//MNIST Fashion

import tensorflow as tf

import matplotlib.pyplot as plt from tensorflow import keras import numpy as np

(x\_train, y\_train), (x\_test, y\_test) = keras.datasets.fashion\_mnist.load\_data()

plt.imshow(x\_train[1])

plt.imshow(x\_train[0])

x\_train = x\_train.astype('float32') / 255.0 x\_test = x\_test.astype('float32') / 255.0

x\_train.shape (60000, 28, 28)

x\_test.shape (10000, 28, 28, 1)

y\_train.shape (60000,)

y\_test.shape (10000,)

model = keras.Sequential([

keras.layers.Conv2D(32, (3,3), activation='relu', input\_shape=(28,28,1)), keras.layers.MaxPooling2D((2,2)),

# It shown 13 \* 13 size image with 32 channel or filter or depth. keras.layers.Dropout(0.25),

# Reduce Overfitting of Training sample drop out 25% Neuron keras.layers.Conv2D(64, (3,3), activation='relu'), keras.layers.MaxPooling2D((2,2)), keras.layers.Dropout(0.25),

keras.layers.Conv2D(128, (3,3), activation='relu'), keras.layers.Flatten(),

keras.layers.Dense(128, activation='relu'), keras.layers.Dropout(0.25), keras.layers.Dense(10, activation='softmax')

**])**

model.summary() Model: "sequential"

model.compile(optimizer='adam', loss='sparse\_categorical\_crossentropy', metrics=['accuracy']) history = model.fit(x\_train, y\_train, epochs=10, validation\_data=(x\_test, y\_test))

test\_loss, test\_acc = model.evaluate(x\_test, y\_test) print('Test accuracy:', test\_acc)