```
import tensorflow as tf

from tensorflow.keras import layers, mode from tensorflow.keras.datasets import mni import matplotlib.pyplot as plt

# Load the MNIST dataset

(x_train, y_train), (x_test, y_test) = mn

# Use only 200 samples for both training

x_train = x_train[:200]

y_train = y_train[:200]

x_test = x_test[:200]

y_test = y_test[:200]

# Preprocess the data

x_train = x_train.reshape((200, 28, 28, 1

x_test = x_test.reshape((200, 28, 28, 1))

#One-hot encode the labels
```

```
y_train = tf.keras.utils.to_categorical(y.

y_test = tf.keras.utils.to_categorical(y_

# Build the CNN model

model = models. Sequential([

layers.Conv2D(32, (3, 3), activation= layers.MaxPooling2D((2, 2)), layers.Conv2D(64, (3, 3),
activation= layers. MaxPooling2D((2, 2)), layers.Conv2D(64, (3, 3), activation= layers.
Flatten(), layers. Dense (64, activation='relu'), layers. Dense (10, activation='softmax' 1)

# Compile the model

model.compile(optimizer='adam', loss='categorical_crossentr metrics=['accuracy'])

# Train the model

history = model.fit(x_train, y_train, epo

# Evaluate the model

test_loss, test_acc = model.evaluate(x_te print(f"Test accuracy on 200 test samples

# Plot training and validation accuracy
```

```
plt.plot(history.history['accuracy'], lab

plt.plot(history.history['val_accuracy'],

plt.xlabel('Epoch')

plt.title('Training and Validation Accura

plt.ylabel('Accuracy')

plt.legend()

plt.show()
```