Comparing CNN and CNN+SVM for MNIST Digit Recognition

Prepared by Gemini

August 14, 2025

Abstract

In this project, I compared two models for recognizing handwritten digits from the MNIST dataset. The first is a regular Convolutional Neural Network (CNN). The second uses the same CNN to extract features, but instead of a softmax layer, it uses a Support Vector Machine (SVM) to classify the digits. I wanted to see if the hybrid CNN+SVM approach could perform better than a plain CNN.

1 Introduction

The MNIST dataset is a well-known benchmark for testing image classification models. CNNs usually perform extremely well on it, but I wanted to test what happens if we take the CNN's learned features and feed them into an SVM for classification. This project compares the two methods side by side, explains how I set them up, and analyzes the results.

2 Method

I built the CNN using PyTorch and trained it normally. For the SVM part, I used Scikit-learn.

2.1 Dataset and Preprocessing

I used the standard MNIST dataset: 60,000 training images and 10,000 test images of digits (0–9). Each image was converted to a tensor and normalized with mean 0.1307 and standard deviation 0.3081.

2.2 CNN Model

The CNN architecture has:

- Two convolutional layers (1→32 channels, then 32→64 channels), each followed by ReLU and max pooling.
- A fully connected layer that outputs 10 scores (one for each digit).

Training settings:

- 6 epochs, batch size 128, learning rate 0.001
- Optimizer: Adam
- Loss: Cross-Entropy

2.3 CNN+SVM Model

For the hybrid model:

1. I trained the CNN as before, then removed the final classification layer.
2. I used the CNN to turn each image into a 128-dimensional feature vector.
3. I trained a linear SVM (C=1.0) on these features.

2.4 Feature Visualization

To see how well the CNN separated the digits, I used t-SNE to reduce a sample of 2000 feature vectors from 128 dimensions to 2D and plotted them.

3 Results

Table 1 shows the test accuracies. The CNN+SVM model was slightly better.

Table 1: Test Accuracy

Model	Accuracy
CNN (softmax)	0.9906
CNN + SVM	0.9915

Figure 1 shows the confusion matrices. Both models predict almost all digits correctly.

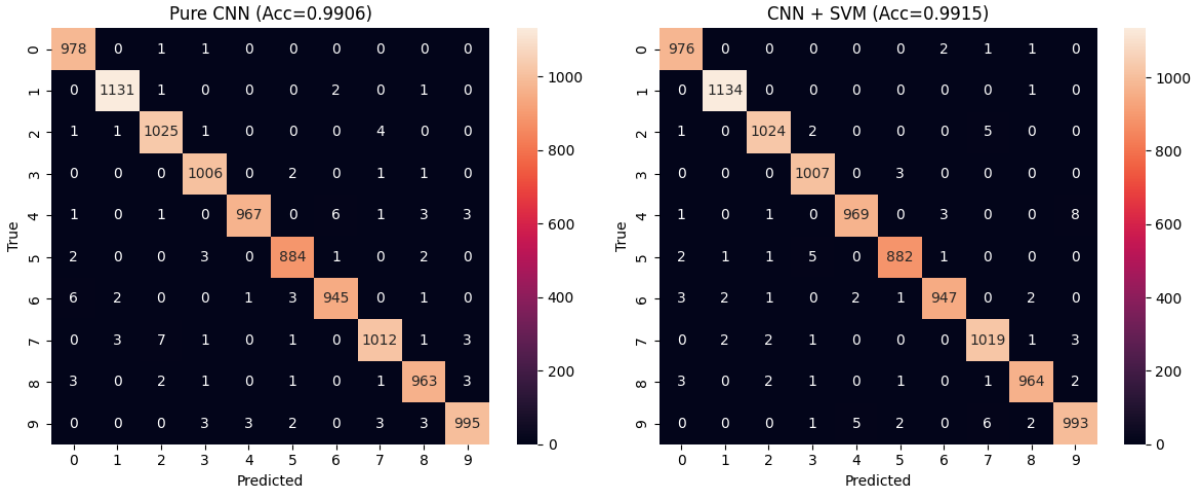

Figure 1: Confusion matrices for CNN (left) and CNN+SVM (right)

The t-SNE plot (Figure 2) shows clear clusters for each digit, meaning the CNN learned features that are easy to separate.

4 Discussion

The CNN alone performed very well at 99.06% accuracy. The CNN+SVM setup gave a small boost to 99.15%. This small improvement is likely because the CNN already produced features that are well separated, making the SVM's job easier. The confusion matrices confirm that both models make very few mistakes.

5 Conclusion

Both models are excellent for MNIST. The hybrid approach only gave a small improvement, but it's still interesting to see that using an SVM on CNN features can slightly increase accuracy when the features are already cleanly separated.

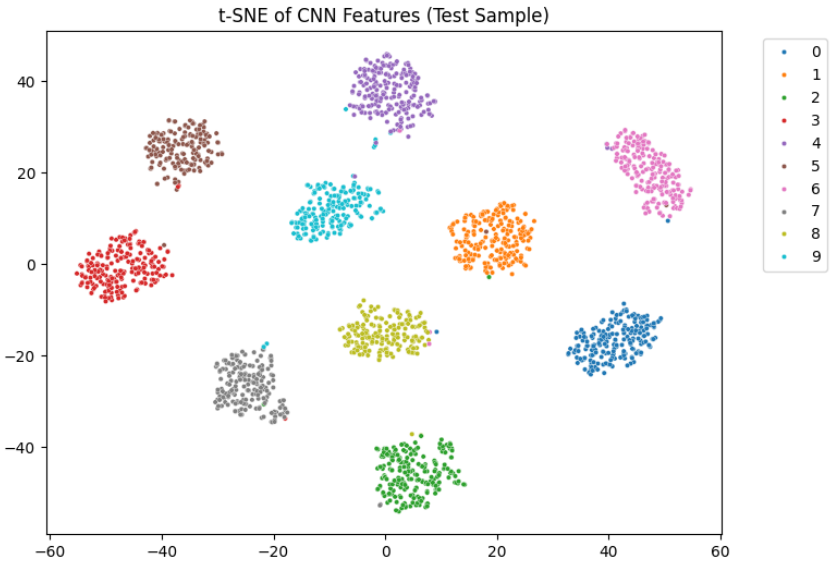

Figure 2: t-SNE of CNN features