

```

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

# Load and preprocess dataset (CIFAR-10 for example)
(x_train, y_train), (x_test, y_test) = keras.datasets.cifar10.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0 # Normalize images
y_train, y_test = y_train.flatten(), y_test.flatten()

# Define CNN model
def create_cnn_model():
    model = keras.Sequential([
        layers.Conv2D(32, (3,3), activation='relu', input_shape=(32, 32, 3)),
        layers.MaxPooling2D((2,2)),
        layers.Conv2D(64, (3,3), activation='relu'),
        layers.MaxPooling2D((2,2)),
        layers.Conv2D(128, (3,3), activation='relu'),
        layers.Flatten(),
        layers.Dense(128, activation='relu'),
        layers.Dense(10, activation='softmax')
    ])
    return model # Corrected indentation: This line should be aligned with the 'model =' statement

# Compile and train the model
model = create_cnn_model()
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))

# Evaluate the model
loss, accuracy = model.evaluate(x_test, y_test)
print(f"Test Accuracy: {accuracy:.4f}")

# Plot training history
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()

```

➔ Downloading data from <https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz>
170498071/170498071 — 3s 0us/step
/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/10
1563/1563 — 79s 50ms/step - accuracy: 0.3797 - loss: 1.6785 - val_accuracy: 0.5841
Epoch 2/10
1563/1563 — 80s 48ms/step - accuracy: 0.6061 - loss: 1.1192 - val_accuracy: 0.6584
Epoch 3/10
1563/1563 — 83s 49ms/step - accuracy: 0.6734 - loss: 0.9259 - val_accuracy: 0.6556
Epoch 4/10
1563/1563 — 76s 49ms/step - accuracy: 0.7263 - loss: 0.7823 - val_accuracy: 0.7084
Epoch 5/10
1563/1563 — 85s 51ms/step - accuracy: 0.7571 - loss: 0.6913 - val_accuracy: 0.7112
Epoch 6/10
1563/1563 — 80s 49ms/step - accuracy: 0.7841 - loss: 0.6145 - val_accuracy: 0.7361
Epoch 7/10
1563/1563 — 81s 49ms/step - accuracy: 0.8069 - loss: 0.5429 - val_accuracy: 0.7312
Epoch 8/10
1563/1563 — 82s 49ms/step - accuracy: 0.8360 - loss: 0.4634 - val_accuracy: 0.7284
Epoch 9/10
1563/1563 — 82s 49ms/step - accuracy: 0.8590 - loss: 0.3972 - val_accuracy: 0.7184
Epoch 10/10
1563/1563 — 76s 48ms/step - accuracy: 0.8754 - loss: 0.3554 - val_accuracy: 0.7156
313/313 — 5s 16ms/step - accuracy: 0.7143 - loss: 0.9526
Test Accuracy: 0.7130

