import tensorflow as tf

from tensorflow.keras import datasets, layers, models

# Load CIFAR-10 dataset

(train\_images, train\_labels), (test\_images, test\_labels) = datasets.cifar10.load\_data()

# Preprocess data

train\_images, test\_images = train\_images / 255.0, test\_images / 255.0

# Create a CNN model

model = models.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(64, (3, 3), activation='relu'),

    layers.Flatten(),

    layers.Dense(64, activation='relu'),

    layers.Dense(10)

])

# Compile the model

model.compile(optimizer='adam',

              loss=tf.keras.losses.SparseCategoricalCrossentropy(from\_logits=True),

              metrics=['accuracy'])

# Train the model

model.fit(train\_images, train\_labels, epochs=3, validation\_data=(test\_images, test\_labels))

# Evaluate the model

test\_loss, test\_acc = model.evaluate(test\_images, test\_labels, verbose=2)

print(f"Test accuracy: {test\_acc}")

model.summary()

TASKS:  
Load a sample image, convert to Gray-scale and perform following tasks:

1. Define a 3x3 convolution kernel, and apply convolution operation on the image.
2. Apply min and max pooling on the image.
3. Apply the padding on the image.
4. Apply convolution with a stride of 4.
5. Display all the results including the original image.