ASSIGNMENT - 12

Designing a **neural network** for image classification on the **CIFAR-10 dataset** involves several steps, Which including data preprocessing, model architecture design, training, and evaluation.

This code sets up a simple convolutional neural network (CNN) architecture for image classification on the CIFAR-10 dataset. The model consists of convolutional layers followed by max-pooling layers, a dense layer, and an output layer. It's compiled with the Adam optimizer and sparse categorical cross-entropy loss. The training history is also plotted to visualize the model's performance over epochs.

```
import tensorflow as tf
from tensorflow.keras import layers, models, datasets
import matplotlib.pyplot as plt
 (train_images, train_labels), (test_images, test_labels) = datasets.cifar10.load_data()
train_images, test_images = train_images / 255.0, test_images / 255.0
model = models.Sequential()


model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)))
model.add(layers.MaxPooling2D((2, 2)))


model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Conv2D(64, (3, 3), activation='relu'))

model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10))  # Output layer with 10 units for 10 classes

model.compile(optimizer='adam',
          loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
          metrics=['accuracy'])
model.summary()
history = model.fit(train_images, train_labels, epochs=10,
              validation_data=(test_images, test_labels))


test_loss, test_acc = model.evaluate(test_images, test_labels, verbose=2)
print('\nTest accuracy:', test_acc)
```

```
plt.plot(history.history['accuracy'], label='Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.ylim([0, 1])
plt.legend(loc='lower right')
plt.show()
```