Project Report: Fashion MNIST

Introduction:

The objective of this project is to create a classification model that can correctly identify various fashion pieces according to their traits. comprising 10,000 examples for the test set and 60,000 examples for the training set. Every example consists of a 28 x 28 grayscale picture with a label from one to ten classes, making up the dataset used for this project. Every image has a height of 28 pixels and a width of 28 pixels, for a total of 784 pixels. Every pixel has a unique pixel-value that represents its level of lightness or darkness; larger values correspond to darker pixels. The pixel value falls between the range of 0 to 255. There are 785 columns in the training and test data sets. The article of clothing is represented by the class labels (see above) in the first column. The pixel values of the related image are displayed in the remaining columns.

Data Preparation:

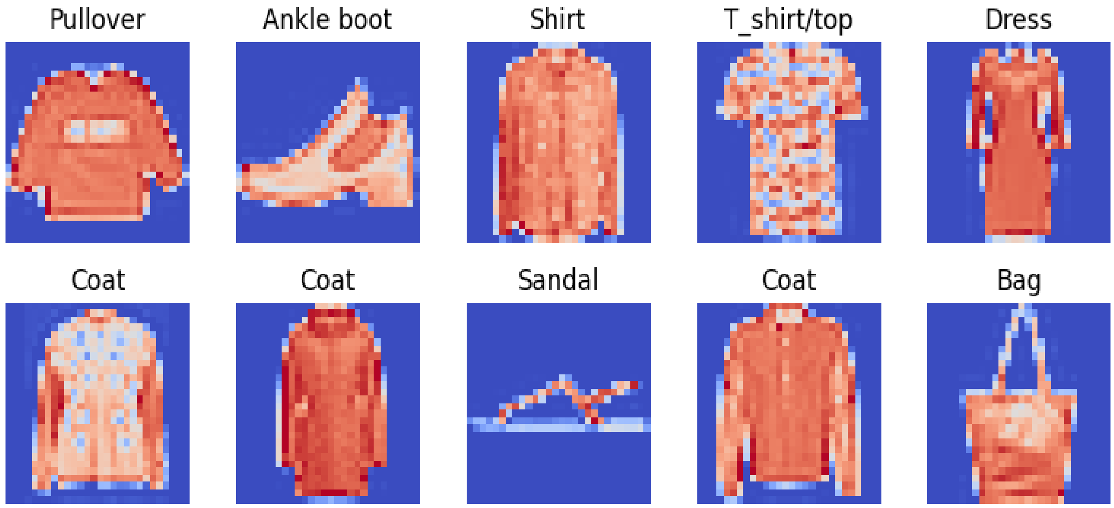
The data preparation phase involves:

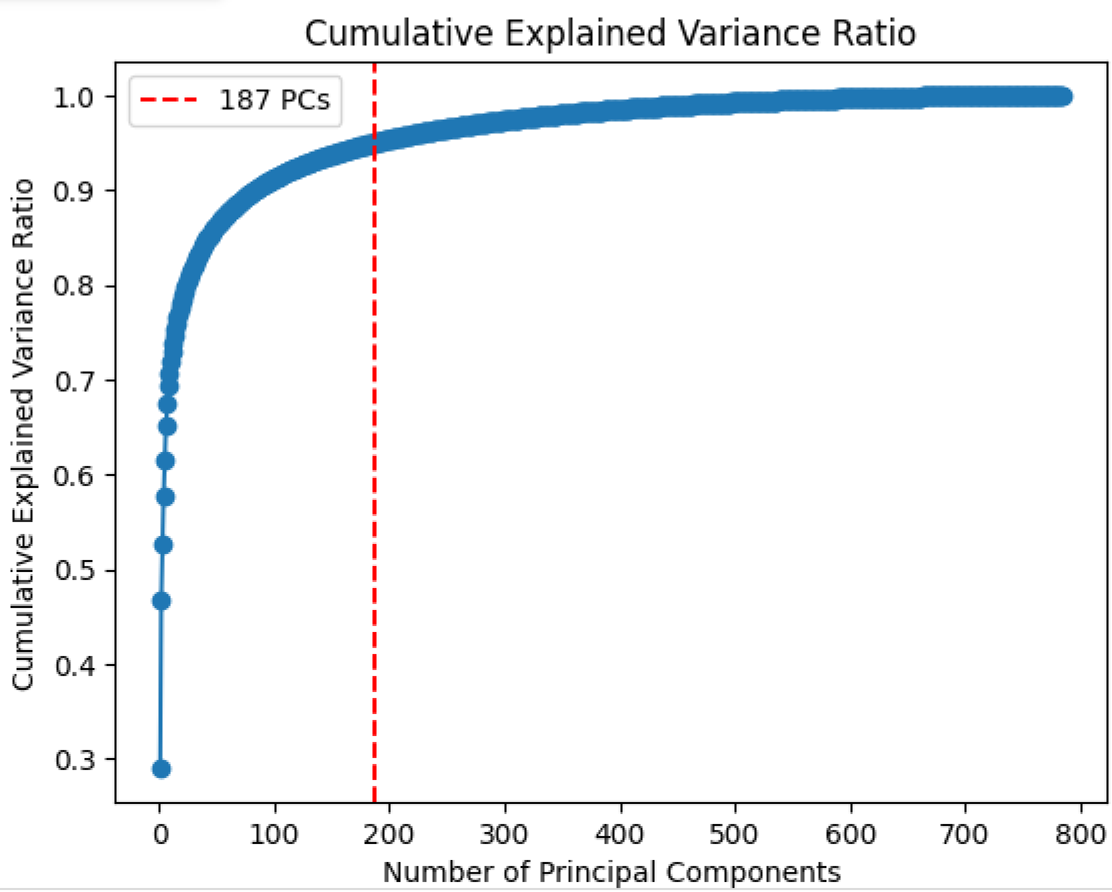
1. importing the required libraries.
2. loading the dataset.
3. Showing data insights:

   The dataset consists of the following labels:

   1. 0 T-shirt/top
   2. 1 Trouser
   3. 2 Pullover
   4. 3 Dress
   5. 4 Coat
   6. 5 Sandal
   7. 6 Shirt
   8. 7 Sneaker
   9. 8 Bag
   10. 9 Ankle boot

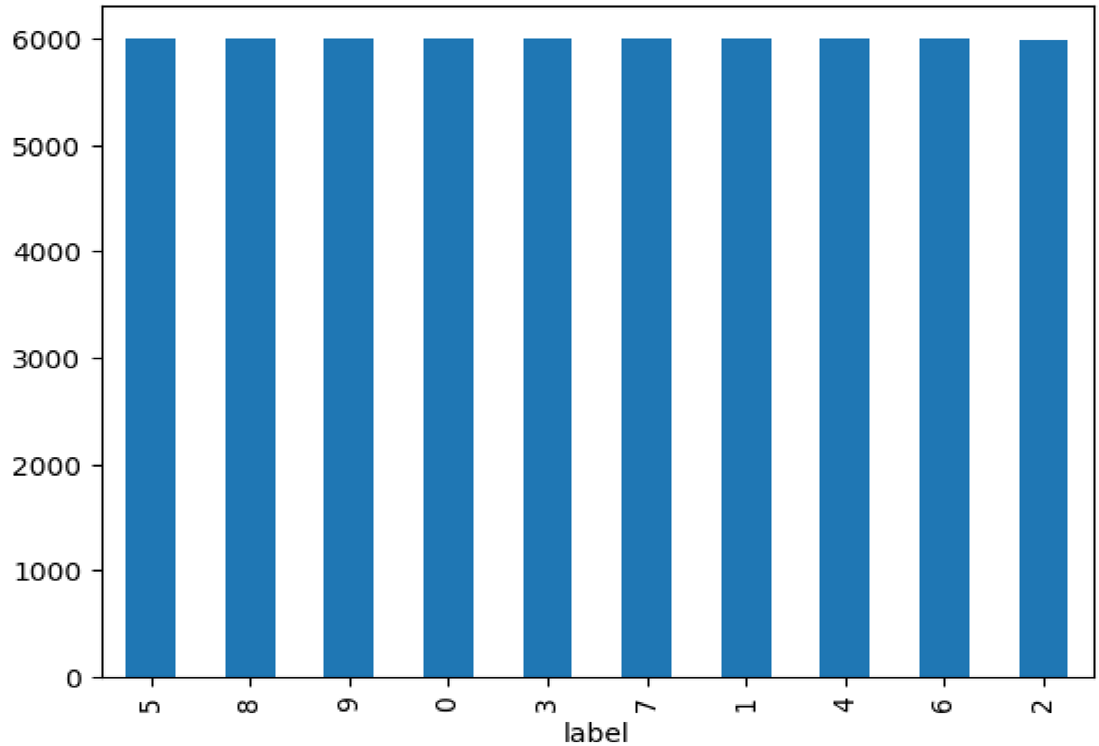
4. Cleaning data by checking for nulls and duplicates and outliers ,and we found that there is no null ,there are small number of duplicates which we can remove ,and some outliers but they have minimal impact.
5. Data Visualization using PCA and drawing some images.



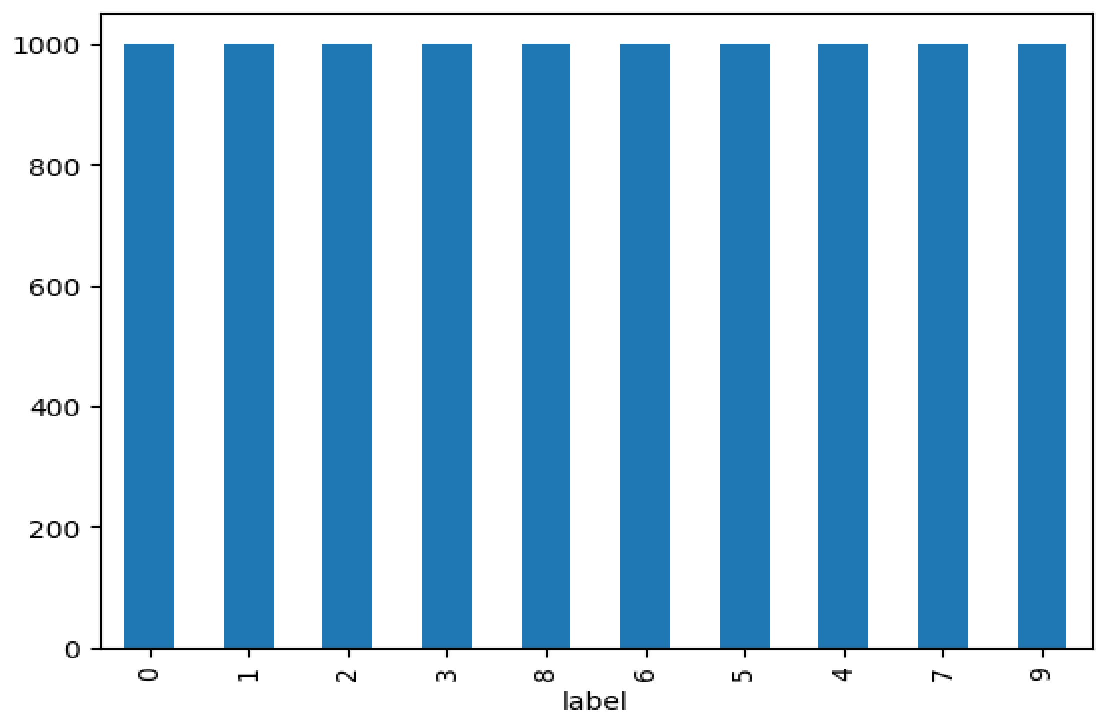


6. Checking distribution of classes and found them balanced.

Train Data



Test Data



7. Carrying out correlation analysis.
8. Data Division to data and target.
9. Data Standardization using standard scaler.
10. Encoding the labels using Label Encoder.

Preprocessing

1. Splitting the data
2. Building the functions we used as model function, evaluation function, and graphing function.
3. Hyperparameters tunning which we will discuss in detail in the next section.
4. Early stopping condition.
5. Learning rate decay function.
6. Images resizing.

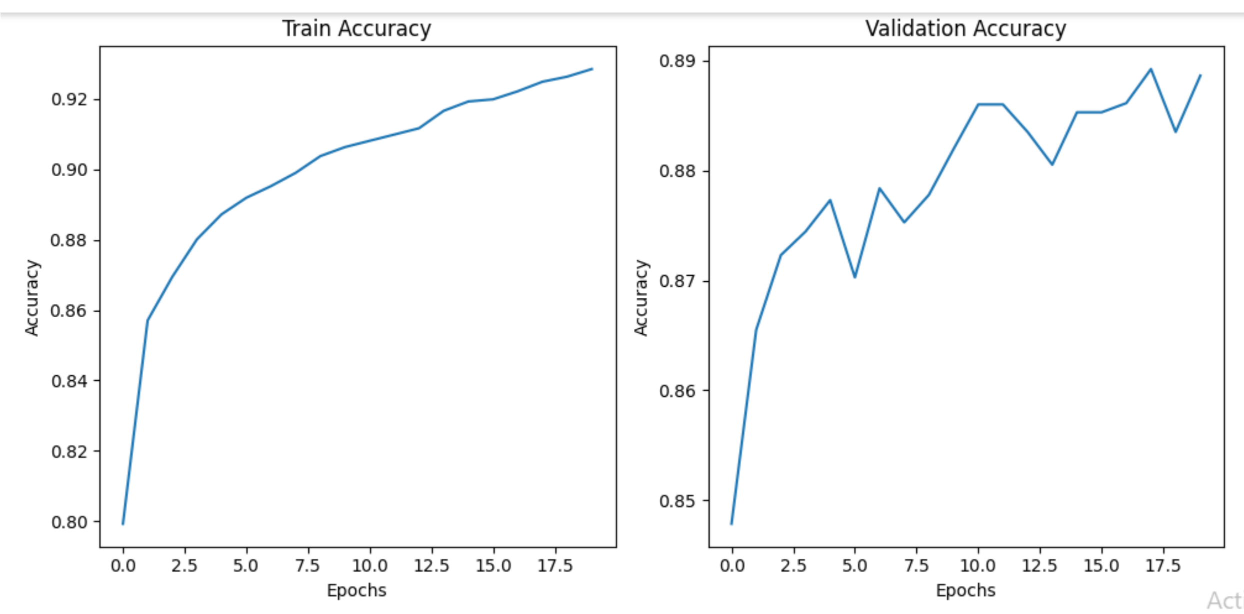
Models

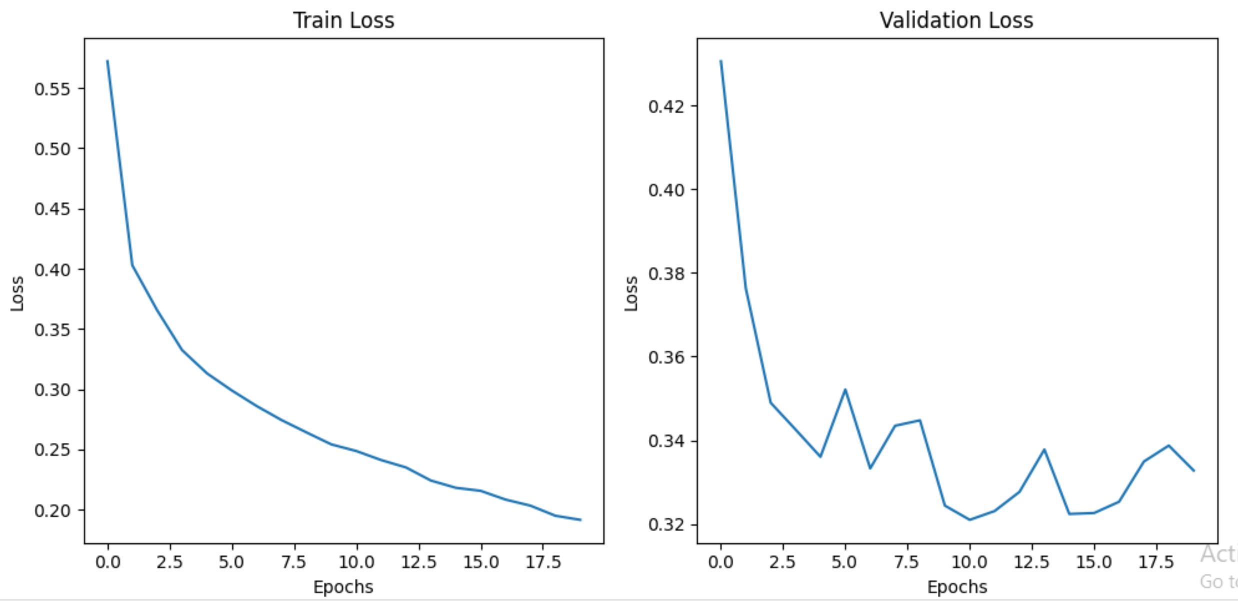
1.MLP

We have worked with some simple layers .

we get that:

model achieved train accuracy= 92.85% ,validation accuracy=88.86%





2.LeNet and Hyperparameter tunning

For the one we used more layers and specifications

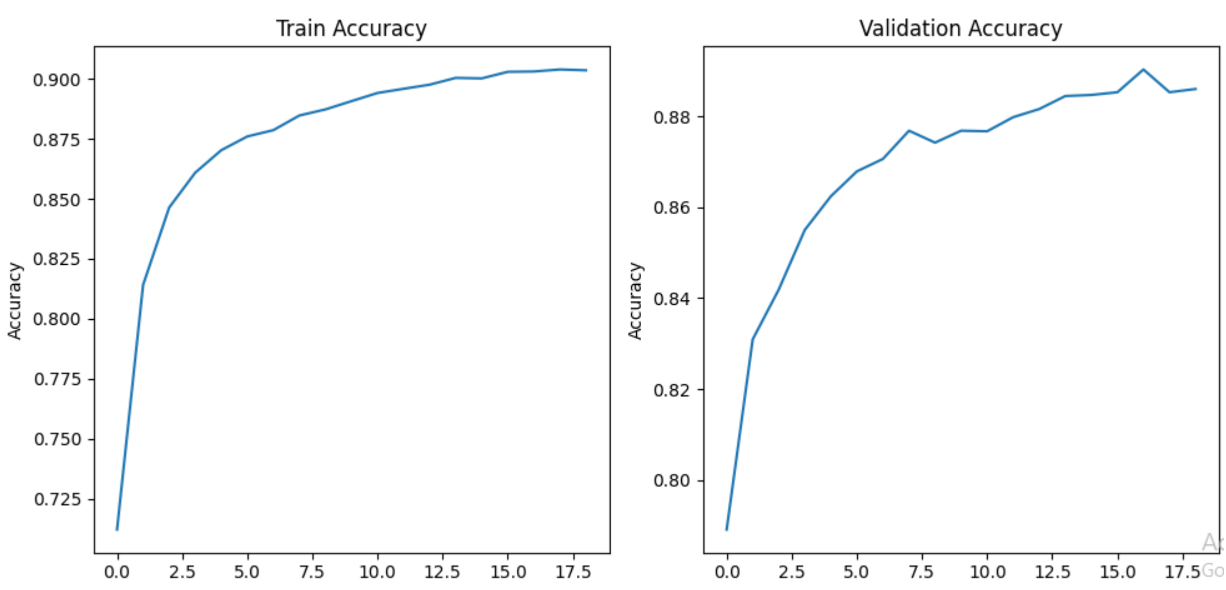
we get that:

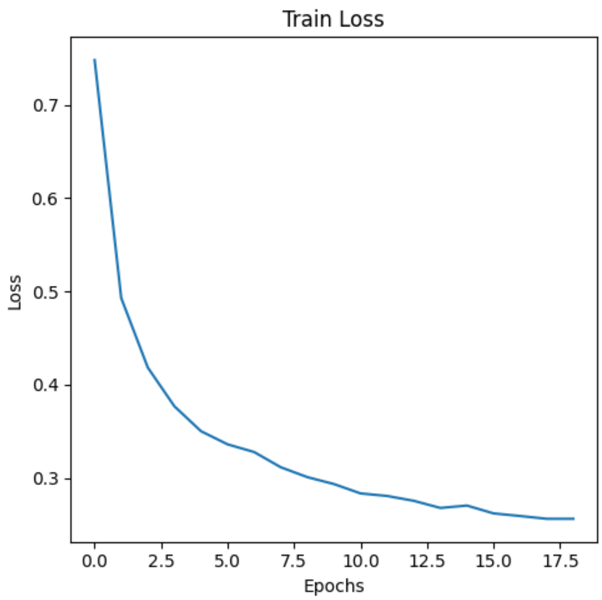
model achieved train accuracy= 90.5% ,validation accuracy=88.6%

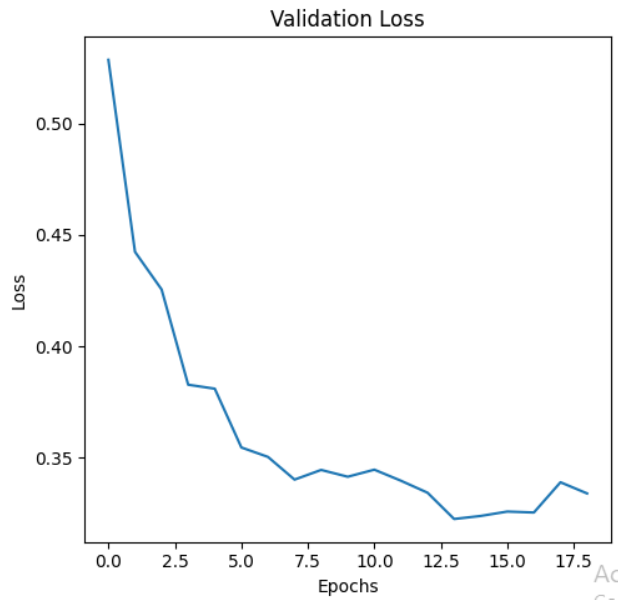
optimal number of units= 120

optimal learning rate= 0.01.

Best Epoch: 17







3. Cross Validation

With 5 folds

we get that:

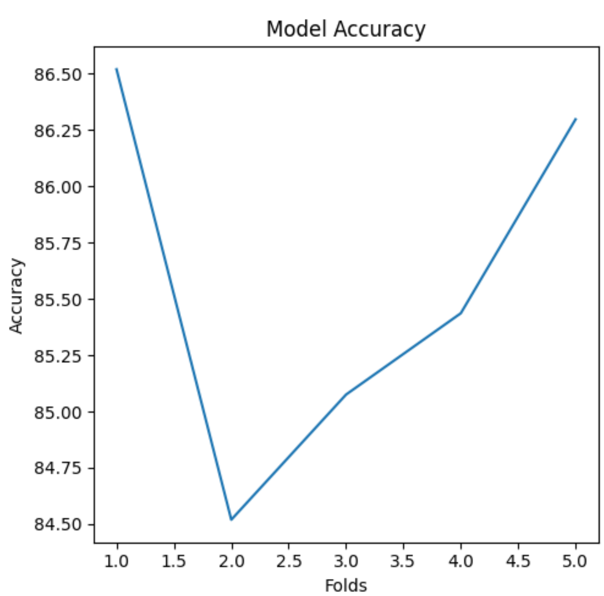
fold 1 achieved accuracy= 86.3%

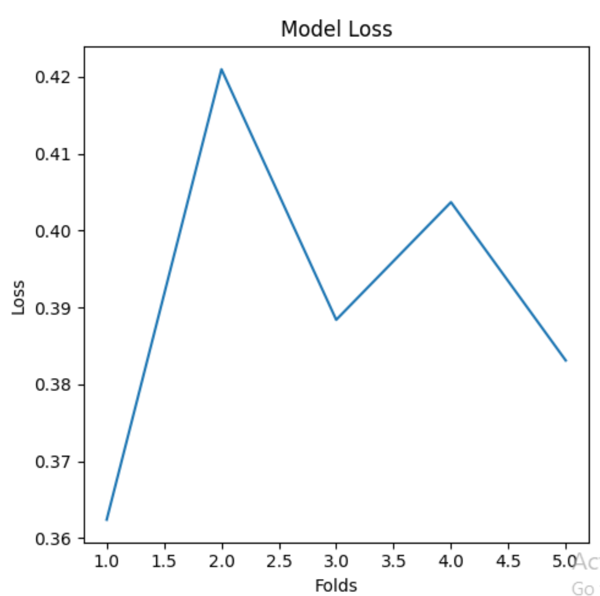
fold 2 achieved accuracy= 84.39%

fold 3 achieved accuracy= 85.3%

fold 4 achieved accuracy= 85.53%

fold 5 achieved accuracy= 86.2%



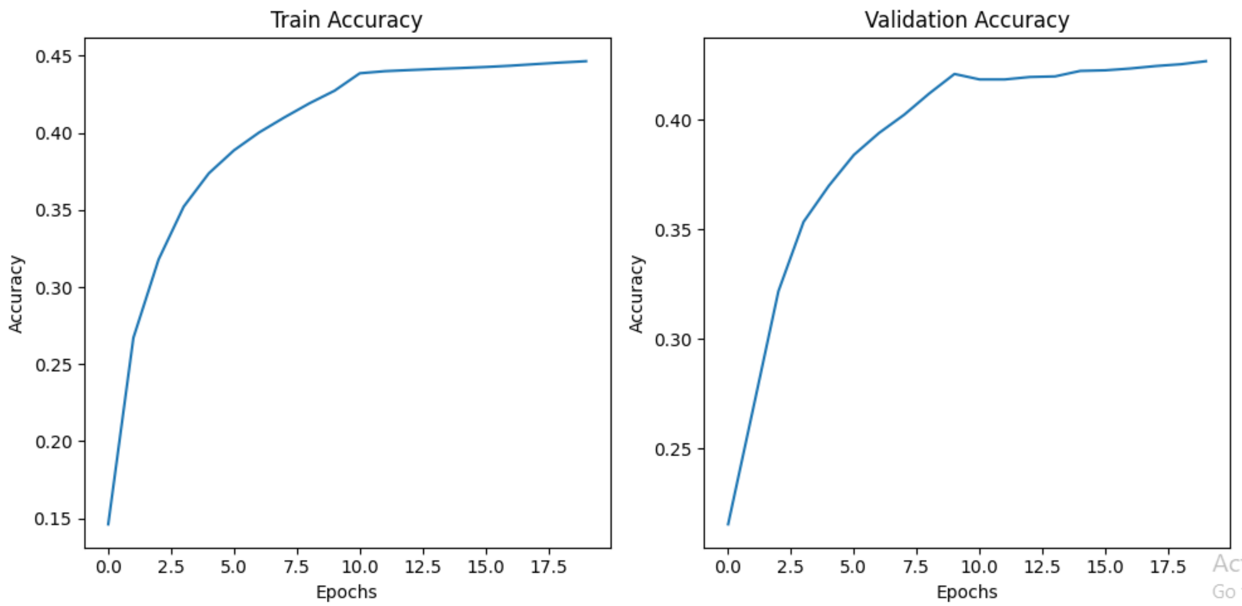


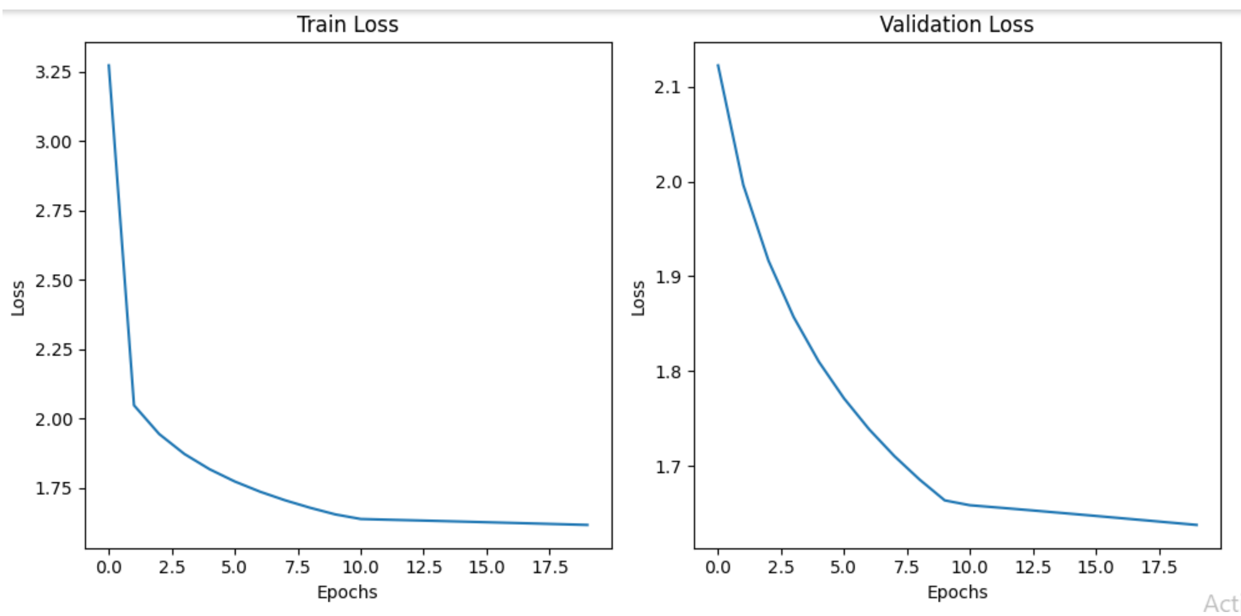
Transfer Learning

1.VGG16

we get that:

model achieved train accuracy= 45.05% ,validation accuracy=42.69%

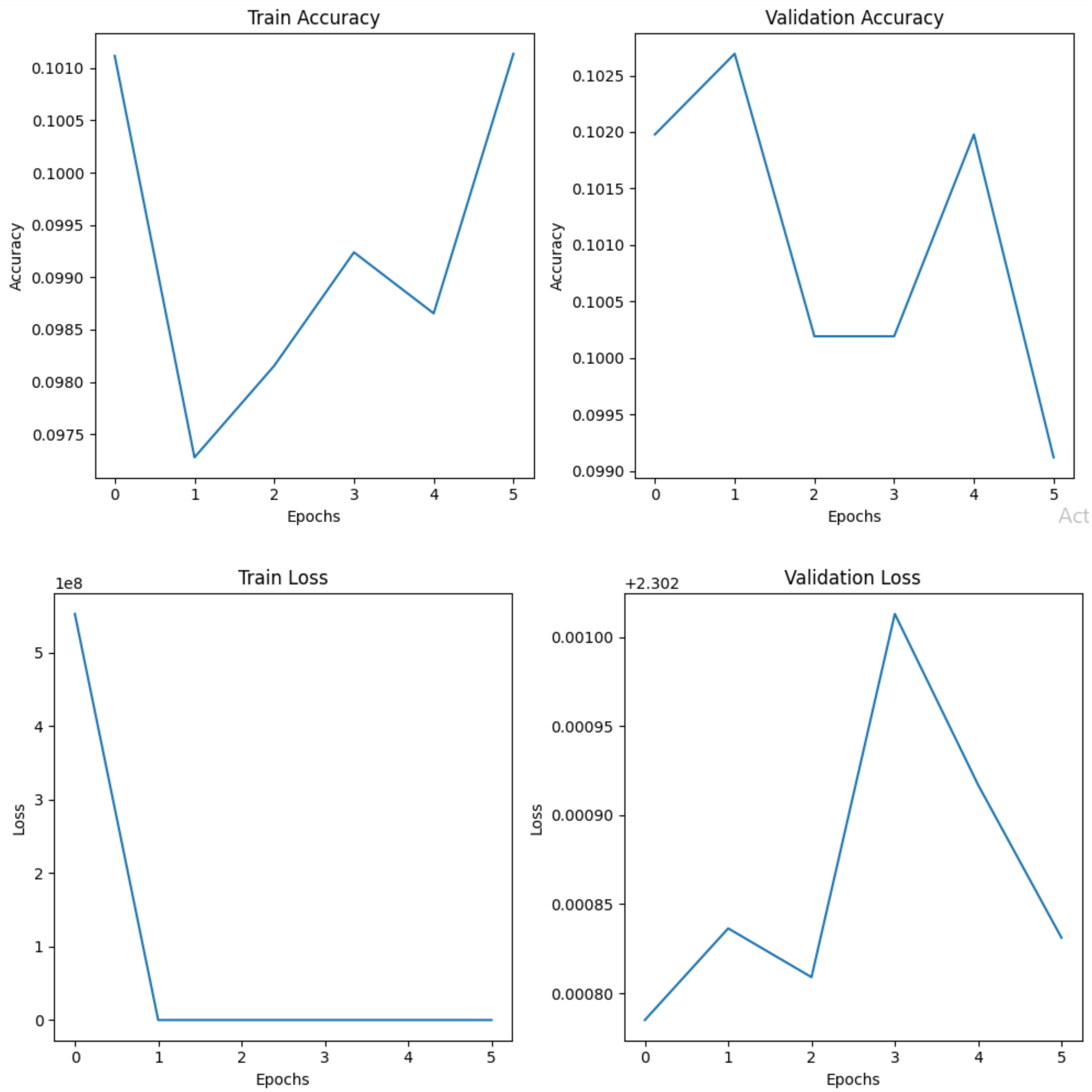




2.VGG19

we get that:

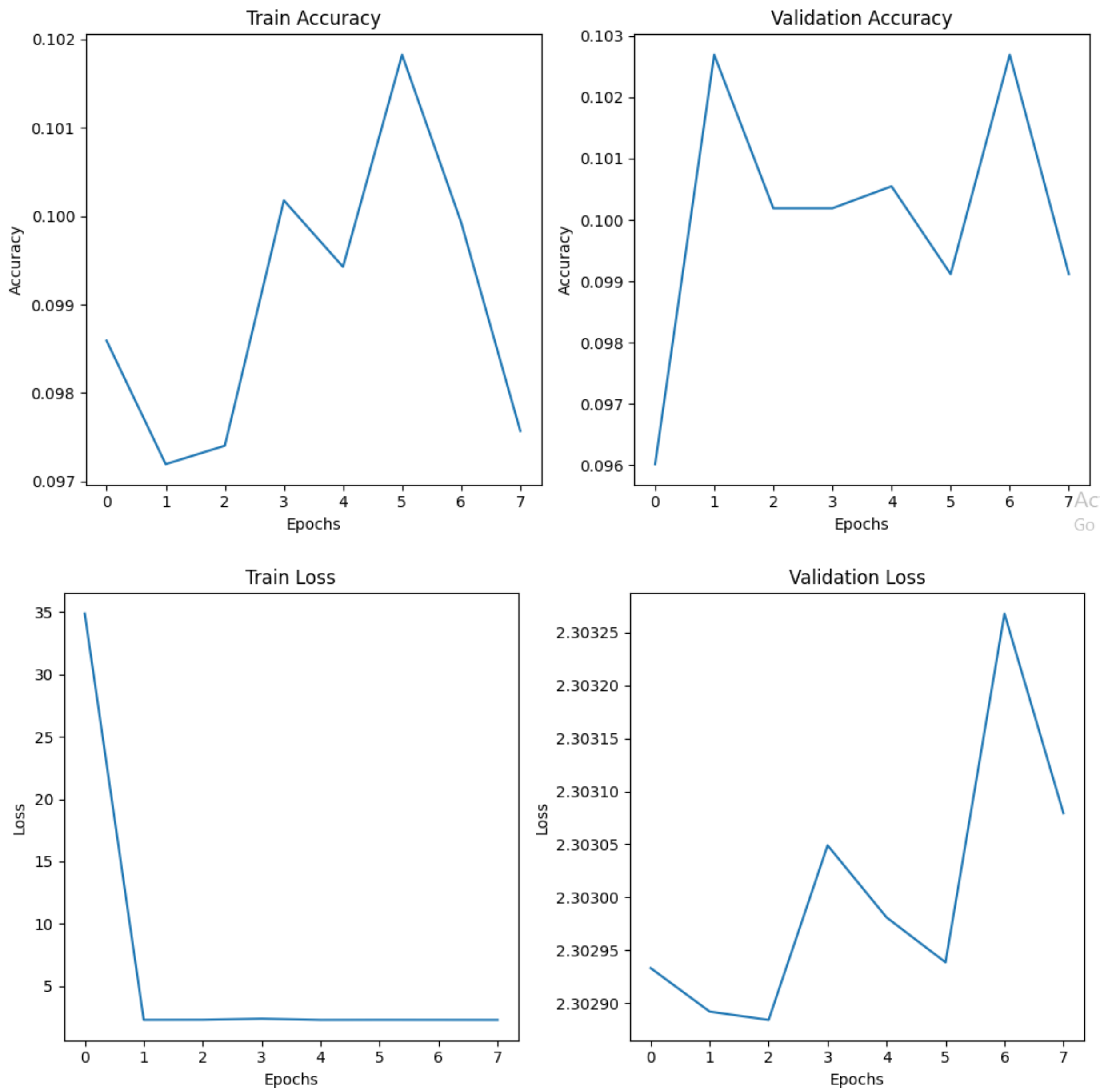
model achieved train accuracy= 10.11% ,validation accuracy=9.91%



3.ResNet152V2

we get that:

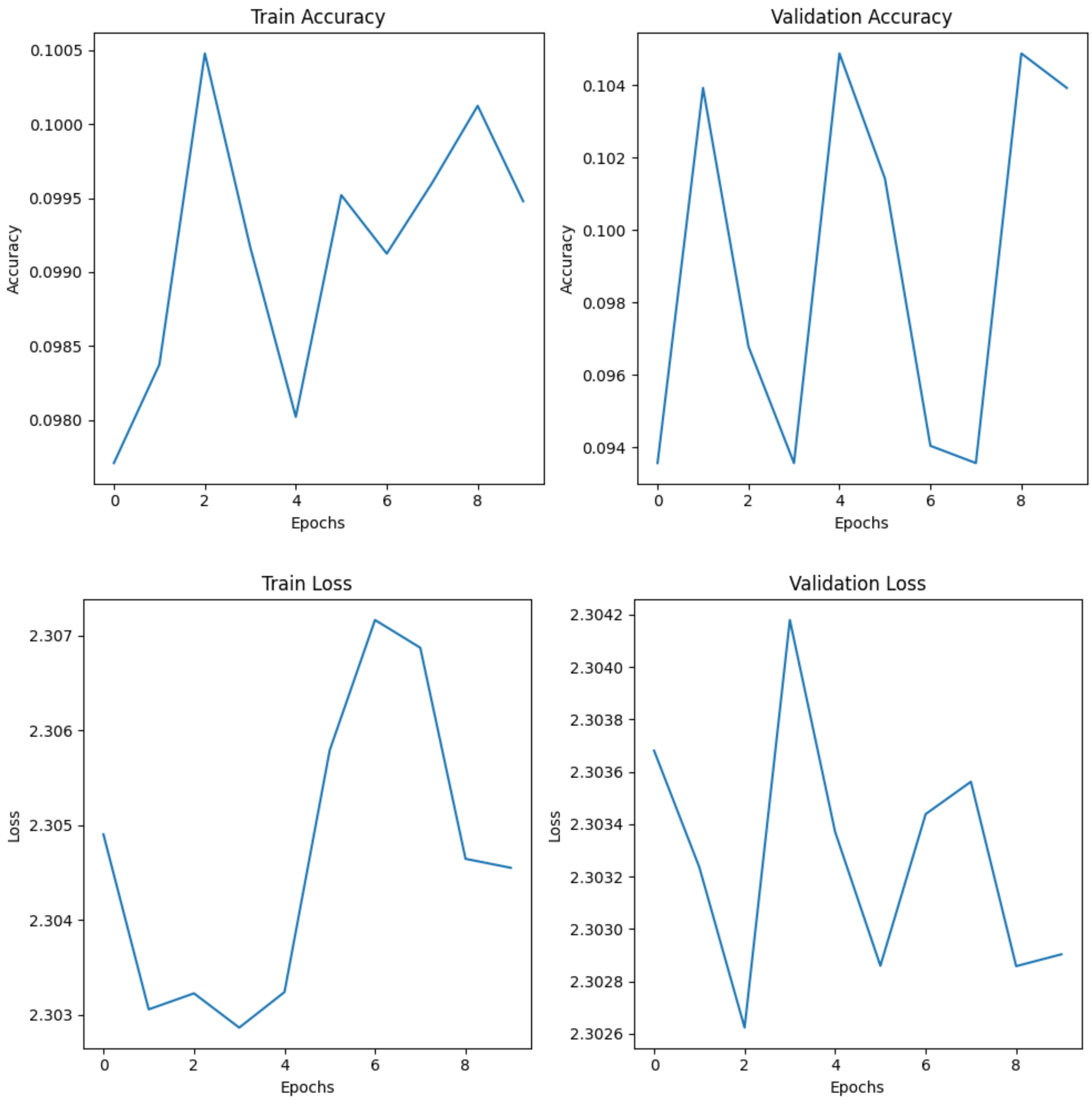
model achieved train accuracy= 9.76% ,validation accuracy=9.91%



4.DenseNet201

we get that:

model achieved train accuracy= 9.95%, validation accuracy=10.39%



Conclusion

Simpler models like MLP (Multi-Layer Perceptron) or LeNet have been found to produce higher and more satisfactory accuracy in the fashion MNIST classification job than more complicated models like VGG16, VGG19, ResNet152V2, and DenseNet201. This result implies that the deeper models' complex architectures might not offer any appreciable advantages over more straightforward methods for this specific purpose. The fashion MNIST dataset's key patterns and features can be efficiently captured by the fully connected MLP model and the convolutional and pooling layers of the LeNet model. This underlines how important it is to take the job and dataset in question into account when assessing the model architecture's complexity. We can obtain competitive accuracy and gain advantages like smaller and less complex models by using simpler models.

References:

https://www.kaggle.com/code/guilhermesdas/fashion-mnist-vgg16/notebook

https://www.kaggle.com/code/anandad/classify-fashion-mnist-with-vgg16/notebook

https://www.reddit.com/r/tensorflow/comments/i4mx5v/classify_fashion_mnist_with_vgg16_question

https://datahacker.rs/lenet-5-implementation-tensorflow-2-0/

https://github.com/python-engineer/tensorflow-course/blob/master/08_09_Star_Wars_Project.ipynb

https://www.youtube.com/watch?v=8cN0PiZQl18&t=503s&ab_channel=PythonEngineer

https://stackoverflow.com/questions/40119743/convert-a-grayscale-image-to-a-3-channel-image

https://stackoverflow.com/questions/39463019/how-to-copy-numpy-array-value-into-higher-dimensions