```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report, confusion_matrix
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense,
Dropout
from tensorflow.keras.optimizers import Adam

train_df = pd.read_csv('fashion-mnist_train.csv')
test_df = pd.read_csv('fashion-mnist_test.csv')

train_df.head()

train_df['label'].unique()

plt.figure(figsize=(12, 8))

for i in range(10):
    img = train_df[train_df['label'] == i].iloc[0, 1:].values.reshape(28,
28)
    plt.subplot(2, 5, i+1)
    plt.imshow(img, cmap='gray')
    plt.title(i)
    plt.axis('off')

plt.suptitle('Fashion MNIST Sample Images for Each Label')
plt.tight_layout()
plt.show()


label_names = {
    0: "T-shirt/top",
    1: "Trouser",
    2: "hoodie",
    3: "Dress",
    4: "Coat",
    5: "Sandal",
    6: "Shirt",
    7: "Sneaker",
    8: "Bag",
    9: "Shoe"
}

plt.figure(figsize=(12, 8))

for i in range(10):
    img = train_df[train_df['label'] == i].iloc[0, 1:].values.reshape(28,
28)
    plt.subplot(2, 5, i+1)
    plt.imshow(img, cmap='gray')
    plt.title(label_names[i])
    plt.axis('off')

plt.suptitle('Fashion MNIST Sample Images for Each Label')
plt.tight_layout()
plt.show()


X_train = train_df.drop('label', axis=1).values
y_train = train_df['label'].values
X_test = test_df.drop('label', axis=1).values
y_test = test_df['label'].values

X_train = X_train / 255.0
X_test = X_test / 255.0
X_train = X_train.reshape(-1, 28, 28, 1)
X_test = X_test.reshape(-1, 28, 28, 1)

y_train_cat = to_categorical(y_train, 10)
y_test_cat = to_categorical(y_test, 10)

model = Sequential()
model.add(Conv2D(32, (3,3), activation='relu', input_shape=(28,28,1)))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(64, (3,3), activation='relu'))
model.add(MaxPooling2D((2,2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(10, activation='softmax'))

model.compile(optimizer=Adam(), loss='categorical_crossentropy',
metrics=['accuracy'])

history = model.fit(X_train, y_train_cat, epochs=10,
validation_data=(X_test, y_test_cat), batch_size=128)

loss, accuracy = model.evaluate(X_test, y_test_cat)
print(f"Test Accuracy: {accuracy:.4f}")

y_pred = model.predict(X_test)
y_pred_classes = np.argmax(y_pred, axis=1)
print(classification_report(y_test, y_pred_classes))

plt.figure(figsize=(8,6))
sns.heatmap(confusion_matrix(y_test, y_pred_classes), annot=True,
fmt='d', cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```