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

# Load the MNIST Fashion dataset
(x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()

# Preprocess the data
x_train = x_train.reshape(x_train.shape[0], 28, 28, 1) # Reshape to 28x28 grayscale images
x_test = x_test.reshape(x_test.shape[0], 28, 28, 1)
x_train = x_train.astype('float32') / 255 # Normalize pixel values to range [0, 1]
x_test = x_test.astype('float32') / 255
y_train = to_categorical(y_train, num_classes=10) # Convert labels to one-hot encoded vectors
y_test = to_categorical(y_test, num_classes=10)
```

```python
# Define the model architecture
model = Sequential([
    Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(28, 28, 1)),
    MaxPooling2D(pool_size=(2, 2)),
    Conv2D(64, kernel_size=(3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(10, activation='softmax')
])

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

# Train the model
model.fit(x_train, y_train, batch_size=128, epochs=10, validation_data=(x_test, y_test))

```

```
Epoch 1/10
469/469 [==============================] - 65s 135ms/step - loss: 0.6409 - accuracy: 0.7709 - val_loss: 0.4312 - val_accuracy: 0.83
Epoch 2/10
469/469 [==============================] - 60s 129ms/step - loss: 0.4217 - accuracy: 0.8485 - val_loss: 0.3649 - val_accuracy: 0.86
Epoch 3/10
469/469 [==============================] - 62s 131ms/step - loss: 0.3633 - accuracy: 0.8697 - val_loss: 0.3282 - val_accuracy: 0.87
Epoch 4/10
469/469 [==============================] - 59s 127ms/step - loss: 0.3288 - accuracy: 0.8822 - val_loss: 0.3029 - val_accuracy: 0.89
Epoch 5/10
469/469 [==============================] - 61s 130ms/step - loss: 0.3067 - accuracy: 0.8887 - val_loss: 0.2901 - val_accuracy: 0.89
Epoch 6/10
469/469 [==============================] - 60s 128ms/step - loss: 0.2856 - accuracy: 0.8956 - val_loss: 0.2770 - val_accuracy: 0.89
Epoch 7/10
469/469 [==============================] - 61s 130ms/step - loss: 0.2687 - accuracy: 0.9021 - val_loss: 0.2743 - val_accuracy: 0.90
Epoch 8/10
469/469 [==============================] - 60s 127ms/step - loss: 0.2530 - accuracy: 0.9083 - val_loss: 0.2582 - val_accuracy: 0.90
Epoch 9/10
469/469 [==============================] - 61s 131ms/step - loss: 0.2442 - accuracy: 0.9109 - val_loss: 0.2607 - val_accuracy: 0.90
Epoch 10/10
469/469 [==============================] - 60s 127ms/step - loss: 0.2305 - accuracy: 0.9150 - val_loss: 0.2538 - val_accuracy: 0.91
<keras.callbacks.History at 0x7f38e36217e0>
```

```python
# Evaluate the model on the testing set
loss, accuracy = model.evaluate(x_test, y_test)
print("Testing loss:", loss)
print("Testing accuracy:", accuracy)

```

```
313/313 [==============================] - 5s 15ms/step - loss: 0.2538 - accuracy: 0.9103
Testing loss: 0.2537824809551239
Testing accuracy: 0.9103000164031982
```