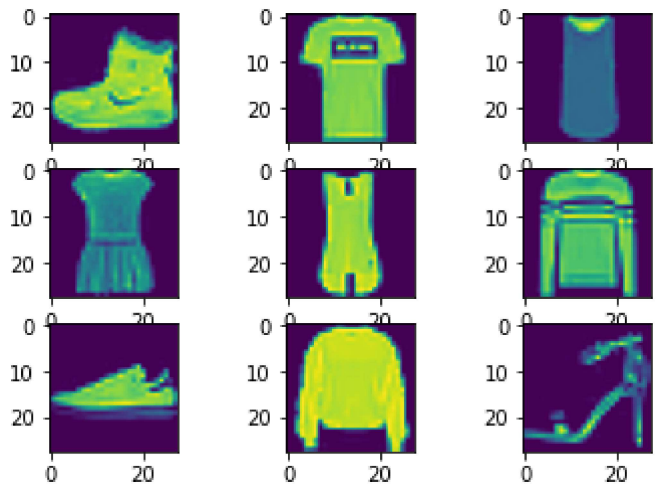
```
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, BatchNormalization, Activation
from keras.layers.convolutional import Conv2D, MaxPooling2D
from keras.utils import np_utils
from keras.datasets import fashion_mnist
import PIL
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
from tensorflow import keras
```

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

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-
32768/29515 [================================] - 0s 0us/step
40960/29515 [========================================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-
26427392/26421880 [==============================] - 0s 0us/step
26435584/26421880 [==============================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-
16384/5148 [================================================================================
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-
4423680/4422102 [==============================] - 0s 0us/step
4431872/4422102 [==============================] - 0s 0us/step
```

```
import matplotlib.pyplot as plt
for i in range(9):
  plt.subplot(330+i+1)
  plt.imshow(x_train[i])
plt.show()
```



```
x = x_test
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train = x_train/255
x_test = x_test/255
```

```
y_train = np_utils.to_categorical(y_train,10)
y_test = np_utils.to_categorical(y_test,10)
```

```
from keras.layers.convolutional import Conv2D, MaxPooling2D
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, BatchNormalization, Activation
from tensorflow.keras.models import Sequential
model=Sequential()
model.add(Conv2D(64,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'
model.add(Conv2D(64,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'
model.add(MaxPooling2D(2,2))
```

```
model.add(MaxPooling2D(2,2))
```

```
model.add(Flatten())
```

```
model.add(Dense(128, activation = 'relu', kernel_initializer = 'he_uniform',input_shape=(2
model.add(Dropout(0.2))
model.add(Dense(10,activation='relu'))
model.add(Dropout(0.1))
```

```
model.add(Dense(10, activation = 'softmax'))
```

```
from tensorflow.keras.optimizers import SGD
```

```
opt = SGD(lr = 0.01, momentum = 0.9)
```

```
/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/gradient_descent.py:102: UserW
  super(SGD, self).__init__(name, **kwargs)
```

```
from keras.backend import categorical_crossentropy
model.compile(optimizer = opt, loss = 'categorical_crossentropy', metrics = ['accuracy'])
```

```
history = model.fit(x_train, y_train, epochs = 50, batch_size = 128, validation_data = (x_
```

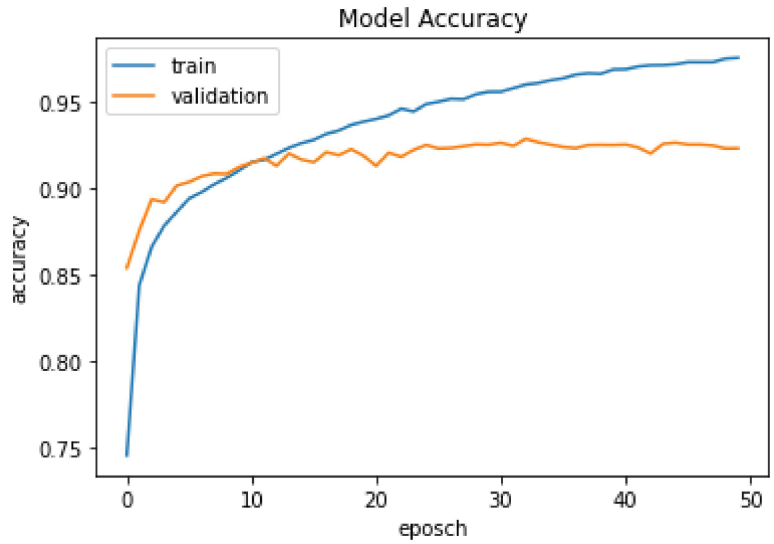
```
Epoch 23/50
469/469 [==============================] - 6s 13ms/step - loss: 0.1527 - accuracy: 0
Epoch 24/50
469/469 [==============================] - 7s 16ms/step - loss: 0.1527 - accuracy: 0
```

```
Epoch 25/50
469/469 [==============================] - 6s 12ms/step - loss: 0.1440 - accuracy: 0
Epoch 26/50
469/469 [==============================] - 6s 12ms/step - loss: 0.1385 - accuracy: 0
Epoch 27/50
469/469 [==============================] - 6s 12ms/step - loss: 0.1355 - accuracy: 0
Epoch 28/50
469/469 [==============================] - 6s 12ms/step - loss: 0.1328 - accuracy: 0
Epoch 29/50
469/469 [==============================] - 6s 12ms/step - loss: 0.1270 - accuracy: 0
Epoch 30/50
469/469 [==============================] - 6s 12ms/step - loss: 0.1227 - accuracy: 0
Epoch 31/50
469/469 [==============================] - 6s 12ms/step - loss: 0.1216 - accuracy: 0
Epoch 32/50
469/469 [==============================] - 6s 12ms/step - loss: 0.1159 - accuracy: 0
Epoch 33/50
469/469 [==============================] - 6s 12ms/step - loss: 0.1115 - accuracy: 0
Epoch 34/50
469/469 [==============================] - 6s 13ms/step - loss: 0.1080 - accuracy: 0
Epoch 35/50
469/469 [==============================] - 6s 12ms/step - loss: 0.1048 - accuracy: 0
Epoch 36/50
469/469 [==============================] - 6s 12ms/step - loss: 0.0999 - accuracy: 0
Epoch 37/50
469/469 [==============================] - 6s 12ms/step - loss: 0.0956 - accuracy: 0
Epoch 38/50
469/469 [==============================] - 6s 12ms/step - loss: 0.0931 - accuracy: 0
Epoch 39/50
469/469 [==============================] - 6s 12ms/step - loss: 0.0932 - accuracy: 0
Epoch 40/50
469/469 [==============================] - 6s 12ms/step - loss: 0.0887 - accuracy: 0
Epoch 41/50
469/469 [==============================] - 6s 12ms/step - loss: 0.0891 - accuracy: 0
Epoch 42/50
469/469 [==============================] - 6s 12ms/step - loss: 0.0843 - accuracy: 0
Epoch 43/50
469/469 [==============================] - 6s 12ms/step - loss: 0.0816 - accuracy: 0
Epoch 44/50
469/469 [==============================] - 6s 12ms/step - loss: 0.0798 - accuracy: 0
Epoch 45/50
469/469 [==============================] - 6s 12ms/step - loss: 0.0801 - accuracy: 0
Epoch 46/50
469/469 [==============================] - 6s 12ms/step - loss: 0.0756 - accuracy: 0
Epoch 47/50
469/469 [==============================] - 6s 12ms/step - loss: 0.0768 - accuracy: 0
Epoch 48/50
469/469 [==============================] - 6s 12ms/step - loss: 0.0738 - accuracy: 0
Epoch 49/50
469/469 [==============================] - 6s 12ms/step - loss: 0.0698 - accuracy: 0
Epoch 50/50
469/469 [==============================] - 6s 12ms/step - loss: 0.0702 - accuracy: 0
```

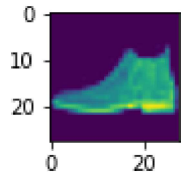
```
model.save('final_fashionmnist_cnn.h5')
```

```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('accuracy')
plt.xlabel('eposch')
plt.legend(['train','validation'], loc = 'upper left')
```

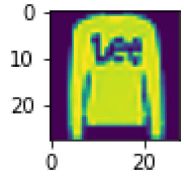
```
<matplotlib.legend.Legend at 0x7f23000c9850>
```



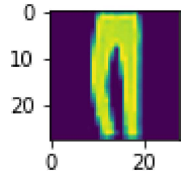

```
y_pred=model.predict(x_test)
for i in range(9):
  plt.subplot(330+i+1)
  plt.imshow(x[i])
  plt.show()
  print(np.round(y_pred[i]))
```



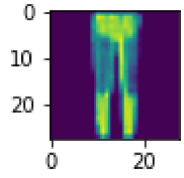
```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
```



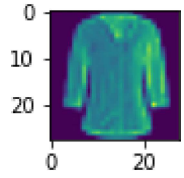
```
[0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
```



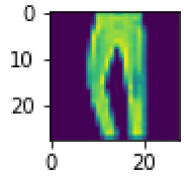
```
[0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
```



```
[0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
```

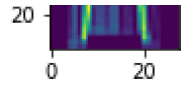


```
[0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
```

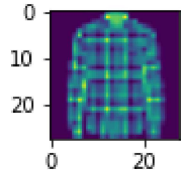


```
[0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
```

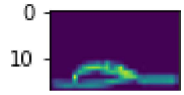

1

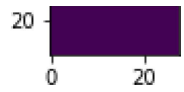


```
[0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]
```



```
[0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
```






```
[0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
```