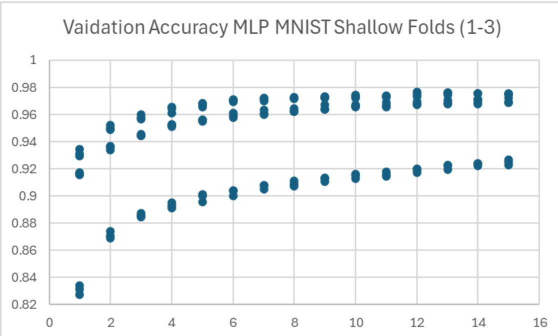
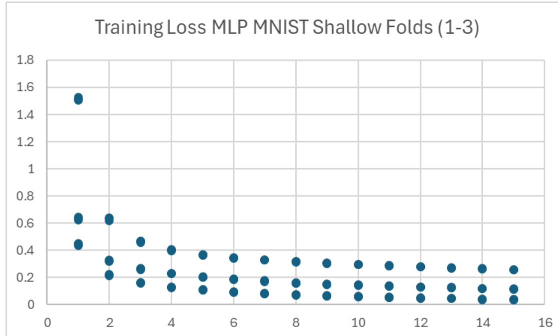
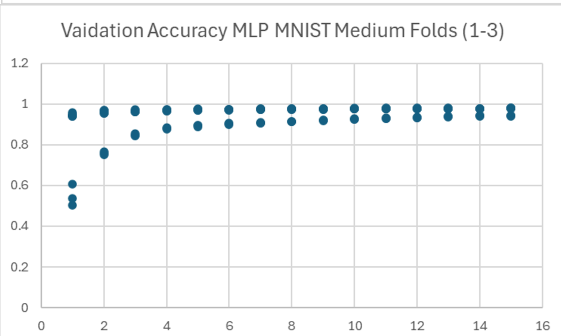
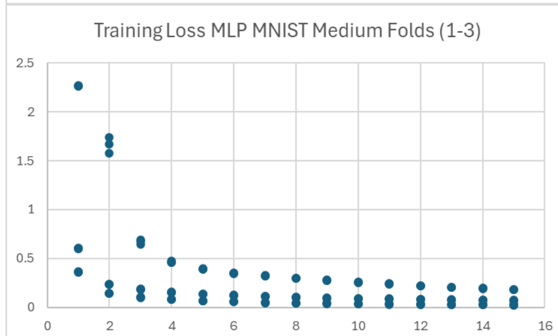
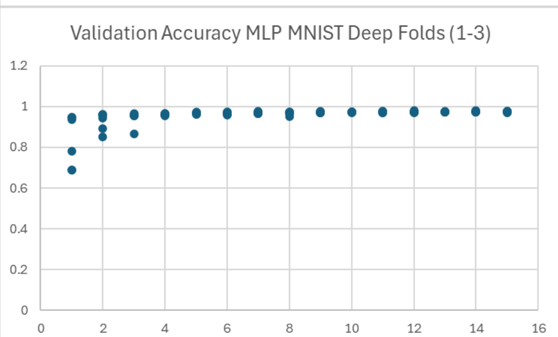
Jerin Mulangan
4375.004
Project 03 Rahman

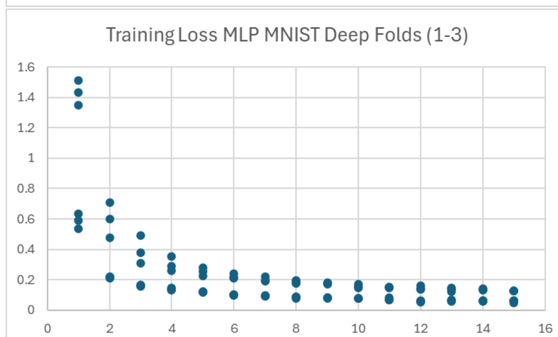






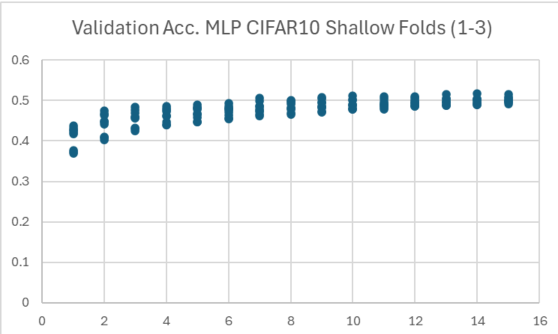


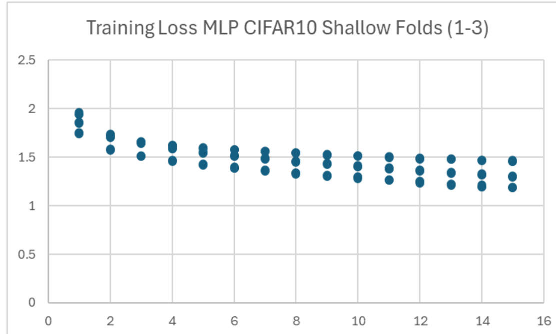


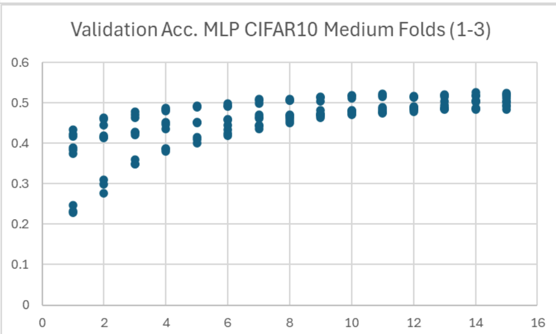


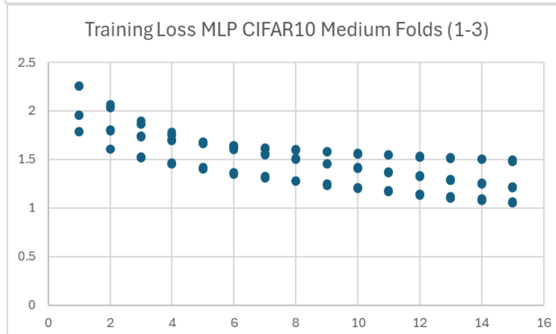
MNIST MLP

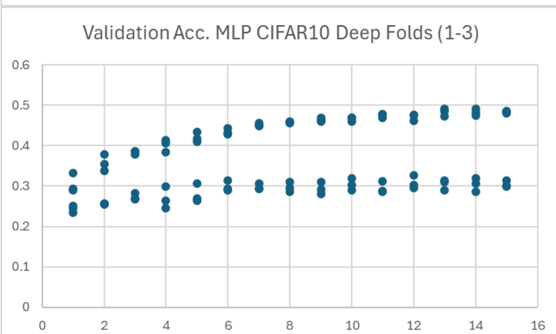
Dataset Fold	Learning Rate	Batch Size	Optimizer	Mean Acc.	Runtime (sec)	STD
Shallow Fold 1	.01	64	SGD	0.924800	287.267176	0.001476
Shallow Fold 2	.001	64	Adam	0.975267	301.361373	0.000452
Shallow Fold 3	.001	128	Adam	0.970033	272.205589	0.001574
Medium Fold 1	.01	64	SGD	0.942983	300.147645	0.001614
Medium Fold 2	.001	64	Adam	0.979150	318.651584	0.000815
Medium Fold 3	.001	128	Adam	0.978817	281.078878	0.000779
Deep Fold 1	.01	64	SGD	0.969517	383.270886	0.002618
Deep Fold 2	.001	64	Adam	0.974267	468.275732	0.002193
Deep Fold 3	.001	128	Adam	0.975100	374.636354	0.001501

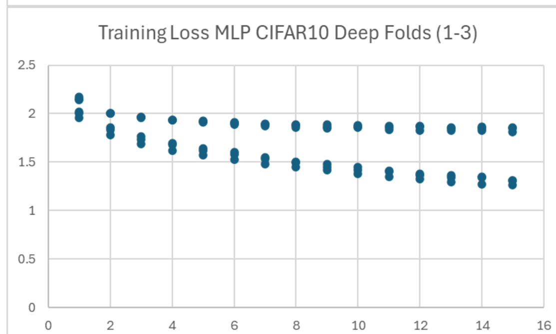






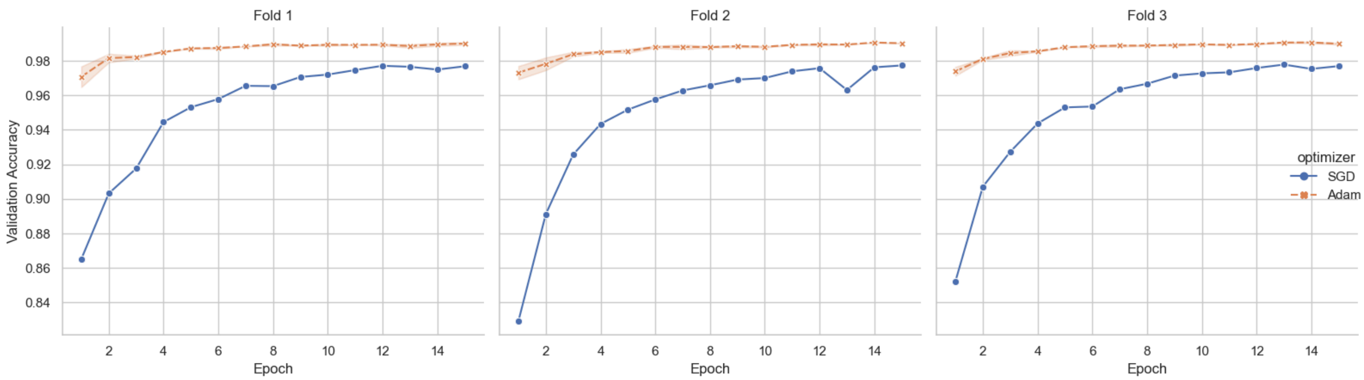


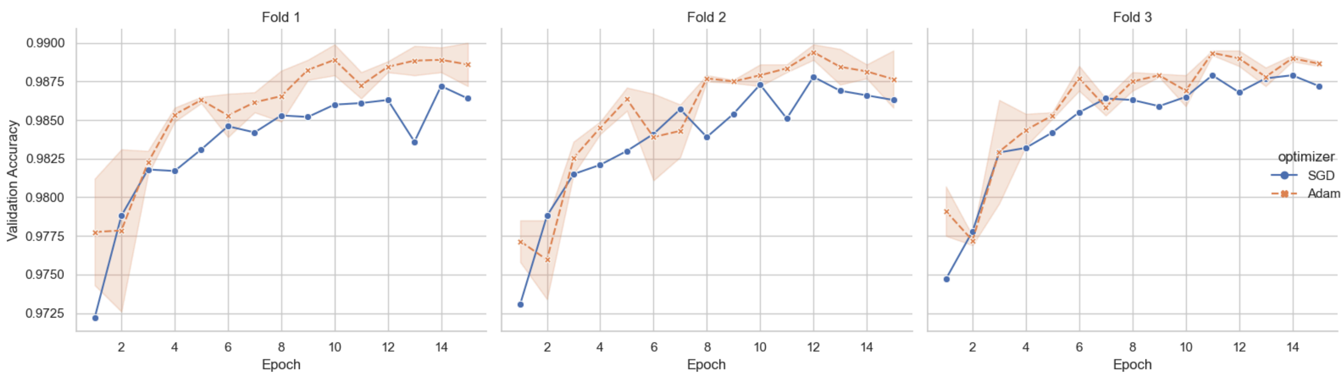


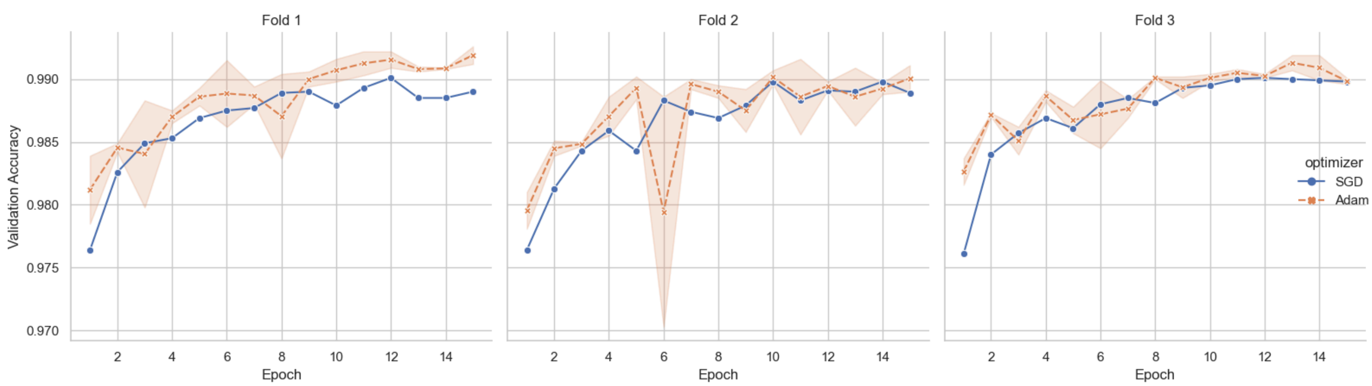


CIFAR10 MLP

Dataset Fold	Learning Rate	Batch Size	Optimizer	Mean Acc.	Runtime (sec)	STD
Shallow Fold 1	.01	64	SGD	0.49796	517.036499	0.002143
Shallow Fold 2	.001	64	Adam	0.50988	522.361284	0.004103
Shallow Fold 3	.001	128	Adam	0.49694	498.982522	0.004015
Medium Fold 1	.01	64	SGD	0.50604	531.261492	0.003839
Medium Fold 2	.001	64	Adam	0.52096	550.165645	0.003099
Medium Fold 3	.001	128	Adam	0.48940	513.392825	0.004593
Deep Fold 1	.01	64	SGD	0.18750	641.815964	0.017013
Deep Fold 2	.001	64	Adam	0.48326	756.058280	0.002062
Deep Fold 3	.001	128	Adam	0.30428	665.356551	0.006750

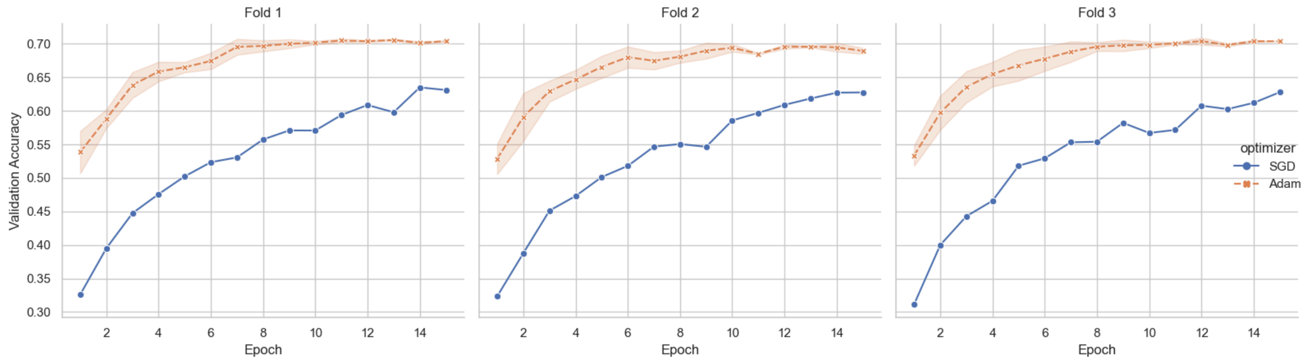


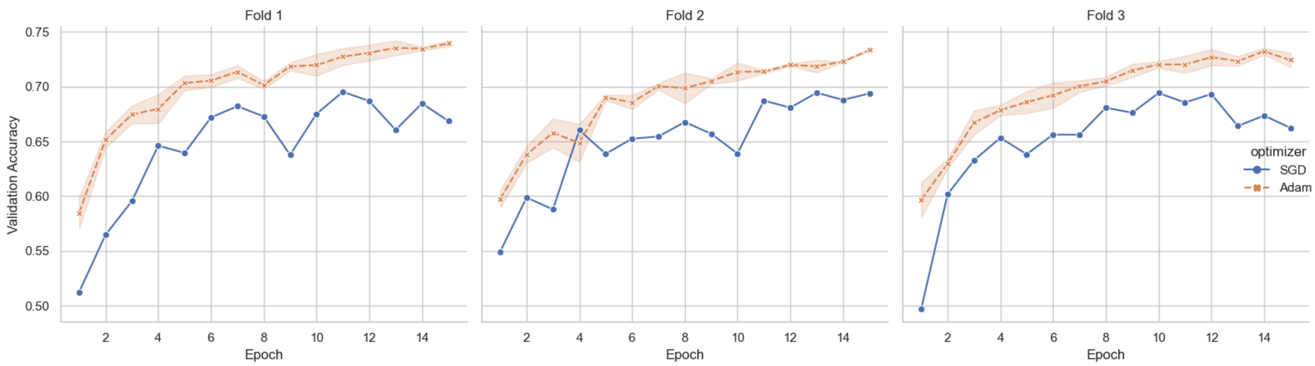


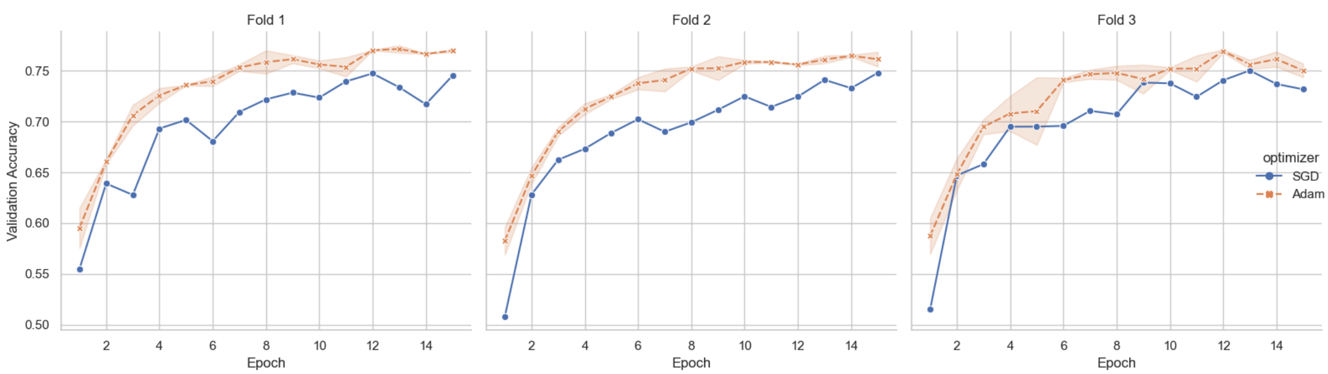


MNIST CNN

Model	Learning Rate	Batch Size	Optimizer	Mean Acc.	STD
CNNBaseline	.01	64	SGD	0.977000	0.002
CNNBaseline	.001	64	Adam	0.990450	0.001
CNNBaseline	.001	128	Adam	0.989533	0.001
CNNEnhanced	.01	64	SGD	0.986667	0.002
CNNEnhanced	.001	64	Adam	0.987183	0.002
CNNEnhanced	.001	128	Adam	0.989417	0.001
CNNDeep	.01	64	SGD	0.989233	0.001
CNNDeep	.001	64	Adam	0.990633	0.001
CNNDeep	.001	128	Adam	0.990583	0.001







CIFAR10 CNN

Model	Learning Rate	Batch Size	Optimizer	Mean Acc.	STD
CNNBaseline	.01	64	SGD	0.62890	0.020
CNNBaseline	.001	64	Adam	0.70050	0.015
CNNBaseline	.001	128	Adam	0.69792	0.017
CNNEnhanced	.01	64	SGD	0.67500	0.022
CNNEnhanced	.001	64	Adam	0.72992	0.018
CNNEnhanced	.001	128	Adam	0.73508	0.016
CNNDeep	.01	64	SGD	0.74160	0.022
CNNDeep	.001	64	Adam	0.76462	0.012
CNNDeep	.001	128	Adam	0.75616	0.015

Analysis

The Convolutional Neural Networks showed significant improvements over the Multilayer Perceptrons in most cases. The big reasons that there was such a large improvement are the specialized architectures of the CNNs. This allows them to use the spatial structure of the image data in a much more meaningful way. They can capture more local patterns like edges and textures as well as hierarchical feature representations.

CNNs are made in a way that makes them far superior than MLPs at detecting local patterns and spatial layers through their convolutional layers, pooling layers and shared weights. This is very essential for tasks like this specific project that relied on image classification, where neighboring pixels contain information that needs to be parsed and then aggregated. MLPs would treat these pixels as independent and would not use spatula structures, which is why the average was lower.

CNNs achieve a significant reduction in the number of parameters by sharing weights across different parts of the image, making them more efficient than MLPs. This reduction in complexity helps CNNs generalize better, reducing overfitting. MLPs, being fully connected, have a large number of parameters, especially for high-dimensional data like images, which can lead to overfitting and slower convergence.

Performance

MNIST

CNNs consistently outperformed the MLPs on the MNIST dataset. At the highest points, even if the MLPs are able to reach largely high accuracies, the average accuracy for the CNNs were higher and had lower standard deviations. This would mean that it had more stable and reliable performances overall.

At its peak, the MLP models reached around 0.99 for a single epoch, but its single best average was around a 0.979 accuracy. Whereas the CNNs usually reached around 0.97-0.99 for most of the datasets.The CNN Deep model with learning rate 0.001 and batch size 64 reached an average accuracy of 0.9906, which just shows CNN’s ability to generalize well on the test set.

CNNBaseline shows high accuracy, with a slight advantage in accuracy when using the Adam optimizer (0.990450 vs 0.977000 with SGD). The learning rate of 0.001 leads to higher accuracy than a learning rate of 0.01. CNNEnhanced and CNNDeep further improve on CNNBaseline, with CNNDeep achieving the highest accuracy of 0.990633. The small improvements are probably because of the deeper network architecture, which can learn more complex patterns.

CIFAR10

CNNs still outperformed the MLPs on the CIFAR10 datasets but not to the extent that it had previously with the MNIST datasets. This is probably because of the increased complexity of the CIFAR-10 dataset, which, after researching the dataset on the internet, seems to involve more challenging, real-world images with higher variance in terms of object types and backgrounds. The highest mean accuracy (0.52096) on the CIFAR10 datasets for MLPs was much lower than the highest mean accuracy for the MNIST datasets.

CNN Deep model still had the best accuracy scores overall for all the datasets. CNNDeep with the Adam optimizer (learning rate 0.001, batch size 64) achieves the best result of 0.76462. While this is a significant improvement over MLPs, it was still lower than what the CNNs had on simpler datasets like MNIST. CNNBaseline achieves 0.62890 accuracy with SGD and 0.70050 with Adam, showing the advantage of Adam in this case. CNNEnhanced and CNNDeep show improvements, especially with the Adam optimizer and larger batch sizes. CNNDeep with a batch size of 64 and Adam optimizer gives the best performance (0.76462), reflecting the added depth and complexity in learning from the CIFAR-10 dataset.

Summary

CNNBaseline: Accuracy improves with the Adam optimizer, and performance is good. Although deeper networks perform better, it appears to capture enough fundamental features for both datasets, as proven by its 0.990450 and 0.70050 scores for MNIST and CIFAR-10.

CNNEnhanced: Shows a slight improvement over CNNBaseline, with greater accuracies in CIFAR-10 (0.73508) and MNIST (0.989417). This suggests that adding enhancements to both datasets—perhaps by adding more layers or changing the architecture—helps, while MNIST shows a greater improvement.

The most intricate architecture, CNNDeep, excels in both datasets. With the maximum accuracy of 0.990633 for MNIST and 0.76462 for CIFAR-10, it is evident that deeper models are more appropriate for managing complicated datasets, even though they have somewhat higher variability (as indicated by the standard deviation).