```
#VAISHNAVI SOLANKAR
#MY PROGRAM
from keras.datasets import fashion_mnist
```

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

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
29515/29515 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26421880/26421880 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
5148/5148 ━━━━━━━━━━━━━━━━━━━━ 0s 1us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4422102/4422102 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
```

```
x_train.shape
```

```
(60000, 28, 28)
```

```
x_test.shape
```

```
(10000, 28, 28)
```

```
cat = ['T-shirt/top','Trouser','Pullover','Dress','Coat','Sandal','Shirt','Sneaker','Bag','Ankle boot']
```

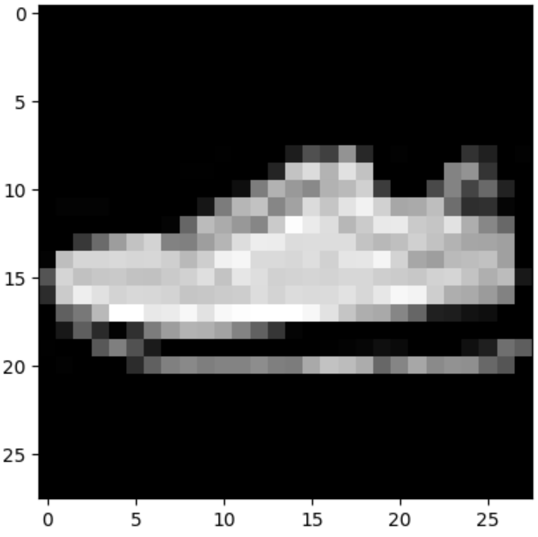
```
set(y_train)
```

```
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
```

```
import matplotlib.pyplot as plt
```

```
plt.imshow(x_train[425],cmap='gray')
```

```
<matplotlib.image.AxesImage at 0x786af7bdb8e0>
```



```
plt.figure(figsize=(16,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(x_train[i])
    plt.xlabel(cat[y_train[i]])
```

```
x_train[1];
```

```
x_train=x_train.astype('float32')/255.0
x_test=x_test.astype('float32')/255.0
```

```
x_train[1].shape
```

```
(28, 28)
```

```
#add colour channel
import numpy as np

x_train=np.expand_dims(x_train,axis=-1)
x_test=np.expand_dims(x_test,axis=-1)
```

```
x_train[1].shape
```

```
(28, 28, 1)
```

```
#one hot encoding

from keras.utils import to_categorical

y_train=to_categorical(y_train)
y_test=to_categorical(y_test)
```

```
#define the model architecture

from keras.models import Sequential
from keras.layers import Conv2D,MaxPooling2D,Flatten,Dense,Dropout
```

```
model=Sequential([
    Conv2D(32,(3,3),activation='relu',input_shape=(28,28,1)),
    MaxPooling2D((2,2)),
    Conv2D(64,(3,3),activation='relu'),
    MaxPooling2D((2,2)),
    Conv2D(64,(3,3),activation='relu'),
    Flatten(),
    Dense(64,activation='relu'),
    Dense(10,activation='softmax')
])
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

```
model.summary()
```

```
Model: "sequential"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━┓
┃ Layer (type)                         ┃ Output Shape                ┃         Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━┩
│ conv2d (Conv2D)                      │ (None, 26, 26, 32)          │             320 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ max_pooling2d (MaxPooling2D)         │ (None, 13, 13, 32)          │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ conv2d_1 (Conv2D)                    │ (None, 11, 11, 64)          │          18,496 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ max_pooling2d_1 (MaxPooling2D)       │ (None, 5, 5, 64)            │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ conv2d_2 (Conv2D)                    │ (None, 3, 3, 64)            │          36,928 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ flatten (Flatten)                    │ (None, 576)                 │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense (Dense)                        │ (None, 64)                  │          36,928 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense_1 (Dense)                      │ (None, 10)                  │             650 │
└──────────────────────────────────────┴─────────────────────────────┴─────────────────┘
 Total params: 93,322 (364.54 KB)
 Trainable params: 93,322 (364.54 KB)
 Non-trainable params: 0 (0.00 B)
```

```
#compile the model
model.compile(optimizer='SGD',loss='categorical_crossentropy', metrics=['accuracy'])
```

```
#train the model
history= model.fit(x_train,y_train,epochs=10,batch_size=10,validation_split=0.2)
```

```
Epoch 1/10
4800/4800 ━━━━━━━━━━━━━━━━━━━━ 78s 16ms/step - accuracy: 0.5775 - loss: 1.1381 - val_accuracy: 0.8191 - val_loss: 0.4999
Epoch 2/10
4800/4800 ━━━━━━━━━━━━━━━━━━━━ 68s 14ms/step - accuracy: 0.8268 - loss: 0.4688 - val_accuracy: 0.8192 - val_loss: 0.5221
Epoch 3/10
4800/4800 ━━━━━━━━━━━━━━━━━━━━ 65s 14ms/step - accuracy: 0.8554 - loss: 0.3981 - val_accuracy: 0.8587 - val_loss: 0.3981
Epoch 4/10
4800/4800 ━━━━━━━━━━━━━━━━━━━━ 83s 14ms/step - accuracy: 0.8682 - loss: 0.3576 - val_accuracy: 0.8717 - val_loss: 0.3544
Epoch 5/10
4800/4800 ━━━━━━━━━━━━━━━━━━━━ 69s 14ms/step - accuracy: 0.8816 - loss: 0.3254 - val_accuracy: 0.8718 - val_loss: 0.3561
Epoch 6/10
4800/4800 ━━━━━━━━━━━━━━━━━━━━ 66s 14ms/step - accuracy: 0.8873 - loss: 0.3027 - val_accuracy: 0.8802 - val_loss: 0.3279
Epoch 7/10
4800/4800 ━━━━━━━━━━━━━━━━━━━━ 81s 14ms/step - accuracy: 0.8973 - loss: 0.2812 - val_accuracy: 0.8687 - val_loss: 0.3525
Epoch 8/10
4800/4800 ━━━━━━━━━━━━━━━━━━━━ 75s 16ms/step - accuracy: 0.9029 - loss: 0.2591 - val_accuracy: 0.8817 - val_loss: 0.3099
Epoch 9/10
4800/4800 ━━━━━━━━━━━━━━━━━━━━ 72s 14ms/step - accuracy: 0.9067 - loss: 0.2558 - val_accuracy: 0.8937 - val_loss: 0.2957
Epoch 10/10
4800/4800 ━━━━━━━━━━━━━━━━━━━━ 81s 13ms/step - accuracy: 0.9109 - loss: 0.2427 - val_accuracy: 0.8965 - val_loss: 0.2890
```

```
loss,accuracy=model.evaluate(x_test,y_test)
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 3s 9ms/step - accuracy: 0.8909 - loss: 0.3029
```