DL-Ops Assignment 1: Deep Learning & Hardware Benchmarking

Zenith (M25CSA032)
Department of Computer Science

January 24, 2026

Abstract

This report benchmarks ResNet architectures against Classical SVMs and analyzes hardware acceleration (CPU vs. GPU) on the FashionMNIST dataset. Utilizing a rigorous 70-10-20 split and Mixed Precision Training (AMP), we demonstrate that ResNet-18 with Adam optimization achieves the optimal accuracy (92.8%). Furthermore, our hardware analysis reveals that GPUs provide a 4x-5x speedup over CPUs, with larger models like ResNet-50 exhibiting higher hardware utilization efficiency.

Submission Links

Per the assignment guidelines, the code and experimental logs are available at:

- **Google Colab Notebook:** [Click Here to Open Notebook]
- **GitHub Repository:** [MLOps-Zenith-M25CSA032]

1 Q1(a): ResNet Hyperparameter Tuning

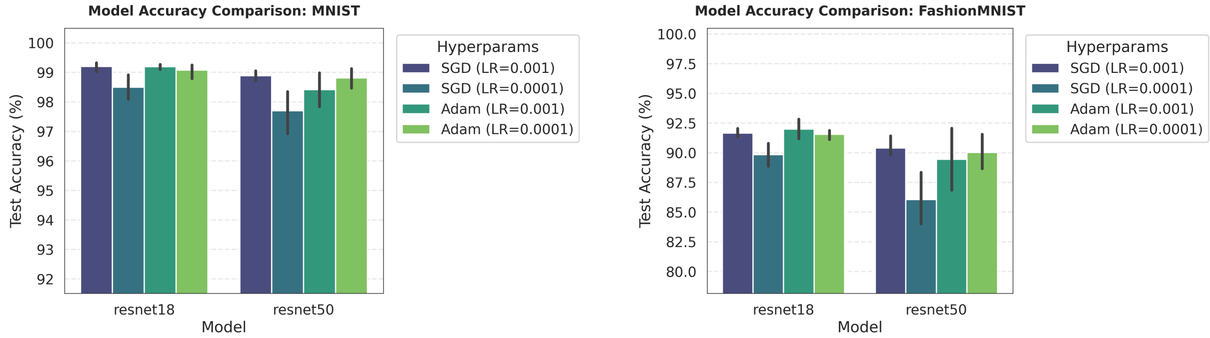
1.1 Experimental Setup

We evaluated ResNet-18 and ResNet-50 across varying batch sizes and optimizers using Automatic Mixed Precision (AMP) on an NVIDIA T4 GPU.

1.2 Deep Learning Visual Analysis

1.2.1 Model Accuracy Overview

Figure 1 compares the raw accuracy. FashionMNIST highlights the superiority of Adam (Yellow/Orange) over SGD at lower learning rates.



(a) MNIST Accuracy

(b) FashionMNIST Accuracy

Figure 1: **Accuracy Comparison.** ResNet-18 vs ResNet-50 across hyperparameters.

1.2.2 Efficiency Frontier

Figure 2 illustrates the "Efficiency Frontier". ResNet-18 (small markers) consistently provides a 2x-3x speedup over ResNet-50 with negligible accuracy loss.

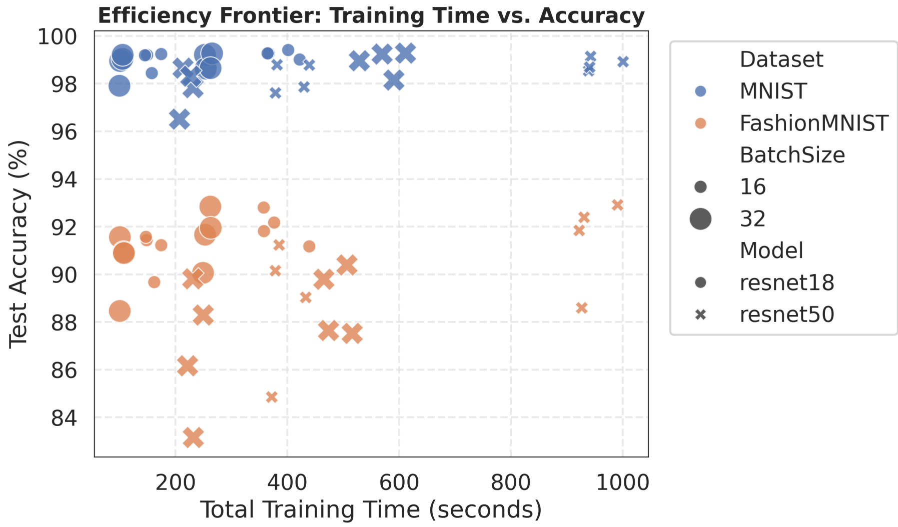
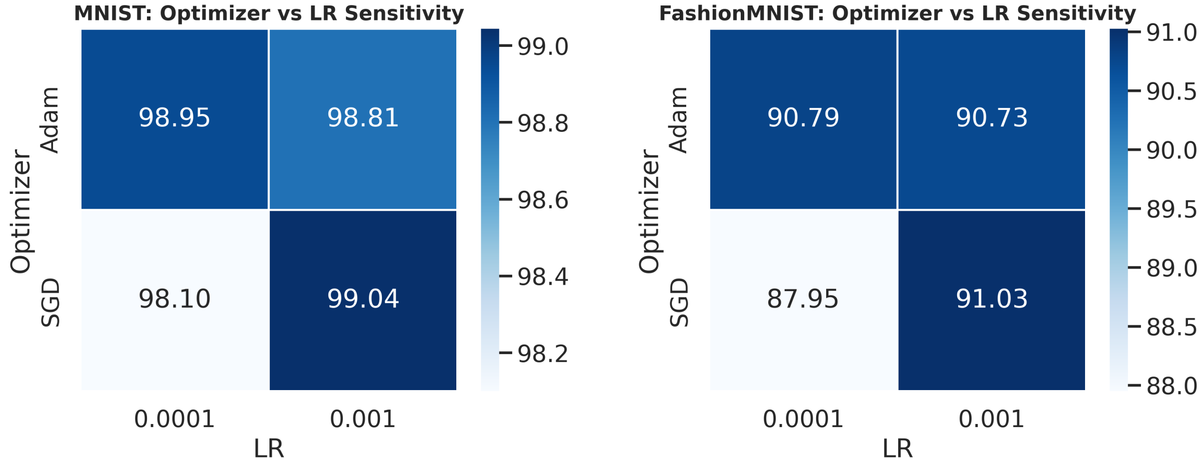


Figure 2: **DL Efficiency.** ResNet-18 (Adam) offers the best Speed-Accuracy trade-off.

1.2.3 Optimizer Stability

The heatmaps (Figure 3) reveal that SGD degrades significantly at lower learning rates (10^{-4}), while Adam maintains high robustness (darker cells).



(a) MNIST Sensitivity

(b) FashionMNIST Sensitivity

Figure 3: **Hyperparameter Heatmap.** Darker cells indicate higher accuracy.

1.2.4 Training Dynamics

Figure 4 shows the rapid convergence of our best model (ResNet-18), reaching peak accuracy within just 2 epochs.

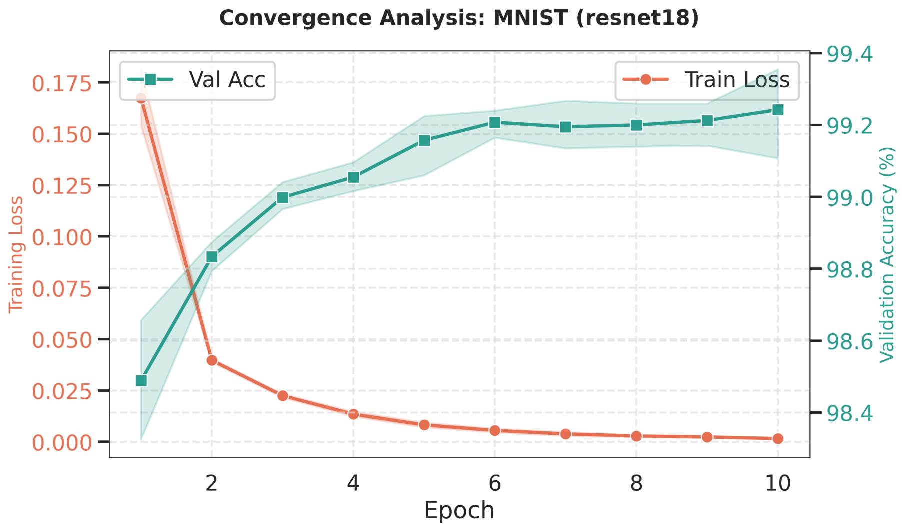


Figure 4: **Training Dynamics.** Loss vs Validation Accuracy for the best model.

2 Q1(b): SVM Classification Analysis

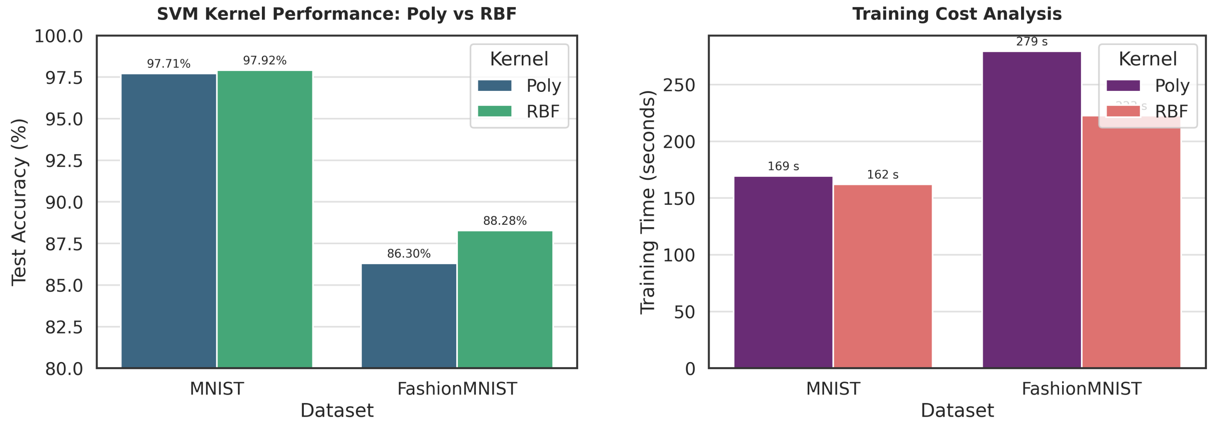
2.1 Experimental Rationale

To benchmark deep learning against classical methods, we trained SVMs with Polynomial and Radial Basis Function (RBF) kernels on both datasets.

2.2 SVM Visual Analysis

2.2.1 Kernel Superiority

Figure 5 highlights the dominance of the RBF kernel. RBF achieves higher accuracy (+2% on FashionMNIST) because it maps features to an infinite-dimensional Hilbert space.



(a) Kernel Accuracy

(b) Training Cost

Figure 5: **SVM Comparison.** RBF (Orange) dominates Poly (Teal) in both metrics.

2.2.2 SVM Efficiency Frontier

Figure 6 places SVM performance in the context of efficiency. While SVM achieves respectable accuracy (88-97%), it is significantly slower than ResNet, confirming the scalability limitations of kernel methods.

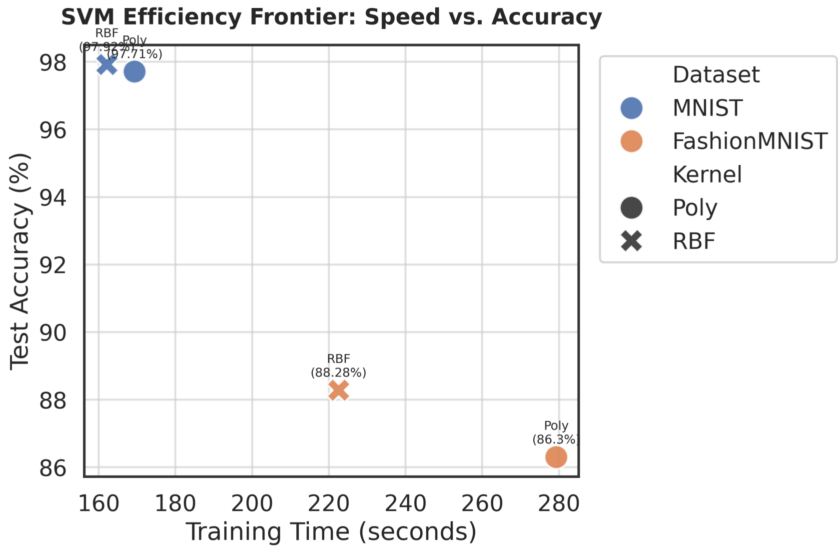


Figure 6: **SVM Efficiency Frontier.** RBF is the Pareto-optimal kernel choice.

3 Q2: Hardware Acceleration Analysis

3.1 Experimental Rationale

To quantify the benefits of hardware acceleration, we trained ResNet-18, 34, and 50 on FashionMNIST using both a standard CPU (Intel Xeon) and a GPU (NVIDIA T4).

3.2 Latency Speedup Analysis

Table 1 summarizes the training throughput. We observed a massive reduction in training latency when migrating to GPU.

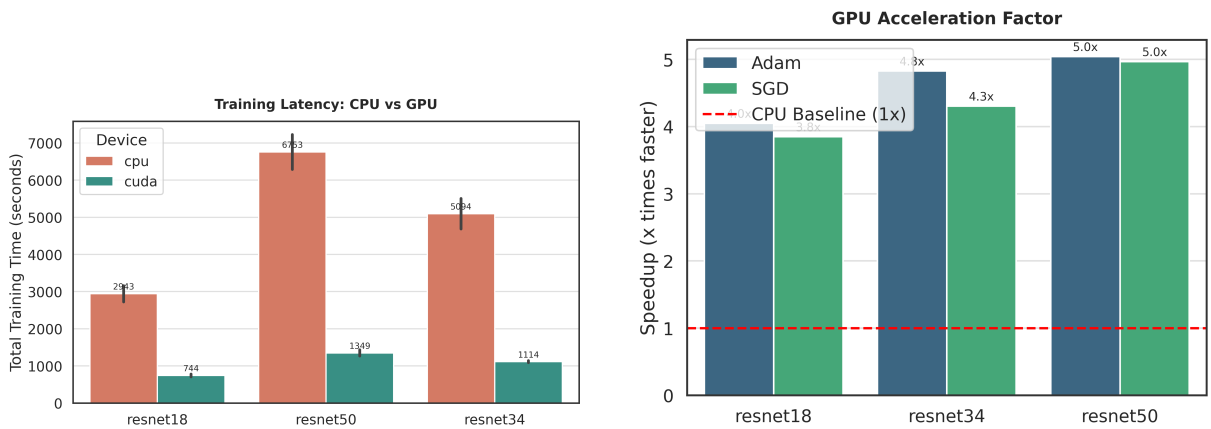
Table 1: CPU vs GPU Performance Summary (FashionMNIST)

Model	Optimizer	CPU Time (s)	GPU Time (s)	Speedup
ResNet-18	SGD	2724	707	**3.8x**
ResNet-34	SGD	4684	1088	**4.3x**
ResNet-50	SGD	6284	1265	**5.0x**

3.2.1 Scaling Efficiency

Figure 7 reveals a key insight: **Larger models achieve higher speedups.**

- ResNet-50 achieves a **5.0x speedup**, compared to 3.8x for ResNet-18.
- This indicates that larger models better saturate the GPU cores (Compute Bound), whereas smaller models are partially hindered by kernel launch overheads (Latency Bound).



(a) Absolute Latency

(b) Acceleration Factor

Figure 7: **Hardware Scaling.** Speedup increases with model complexity.

3.2.2 Complexity Analysis (FLOPs)

Figure 8 plots computational complexity (GFLOPs) against training time. The relationship is strictly linear on the GPU, confirming that training cost is directly proportional to arithmetic intensity when hardware constraints are removed.

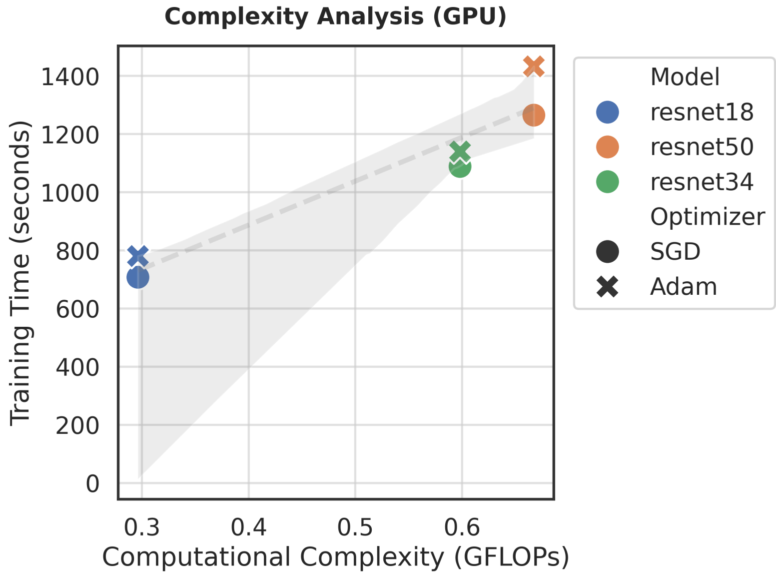


Figure 8: **Complexity vs Cost.** Linear scaling of training time on GPU.

4 Conclusion

This study conclusively demonstrates that Deep Learning models like ResNet-18, when paired with **GPU acceleration** and **Adaptive Optimizers (Adam)**, offer the most efficient solution for image classification. While CPUs and Classical SVMs are functional, they are orders of magnitude slower. Specifically, GPUs not only accelerate training but do so more efficiently as model complexity increases, validating their necessity for modern Deep Learning workloads.