# Week 2 Report

## Objective

The primary focus of Week 2 was to gain a practical understanding of Neural Networks, especially Convolutional Neural Networks (CNNs), and apply them to real-world image classification tasks using PyTorch.

## Tools & Libraries Used

* PyTorch – for building and training the neural network
* Torchvision – for loading popular datasets like MNIST/CIFAR and data preprocessing
* Matplotlib – for visualizing training metrics (loss/accuracy)
* scikit-learn – for evaluation metrics like confusion matrix and classification report

## Tasks Completed

* Environment Setup: Installed required libraries using pip.
* Dataset Loading: Loaded and transformed the MNIST dataset using torchvision.
* Model Design: Built a neural network using PyTorch with layers, ReLU activation, and a CrossEntropyLoss function.
* Model Training: Trained the model using an optimizer (SGD/Adam) and visualized loss and accuracy.
* Evaluation: Evaluated using accuracy, confusion matrix, and classification report.

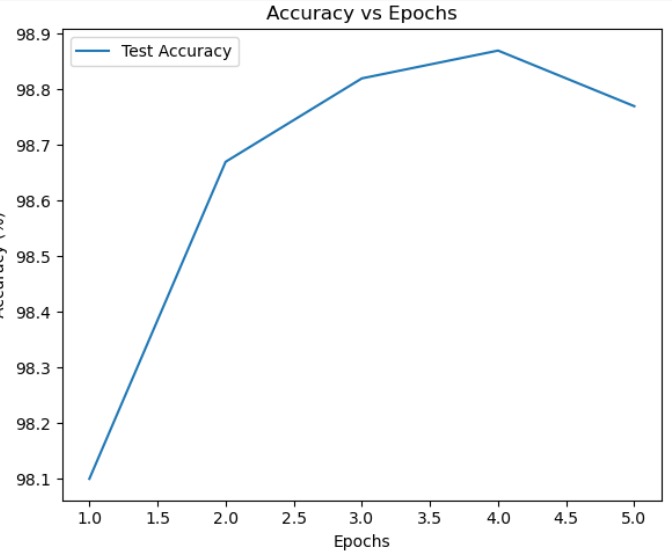
## Results & Observations

The model achieved test accuracy between ~97% and ~98.8% over 5 epochs depending on model type. Loss steadily decreased, showing good convergence. Misclassifications occurred mainly between similar-looking digits. Techniques like dropout could improve performance.

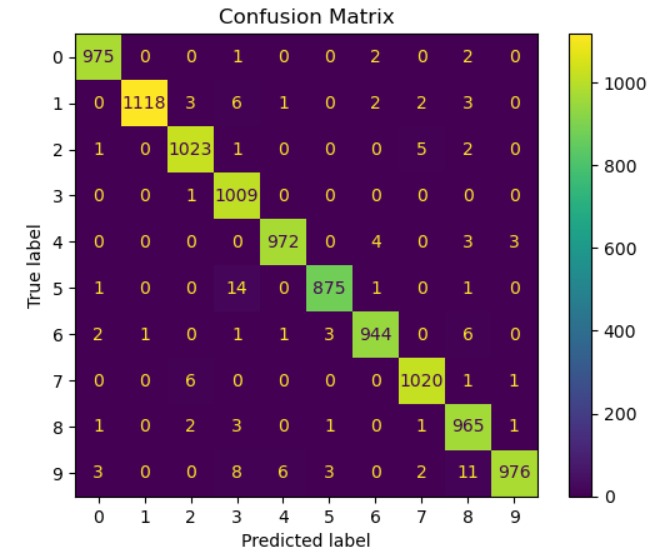
## Conclusion

This assignment provided practical experience in neural network implementation, dataset handling, and result interpretation. Code, visualizations, and results are included in the Week 2 folder.

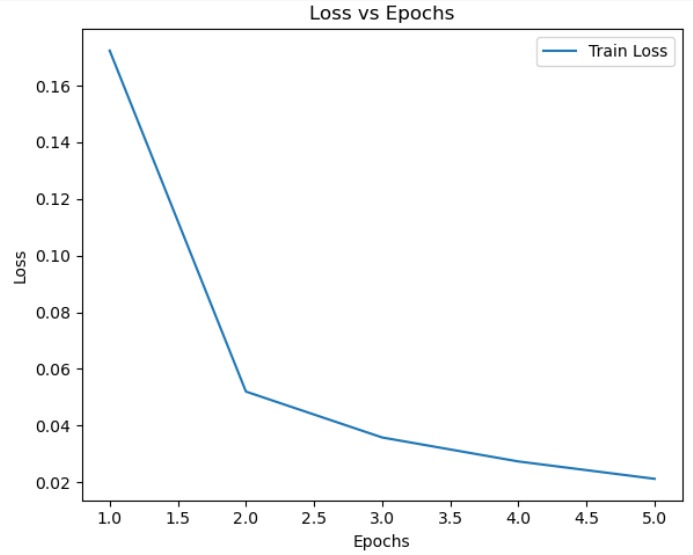
## Visual Results



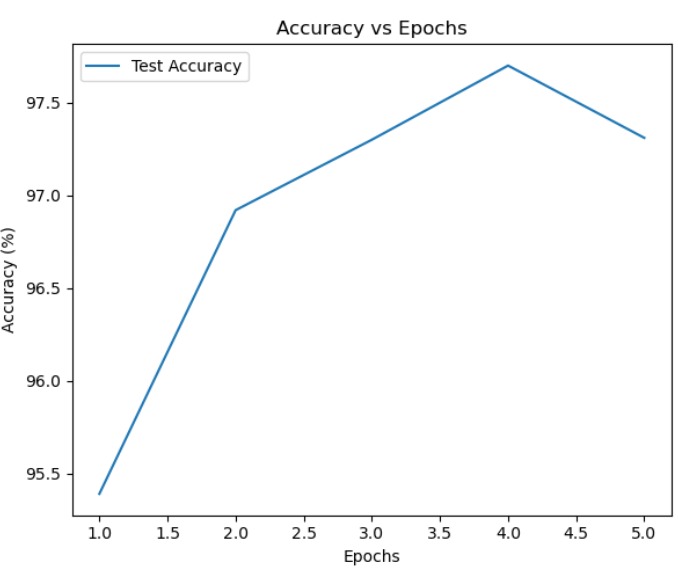
Accuracy vs Epochs – CNN Model



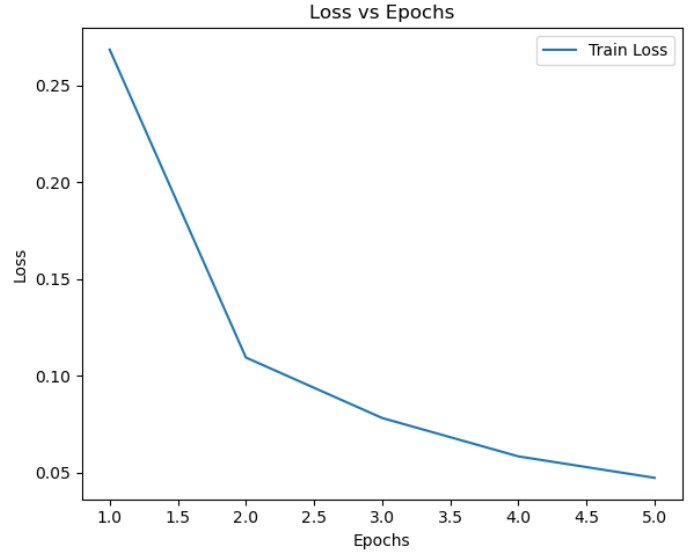
Confusion Matrix – CNN Model



Loss vs Epochs – CNN Model



Accuracy vs Epochs – Feedforward NN



Loss vs Epochs – Feedforward NN