

COMPUTER VISION (CSCI -631)

Homework-4

Name: Rajkumar Lenin Pillai

Note : Run the code in file Pillai_Rajkumar_HW4.py file and the results of convolutional neural network on Fashion-MNIST dataset can be observed in different plots.

The model:

The model which is used has two 2-d convolutional layers and kernel of size 3 x 3 which can be observed in the code.

Training:

Before training all the samples, the training dataset is split into 80% for training and 20% for validation set with each set containing equally balanced samples from each class.

Following are the results of training the model:

Epoch 1, Train_loss: 1.34790744860967
Epoch 1, Val_loss: 0.19387312602996826
Epoch 2, Train_loss: 0.7094432459672292
Epoch 2, Val_loss: 0.16148913725217184
Epoch 3, Train_loss: 0.627325865983963
Epoch 3, Val_loss: 0.14517351869742076
Epoch 4, Train_loss: 0.5798468462228775
Epoch 4, Val_loss: 0.13978572142124177
Epoch 5, Train_loss: 0.5459415632486343
Epoch 5, Val_loss: 0.13089000837008158
Epoch 6, Train_loss: 0.5196541951497395
Epoch 6, Val_loss: 0.1237006238301595
Epoch 7, Train_loss: 0.4963210511207581
Epoch 7, Val_loss: 0.12114469929536184
Epoch 8, Train_loss: 0.47707136195898053
Epoch 8, Val_loss: 0.11757868607838949
Epoch 9, Train_loss: 0.4609502405524254
Epoch 9, Val_loss: 0.11651479748884837
Epoch 10, Train_loss: 0.4457222165465355
Epoch 10, Val_loss: 0.10897519524892171
Epoch 11, Train_loss: 0.4327495127717654
Epoch 11, Val_loss: 0.10802022208770116
Epoch 12, Train_loss: 0.42314525906244915
Epoch 12, Val_loss: 0.10412661006053289
Epoch 13, Train_loss: 0.4117505207459132
Epoch 13, Val_loss: 0.10128121399879456
Epoch 14, Train_loss: 0.4031962640484174
Epoch 14, Val_loss: 0.09979696935415268

Epoch 15, Train_loss: 0.3939326184193293
Epoch 15, Val_loss: 0.09952162144581476
Epoch 16, Train_loss: 0.3864604932665825
Epoch 16, Val_loss: 0.098591031173865
Epoch 17, Train_loss: 0.3797690398693085
Epoch 17, Val_loss: 0.09755712393919627
Epoch 18, Train_loss: 0.37370280557870866
Epoch 18, Val_loss: 0.0935914588769277
Epoch 19, Train_loss: 0.3665297553340594
Epoch 19, Val_loss: 0.09744623746474584
Prev loss_validation 0.0935914588769277 Runnning loss_validation 0.09744623746474584

Time to converge 624.4748222827911 seconds

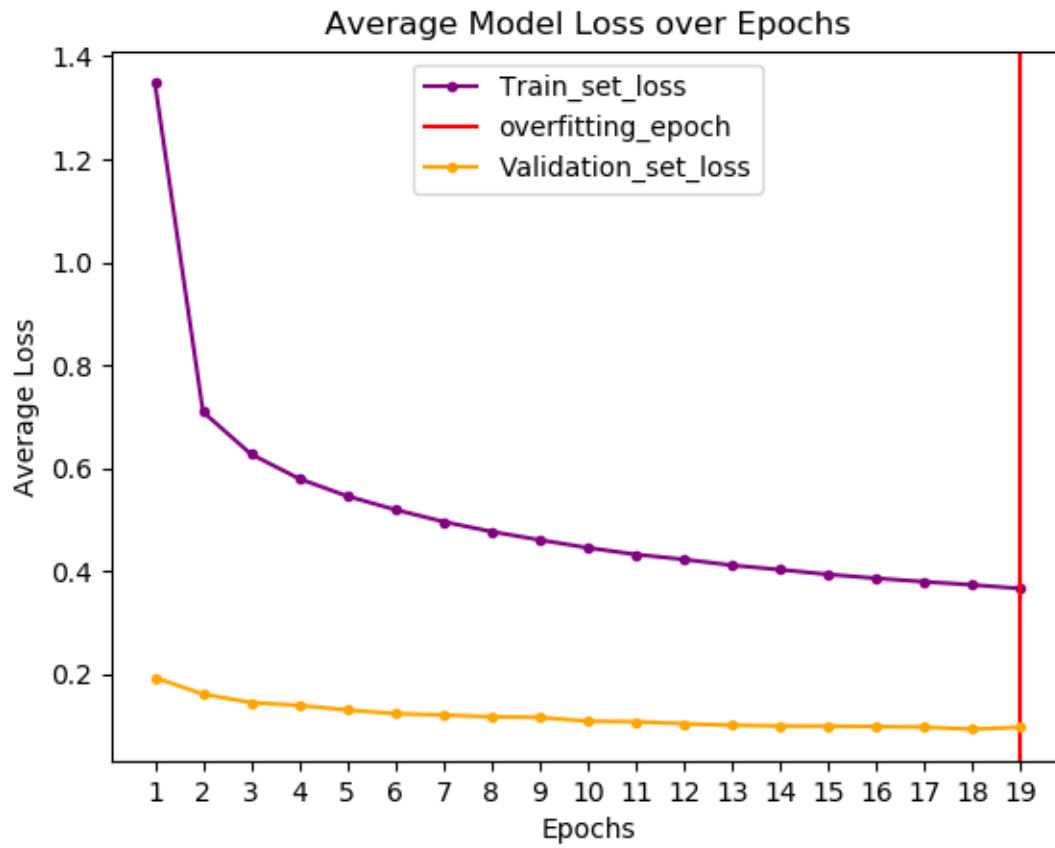
Accuracy calculation

Accuracy of the network on the test images: 85 %

Accuracy of	0 : 85 %	Incorrect samples	149	correct samples	851
Accuracy of	1 : 95 %	Incorrect samples	48	correct samples	952
Accuracy of	2 : 88 %	Incorrect samples	117	correct samples	883
Accuracy of	3 : 90 %	Incorrect samples	91	correct samples	909
Accuracy of	4 : 71 %	Incorrect samples	289	correct samples	711
Accuracy of	5 : 96 %	Incorrect samples	37	correct samples	963
Accuracy of	6 : 49 %	Incorrect samples	506	correct samples	494
Accuracy of	7 : 90 %	Incorrect samples	91	correct samples	909
Accuracy of	8 : 94 %	Incorrect samples	57	correct samples	943
Accuracy of	9 : 96 %	Incorrect samples	40	correct samples	960

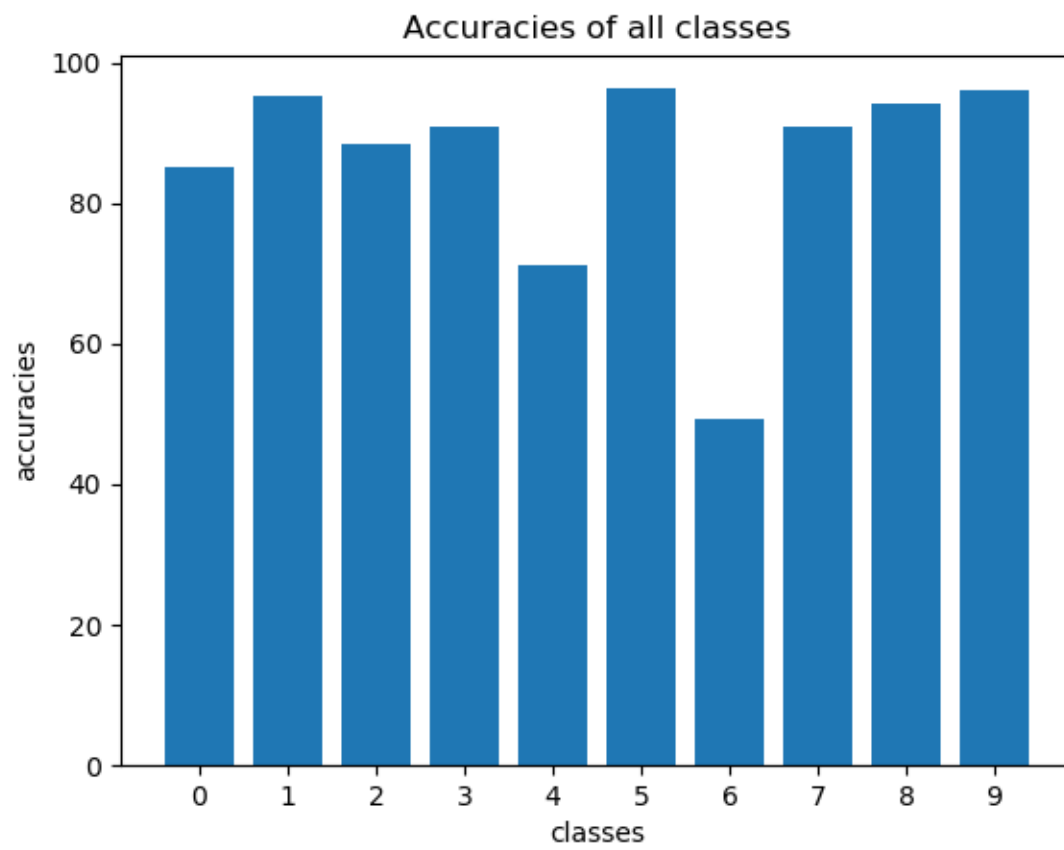
From the above observations we can say Class-5 and Class-9 have highest accuracy of 96% and Class-6 has lowest accuracy of 49%. So it means that model performs better in predicting the images of sandal and ankle boot and it is less efficient in predicting the image of shirt.

Training and validation loss curve:



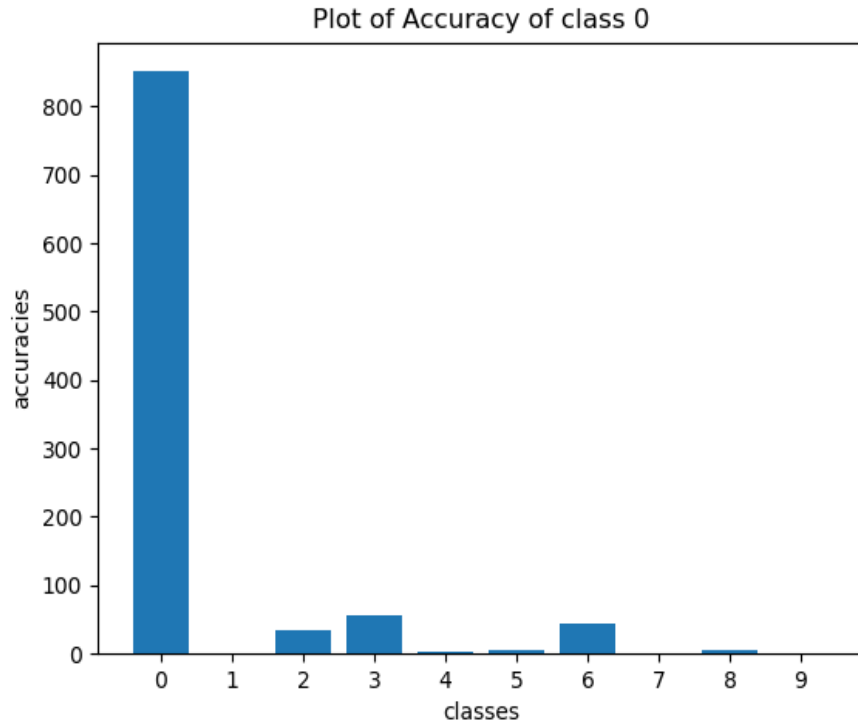
The model was run for 20 epochs but since it starts overfitting after 19th epoch the model was not trained for 20th epoch

Plot of accuracy of all classes:

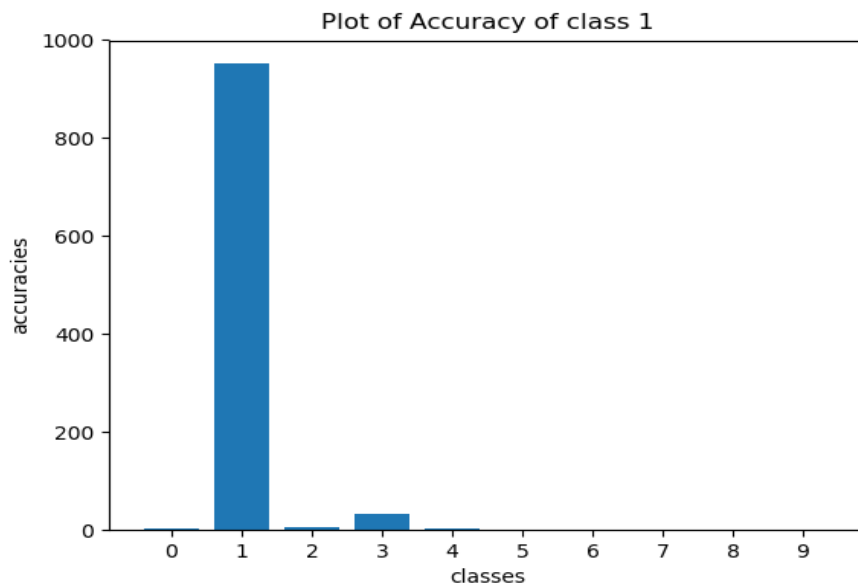


Plot of accuracies of individual classes:

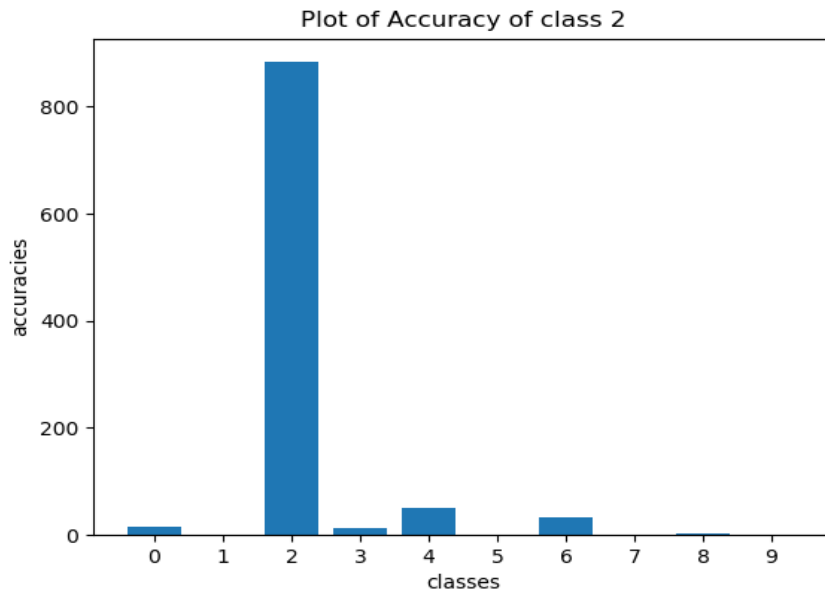
Plot of accuracy of class – 0:



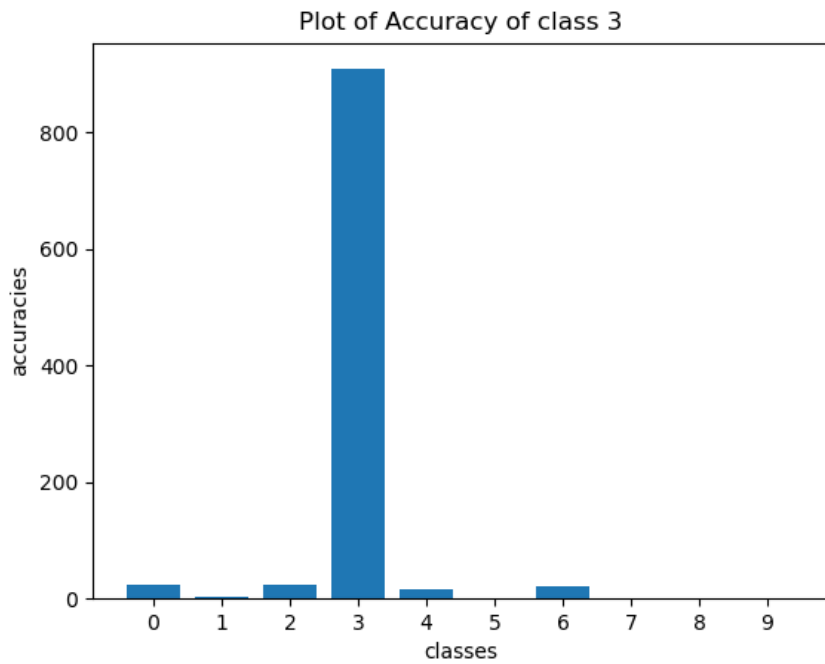
Plot of accuracy of class – 1:



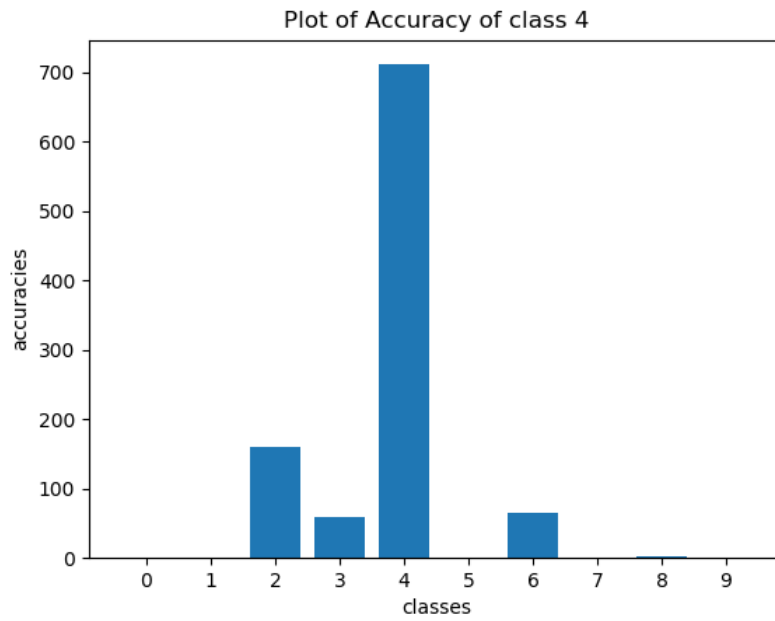
Plot of accuracy of class – 2:



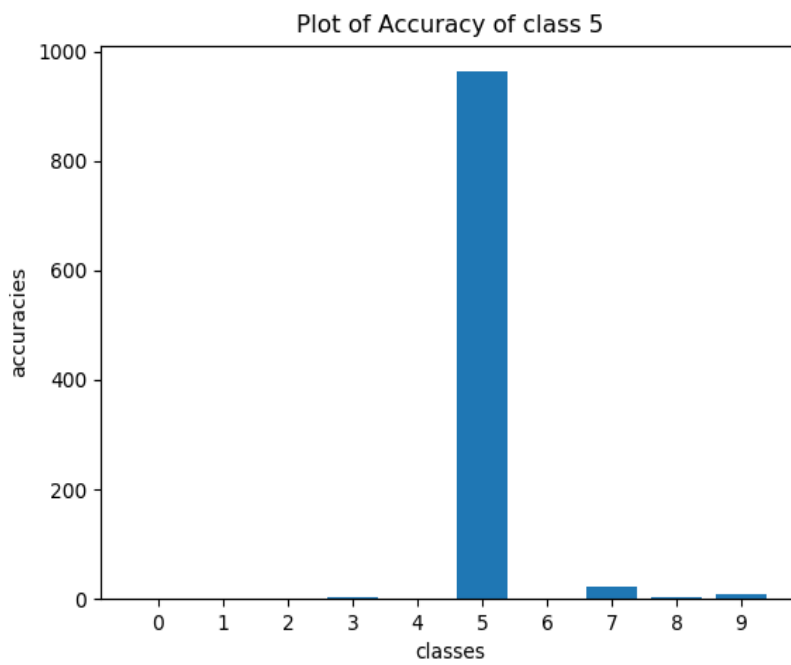
Plot of accuracy of class – 3:



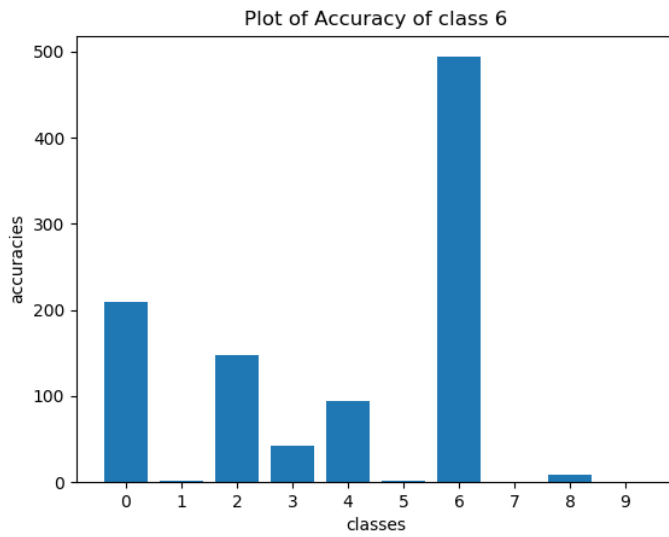
Plot of accuracy of class – 4:



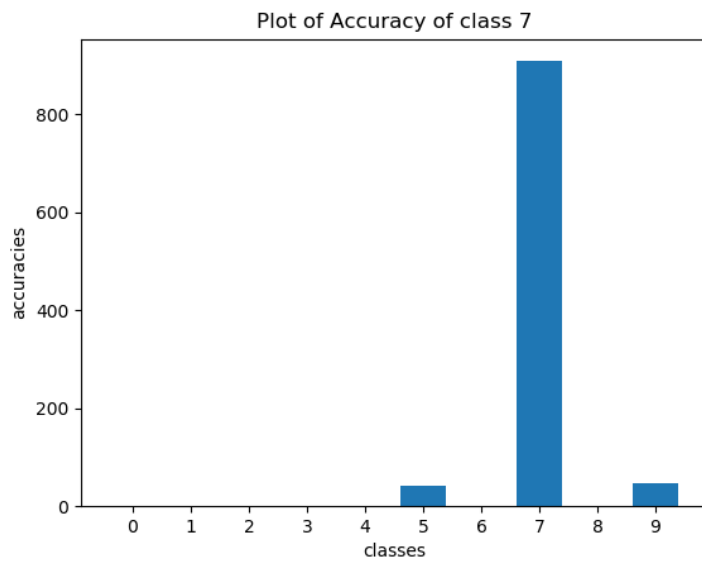
Plot of accuracy of class – 5:



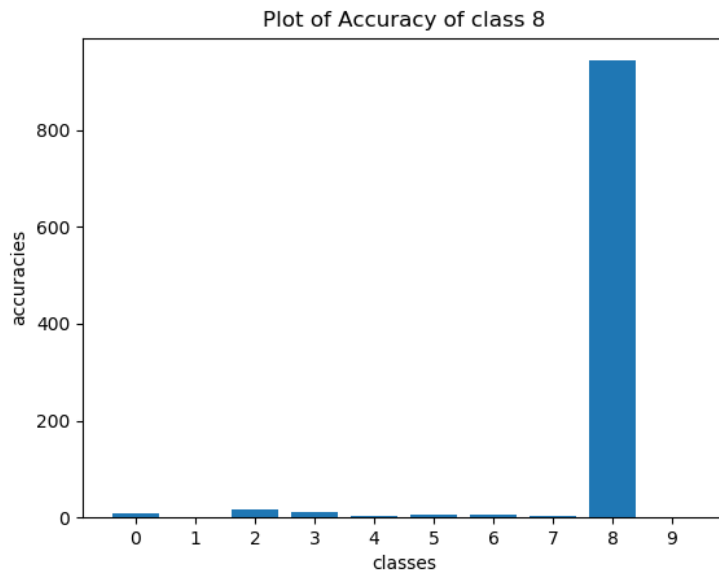
Plot of accuracy of class – 6:



Plot of accuracy of class – 7:



Plot of accuracy of class – 8:



Plot of accuracy of class – 9:

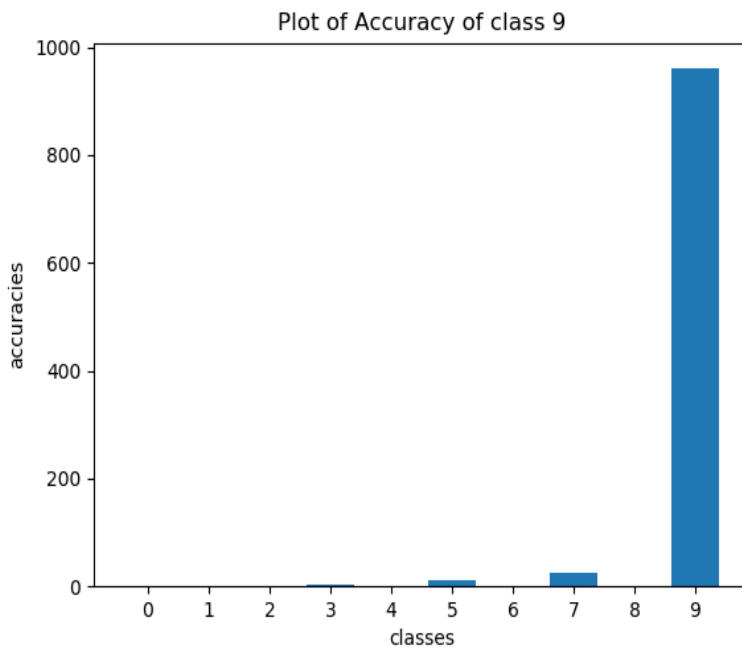
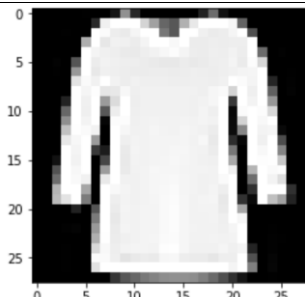
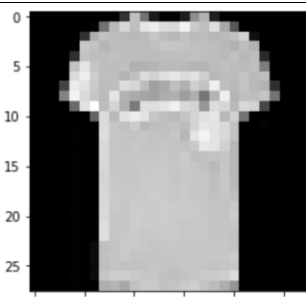
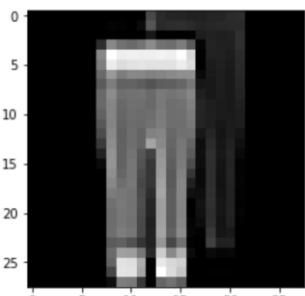
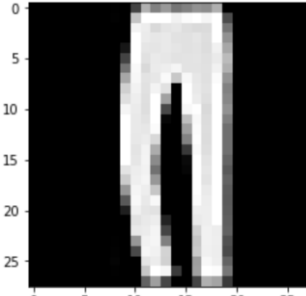
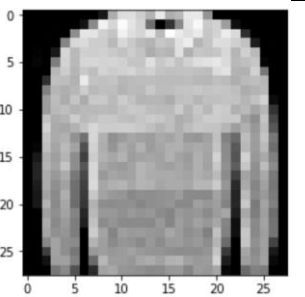
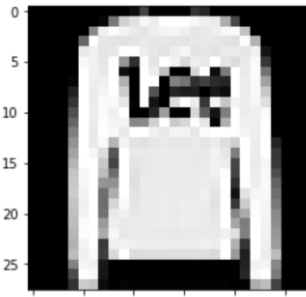
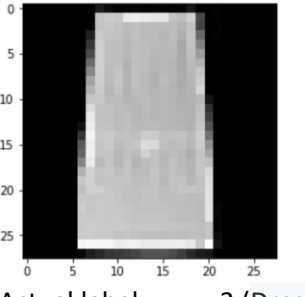
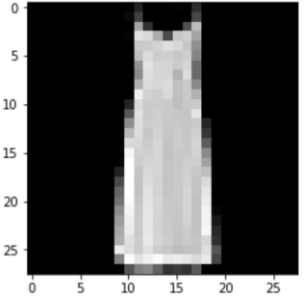
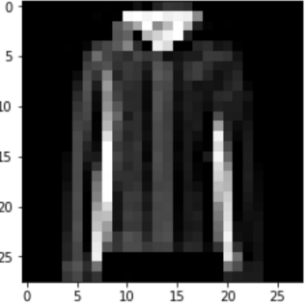
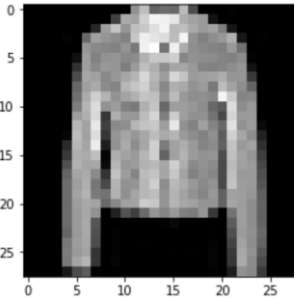
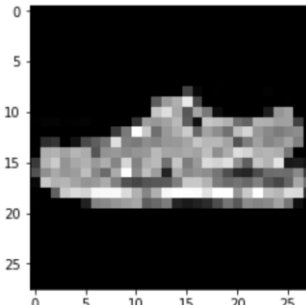
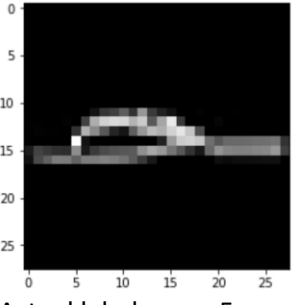
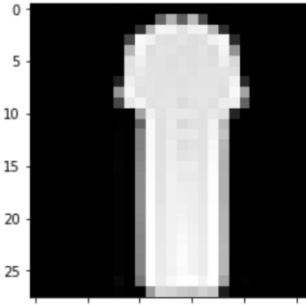
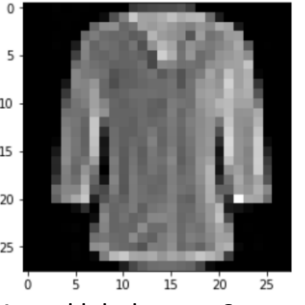
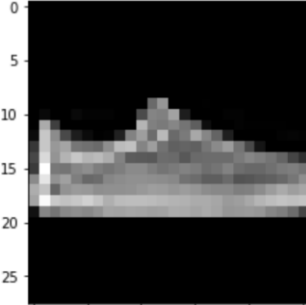
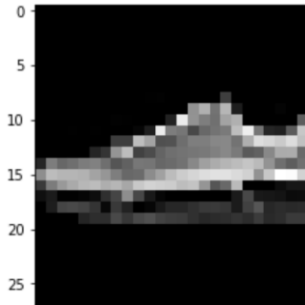
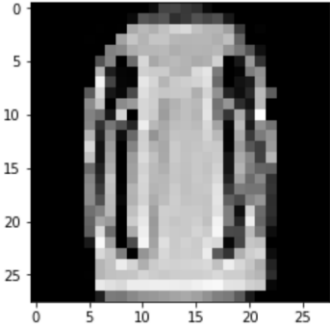
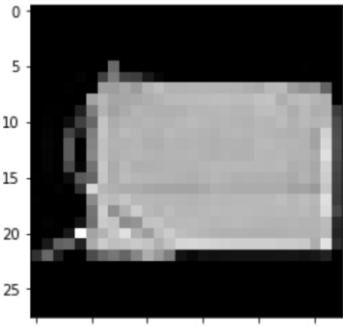
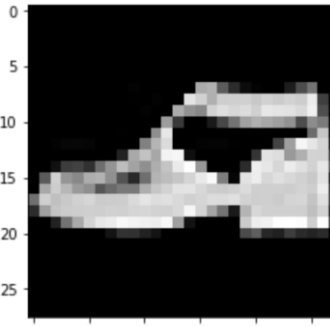
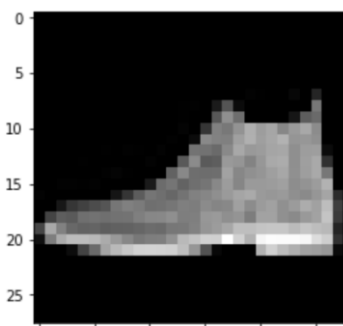


Table of predictions of test samples:-

Class labels	Incorrect Image samples	Correct Image samples
0	 <p>Actual label - 0 (T-shirt/top) Predicted label- 6</p>	 <p>Actual label - 0 Predicted label- 0</p>
1	 <p>Actual label - 1 (Trouser) Predicted label- 3</p>	 <p>Actual label - 1 Predicted label- 1</p>
2	 <p>Actual label - 2 (Pullover) Predicted label- 6</p>	 <p>Actual label - 2 Predicted label- 2</p>
3	 <p>Actual label - 3 (Dress) Predicted label- 6</p>	 <p>Actual label - 3 Predicted label- 3</p>

4	 <p>Actual label - 4 (Coat) Predicted label- 6</p>	 <p>Actual label - 4 Predicted label- 4</p>
5	 <p>Actual label - 5 (Sandal) Predicted label- 7</p>	 <p>Actual label - 5 Predicted label- 5</p>
6	 <p>Actual label - 6 (Shirt) Predicted label- 3</p>	 <p>Actual label - 6 Predicted label- 6</p>
7	 <p>Actual label - 7 (Sneaker) Predicted label- 8</p>	 <p>Actual label - 7 Predicted label- 7</p>

8	 <p>Actual label - 8 (Bag) Predicted label- 3</p>	 <p>Actual label - 8 Predicted label- 8</p>
9	 <p>Actual label - 9 (Ankle boot) Predicted label- 5</p>	 <p>Actual label - 9 Predicted label- 9</p>

From the above table of correct and incorrect samples we can observe that the model can predict the sample if the image is clear and those images which can be easily interpreted visually.