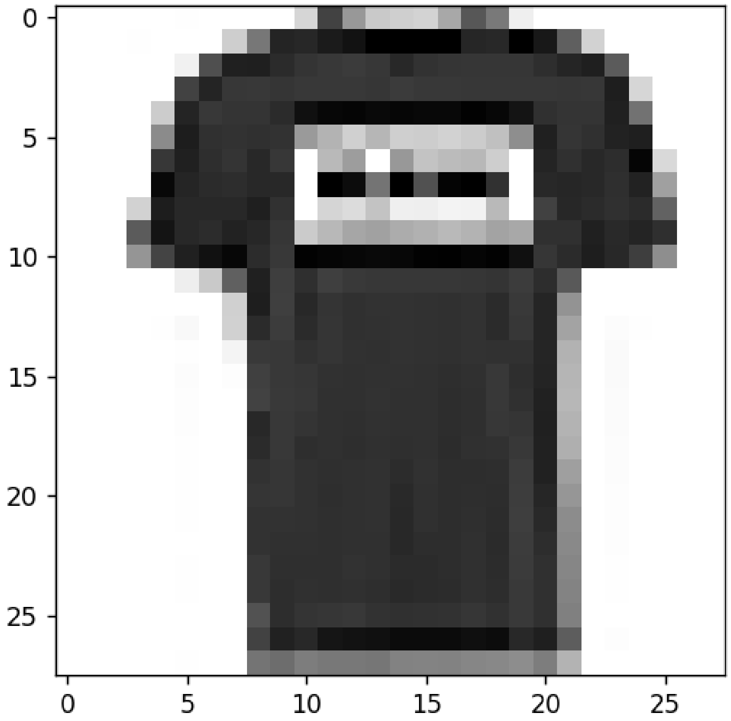
Program No: 13

Aim: Programs on convolutional neural network to classify images from any standard dataset in the public domain.

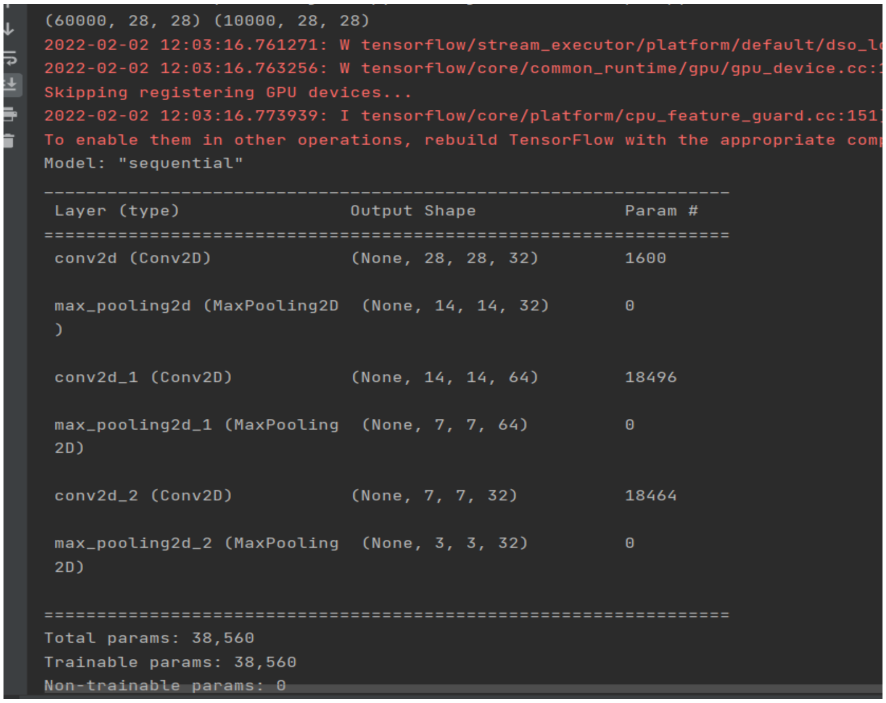
Program

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import tensorflow as tf

from tensorflow import keras

np.random.seed(42)

fashion_mnist=keras.datasets.fashion_mnist

(x_train,y_train),(x_test,y_test)=fashion_mnist.load_data()

print(x_train.shape,x_test.shape)

x_train=x_train/255.0

x_test=x_test/255.0

plt.imshow(x_train[1],cmap='binary')

plt.show()

np.unique(y_test)

class_names=['T-shirt/Top','Trouser','Pullover','Dress','Coat','Sandal','Shirt','Sneaker','Bag','Ankle Boot']

n_rows=5

n_cols=10

plt.figure(figsize=(n_cols * 1.4,n_rows * 1.6))

for row in range(n_rows):
    for col in range(n_cols):
        index=n_cols * row +col
        plt.subplot(n_rows,n_cols,index+1)
        plt.imshow(x_train[index],cmap='binary',interpolation='nearest')
        plt.axis('off')
        plt.title(class_names[y_train[index]])
plt.show()
model_CNN=keras.models.Sequential()
model_CNN.add(keras.layers.Conv2D(filters=32,kernel_size=7,padding='same',activation='relu',
input_shape=[28,28,1]))
model_CNN.add(keras.layers.MaxPooling2D(pool_size=2))
model_CNN.add(keras.layers.Conv2D(filters=64,kernel_size=3,padding='same',activation='relu'
))
model_CNN.add(keras.layers.MaxPooling2D(pool_size=2))

model_CNN.add(keras.layers.Conv2D(filters=32,kernel_size=3,padding='same',activation='relu'
))
model_CNN.add(keras.layers.MaxPooling2D(pool_size=2))
model_CNN.summary()
model_CNN.add(keras.layers.Flatten())
model_CNN.add(keras.layers.Dense(units=128,activation='relu'))
model_CNN.add(keras.layers.Dense(units=64,activation='relu'))
model_CNN.add(keras.layers.Dense(units=10,activation='softmax'))
model_CNN.summary()
model_CNN.compile(loss='sparse_categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

x_train=x_train[...,np.newaxis]

x_test=x_test[...,np.newaxis]

history_CNN=model_CNN.fit(x_train,y_train,epochs=2,validation_split=0.1)

pd.DataFrame(history_CNN.history).plot()

plt.grid(True)

plt.xlabel('epochs')

plt.ylabel('loss/accuracy')

plt.title('Training and validation plot')

plt.show()

test_loss,test_accuracy=model_CNN.evaluate(x_test,y_test)

print('Test Loss:{}','Test Accuracy:{}'.format(test_loss,test_accuracy))
```

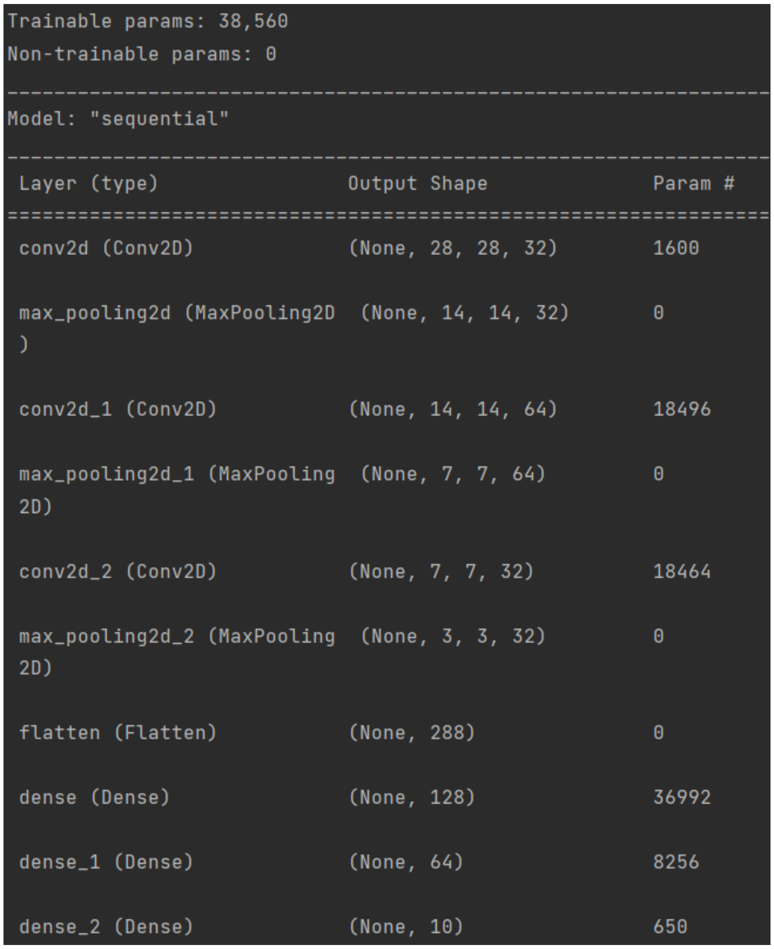
OUTPUT


0
5
10
15
20
25
0
5
10
15
20
25

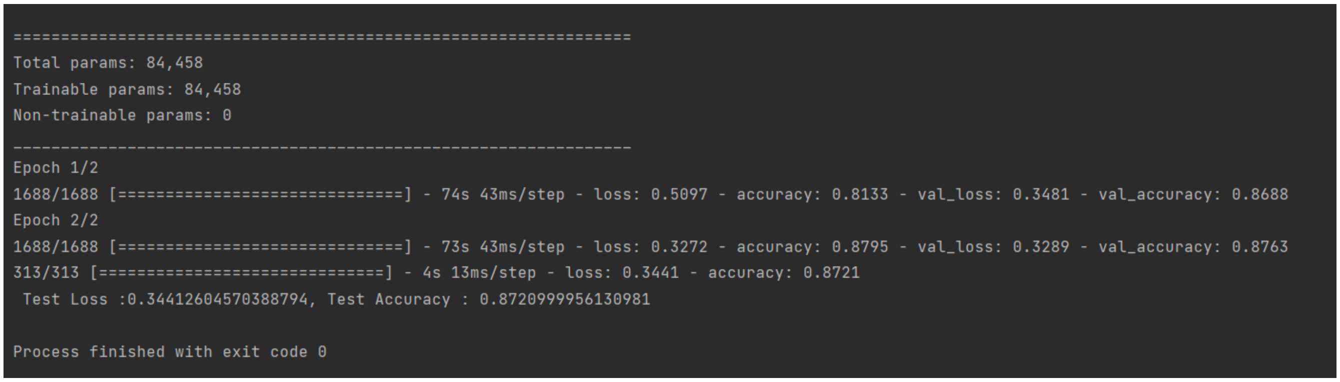
Ankle Boot
T-shirt/Top
T-shirt/Top
Dress
T-shirt/Top
Pullover
Sneaker
Pullover
Sandal
Sandal
T-shirt/Top
Ankle Boot
Sandal
Sandal
Sneaker
Ankle Boot
Trouser
T-shirt/Top
Shirt
Coat
Dress
Trouser
Coat
Bag
Coat
Dress
T-shirt/Top
Pullover
Coat
Coat
Sandal
Dress
Shirt
Shirt
T-shirt/Top
Bag
Sandal
Pullover
Trouser
Shirt
Shirt
Sneaker
Ankle Boot
Sandal
Ankle Boot
Pullover
Sneaker
Dress
T-shirt/Top
Dress



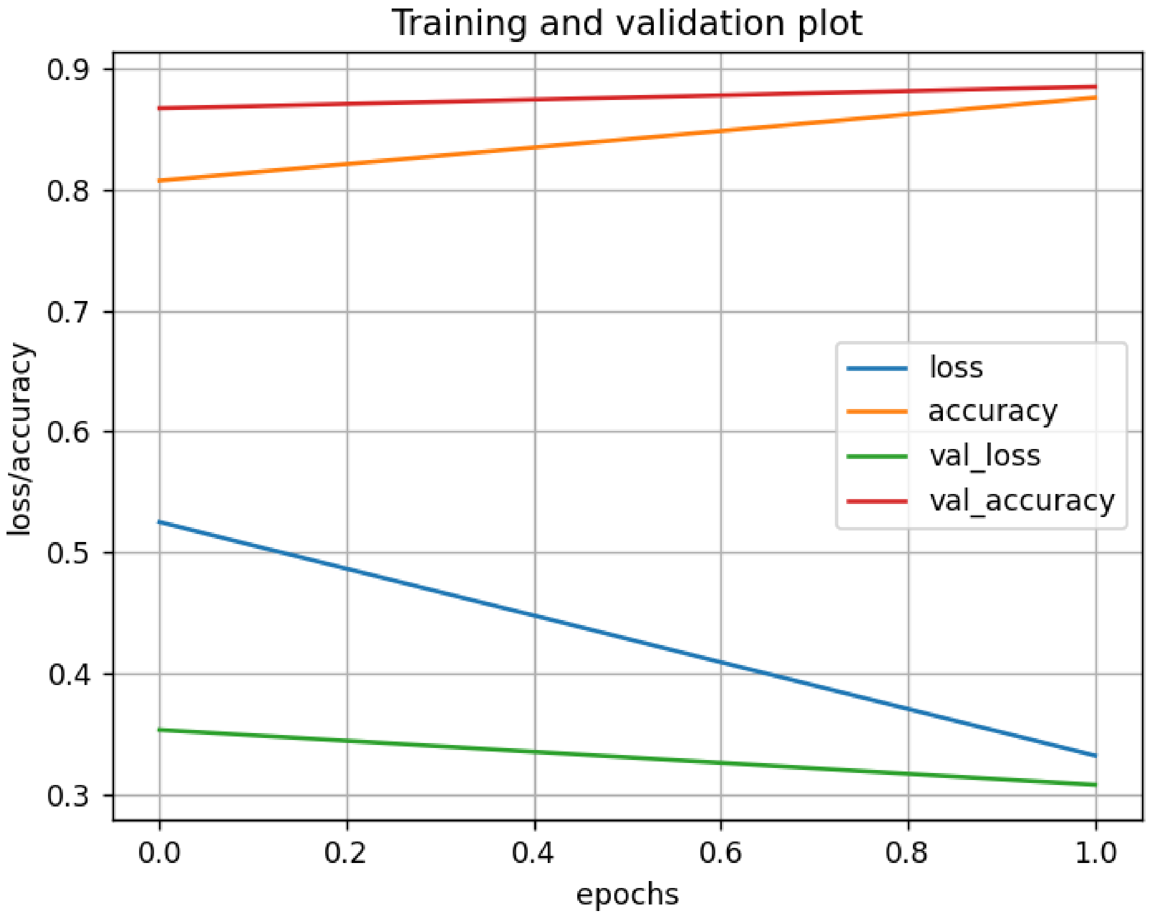
```
(60000, 28, 28) (10000, 28, 28)
2022-02-02 12:03:16.761271: W tensorflow/stream_executor/platform/default/dso_l
2022-02-02 12:03:16.763256: W tensorflow/core/common_runtime/gpu/gpu_device.cc:
Skipping registering GPU devices...
2022-02-02 12:03:16.773939: I tensorflow/core/platform/cpu_feature_guard.cc:151
To enable them in other operations, rebuild TensorFlow with the appropriate comp
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 28, 28, 32)        1600

 max_pooling2d (MaxPooling2D  (None, 14, 14, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 14, 14, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 7, 7, 64)         0
 2D)

 conv2d_2 (Conv2D)           (None, 7, 7, 32)          18464

 max_pooling2d_2 (MaxPooling  (None, 3, 3, 32)         0
 2D)

=================================================================
Total params: 38,560
Trainable params: 38,560
Non-trainable params: 0
```



```
Trainable params: 38,560
Non-trainable params: 0
_________________________________________________________________
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 28, 28, 32)        1600

 max_pooling2d (MaxPooling2D  (None, 14, 14, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 14, 14, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 7, 7, 64)         0
 2D)

 conv2d_2 (Conv2D)           (None, 7, 7, 32)          18464

 max_pooling2d_2 (MaxPooling  (None, 3, 3, 32)         0
 2D)

 flatten (Flatten)           (None, 288)               0

 dense (Dense)               (None, 128)               36992

 dense_1 (Dense)             (None, 64)                8256

 dense_2 (Dense)             (None, 10)                650
```



```
=================================================================
Total params: 84,458
Trainable params: 84,458
Non-trainable params: 0
_________________________________________________________________
Epoch 1/2
1688/1688 [==============================] - 74s 43ms/step - loss: 0.5097 - accuracy: 0.8133 - val_loss: 0.3481 - val_accuracy: 0.8688
Epoch 2/2
1688/1688 [==============================] - 73s 43ms/step - loss: 0.3272 - accuracy: 0.8795 - val_loss: 0.3289 - val_accuracy: 0.8763
313/313 [==============================] - 4s 13ms/step - loss: 0.3441 - accuracy: 0.8721
 Test Loss :0.34412604570388794, Test Accuracy : 0.8720999956130981

Process finished with exit code 0
```


Training and validation plot
loss/accuracy
epochs
loss
accuracy
val_loss
val_accuracy