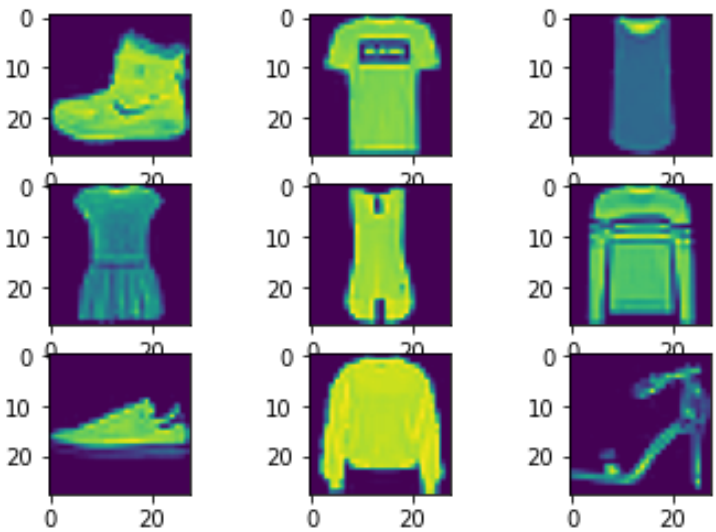
CNN_fashion

```
from keras.datasets import fashion_mnist
import matplotlib.pyplot as plt
from tensorflow.keras.utils import to_categorical


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

```
    (60000, 28, 28)
    (10000, 28, 28)
```

```
for i in range(9):
  plt.subplot(330+i+1)    # 330 mean: 3 hang 3 cot
  plt.imshow(x_train[i])
plt.show()
```



```
x = x_test
x_train = x_train.reshape((x_train.shape[0], x_train.shape[1], x_train.shape[2], 1))
x_test = x_test .reshape((x_test.shape[0], x_test.shape[1], x_test.shape[2], 1))
print(x_train.shape, y_train.shape)
print(x_test.shape, y_test.shape)
```

```
    (60000, 28, 28, 1) (60000,)
    (10000, 28, 28, 1) (10000,)
```

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


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

```
model = Sequential()

from tensorflow.keras.layers import Dense, Activation, Dropout
model.add(Conv2D(32, (3,3), activation='relu', input_shape= (28,28,1)))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(48, (3,3), activation='relu'))
model.add(MaxPooling2D((2,2)))
model.add(Dropout(0.5))
model.add(Flatten())
model.add(Dense(500, activation='relu'))
model.add(Dense(10, activation='softmax'))
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, 48)        13872

 max_pooling2d_1 (MaxPooling  (None, 5, 5, 48)         0
 2D)

 dropout (Dropout)           (None, 5, 5, 48)          0

 flatten (Flatten)           (None, 1200)              0

 dense (Dense)               (None, 500)               600500

 dense_1 (Dense)             (None, 10)                5010

=================================================================
Total params: 619,702
Trainable params: 619,702
Non-trainable params: 0
_________________________________________________________________
```

```
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'

history = model.fit(x_train, y_train, epochs=10, batch_size = 128, verbose= 2, validation_s
```

```
Epoch 1/10
422/422 - 15s - loss: 0.5899 - accuracy: 0.7848 - val_loss: 0.4024 - val_accuracy: 0.
Epoch 2/10
422/422 - 2s - loss: 0.4136 - accuracy: 0.8513 - val_loss: 0.3494 - val_accuracy: 0.8
Epoch 3/10
422/422 - 2s - loss: 0.3621 - accuracy: 0.8671 - val_loss: 0.3236 - val_accuracy: 0.8
Epoch 4/10
422/422 - 2s - loss: 0.3321 - accuracy: 0.8777 - val_loss: 0.3105 - val_accuracy: 0.8
```

```
Epoch 5/10
422/422 - 2s - loss: 0.3093 - accuracy: 0.8867 - val_loss: 0.2802 - val_accuracy: 0.8
Epoch 6/10
422/422 - 2s - loss: 0.2867 - accuracy: 0.8934 - val_loss: 0.2633 - val_accuracy: 0.9
Epoch 7/10
422/422 - 2s - loss: 0.2768 - accuracy: 0.8966 - val_loss: 0.2553 - val_accuracy: 0.9
Epoch 8/10
422/422 - 2s - loss: 0.2598 - accuracy: 0.9030 - val_loss: 0.2464 - val_accuracy: 0.9
Epoch 9/10
422/422 - 2s - loss: 0.2519 - accuracy: 0.9053 - val_loss: 0.2391 - val_accuracy: 0.9
Epoch 10/10
422/422 - 2s - loss: 0.2421 - accuracy: 0.9095 - val_loss: 0.2361 - val_accuracy: 0.9
```

```
score = model.evaluate(x_test, y_test,verbose=0)
print('Sai so kiem tra la:', score[0])
print('Do chinh xac kiem tra la:', score[1]*100)
```

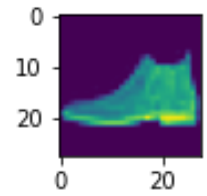
```
Sai so kiem tra la: 0.25060924887657166
Do chinh xac kiem tra la: 90.6000018119812
```

```
model.save('CNN_FASHION_MNIST.h5')
from keras.models import load_model
model1 = load_model('CNN_FASHION_MNIST.h5')
```

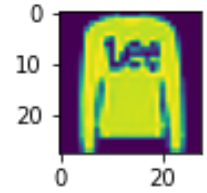
```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train','validation'], loc='upper-left')
```

```
    /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6: MatplotlibDeprecation
            best
            upper right
            upper left
            lower left
            lower right
            right
            center left
            center right
            lower center
            upper center
            center
```

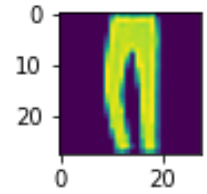
```
import numpy as np
y_pred = model.predict(x_test)
for i in range (9):
  plt.subplot(330+i+1)    # 330 mean: 3 hang 3 cot
  plt.imshow(x[i])
  plt.show()
  print(np.round(y_pred[i]))
```



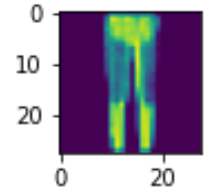
```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
```

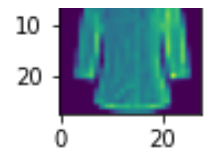


```
[0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
```

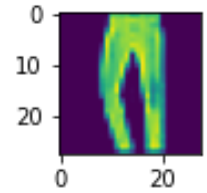


```
[0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
```

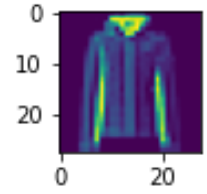




```
[0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
```



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



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