#Assi3

#Build the Image classification model by dividing the model into following 4 stages:

    #a. Loading and preprocessing the image data

    #b. Defining the model’s architecture

    #c. Training the model

    #d. Estimating the model’s performance

import tensorflow as tf

from tensorflow.keras.datasets import cifar10

from tensorflow.keras.utils import to\_categorical

# Load and preprocess the CIFAR-10 dataset

(x\_train, y\_train), (x\_test, y\_test) = cifar10.load\_data()

# Normalize pixel values to be between 0 and 1

x\_train, x\_test = x\_train / 255.0, x\_test / 255.0

# Convert labels to one-hot encoding

y\_train = to\_categorical(y\_train, num\_classes=10)

y\_test = to\_categorical(y\_test, num\_classes=10)

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

# Define the model architecture

model = Sequential([

    Conv2D(32, (3, 3), activation='relu', input\_shape=(32, 32, 3)),

    MaxPooling2D((2, 2)),

    Conv2D(64, (3, 3), activation='relu'),

    MaxPooling2D((2, 2)),

    Conv2D(64, (3, 3), activation='relu'),

    Flatten(),

    Dense(64, activation='relu'),

    Dense(10, activation='softmax')

])

model.summary()

# Compile the model

model.compile(optimizer='adam',

              loss='categorical\_crossentropy',

              metrics=['accuracy'])

# Train the model

model.fit(x\_train, y\_train, epochs=10, batch\_size=64, validation\_split=0.2)

# Evaluate the model on the test data

test\_loss, test\_accuracy = model.evaluate(x\_test, y\_test)

print(f'Test loss: {test\_loss}')

print(f'Test accuracy: {test\_accuracy}')