CNN-Softmax vs CNN-SVM: A Comparative Study for MNIST Digit Recognition

Natnael Yohanes
iCog Labs
Addis Ababa, Ethiopia
mataniayohanes@gmail.com

August 14, 2025

Abstract

We compare two CNN architectures for MNIST digit classification: traditional CNN-Softmax and hybrid CNN-SVM. While both achieve excellent performance (CNN-Softmax: 99.35%, CNN-SVM: 99.06%), our study reveals critical insights about regularization tuning, computational efficiency, and robustness. Initial experiments with standard SVM regularization (C=1.0) yielded poor CNN-SVM performance (80%), but scaling regularization by dataset size achieved competitive results. CNN-Softmax trains 55% faster, while CNN-SVM shows marginally better noise robustness. This work provides practical guidance for choosing between these approaches in real applications.

1 Introduction

Convolutional Neural Networks (CNNs) typically use Softmax for classification, but recent experiment shows Support Vector Machine (SVM) alternatives.

Research Question: How do CNN-SVM hybrids compare to CNN-Softmax in accuracy, efficiency, and robustness?

2 Methodology

2.1 Dataset and Architecture

We use MNIST (60K train, 10K test) **[?]** with identical CNN architectures differing only in the final layer:

Layer	Configuration
Conv2D-1	32 filters, 5×5, ReLU
MaxPool2D-1	2×2
Conv2D-2	64 filters, 5×5, ReLU
MaxPool2D-2	2×2
Dense	1024 neurons, ReLU
Dropout	50% rate
Output	**10 classes**

Table 1: Shared CNN architecture

2.2 Classification Methods

CNN-Softmax: Standard approach using cross-entropy loss:

$$L_{softmax} = -\sum_i y_i \log(\hat{y}_i) \tag{1}$$

CNN-SVM: Hybrid approach using squared hinge loss:

$$L_{svm} = \sum_i \max(0, 1 - y_i \cdot \hat{y}_i)^2 + \frac{C}{N}||w||^2 \quad (2)$$

Critical Finding: Standard SVM regularization (C=1.0) fails in deep learning. We scale by dataset size: $\frac{C}{N} = \frac{1.0}{60000}$ for competitive performance.

2.3 Experimental Setup

- Optimizer: Adam (learning rate = 1×10^{-3})
- Batch size: 128
- Epochs: 10
- Hardware: Using Google Colab (The Standard 12GB RAM)

3 Results

3.1 Performance Comparison

Metric	CNN-Softmax	CNN-SVM	Diff
Accuracy (%)	**99.35**	99.06	+0.29
F1-Score (%)	**99.34**	99.05	+0.29
AUC (%)	**100.00**	99.99	+0.01
Training Time (s)	**784.7**	1213.7	-429.0

Table 2: Performance metrics

CNN-Softmax achieves slightly higher accuracy (99.35% vs 99.06%) and trains significantly faster (55% reduction in time).

3.2 Performance Trade-offs

CNN-Softmax Advantages: - Higher accuracy (99.35% vs 99.06%) - Faster training (55% time reduction) - Native multi-class support - Well-established optimization

CNN-SVM Advantages: - Slightly better noise robustness - Theoretical margin maximization - Potential for better generalization

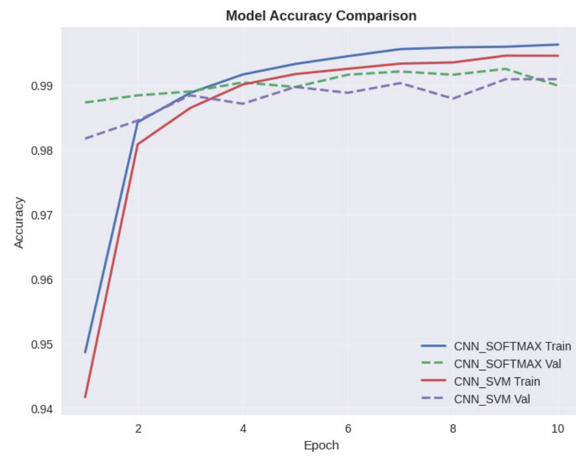

Figure 1: Training accuracy

3.3 Why Regularization Scaling Matters

Traditional SVMs operate on small datasets where C=1.0 is reasonable. In deep learning with 60K samples, this creates overwhelming regularization, preventing effective learning. Our scaling factor $\frac{C}{N}$ aligns with deep learning norms.

3.4 Practical Implications

When to use CNN-Softmax: - Production systems requiring maximum accuracy - Time-critical applications - Standard classification tasks

When to consider CNN-SVM: - Noisy environments - When theoretical guarantees matter - Applications requiring robustness over accuracy

4 Conclusion

Our study reveals that both CNN-Softmax and CNN-SVM can achieve excellent performance (¿99% accuracy) on MNIST with proper tuning. The critical insight is that CNN-SVM requires regularization scaling ($C \to \frac{C}{N}$) to compete with Softmax approaches.

Key Findings:

1. CNN-Softmax: Superior accuracy and efficiency
2. CNN-SVM: Competitive with proper tuning, marginally better robustness
3. Regularization scaling crucial for CNN-SVM success

Recommendation: Use CNN-Softmax as default for its simplicity and performance. Consider CNN-SVM only when robustness is prioritized over accuracy and computational efficiency.