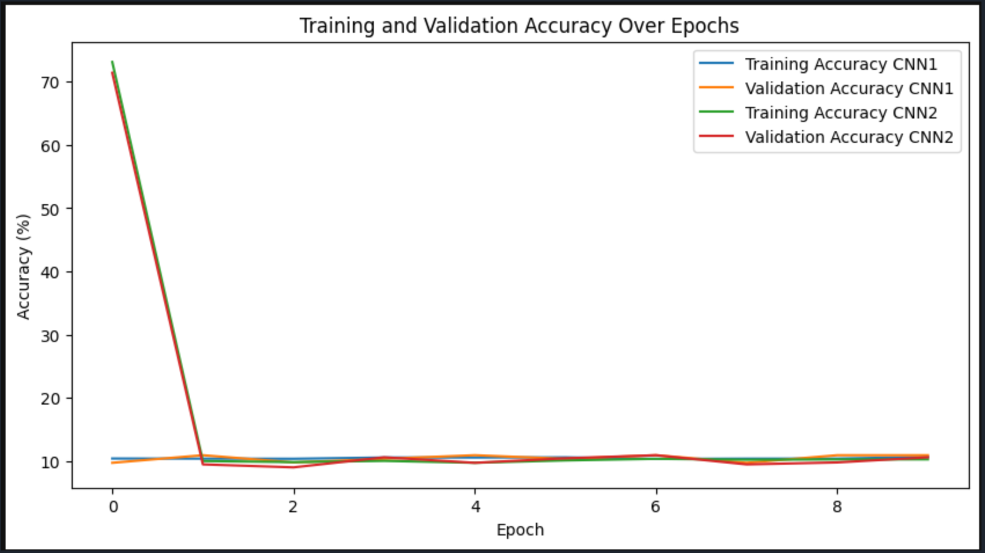
Loss:

Training: (2.2980 to 3.5104)
Validation: (2.6007 to 3.0073)

Graphs Overview

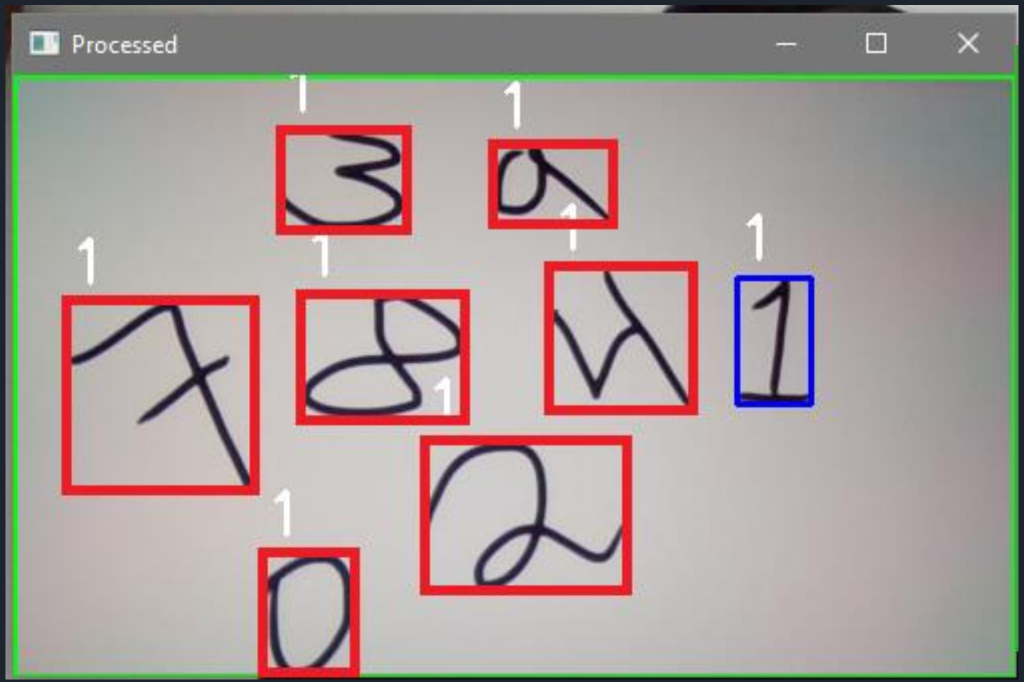
- Both CNN1 and CNN2 exhibit poor performance at a learning rate of 0.1.
- CNN1 fails to improve significantly over epochs, indicating that the model may not be learning effectively.
- CNN2 shows severe instability, with accuracy dropping after the first epoch and highly variable loss values.
- A learning rate of 0.1 is too high for both models, causing learning instability and poor convergence. Lower learning rates are more suitable for these CNN architectures.



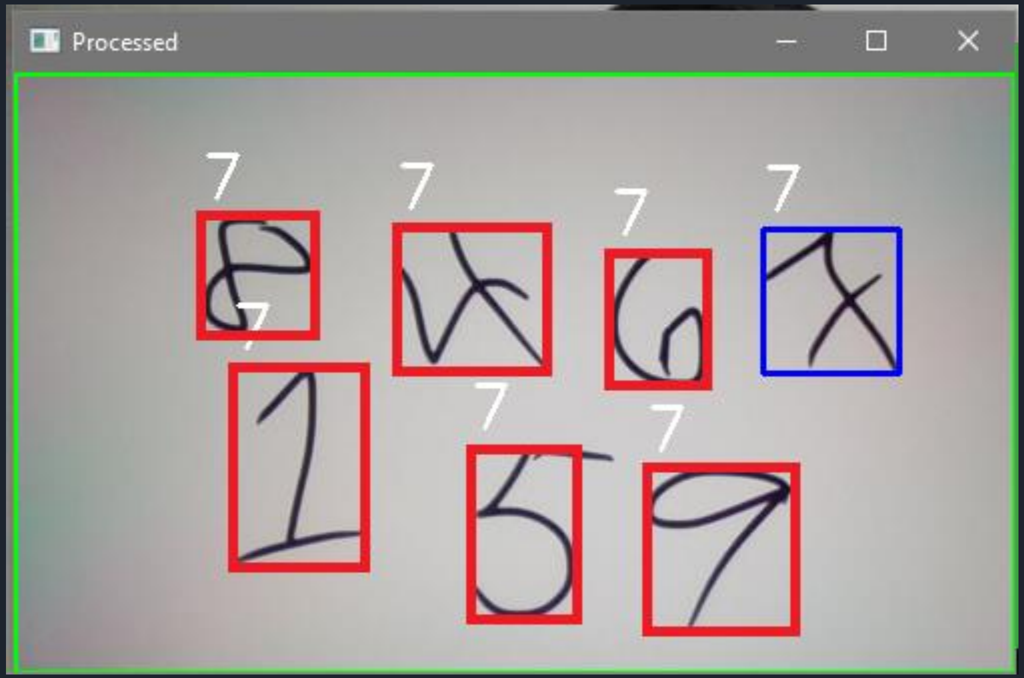


Results of CNN-1 & CNN-2 Learning Rate 0.1

RESULT OF CNN-1

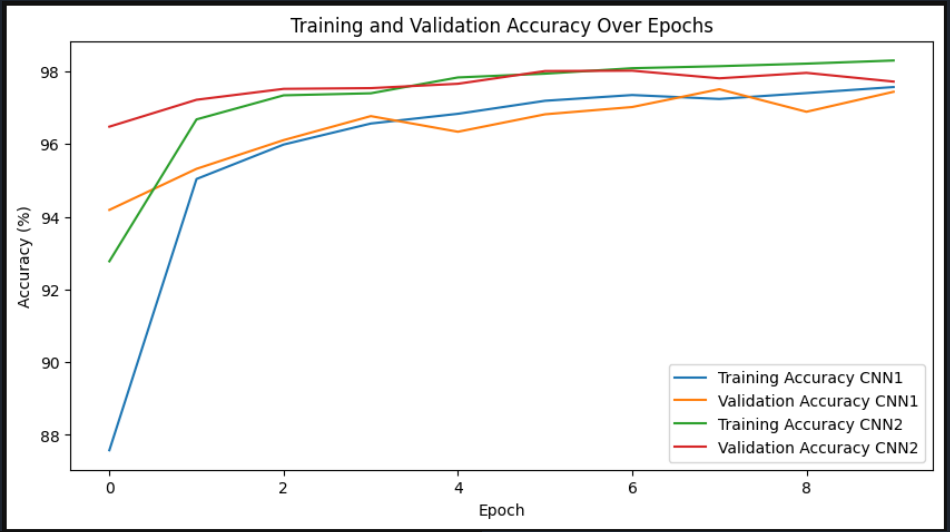


RESULT OF CNN-2



High learning rate (0.1) causes significant instability in both models.
CNN1: Fails to distinguish different digits, defaulting to predicting '1' for all inputs.
CNN2: Similarly fails, predicting '7' for all inputs.

Comparison Between CNN-1 & CNN-2 Learning Rate 0.001



CNN-1

Accuracy:

Training: (87.58%% to 97.57%%)

Validation: (94.19% to 97.44%)

Loss:

Training: (0.4018 to 0.0771)

Validation: (0.1939 to 0.0788)

CNN-2

Accuracy:

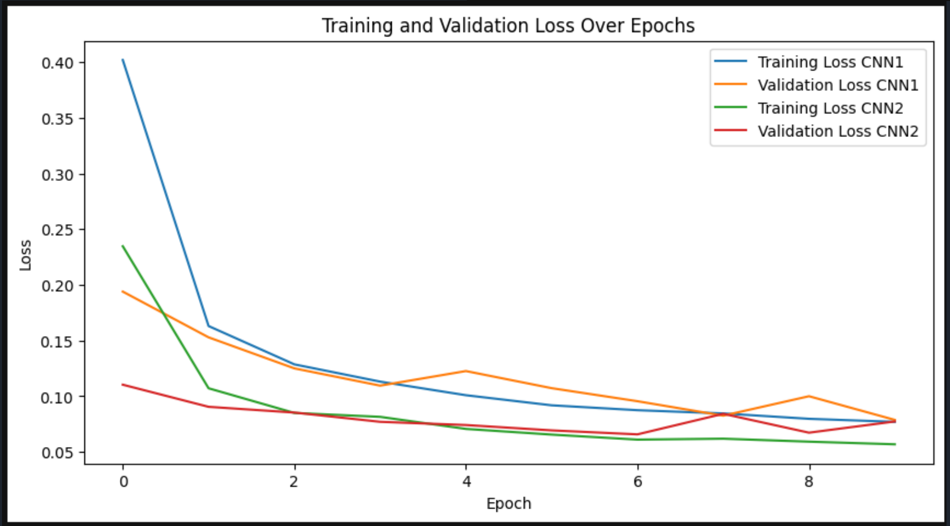
Training: (92.78% to 98.30%)

Validation: (96.48% to 97.72%)

Loss:

Training: (0.2346 to 0.0569)

Validation: (0.1104 to 0.0774)



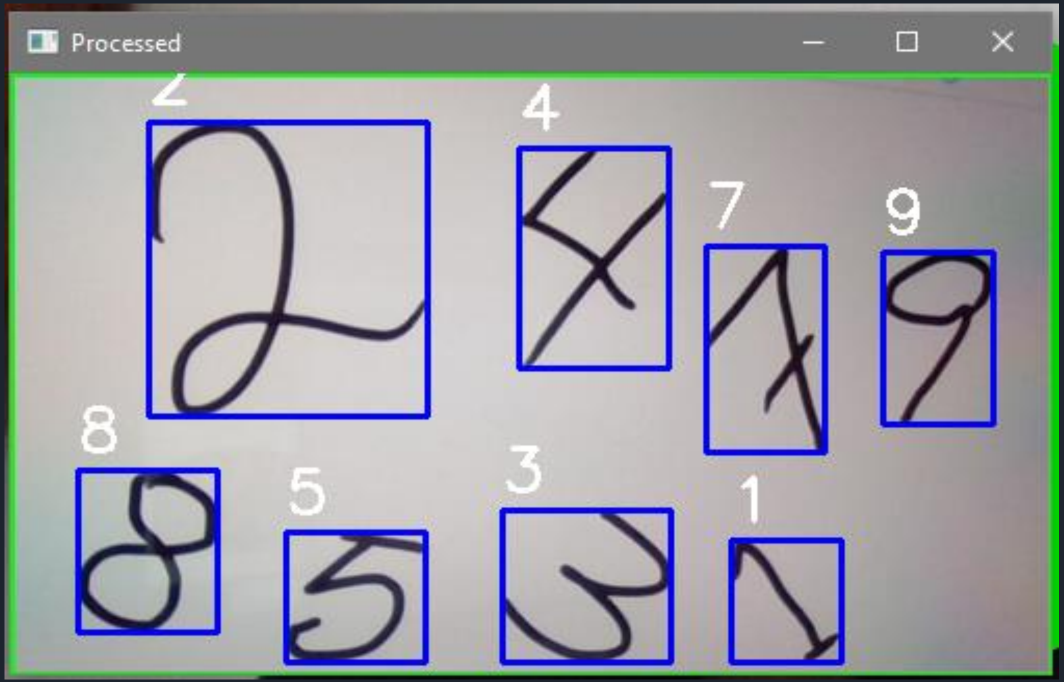
Graphs Overview

CNN-1:
Shows a steady increase in accuracy and a decrease in loss, indicating effective learning.
Final accuracy and loss values are high and low respectively, showing good model performance.

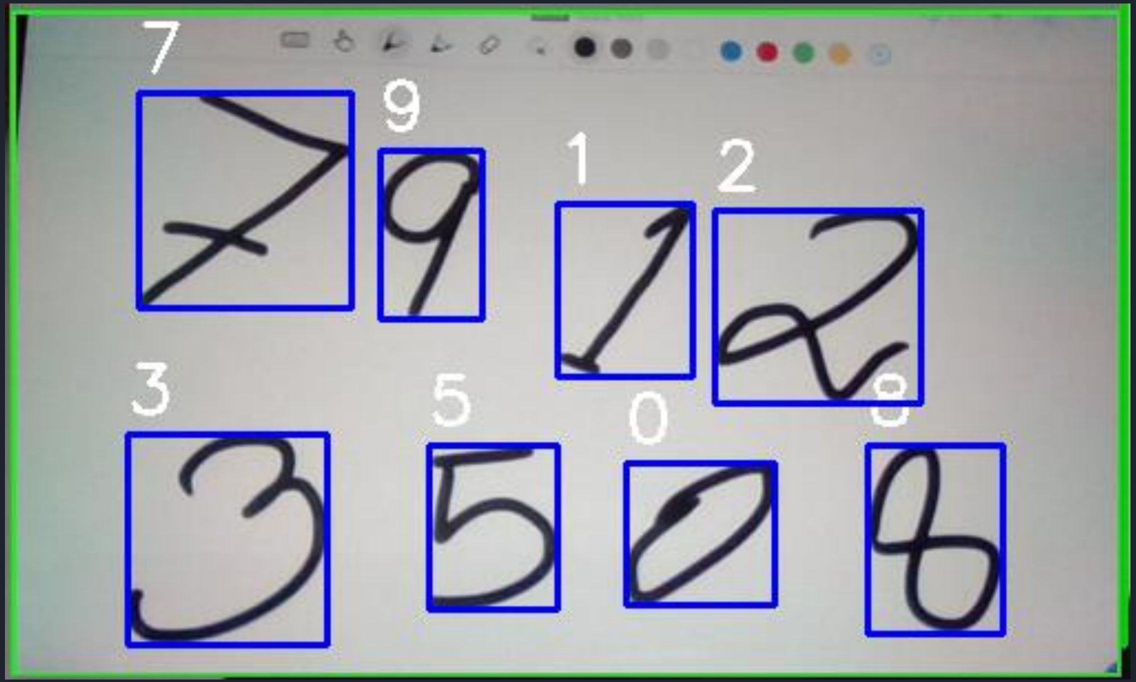
CNN-2:
Achieves high accuracy quickly and maintains a low loss, indicating very effective learning and better performance.
More stable training and validation loss curves compared to CNN1, showing robustness.

Results of CNN-1 & CNN-2
Learning Rate 0.001

RESULT OF CNN-1



RESULT OF CNN-2



Both CNN1 and CNN2 perform perfectly at a learning rate of 0.001, with 100% accuracy in classifying the digits.

Conclusion:

"Dual CNNs for Real-Time Handwritten Digit Recognition via Webcam", we developed and evaluated two CNN models (CNN1 and CNN2) for digit recognition using the MNIST dataset. By applying data augmentation techniques such as random rotations, perspective transformations, and affine transformations, we enhanced the models' generalization capabilities. Both models were trained and validated, with CNN2 showing superior performance due to its deeper architecture and regularization techniques. The models were then evaluated on an augmented test dataset, demonstrating robust performance. Additionally, we implemented a real-time digit recognition system using a webcam, which successfully captured and predicted digits on-the-fly. This project highlights the effectiveness of CNNs in digit recognition tasks and their practical application in real-time scenarios. Future work includes further model optimization, deployment, and expanding the dataset to improve robustness.