

**R.YUVASRI**

**B. TECH**

**INFORMATION TECHNOLOGY**

**SASTRA DEEMED UNIVERSITY**

## **DOCUMENTATION**

### **Teachnook Project**

**Machine learning Major project**

**Topic: Image Classification using CNNs**

**Objective:** Building an image classification system using CNN.

**Steps:**

#### **1. Dataset Collection:**

Gathered a large dataset of labelled images from fashion\_mnist dataset using tensorflow.

#### **2. Data Preprocessing:**

Processed the data by resizing all the images, normalising pixel values and splitting them into training and testing data.

**Code:**

```
train_images=train_images.astype('float32')/255
```

```
test_images=test_images.astype('float32')/255
```

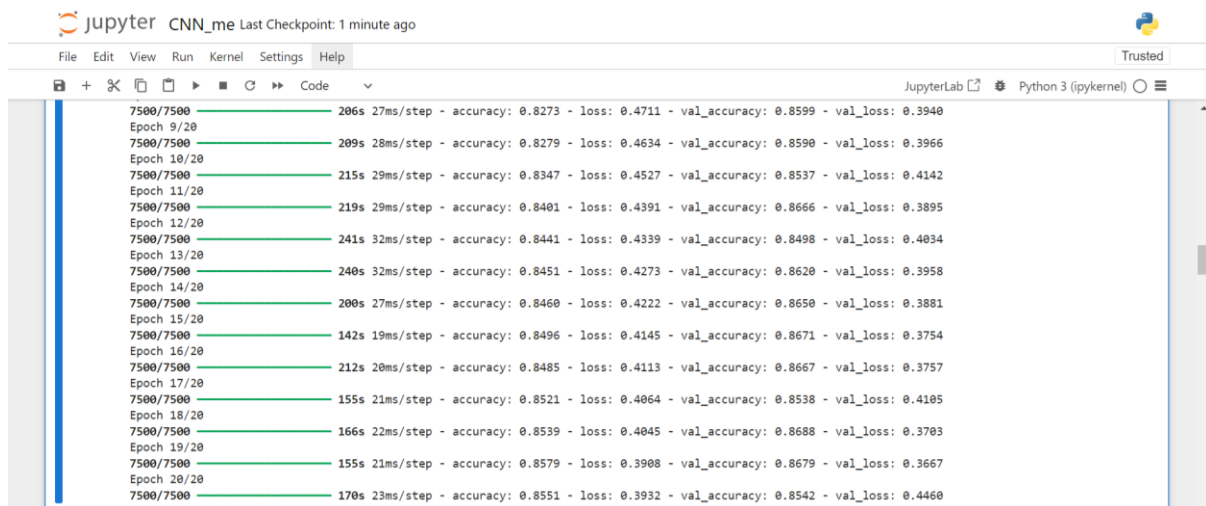
```
train_images=train_images.reshape(train_images.shape[0],28,28,1)
```

#### **3. CNN Model :**

Built the model by adding different layers like-input, hidden,output layers with neurons,activation function like **relu** and **softmax**.

## 4. Model Training:

Trained the model using optimizer and running the epoch 20 times.



## 5. Model Evaluation

Evaluated the trained model using the testing dataset. Calculated metrics such as accuracy, precision, recall, and F1-score to assess the model's performance.

```
File Edit View Run Kernel Settings Help
+ ✂ 📄 ▶ ⏮ ⏭ ⏪ ⏩ Code
JupyterLab Python 3

[20]: y_pred = model.predict(test_images).argmax(axis=-1)
313/313 ————— 3s 10ms/step

[21]: pprint.pprint(classification_report(test_labels, y_pred))

('      precision    recall  f1-score   support\n'
 '\n'
 '      0      0.86      0.78      0.82     1000\n'
 '      1      0.98      0.97      0.98     1000\n'
 '      2      0.92      0.68      0.78     1000\n'
 '      3      0.86      0.91      0.88     1000\n'
 '      4      0.81      0.75      0.78     1000\n'
 '      5      0.94      0.97      0.95     1000\n'
 '      6      0.59      0.78      0.67     1000\n'
 '      7      0.93      0.92      0.93     1000\n'
 '      8      0.94      0.98      0.96     1000\n'
 '      9      0.96      0.94      0.95     1000\n'
 '\n'
 ' accuracy      0.87     10000\n'
 ' macro avg      0.88      0.87      0.87     10000\n'
 'weighted avg      0.88      0.87      0.87     10000\n')

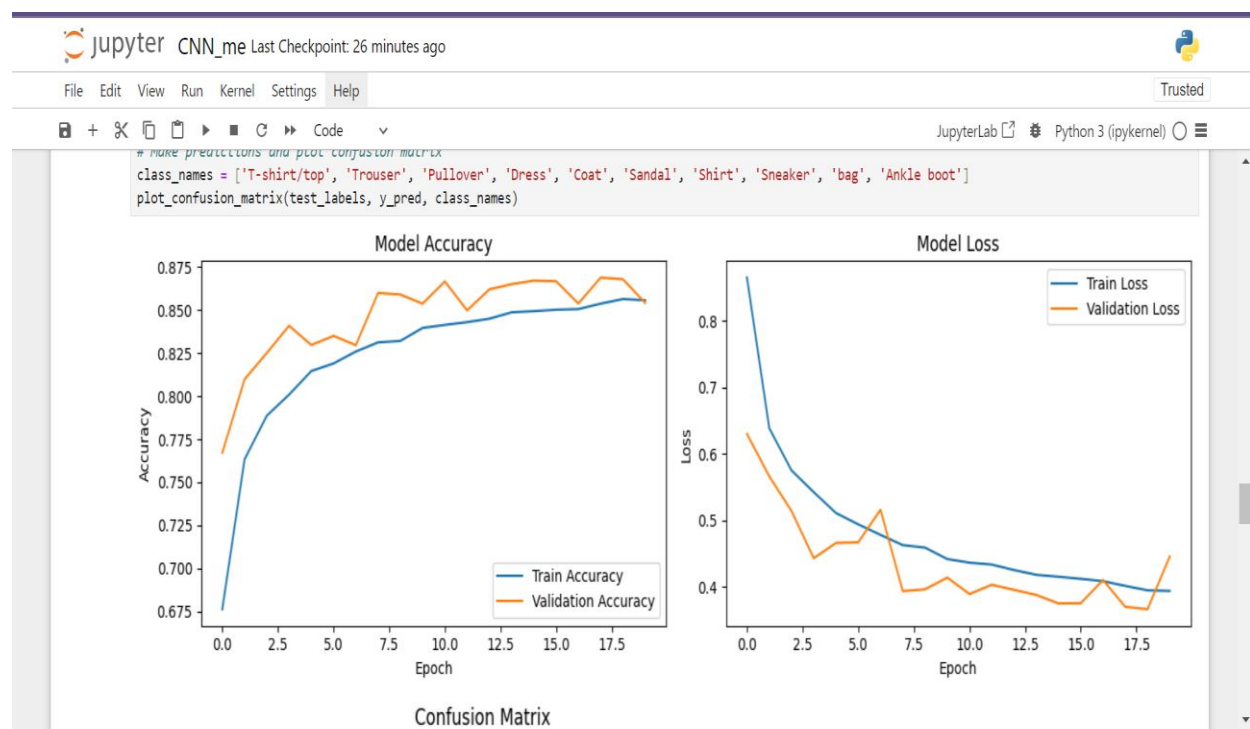
[22]: import seaborn as sns
```

## 6. Model Optimization

Fine-tuned the model by applying techniques like:

**data agumentation to flip and zoom data:**

```
data_agumentation = tf.keras.Sequential([  
    layers.RandomRotation(0.1),  
    layers.RandomFlip(),  
    layers.RandomZoom(0.1),  
    layers.RandomTranslation(0.1,0.1)  
])
```



**7. Deployment and Interface:** Input image is given to the model and the trained model predicts it.

