

## CSAI 867 Project 2

Adham Mokhtar - 20398545

### Project Description:

In this project, you will use the Fashion-MNIST dataset using a CNN neural network architecture.

### Dataset Description:

Fashion MNIST: MNIST-like dataset of 60,000 28x28 labeled fashion images.

label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel775	pixel776	pixel777	pixel778	pixel779	pixel780	pixel781	pixel782
0	2	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
1	9	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
2	6	0	0	0	0	0	0	0	5	...	0	0	0	30	43	0	0	0
3	0	0	0	0	1	2	0	0	0	...	3	0	0	0	0	0	1	0
4	3	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0

5 rows × 785 columns

After searching and exploring the data, we can say that the data is clean:

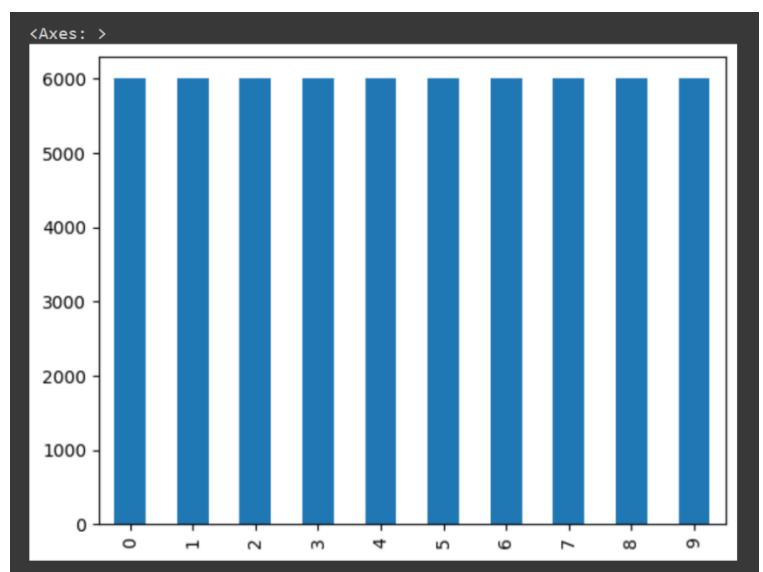
- Zero nulls
- Zero missing values
- Just 43 duplicated rows



Data consist of 10 Class as shown here:

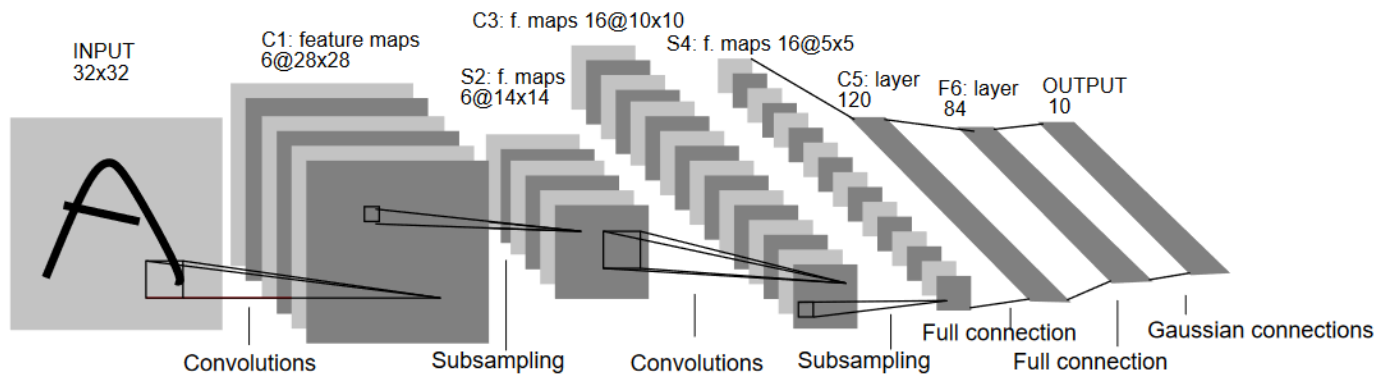
Each class has only 10 samples.

- T-shirt/top
- Trouser
- Pullover
- Dress
- Coat
- Sandal
- Shirt
- Sneaker
- Bag
- Ankle boot



## LeNet 5 Model:

I Implement the LeNet 5 model architecture:



I tried some hyperparameter with LeNet 5 and Adam Optimizer for best results.

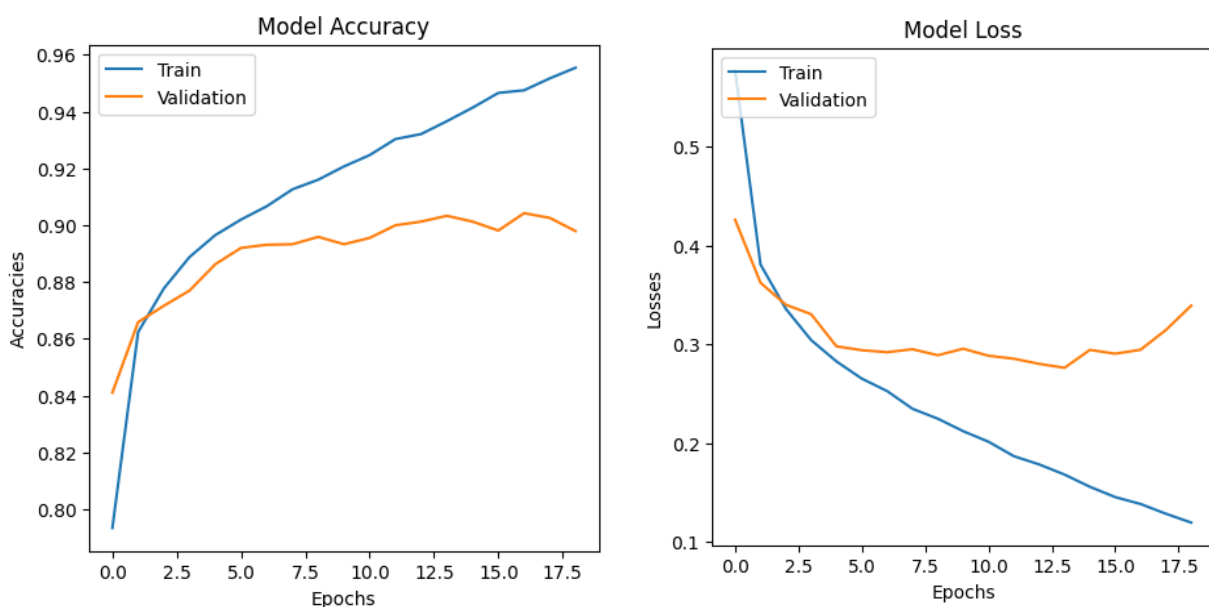
- Hidden Size
- Learning Rate

I used Keras Tuner to search for the best hyper-parameter as a recommendation from Dr. Steven.

### Observation:

I trained the model with the best parameters I had found before and I tested it.

That is the graphs of the whole process:



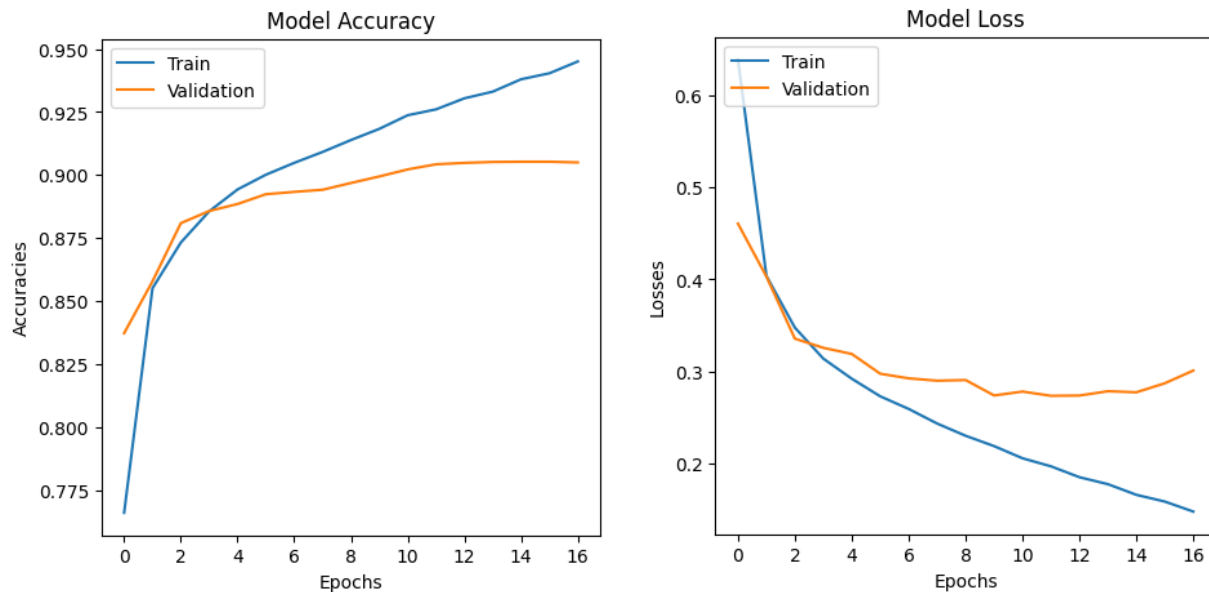
## Cross Validation:

I had worked on implementing cross validation using KFold for splitting the data into folds but I decided to use StratifiedKFold to keep the balance ratio of the data into the folds.

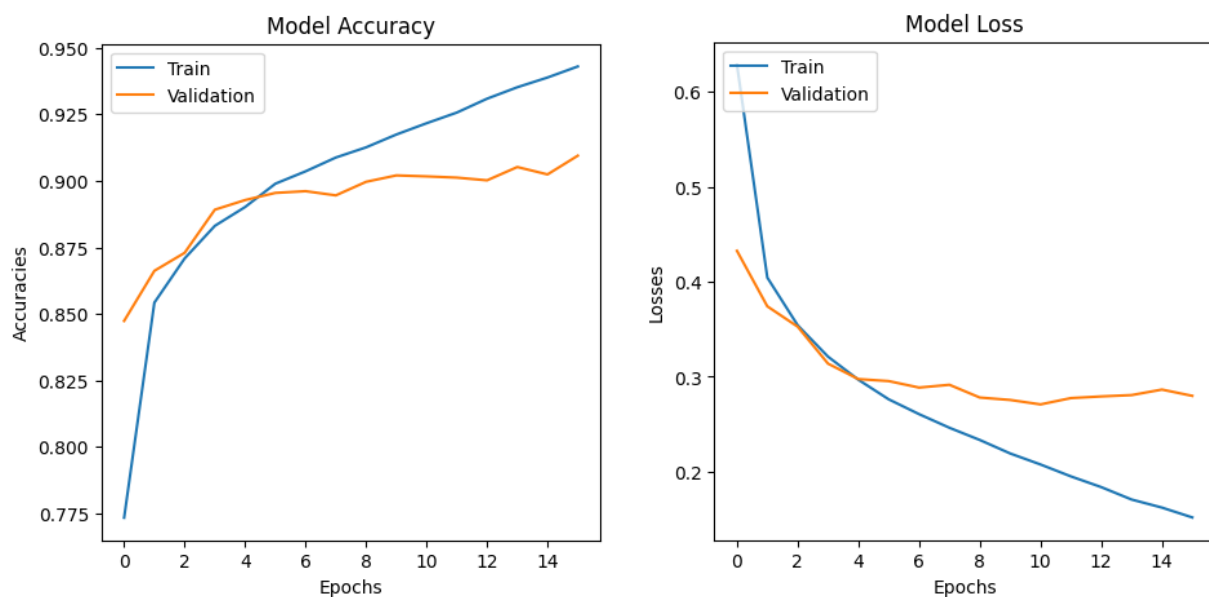
I trained with 5 folds

### The convergence curves for LeNet-5.

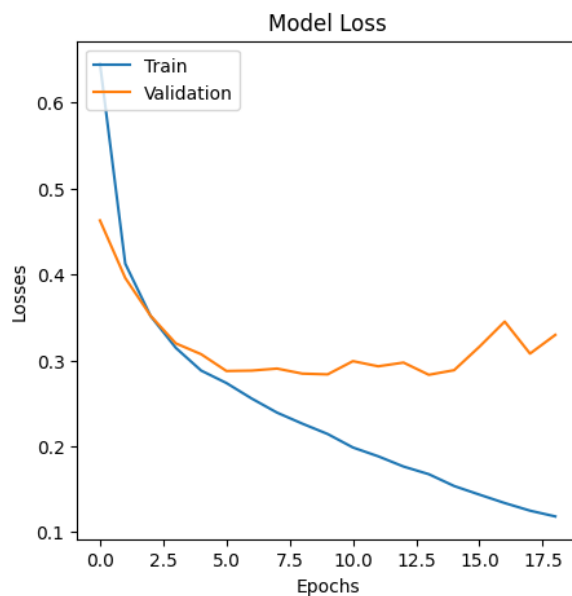
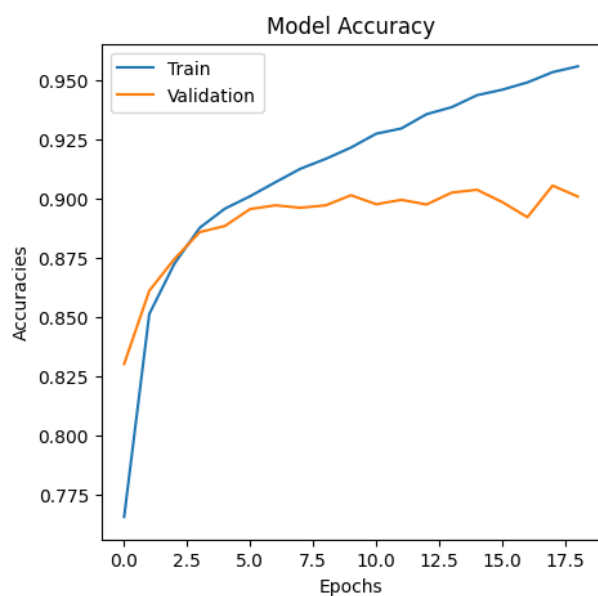
#### - Fold 1:



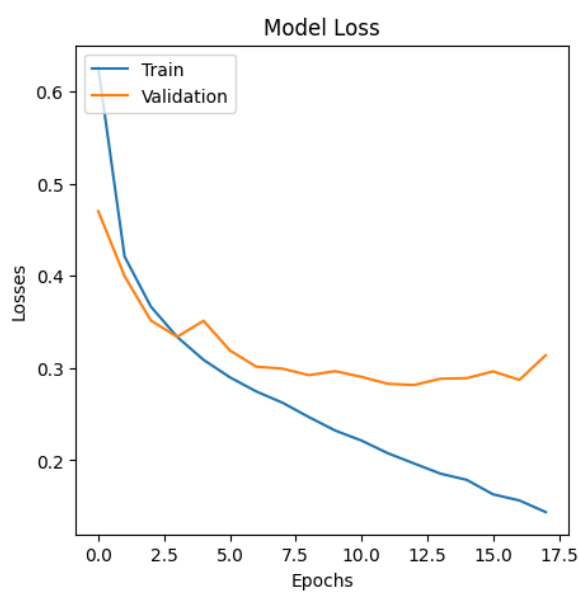
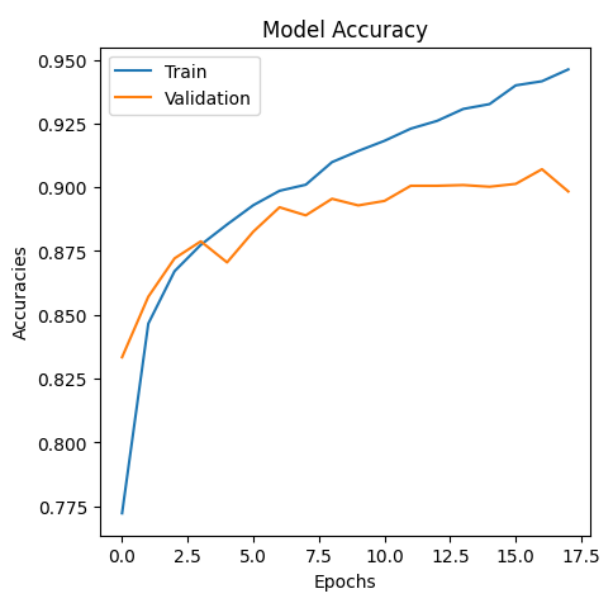
#### - Fold 2:



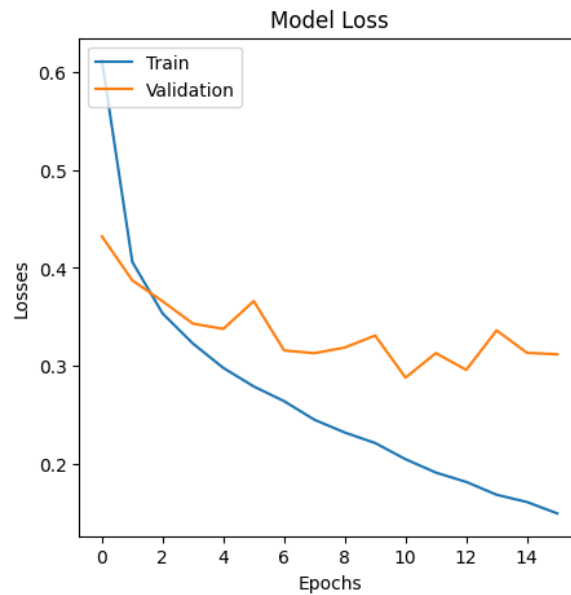
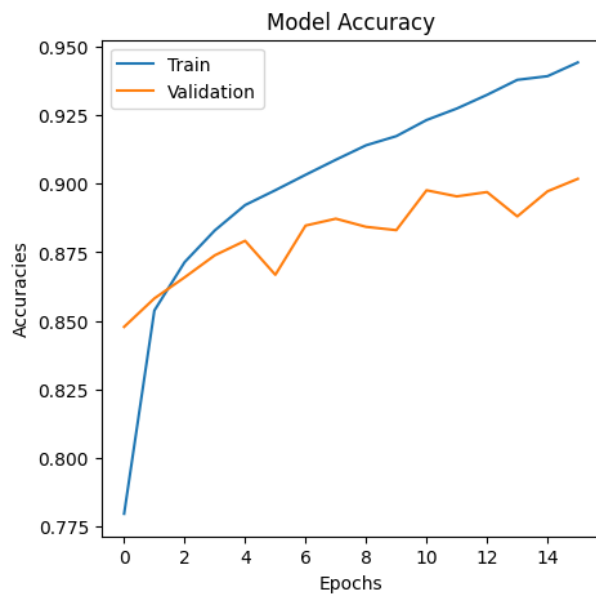
- **Fold 3:**



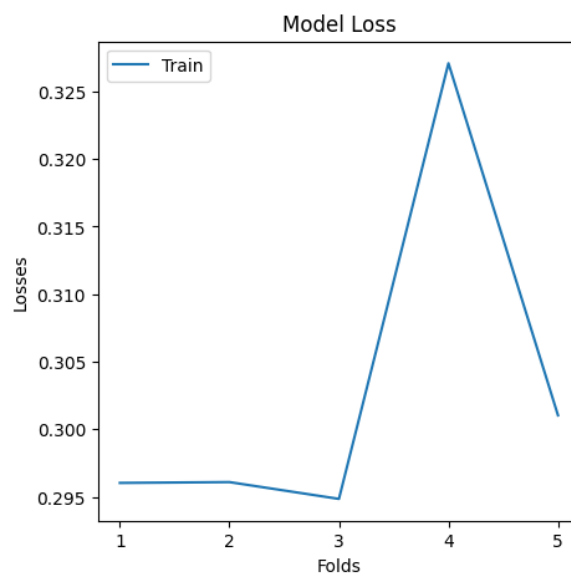
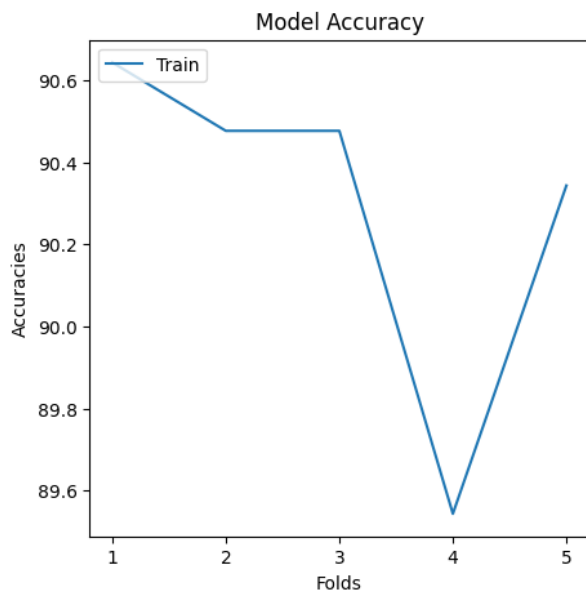
- **Fold 4:**



### - Fold 5:



### Comparison of Models Evaluation:



### Observation:

From the previous evaluation graphs we can see that the first model is the best but we can also observe that the data is biased because the curve should be near to a straight line.

## Transfer learning:

I used several pretrained models as shown in the notebook.

- VGG19
- RESNet152V2
- DenseNet201

I faced several problems while implementing this project and the pretrained models

- Most of the Pretrained models take only 3 channels
- Most of the Pretrained models take only 32×32 images or above
- Finding a good fully connected architecture.
- Finding a good hyperparameters.

I tried my best and find the following:

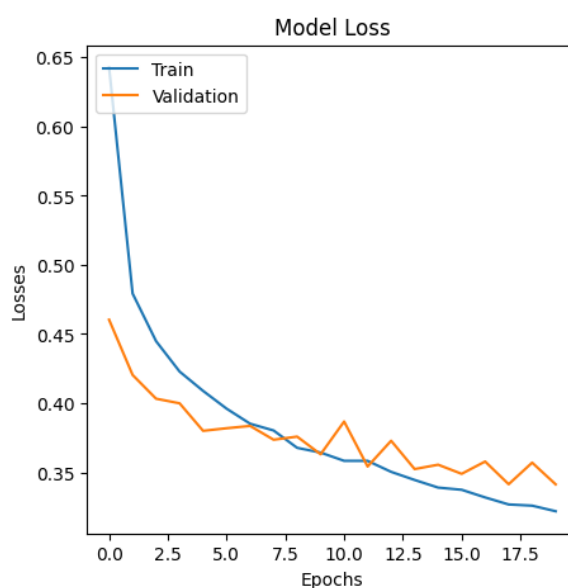
Model	accuracy	val_accuracy	test_accuracy
VGG19	88.06%	87.47%	87.69%
RESNet152V2	84.34%	79.90%	79.56%
DenseNet201	83.86%	80.01%	80.15%

## Observation:

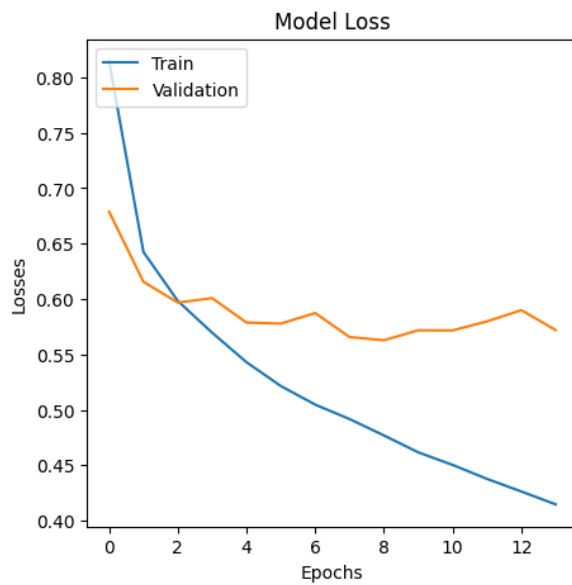
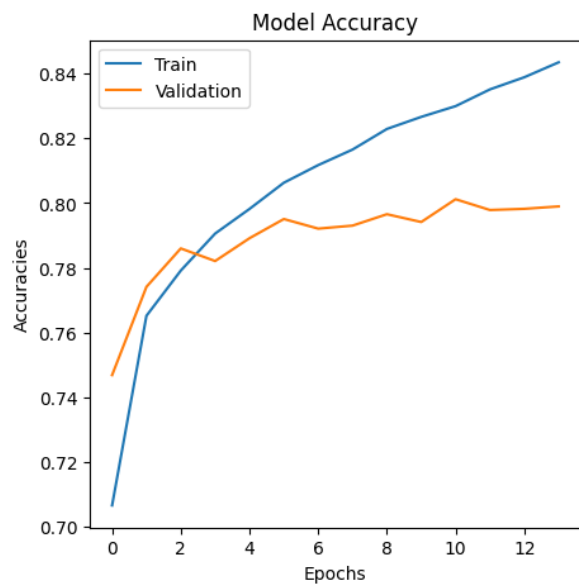
The VGG19 is better in than the other models with the almost the same fully connected network.

## Models Training Graphs:

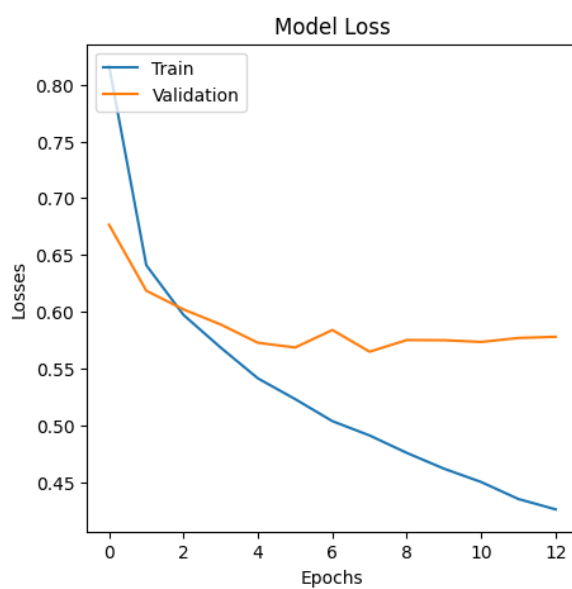
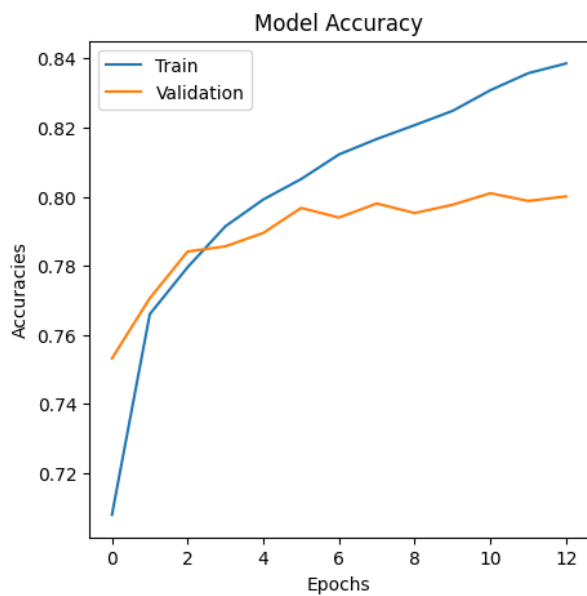
### - VGG19:



## - RESNet152V2:



## - DenseNet201:



## Notes:

I had implemented this Project on google **Colab** Notebook, the data attached here: [Fashion MNIST](#) | [Kaggle](#)

Thank you.  
Adham Mokhtar  
20398545