**In-Lab**

**In-Lab Task 1**

|  |
| --- |
| **Code:**  !pip install kaggle  !mkdir -p ~/.kaggle  !cp kaggle.json ~/.kaggle/  !chmod 600 ~/.kaggle/kaggle.json  **Output:** |

**In-Lab Task 2**

|  |
| --- |
| **Code:**  from google.colab import files  # Upload a file  uploaded = files.upload()  # Print the uploaded file  print(uploaded)  **Output:** |

**In-Lab Task 3**

|  |
| --- |
| **Code:**  #Loading Dataset  !kaggle datasets download -d moltean/fruits  **Output:** |

**In-Lab Task 4**

|  |
| --- |
| **Code:**  # Unzipping the Fruits-360 Folder  !unzip -q fruits.zip  **Output:** |

**In-Lab Task 5**

|  |
| --- |
| **Code:**  import tensorflow as tf  from tensorflow.keras import layers, models  # Set the path to the dataset  dataset\_path = "fruits-360\_dataset/fruits-360/Training"  # Define the CNN model  model = models.Sequential()  # Convolutional and Pooling Layers  model.add(layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(50, 50, 3)))  model.add(layers.MaxPooling2D((2, 2)))  model.add(layers.Conv2D(64, (3, 3), activation='relu'))  model.add(layers.MaxPooling2D((2, 2)))  # Flatten Layer  model.add(layers.Flatten())  # Dense Layers  model.add(layers.Dense(128, activation='relu'))  model.add(layers.Dense(131, activation='softmax'))  # Compile the model  model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])  # Display the model summary  model.summary()  # Image data augmentation  train\_datagen = tf.keras.preprocessing.image.ImageDataGenerator(      rescale=1./255,      shear\_range=0.2,      zoom\_range=0.2,      horizontal\_flip=True,      validation\_split=0.2  )  # Training and validation generators  batch\_size = 64  # Increased batch size for faster processing  train\_generator = train\_datagen.flow\_from\_directory(      dataset\_path,      target\_size=(50, 50),  # Adjusted to the input size      batch\_size=batch\_size,      class\_mode='categorical',      subset='training'  )  validation\_generator = train\_datagen.flow\_from\_directory(      dataset\_path,      target\_size=(50, 50),      batch\_size=batch\_size,      class\_mode='categorical',      subset='validation'  )  # Train the model  model.fit(      train\_generator,      epochs=5,  # Reduced the number of epochs for demonstration      validation\_data=validation\_generator  )  **Output:** |

**In-Lab Task 6**

|  |
| --- |
| **Code:**  # Assuming you have a separate test dataset  test\_datagen = tf.keras.preprocessing.image.ImageDataGenerator(rescale=1./255)  test\_generator = test\_datagen.flow\_from\_directory(      "fruits-360\_dataset/fruits-360/Test",  # Replace with the path to your test dataset      target\_size=(50, 50),      batch\_size=batch\_size,      class\_mode='categorical',      shuffle=False  # Set shuffle to False to maintain the order for evaluation  )  # Evaluate the model on the test set  test\_loss, test\_accuracy = model.evaluate(test\_generator)  print(f'Test Loss: {test\_loss:.4f}')  print(f'Test Accuracy: {test\_accuracy \* 100:.2f}%')  **Output:** |