

```
import tensorflow as tf
from tensorflow.keras.datasets import fashion_mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.utils import to_categorical
```

```
(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 ————— 2s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
5148/5148 ————— 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4422102/4422102 ————— 1s 0us/step
```

```
X_train = X_train.reshape(-1, 28, 28, 1) / 255.0
X_test = X_test.reshape(-1, 28, 28, 1) / 255.0
y_train = to_categorical(y_train, 10)
y_test = to_categorical(y_test, 10)
```

```

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)),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(10, activation='softmax')
])
```

```

↳ /usr/local/lib/python3.11/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.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

```
model.fit(X_train, y_train, epochs=5, batch_size=64, validation_data=(X_test, y_test))
```

```

↳ Epoch 1/5
938/938 ————— 10s 6ms/step - accuracy: 0.7509 - loss: 0.6949 - val_accuracy: 0.8583 - val_loss: 0.4061
Epoch 2/5
938/938 ————— 3s 4ms/step - accuracy: 0.8747 - loss: 0.3483 - val_accuracy: 0.8824 - val_loss: 0.3270
Epoch 3/5
938/938 ————— 4s 4ms/step - accuracy: 0.8988 - loss: 0.2851 - val_accuracy: 0.8793 - val_loss: 0.3196
Epoch 4/5
938/938 ————— 4s 4ms/step - accuracy: 0.9074 - loss: 0.2515 - val_accuracy: 0.8925 - val_loss: 0.2923
Epoch 5/5
938/938 ————— 4s 5ms/step - accuracy: 0.9158 - loss: 0.2271 - val_accuracy: 0.8974 - val_loss: 0.2798
<keras.src.callbacks.history.History at 0x7d67d95e4f50>
```

```
test_loss, test_acc = model.evaluate(X_test, y_test)
print(f"Test Accuracy: {test_acc}")
```

```

↳ 313/313 ————— 1s 2ms/step - accuracy: 0.8973 - loss: 0.2851
Test Accuracy: 0.8974000215530396
```

Start coding or [generate](#) with AI.

