INFO-6146 Capstone Project Report
FasionMNIST classification using CNNs in PyTorch

Abstract:

In this project I built a convolutional neural network (CNN) in PyTorch to classify fashion items using the Fashion-MNIST dataset. The dataset has grayscale images of clothes like shirts, sneakers, and bags. I did the full process: loading the data, setting up the model, training, testing, and checking how well it performs. I also added some visualizations like accuracy/loss curves, a confusion matrix, and misclassified examples to see how the model did and where it messed up.

Introduction:

Image classification is a big part of deep learning. You see it in stuff like online shopping, security systems, and healthcare. For this project I wanted to keep it simple and use a dataset that's good for learning but still realistic. That's why I picked Fashion-MNIST. The goal was to apply what we learned in this course using PyTorch, build a working CNN, and see how it handles real data.

Methodology:

I used the Fashion-MNIST dataset from torchvision.datasets. It has 70,000 images — 60,000 for training and 10,000 for testing. I split the training set into 80% training and 20% validation. I normalized the images and used standard augmentations to help the model generalize better.

The model is a CNN with two convolutional layers, ReLU activations, max pooling, dropout, and two linear layers. I trained it using the Adam optimizer and cross-entropy loss. I used a batch size of 64 and trained for 10 epochs. During training I tracked the loss and accuracy for both training and validation sets.

Results:

The model trained smoothly. Here's a summary:
- Training accuracy reached over 93%
- Validation and test accuracy stayed around 89–91%
- Loss values steadily decreased without signs of overfitting

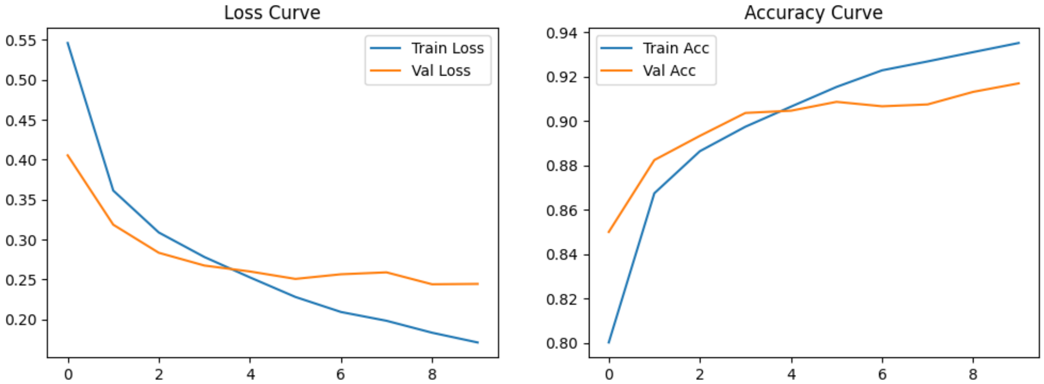

Figure: Loss and Accuracy Curves

I also created a confusion matrix to see which classes were doing well and where the mistakes happened.

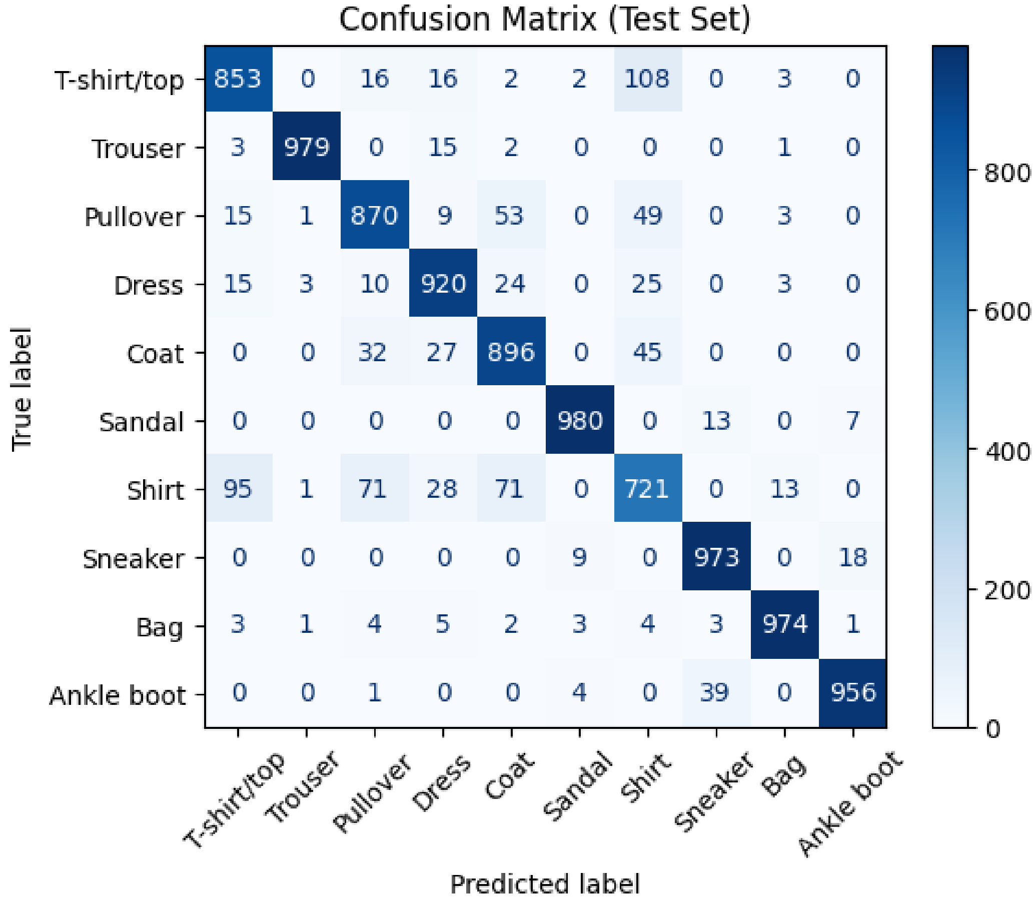

Figure: Confusion Matrix

Lastly, I visualized some of the misclassified examples to understand what kinds of items confused the model the most.

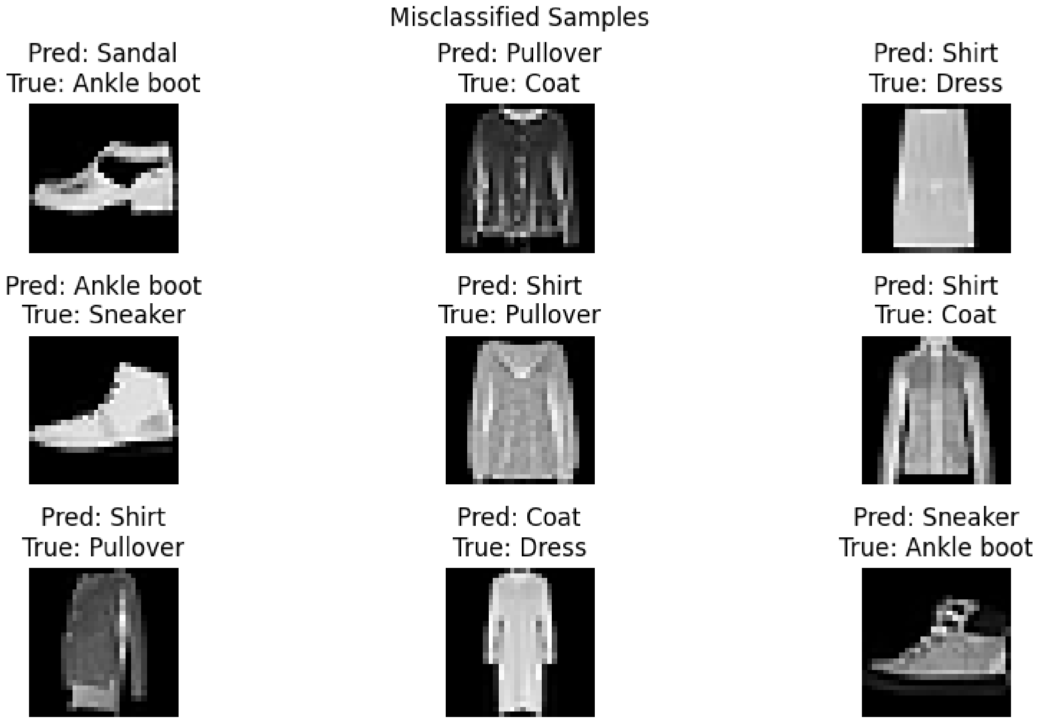

Figure: Misclassified Examples

Discussion:

The CNN did pretty well overall. The Fashion-MNIST dataset is clean and balanced, which helped. Basic augmentations gave a small boost. Tuning the learning rate and dropout helped improve generalization.

Most of the mistakes happened between similar items like shirts vs pullovers or coats vs dresses. This is expected since those categories can look very similar. To improve further I could:

- Add more convolutional layers and batch norm
- Try pretrained models (transfer learning)
- Use more diverse augmentations

Conclusion:

I built an image classifier using PyTorch and CNNs. It was able to classify clothes in Fashion-MNIST with about 90% accuracy. The workflow followed everything we covered: loading data, building and training a model, tuning it, evaluating results, and visualizing performance. With more time, I'd test deeper networks or datasets with RGB images to push the limits more.