

MLOps Assignment 1

Deep Learning Experiments on MNIST and FashionMNIST

Submitted by:

Vasishth Bhatt

Roll Number: M25CSA007

Date: January 24, 2026

Project Links

GitHub Pages: <https://duckquack123.github.io/MLOps-Vasishth-M25CSA007/>

Kaggle Notebook: <https://www.kaggle.com/code/bhattvasishth/m25csa007-mldlops-ass>

GitHub Repository: <https://github.com/duckquack123/MLOps-Vasishth-M25CSA007>

Contents

1	Introduction	2
1.1	Scope of Experiments	2
1.2	Experimental Setup	2
1.3	Datasets	2
2	Q1(a): ResNet Classification Results	3
2.1	MNIST Dataset Results	3
2.2	FashionMNIST Dataset Results	3
2.3	Accuracy Comparison Visualization	4
2.4	Hyperparameter Heatmaps	4
2.5	ResNet-18 vs ResNet-50 Comparison	5
2.6	Training Time Analysis	6
2.7	Effect of Epochs and Pin Memory	7
2.8	Q1(a) Key Findings	7
3	Q1(b): SVM Classification Results	8
3.1	MNIST SVM Results	8
3.2	FashionMNIST SVM Results	8
3.3	SVM Visualization	9
3.4	SVM vs Deep Learning Comparison	9
3.5	Q1(b) Key Findings	9
4	Q2: CPU vs GPU Performance Comparison	10
4.1	Performance Results	10
4.2	GPU Speedup Analysis	10
4.3	FLOPs Comparison	10
4.4	Performance Visualizations	11
4.5	Q2 Key Findings	12
5	Conclusions	13
5.1	Summary of Best Results	13
5.2	Key Takeaways	13

1 Introduction

This report presents the results of deep learning experiments conducted on the MNIST and FashionMNIST datasets using ResNet architectures. The experiments compare different model configurations, hyperparameters, and training environments.

1.1 Scope of Experiments

The experiments are organized into three main questions:

- **Q1(a):** ResNet-18 and ResNet-50 classification with grid search over hyperparameters
- **Q1(b):** SVM classifier baseline with polynomial and RBF kernels
- **Q2:** CPU vs GPU (CUDA) performance comparison

1.2 Experimental Setup

Table 1: Experimental Configuration

Parameter	Value
Data Split	70% Train / 10% Validation / 20% Test
Models	ResNet-18 (11.18M params), ResNet-50 (23.52M params)
Pretrained	False
Mixed Precision (AMP)	True
Batch Sizes	16, 32
Optimizers	SGD, Adam
Learning Rates	0.001, 0.0001
Epochs	2, 5
pin_memory	True, False

1.3 Datasets

- **MNIST:** Handwritten digits (0-9), 28×28 grayscale images, 70,000 samples
- **FashionMNIST:** Fashion items (10 classes), 28×28 grayscale images, 70,000 samples

2 Q1(a): ResNet Classification Results

This section presents the classification performance of ResNet-18 and ResNet-50 models trained with different hyperparameter configurations.

2.1 MNIST Dataset Results

Table 2: MNIST Test Classification Accuracy (%)

Batch Size	Optimizer	Learning Rate	ResNet-18	ResNet-50
16	SGD	0.001	98.95	98.55
16	SGD	0.0001	98.48	97.91
16	Adam	0.001	99.01	98.31
16	Adam	0.0001	98.76	97.74
32	SGD	0.001	98.82	98.70
32	SGD	0.0001	97.91	96.87
32	Adam	0.001	98.78	98.54
32	Adam	0.0001	98.66	97.02

Best MNIST Result: ResNet-18 with Adam optimizer, learning rate 0.001, batch size 16 achieves **99.01%** accuracy.

2.2 FashionMNIST Dataset Results

Table 3: FashionMNIST Test Classification Accuracy (%)

Batch Size	Optimizer	Learning Rate	ResNet-18	ResNet-50
16	SGD	0.001	90.06	89.02
16	SGD	0.0001	88.66	85.60
16	Adam	0.001	90.57	87.42
16	Adam	0.0001	89.35	87.91
32	SGD	0.001	89.93	89.29
32	SGD	0.0001	87.86	83.60
32	Adam	0.001	90.76	89.40
32	Adam	0.0001	89.69	86.91

Best FashionMNIST Result: ResNet-18 with Adam optimizer, learning rate 0.001, batch size 32 achieves **90.76%** accuracy.

2.3 Accuracy Comparison Visualization

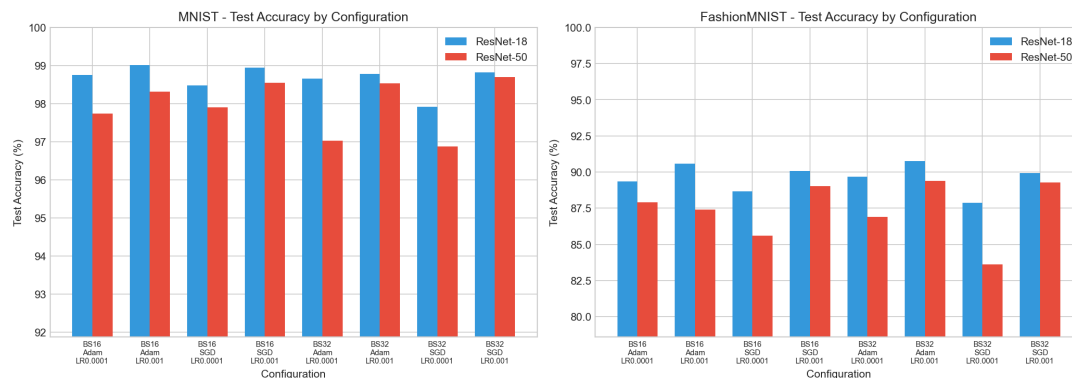
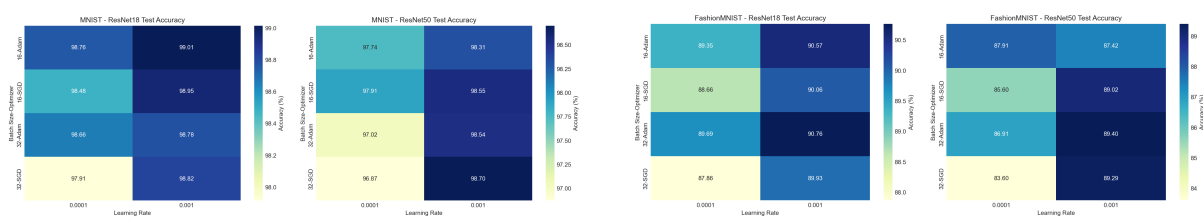


Figure 1: Accuracy comparison across different hyperparameter configurations for MNIST and FashionMNIST datasets.

2.4 Hyperparameter Heatmaps



(a) MNIST Accuracy Heatmap

(b) FashionMNIST Accuracy Heatmap

Figure 2: Heatmaps showing accuracy distribution across hyperparameter combinations.

2.5 ResNet-18 vs ResNet-50 Comparison

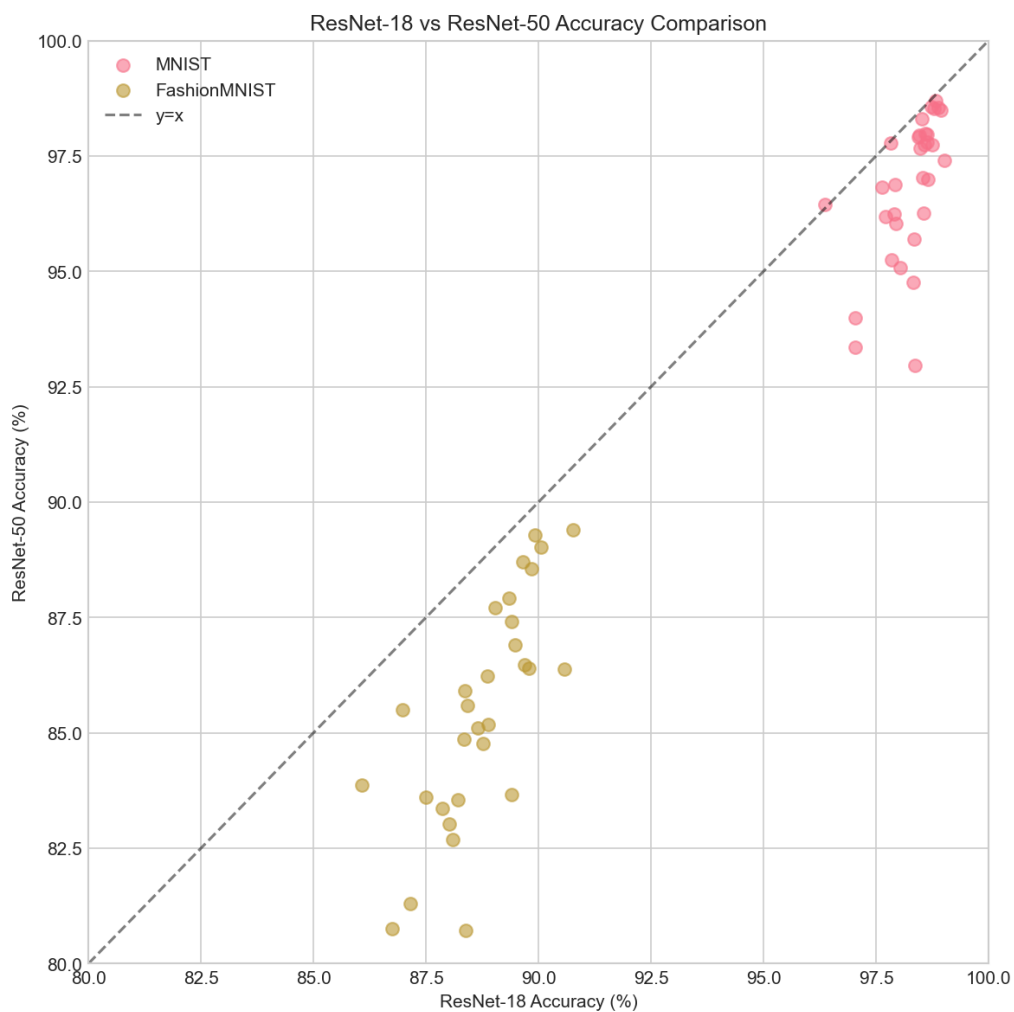


Figure 3: Scatter plot comparing ResNet-18 and ResNet-50 performance across configurations.

2.6 Training Time Analysis

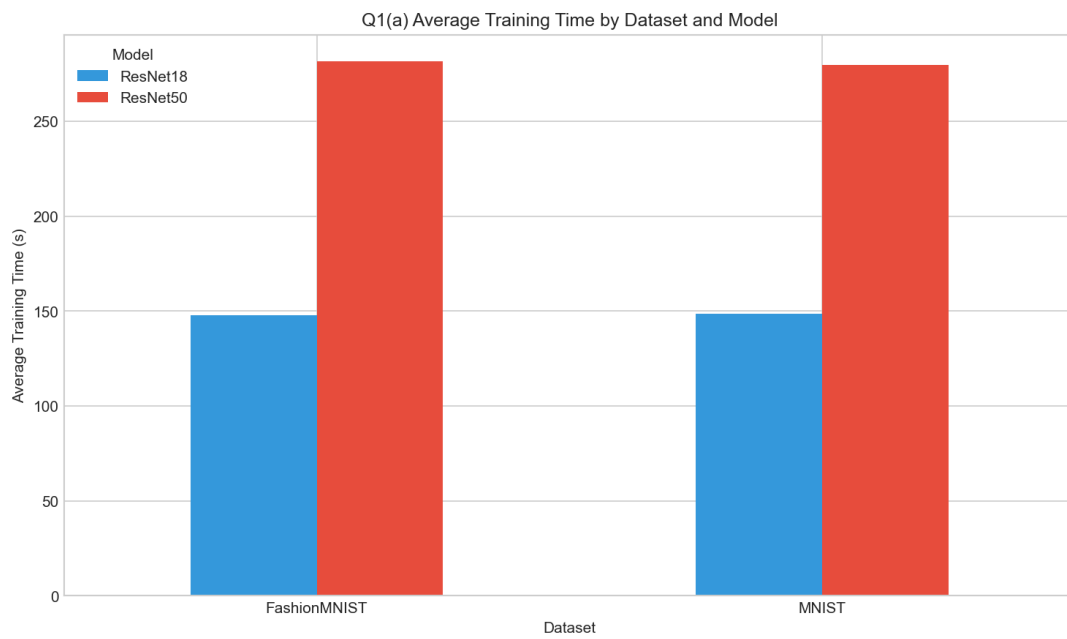


Figure 4: Training time comparison for different model and hyperparameter configurations.

2.7 Effect of Epochs and Pin Memory

Additional experiments were conducted to study the impact of training duration and memory pinning.

Table 4: Effect of Epochs and Pin Memory (MNIST, ResNet-18, lr=0.001)

Batch Size	Optimizer	pin_memory	Epochs	Accuracy (%)
16	Adam	False	2	98.36
16	Adam	False	5	99.01
16	Adam	True	2	98.32
16	Adam	True	5	98.52
16	SGD	False	2	98.46
16	SGD	False	5	98.89
16	SGD	True	2	98.59
16	SGD	True	5	98.95

2.8 Q1(a) Key Findings

1. **ResNet-18 vs ResNet-50:** Despite having fewer parameters (11.18M vs 23.52M), ResNet-18 consistently outperforms ResNet-50. This suggests potential overfitting with the larger model on these relatively simple datasets.
2. **Learning Rate Impact:** A learning rate of 0.001 consistently outperforms 0.0001 across all configurations.
3. **Optimizer Comparison:** Adam optimizer achieves the best peak accuracies (99.01% on MNIST, 90.76% on FashionMNIST).
4. **Batch Size Effect:** Minimal difference (<1%) between batch sizes 16 and 32.
5. **Epochs:** Training for 5 epochs yields better results than 2 epochs.

3 Q1(b): SVM Classification Results

Support Vector Machine classifiers were trained as traditional machine learning baselines using polynomial and RBF kernels with varying regularization parameters.

3.1 MNIST SVM Results

Table 5: SVM Results on MNIST Dataset

Kernel	C	Gamma	Test Acc (%)	F1 Score	Train Time (s)
poly	0.1	scale	54.75	0.579	39.6
poly	1.0	scale	91.50	0.919	22.8
poly	10.0	scale	95.50	0.955	16.0
poly	0.1	auto	38.30	0.415	41.3
poly	1.0	auto	87.95	0.888	25.9
poly	10.0	auto	95.35	0.954	16.2
rbf	0.1	scale	89.35	0.895	20.0
rbf	1.0	scale	94.85	0.948	11.4
rbf	10.0	scale	95.60	0.956	10.7
rbf	0.1	auto	89.85	0.899	19.7
rbf	1.0	auto	95.20	0.952	10.7
rbf	10.0	auto	95.55	0.955	9.8

3.2 FashionMNIST SVM Results

Table 6: SVM Results on FashionMNIST Dataset

Kernel	C	Gamma	Test Acc (%)	F1 Score	Train Time (s)
poly	0.1	scale	69.05	0.716	21.2
poly	1.0	scale	82.15	0.827	11.3
poly	10.0	scale	85.15	0.852	9.4
poly	0.1	auto	69.05	0.716	21.3
poly	1.0	auto	82.15	0.827	11.5
poly	10.0	auto	85.15	0.852	9.2
rbf	0.1	scale	78.65	0.783	14.5
rbf	1.0	scale	85.00	0.849	9.4
rbf	10.0	scale	86.30	0.862	9.1
rbf	0.1	auto	78.65	0.783	14.8
rbf	1.0	auto	85.00	0.849	9.1
rbf	10.0	auto	86.30	0.862	9.0

3.3 SVM Visualization

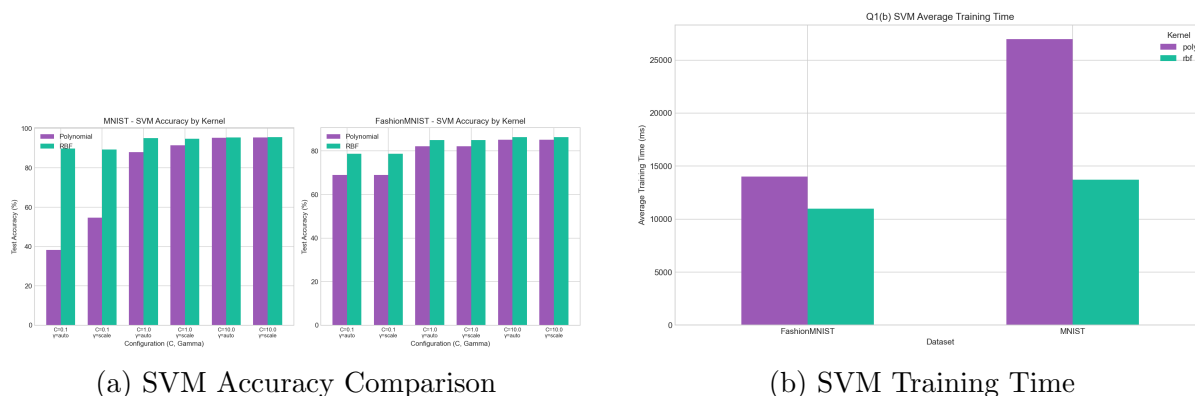


Figure 5: SVM performance metrics across different kernel and parameter configurations.

3.4 SVM vs Deep Learning Comparison

Table 7: Best Results: SVM vs ResNet Comparison

Dataset	Method	Configuration	Accuracy (%)	Improvement
MNIST	SVM	RBF, C=10.0, gamma=scale	95.60	–
MNIST	ResNet-18	Adam, lr=0.001, bs=16	99.01	+3.41%
FashionMNIST	SVM	RBF, C=10.0, gamma=scale	86.30	–
FashionMNIST	ResNet-18	Adam, lr=0.001, bs=32	90.76	+4.46%

3.5 Q1(b) Key Findings

- Kernel Performance:** RBF kernel consistently outperforms polynomial kernel on both datasets.
- Regularization (C):** Higher C values (C=10.0) provide the best accuracy.
- Gamma Setting:** “scale” gamma generally performs comparably or better than “auto”.
- Training Time:** Higher C values also result in faster training times.
- Deep Learning Superiority:** ResNet models significantly outperform SVM baselines by **+3.41%** on MNIST and **+4.46%** on FashionMNIST.

4 Q2: CPU vs GPU Performance Comparison

This section compares training performance between CPU and GPU (CUDA) on the FashionMNIST dataset.

4.1 Performance Results

Table 8: CPU vs GPU Training Comparison on FashionMNIST

Device	BS	Opt	LR	R18 Acc	R50 Acc	R18 Time	R50 Time
CPU	32	SGD	0.001	88.54%	84.84%	341.6s	1064.1s
CPU	32	Adam	0.001	88.23%	87.11%	470.8s	1281.0s
CUDA	32	SGD	0.001	87.97%	84.18%	63.0s	111.7s
CUDA	32	Adam	0.001	86.14%	86.96%	69.8s	125.7s

4.2 GPU Speedup Analysis

Table 9: GPU Speedup Over CPU

Model	Optimizer	CPU Time (s)	GPU Time (s)	Speedup
ResNet-18	SGD	341.6	63.0	5.43×
ResNet-18	Adam	470.8	69.8	6.75×
ResNet-50	SGD	1064.1	111.7	9.53×
ResNet-50	Adam	1281.0	125.7	10.19×

Maximum GPU Speedup: ResNet-50 with Adam achieves **10.19×** speedup on GPU compared to CPU training.

4.3 FLOPs Comparison

Table 10: Model Complexity (FLOPs)

Model	Parameters	FLOPs	Relative Complexity
ResNet-18	11.18M	33.18M	1.0×
ResNet-50	23.52M	78.76M	2.37×

4.4 Performance Visualizations

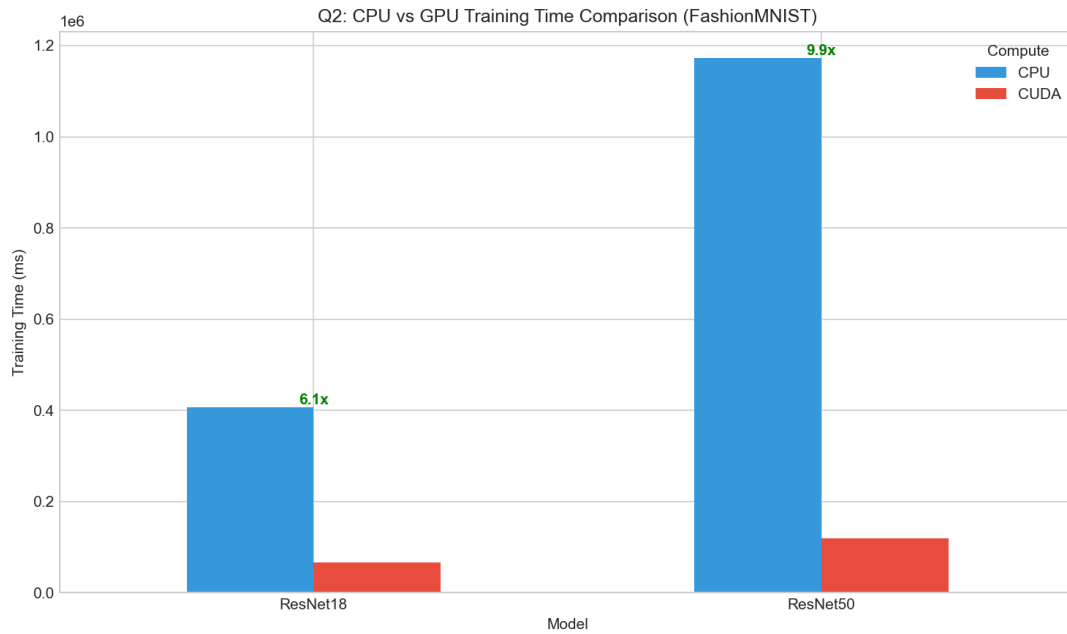


Figure 6: Training time comparison between CPU and GPU for different model configurations.

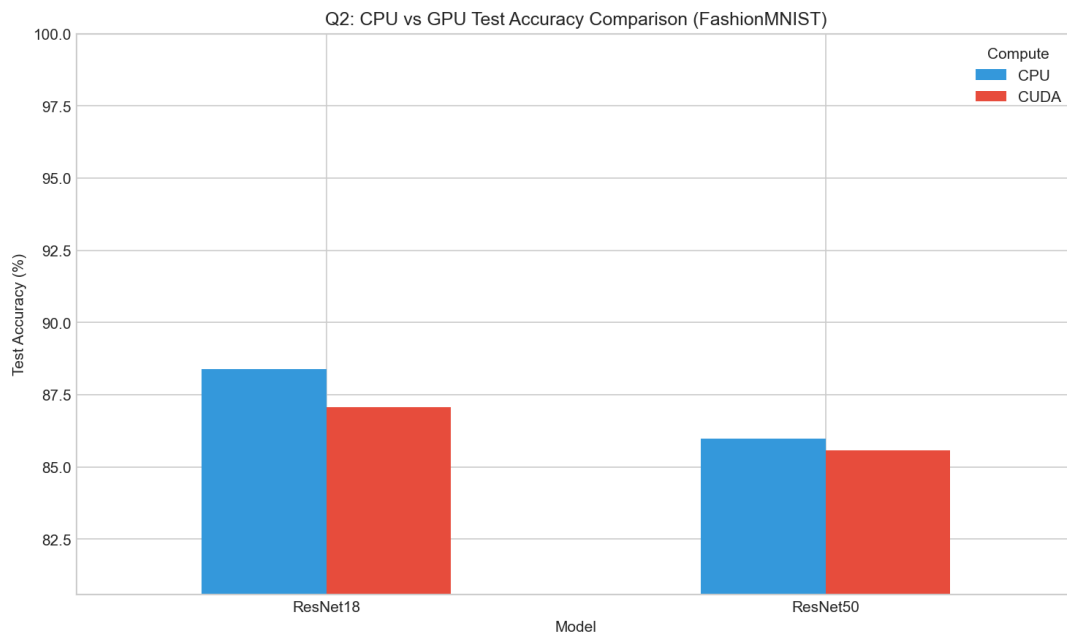


Figure 7: Accuracy comparison between CPU and GPU training.

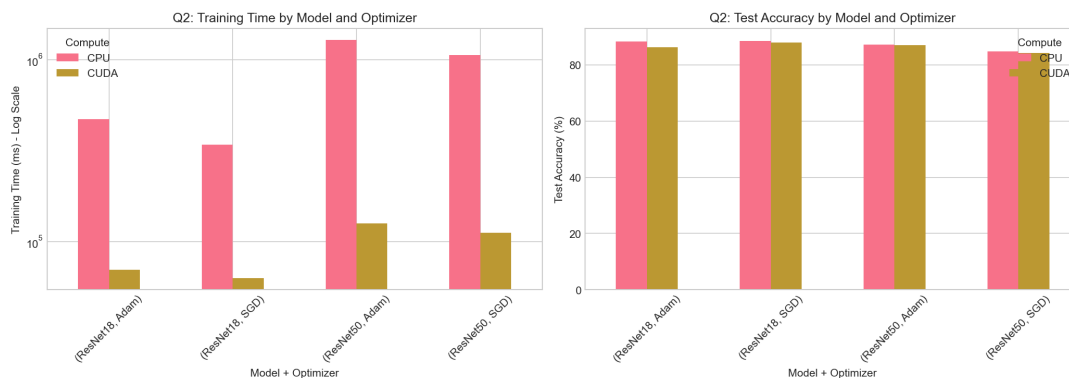


Figure 8: Combined view of CPU vs GPU performance metrics.

4.5 Q2 Key Findings

1. **GPU Acceleration:** GPU provides significant speedup ranging from $5.43\times$ to $10.19\times$ compared to CPU training.
2. **Model Complexity Benefit:** Larger models (ResNet-50) benefit more from GPU acceleration than smaller models (ResNet-18).
3. **Accuracy Consistency:** Both CPU and GPU training achieve comparable accuracies, confirming that GPU training doesn't compromise model quality.
4. **FLOPs Impact:** ResNet-50 requires $2.37\times$ more FLOPs than ResNet-18, explaining the greater GPU speedup benefit.
5. **Optimizer Overhead:** Adam optimizer has slightly higher overhead than SGD on both CPU and GPU.

5 Conclusions

5.1 Summary of Best Results

Table 11: Best Configurations Summary

Task	Best Configuration	Accuracy	F1 Score
MNIST (Deep Learning)	ResNet-18, Adam, lr=0.001, bs=16	99.01%	0.990
FashionMNIST (Deep Learning)	ResNet-18, Adam, lr=0.001, bs=32	90.76%	0.907
MNIST (SVM)	RBF, C=10.0, gamma=scale	95.60%	0.956
FashionMNIST (SVM)	RBF, C=10.0, gamma=scale	86.30%	0.862

5.2 Key Takeaways

- Model Selection:** ResNet-18 is the optimal choice for MNIST and FashionMNIST, balancing computational efficiency with high accuracy. The larger ResNet-50 shows signs of overfitting on these relatively simple datasets.
- Hyperparameter Insights:**
 - Adam optimizer with learning rate 0.001 yields the best results
 - Batch size has minimal impact (16 vs 32)
 - 5 epochs provide better convergence than 2 epochs
- Deep Learning vs Traditional ML:** ResNet significantly outperforms SVM baselines (+3.41% on MNIST, +4.46% on FashionMNIST), justifying the additional computational cost.
- Hardware Acceleration:** GPU training is essential for efficiency, offering **5-10× speedups**. This benefit increases with model complexity, making GPU crucial for training larger architectures.
- Practical Recommendations:**
 - Use ResNet-18 for similar image classification tasks
 - Prefer Adam optimizer with lr=0.001
 - Always use GPU when available for training

End of Report

Vasishth Bhatt (M25CSA007)
January 2026