CS770 Machine Learning

Assignment 3

05/03/2025

Submitted by,

S M Asif Hossain

myWSU: Y935M825

Email: sxhossain10@shockers.wichita.edu

Introduction

In this project, the Fashion MNIST dataset was used to classify grayscale images of clothing items into ten categories. The dataset consists of 60,000 training images and 10,000 testing images, each with a resolution of 28×28 pixels. A variety of deep learning models, including a shallow neural network and several convolutional neural networks were developed, trained, and evaluated. The primary objective was to build and assess these models individually while applying hyperparameter tuning techniques to improve performance. Accuracy was used as the main evaluation metric, and confusion matrices along with classification reports were generated to provide a detailed evaluation.

Methodology

Three models were built and tested using the Fashion MNIST dataset. Each model was trained separately and its performance was evaluated. The models were designed with increasing complexity. Hyperparameter tuning was also done to improve the results. All models were trained for 10 epochs and tested using the same data to keep the comparison fair.

Before training the models, the dataset was prepared through several steps. The Fashion MNIST dataset consisted of 60,000 training images and 10,000 test images. From the training set, 20% of the data was separated to create a validation set. Each image was grayscale with a size of 28×28 pixels. The pixel values were normalized to a range between 0 and 1. Sample images from each class were visualized. The images were reshaped to fit the input format required for convolutional layers, and the class labels were converted to one-hot encoded vectors to be used for classification.

Model 1 was a shallow neural network. It had only one Dense layer with 10 output units and softmax activation. Initially, stochastic gradient descent (SGD) was used as optimizer. No hidden layer was used. This model was built as a basic starting point. Hyperparameter tuning was done by testing different optimizers. At first, Adam optimizer was used. Later, the SGD was applied. The learning rate and batch size were also adjusted to observe their effect on performance.

Model 2 was a basic CNN. It had one Conv2D layer with 32 filters and a 3×3 kernel, followed by a Flatten layer and a Dense output layer with 10 units using softmax activation. ReLU was used as the activation function in the Conv2D layer. In the tuned versions, dropout was added after the convolutional layer. The optimizer was changed from Adam to RMSprop in one version, and the number of filters was increased to 64 in another. The batch size remained consistent at 64 across all models, and the learning rate was set to 0.001 in the tuned versions.

Model 3 was a deep CNN where multiple Conv2D layers. Initially, two Conv2D layers were used, followed by a MaxPooling2D layer to reduce spatial dimensions. In the tuned versions, dropout was added after the Conv2D layers, and the number of units in the dense layers was increased. In a further tuned version, a third Conv2D layer was added to deepen the architecture and enhance feature learning. The number of filters was also increased in deeper layers. A Flatten layer was applied before the dense layers. Additionally, the learning rate was reduced. Different combinations were tested to find the best results.

Methodology of Extra Task (Q2)

For the extra task, a custom neural network was built using a combination of fully connected layers, batch normalization, dropout, and input concatenation. Before training, data augmentation was applied to the training set. Techniques such as random rotations, shifts, zoom, and horizontal flips were used. The input image was first flattened to convert the spatial format into a 1D vector. Two Dense layers with 30 units each and ReLU activation were added. Batch normalization was applied after each Dense layer and dropout was used. The original flattened input was concatenated with the output of the second hidden layer. This combined vector was then passed through a final Dense layer with 10 units and softmax activation to produce class probabilities.

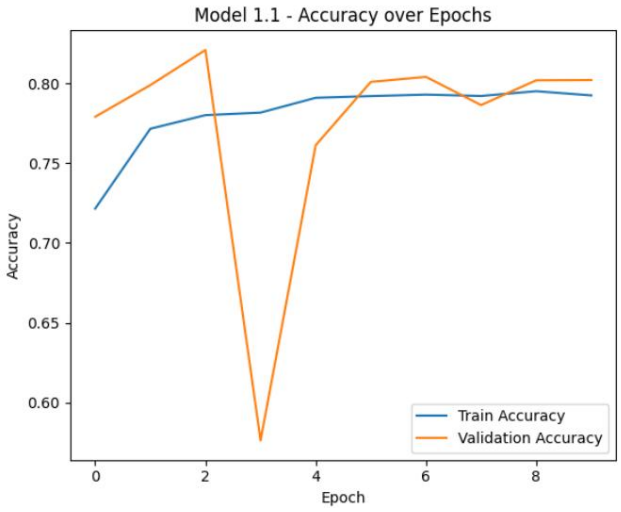
Code of Assignment 3 in Google Colab:
https://colab.research.google.com/drive/1ELQe8QKDthVaPwgP_KSIWaZ1PY0t_2Zk?usp=sharing

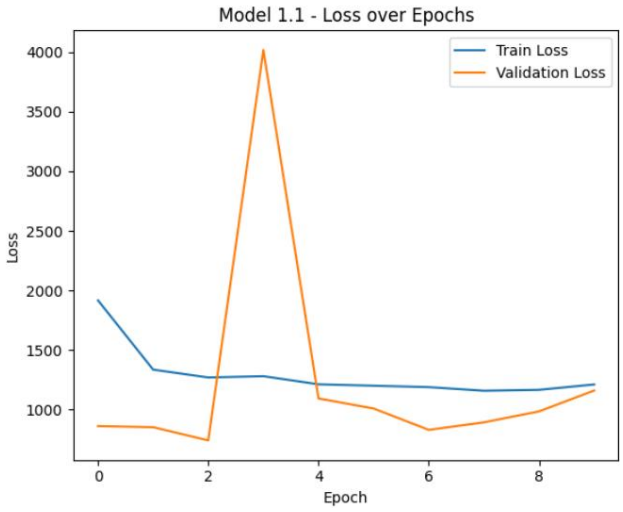
Results

Model 1 was a shallow neural network. Model 1.1 used a basic architecture with the SGD optimizer. Model 1.2 applied the Adam optimizer with lower learning rate, and Model 1.3 experimented with a larger batch size using SGD. The comparison table, training curves, confusion matrix, and classification reports of each model are provided below.

Model	Optimizer	Learning Rate	Batch Size	Dropout	Accuracy
1.1	SGD	0.01	64	No	0.7885
1.2	Adam	0.001	64	No	0.8456
1.3	SGD	0.01	128	No	0.8057

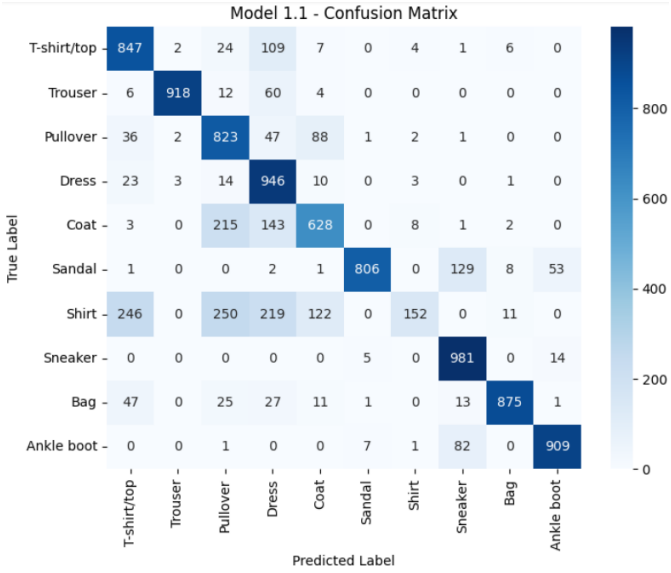
Training curves, Classification Report and Confusion Matrix of **Model 1.1** are listed below:



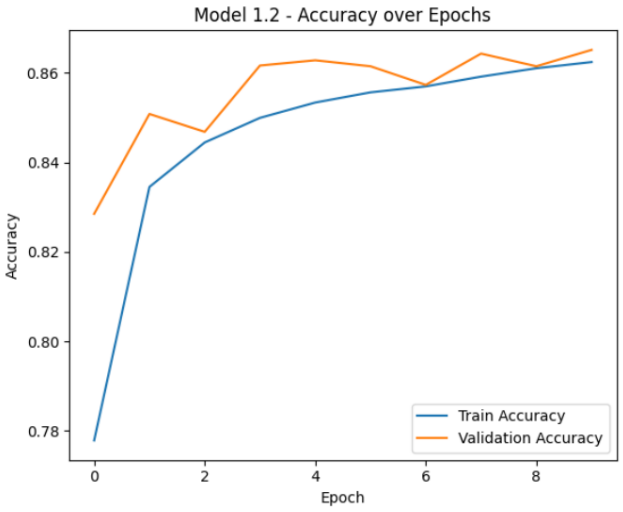


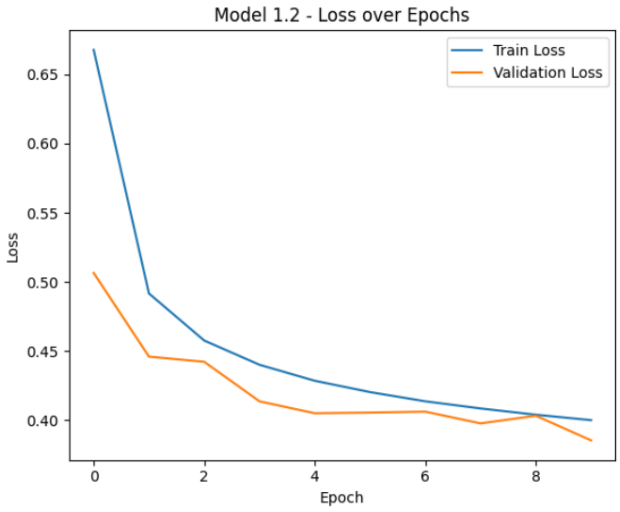
Classification Report for Model 1.1:

	precision	recall	f1-score	support
T-shirt/top	0.70	0.85	0.77	1000
Trouser	0.99	0.92	0.95	1000
Pullover	0.60	0.82	0.70	1000
Dress	0.61	0.95	0.74	1000
Coat	0.72	0.63	0.67	1000
Sandal	0.98	0.81	0.89	1000
Shirt	0.89	0.15	0.26	1000
Sneaker	0.81	0.98	0.89	1000
Bag	0.97	0.88	0.92	1000
Ankle boot	0.93	0.91	0.92	1000
accuracy			0.79	10000
macro avg	0.82	0.79	0.77	10000
weighted avg	0.82	0.79	0.77	10000



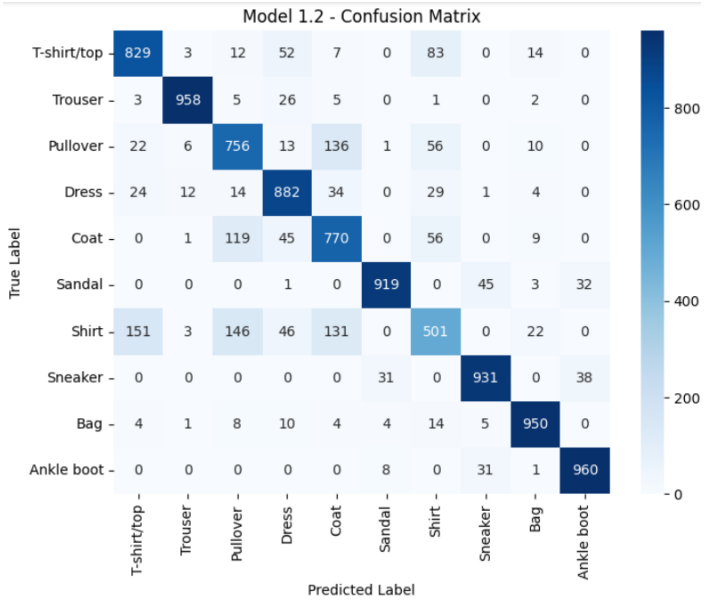
Training curves, Classification Report and Confusion Matrix of **Model 1.2** are listed below:



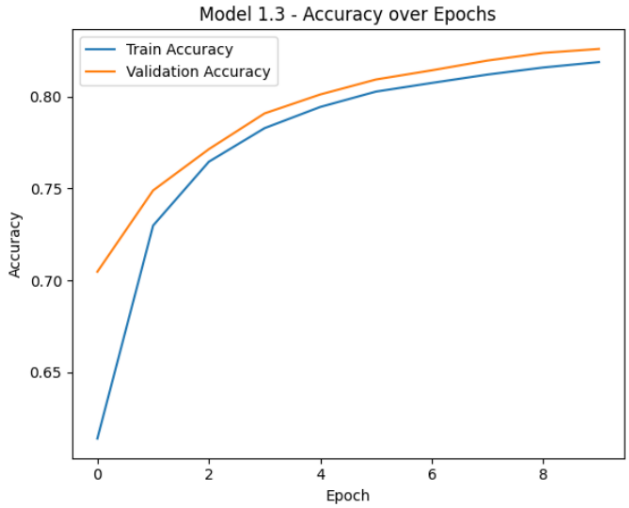


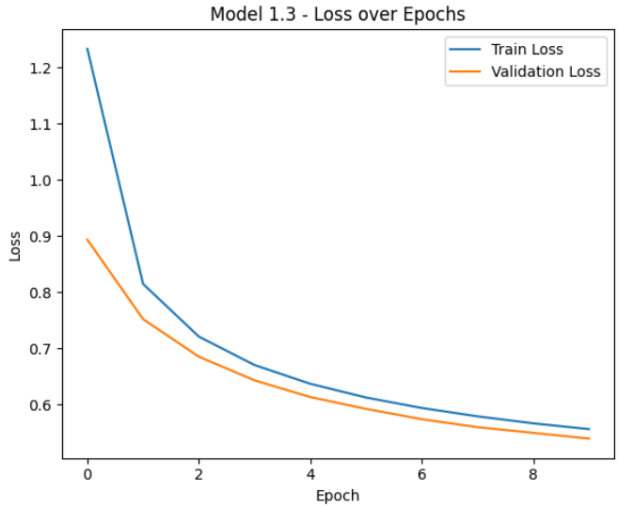
Classification Report for Model 1.2:

	precision	recall	f1-score	support
T-shirt/top	0.80	0.83	0.82	1000
Trouser	0.97	0.96	0.97	1000
Pullover	0.71	0.76	0.73	1000
Dress	0.82	0.88	0.85	1000
Coat	0.71	0.77	0.74	1000
Sandal	0.95	0.92	0.94	1000
Shirt	0.68	0.50	0.58	1000
Sneaker	0.92	0.93	0.92	1000
Bag	0.94	0.95	0.94	1000
Ankle boot	0.93	0.96	0.95	1000
accuracy			0.85	10000
macro avg	0.84	0.85	0.84	10000
weighted avg	0.84	0.85	0.84	10000

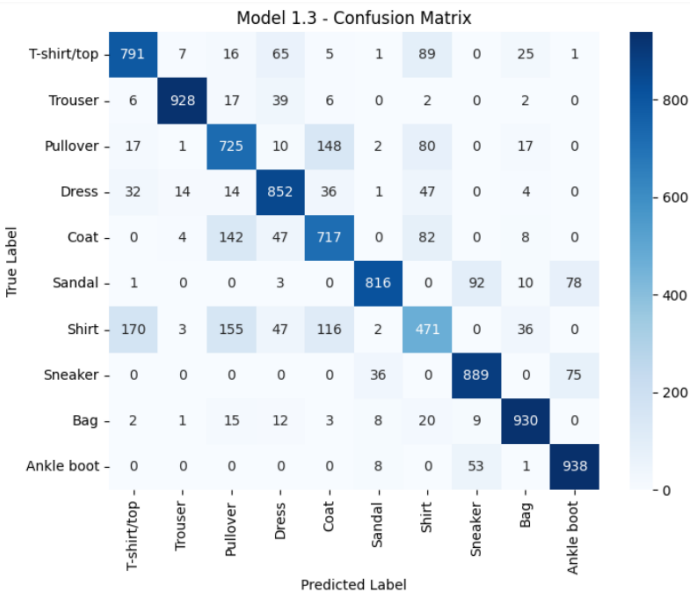


Training curves, Classification Report and Confusion Matrix of **Model 1.3** are listed below:





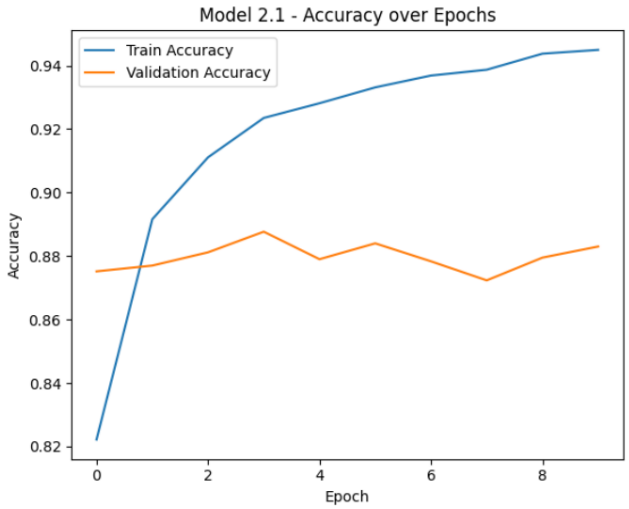

```
Classification Report for Model 1.3:

              precision    recall  f1-score   support

 T-shirt/top       0.78      0.79      0.78      1000
     Trouser       0.97      0.93      0.95      1000
    Pullover       0.67      0.72      0.70      1000
       Dress       0.79      0.85      0.82      1000
        Coat       0.70      0.72      0.71      1000
      Sandal       0.93      0.82      0.87      1000
       Shirt       0.60      0.47      0.53      1000
     Sneaker       0.85      0.89      0.87      1000
         Bag       0.90      0.93      0.91      1000
  Ankle boot       0.86      0.94      0.90      1000

    accuracy                           0.81     10000
   macro avg       0.80      0.81      0.80     10000
weighted avg       0.80      0.81      0.80     10000
```

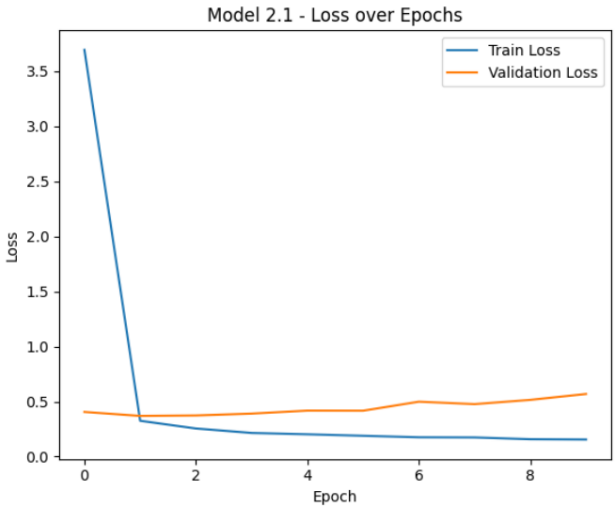


Model 2 was a basic CNN. Model 2.1 used a single Conv2D layer followed by Flatten and Dense layers. Model 2.2 and Model 2.3 were tuned versions where different hyperparameters were adjusted, including optimizer and dropout. The comparison table, training curves, confusion matrix, and classification reports of each model are provided below.

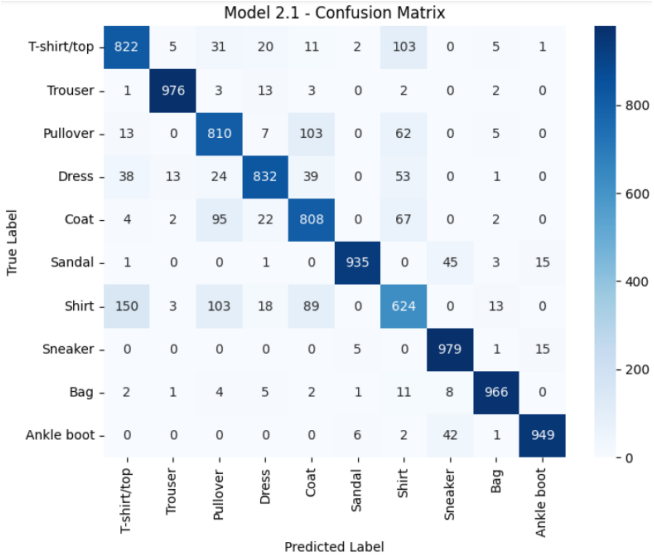
Model	Optimizer	Learning Rate	Batch Size	Dropout	Accuracy
2.1	Adam	0.001	64	No	0.8701
2.2	RMSprop	0.001	64	0.3	0.8414
2.3	Adam	0.001	64	0.3	0.8729

Training curves, Classification Report and Confusion Matrix of **Model 2.1** are listed below:

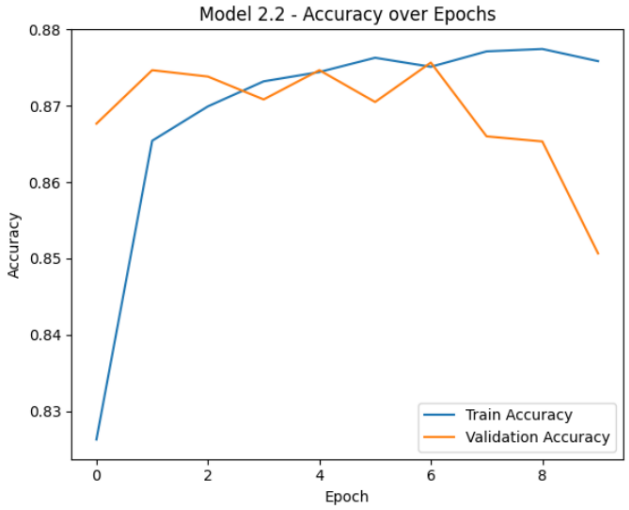


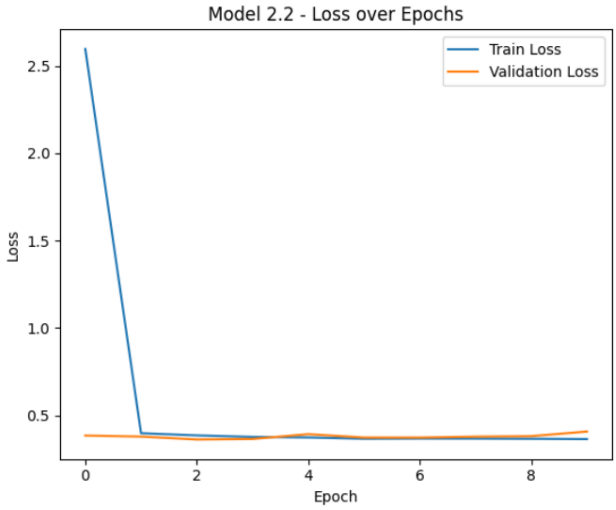



```
Classification Report for Model 2.1:

              precision    recall  f1-score   support

 T-shirt/top       0.80      0.82      0.81      1000
     Trouser       0.98      0.98      0.98      1000
    Pullover       0.76      0.81      0.78      1000
       Dress       0.91      0.83      0.87      1000
        Coat       0.77      0.81      0.79      1000
      Sandal       0.99      0.94      0.96      1000
       Shirt       0.68      0.62      0.65      1000
     Sneaker       0.91      0.98      0.94      1000
         Bag       0.97      0.97      0.97      1000
  Ankle boot       0.97      0.95      0.96      1000

    accuracy                           0.87     10000
   macro avg       0.87      0.87      0.87     10000
weighted avg       0.87      0.87      0.87     10000
```

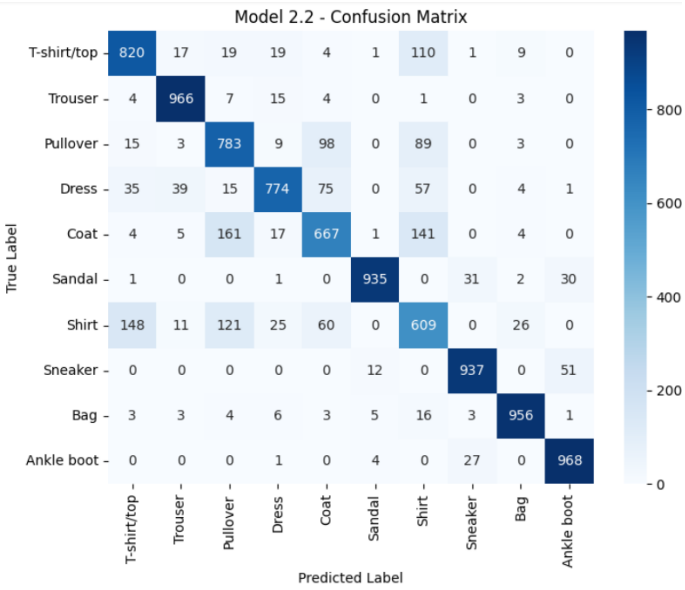


Training curves, Classification Report and Confusion Matrix of **Model 2.2** are listed below:

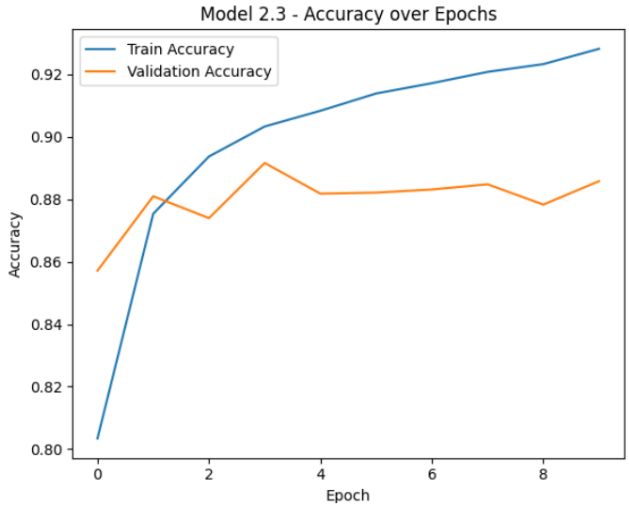


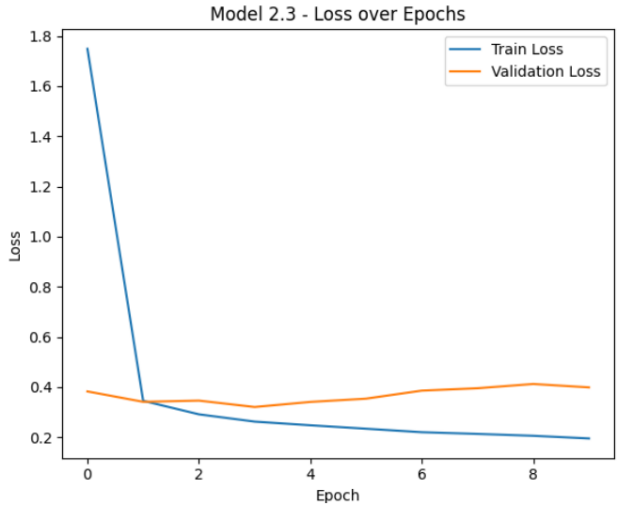



```
Classification Report for Model 2.2:

              precision    recall  f1-score   support

 T-shirt/top       0.80      0.82      0.81      1000
     Trouser       0.93      0.97      0.95      1000
    Pullover       0.71      0.78      0.74      1000
       Dress       0.89      0.77      0.83      1000
        Coat       0.73      0.67      0.70      1000
      Sandal       0.98      0.94      0.96      1000
       Shirt       0.60      0.61      0.60      1000
     Sneaker       0.94      0.94      0.94      1000
         Bag       0.95      0.96      0.95      1000
  Ankle boot       0.92      0.97      0.94      1000

    accuracy                           0.84     10000
   macro avg       0.84      0.84      0.84     10000
weighted avg       0.84      0.84      0.84     10000
```



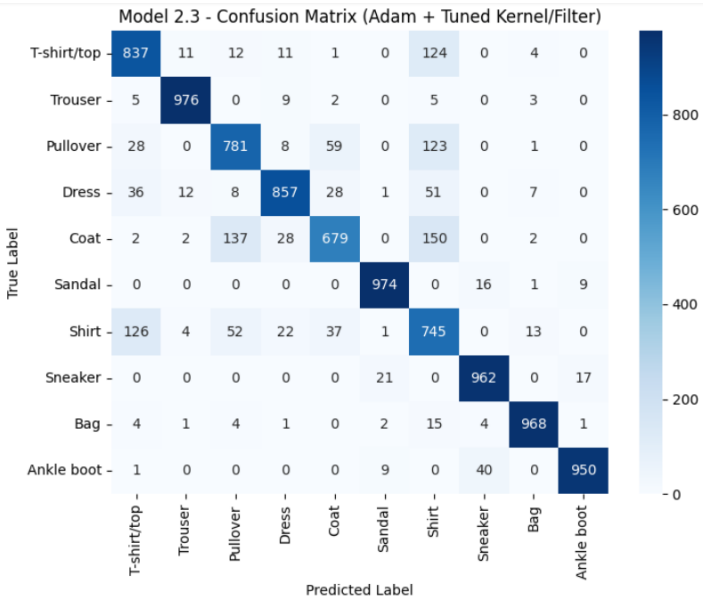
Training curves, Classification Report and Confusion Matrix of **Model 2.3** are listed below:





Classification Report for Model 2.3:

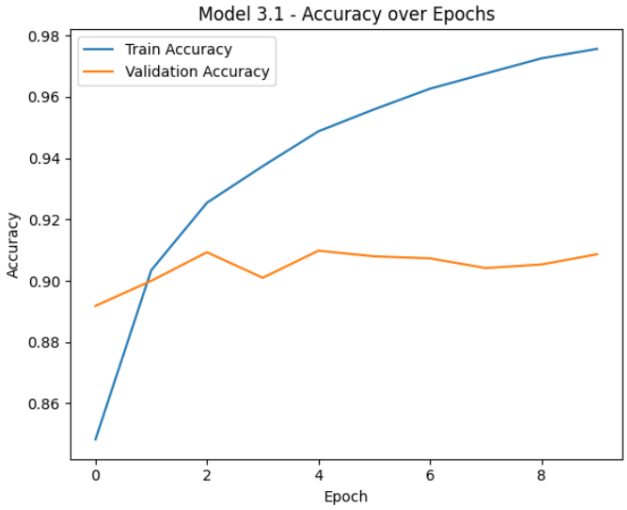
	precision	recall	f1-score	support
T-shirt/top	0.81	0.84	0.82	1000
Trouser	0.97	0.98	0.97	1000
Pullover	0.79	0.78	0.78	1000
Dress	0.92	0.86	0.89	1000
Coat	0.84	0.68	0.75	1000
Sandal	0.97	0.97	0.97	1000
Shirt	0.61	0.74	0.67	1000
Sneaker	0.94	0.96	0.95	1000
Bag	0.97	0.97	0.97	1000
Ankle boot	0.97	0.95	0.96	1000
accuracy			0.87	10000
macro avg	0.88	0.87	0.87	10000
weighted avg	0.88	0.87	0.87	10000

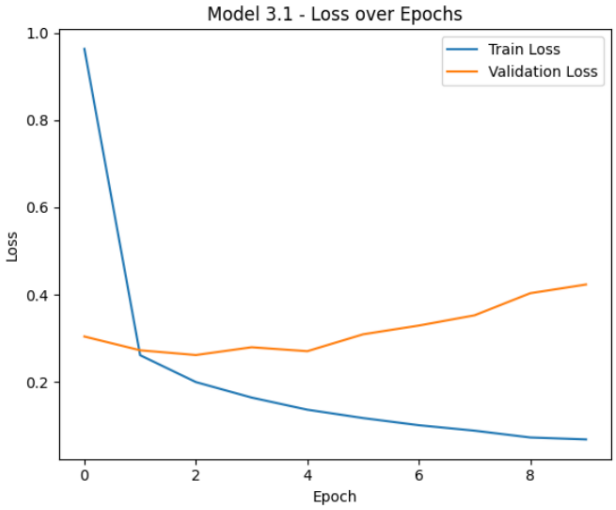


Model 3 was a deep convolutional neural network. Model 3.1 used a basic deep CNN structure with two Conv2D layers, one MaxPooling2D layer, and dense layers. Model 3.2 introduced dropout layers and increased the number of filters. Model 3.3 dropped the learning rate to improve generalization. The comparison table, training curves, confusion matrix, and classification reports of each model are provided below.

Model	Optimizer	Learning Rate	Batch Size	Dropout	Accuracy
3.1	Adam	0.001	64	No	0.8960
3.2	Adam	0.001	64	0.3	0.9016
3.3	Adam	0.0005	64	0.3	0.9180

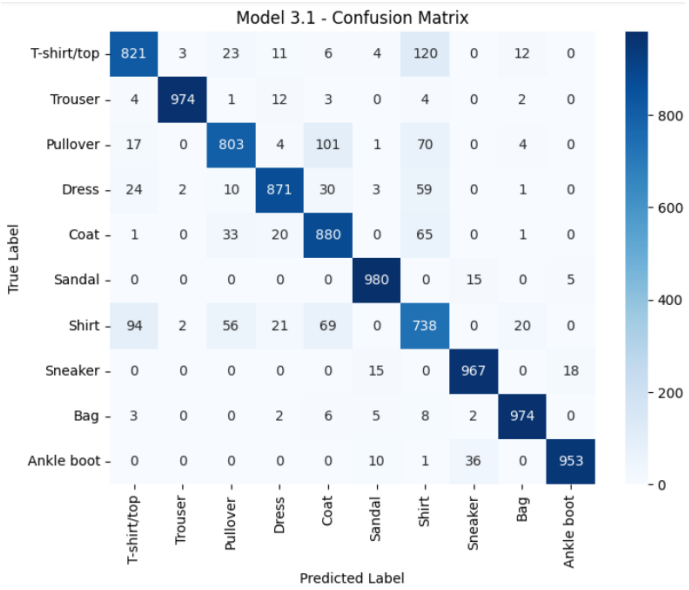
Training curves, Classification Report and Confusion Matrix of **Model 3.1** are listed below:



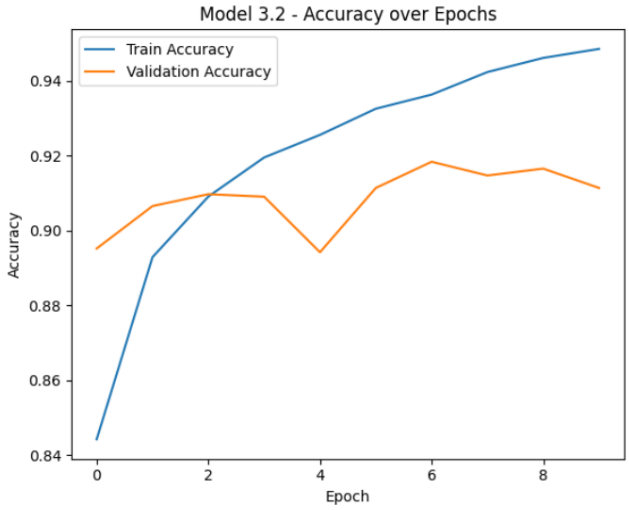


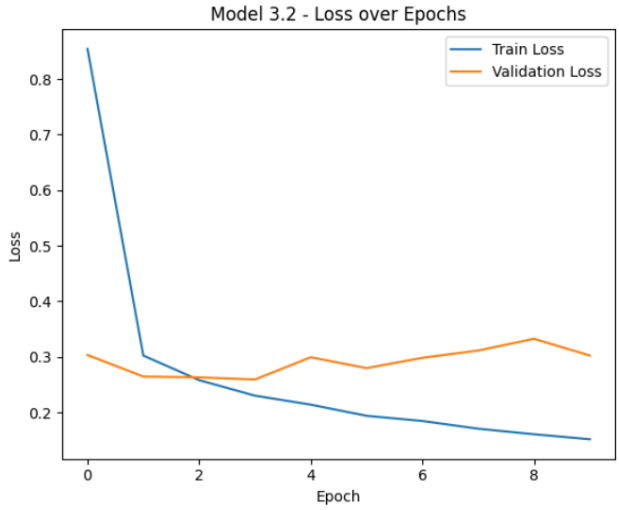
Classification Report for Model 3.1:

	precision	recall	f1-score	support
T-shirt/top	0.85	0.82	0.84	1000
Trouser	0.99	0.97	0.98	1000
Pullover	0.87	0.80	0.83	1000
Dress	0.93	0.87	0.90	1000
Coat	0.80	0.88	0.84	1000
Sandal	0.96	0.98	0.97	1000
Shirt	0.69	0.74	0.71	1000
Sneaker	0.95	0.97	0.96	1000
Bag	0.96	0.97	0.97	1000
Ankle boot	0.98	0.95	0.96	1000
accuracy			0.90	10000
macro avg	0.90	0.90	0.90	10000
weighted avg	0.90	0.90	0.90	10000



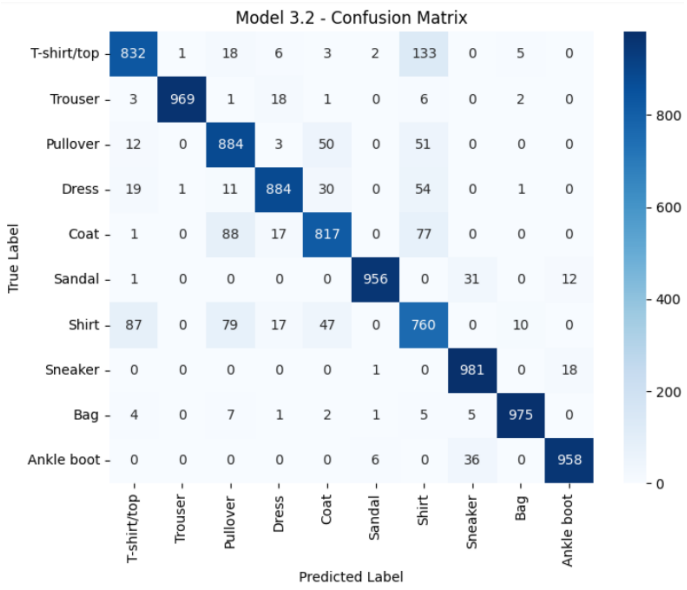
Training curves, Classification Report and Confusion Matrix of **Model 3.2** are listed below:



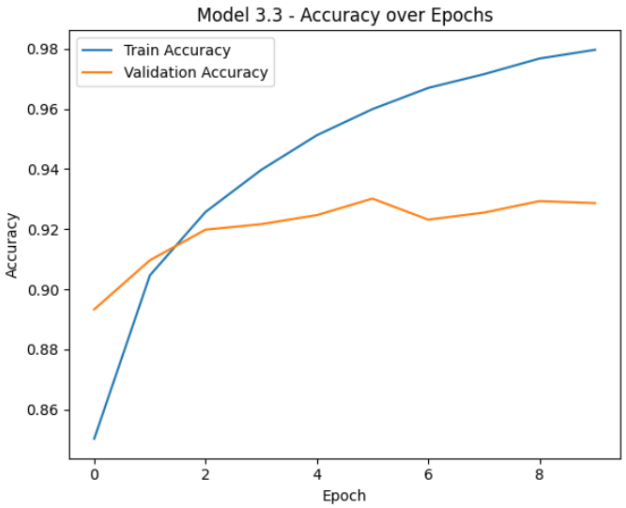


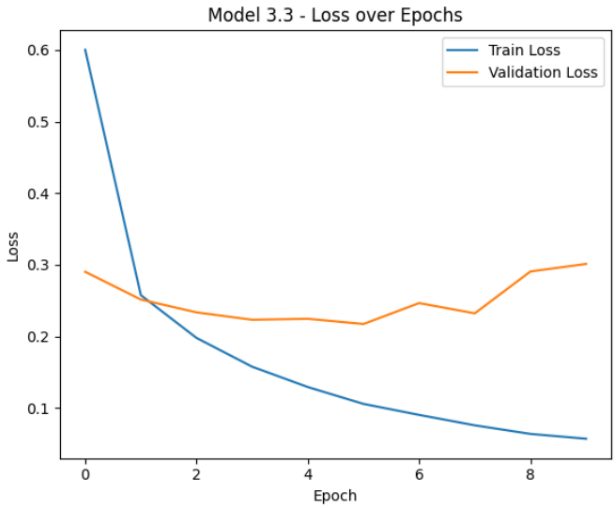
Classification Report for Model 3.2:

	precision	recall	f1-score	support
T-shirt/top	0.87	0.83	0.85	1000
Trouser	1.00	0.97	0.98	1000
Pullover	0.81	0.88	0.85	1000
Dress	0.93	0.88	0.91	1000
Coat	0.86	0.82	0.84	1000
Sandal	0.99	0.96	0.97	1000
Shirt	0.70	0.76	0.73	1000
Sneaker	0.93	0.98	0.96	1000
Bag	0.98	0.97	0.98	1000
Ankle boot	0.97	0.96	0.96	1000
accuracy			0.90	10000
macro avg	0.90	0.90	0.90	10000
weighted avg	0.90	0.90	0.90	10000



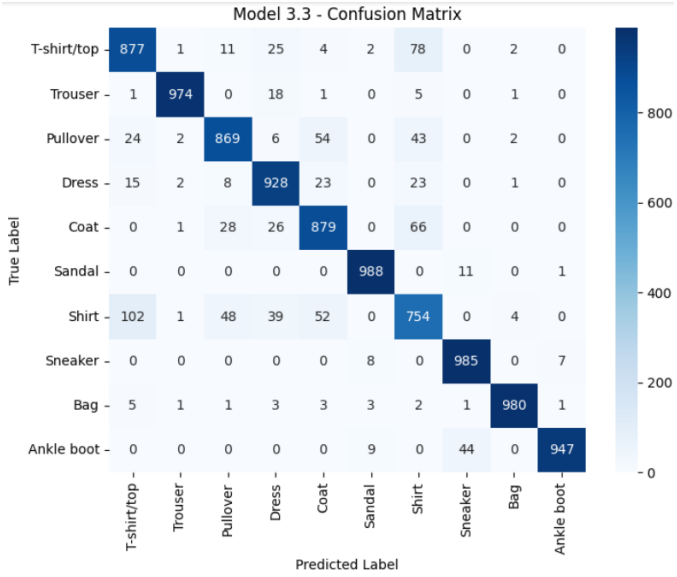
Training curves, Classification Report and Confusion Matrix of **Model 3.3** are listed below:





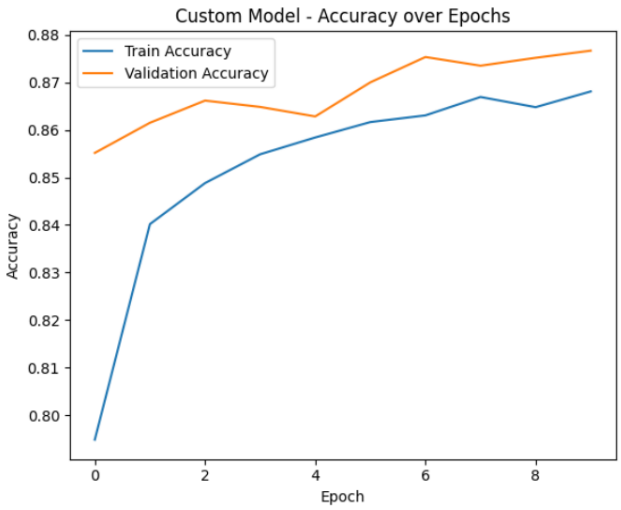
Classification Report for Model 3.3:

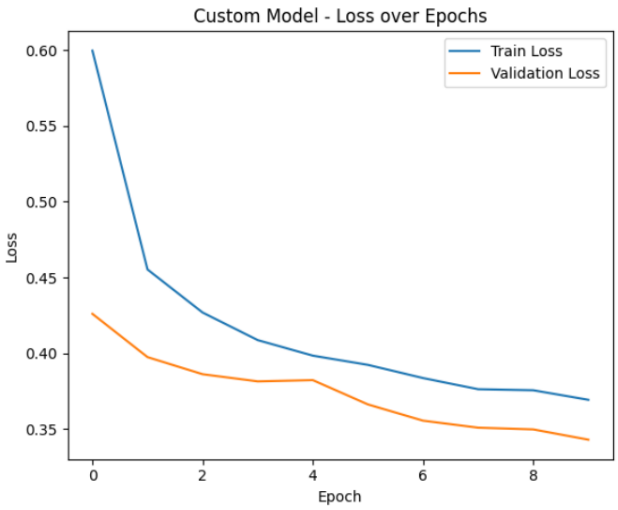
	precision	recall	f1-score	support
T-shirt/top	0.86	0.88	0.87	1000
Trouser	0.99	0.97	0.98	1000
Pullover	0.90	0.87	0.88	1000
Dress	0.89	0.93	0.91	1000
Coat	0.87	0.88	0.87	1000
Sandal	0.98	0.99	0.98	1000
Shirt	0.78	0.75	0.77	1000
Sneaker	0.95	0.98	0.97	1000
Bag	0.99	0.98	0.98	1000
Ankle boot	0.99	0.95	0.97	1000
accuracy			0.92	10000
macro avg	0.92	0.92	0.92	10000
weighted avg	0.92	0.92	0.92	10000



Results of Extra Task (Q2)

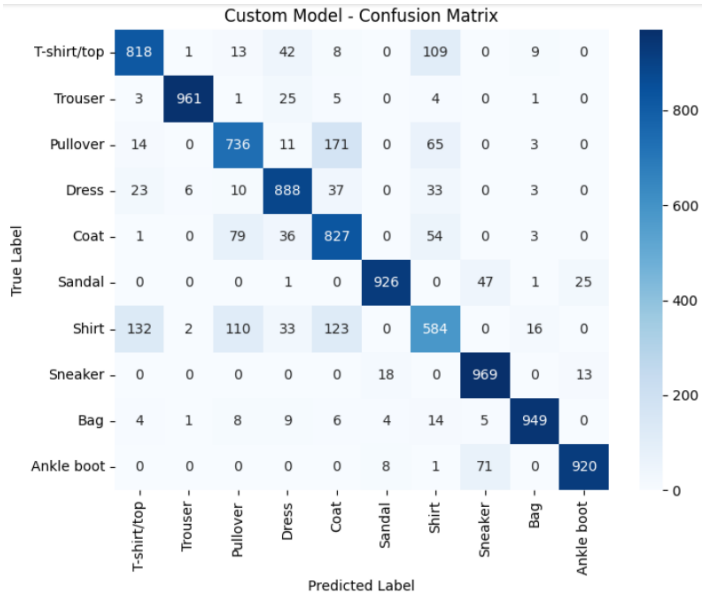
The model achieved an accuracy of 0.8578. Training curves, Classification Report and Confusion Matrix of the model is listed below:





Classification Report for Custom Model:

	precision	recall	f1-score	support
T-shirt/top	0.82	0.82	0.82	1000
Trouser	0.99	0.96	0.98	1000
Pullover	0.77	0.74	0.75	1000
Dress	0.85	0.89	0.87	1000
Coat	0.70	0.83	0.76	1000
Sandal	0.97	0.93	0.95	1000
Shirt	0.68	0.58	0.63	1000
Sneaker	0.89	0.97	0.93	1000
Bag	0.96	0.95	0.96	1000
Ankle boot	0.96	0.92	0.94	1000
accuracy			0.86	10000
macro avg	0.86	0.86	0.86	10000
weighted avg	0.86	0.86	0.86	10000



Analysis

The performances of all models were compared using accuracy, precision, recall, and F1 score. Among them, the highest overall performance was achieved by Model 3.3, which was a deep CNN that used three convolutional layers, dropout for regularization, and a reduced learning rate. It reached an accuracy of 0.9180. In contrast, Model 2.3, which was a basic CNN with the Adam optimizer with dropout, achieved the highest accuracy of 0.8729 within Model 2. The shallow neural network Model 1.2, which used the Adam optimizer, achieved the highest accuracy of 0.8456 within Model 1.

Model 1, which was a shallow neural network with no convolutional layers, showed the weakest performance overall. In its baseline version, Model 1.1, an accuracy of 0.7885 was recorded. After tuning the optimizer and learning rate in Model 1.2 got an accuracy of 0.8456 and Model 1.3 got an accuracy of 0.8057, but still remained lower than the other CNN-based models' maximum accuracy. Misclassification was most common in the class “Shirt”, which had an F1 score of only 0.53 in Model 1.3. Misclassifications were also observed in the "Pullover" class with an F1 score of 0.70 and in the "Coat" class with an F1 score of 0.71.

Model 2 was a basic CNN. Its baseline, Model 2.1, used a single convolutional layer with Adam optimizer, achieved an accuracy of 0.8701. In the tuned versions, Model 2.2 used the RMSprop optimizer and dropout, but this change led to a decrease in accuracy to 0.8414. Model 2.3 returned to the Adam optimizer, added dropout, and achieved a slightly better accuracy of 0.8729 and helped in balancing performance across classes. For example, in Model 2.3, “Shirt” had an F1 score of 0.67, which was better than the shallow model's performance.

Model 3 was a deep CNN with multiple convolutional layers. In the baseline version, Model 3.1, two convolutional layers were used with the Adam optimizer, and an accuracy of 0.8960 was obtained. In Model 3.2, dropout was introduced, resulting in a slightly better accuracy of 0.9016. The best outcome came from Model 3.3, where a third convolutional layer was added, dropout was applied, and the learning rate was decreased. It achieved an accuracy of 0.9180 and improved F1 scores for most classes, including “Shirt”, which rose to 0.77.

In short, it was observed that the deep CNN outperformed both the basic CNN and the shallow neural network. However, it was also noticed that deeper models required more training time and computational resources due to their higher number of parameters.

Analysis of Extra Task (Q2)

The model achieved a test accuracy of 0.8578. Throughout training, both training and validation accuracy improved steadily, with validation accuracy reaching 0.8767 by the final epoch. The training and validation loss decreased consistently over time. The model performed well across most classes. The lowest F1 score was recorded for the “Shirt” class, which was 0.63. The confusion matrix showed notable misclassifications between “Shirt,” “T-shirt/top,” and “Coat.”

Conclusion

This project explored different neural network architectures for classifying images from the Fashion MNIST dataset. Three types of models were built, including a shallow neural network, a basic CNN, and a deeper CNN. As the models became more complex, their performance improved. Hyperparameter tuning, such as adjusting optimizers, learning rates, and adding dropout, led to better generalization. The deep CNN with proper tuning showed the best performance overall. In future experiments, further improvements could be made by applying early stopping, transfer learning, and more advanced optimization techniques.

Code of Assignment 3 in Google Colab:
https://colab.research.google.com/drive/1ELQe8QKDthVaPwgP_KSIWaZ1PY0t_2Zk?usp=sharing