**LAB 10**

**Task 1:**

|  |
| --- |
| from google.colab import files  files.upload() |

**Output:**

|  |
| --- |
| **A screenshot of a computer  Description automatically generated** |

**Task 2:**

|  |
| --- |
| !pip install -q kaggle  !mkdir ~/.kaggle  !cp kaggle.json ~/.kaggle/  !chmod 600 ~/.kaggle/kaggle.json  !kaggle datasets download -d moltean/fruits |

**Output:**

|  |
| --- |
|  |

**Task 3:**

|  |
| --- |
| !unzip fruits.zip |

**Output:**

|  |
| --- |
|  |

**Task 4:**

|  |
| --- |
| import tensorflow as tf  from tensorflow.keras.preprocessing.image import ImageDataGenerator |

**Task 5:**

|  |
| --- |
| train\_datagen = ImageDataGenerator(rescale=1./255, validation\_split=0.2) # Splitting data  train\_generator = train\_datagen.flow\_from\_directory(      directory='./fruits-360\_dataset',      target\_size=(224, 224),      batch\_size=32,      class\_mode='categorical',      subset='training')  validation\_generator = train\_datagen.flow\_from\_directory(      directory='./fruits-360-original-size',      target\_size=(224, 224),      batch\_size=32,      class\_mode='categorical',      subset='validation') |

**Task 6:**

|  |
| --- |
| model = tf.keras.models.Sequential([      tf.keras.layers.Conv2D(32, (3,3), activation='relu', input\_shape=(224, 224, 3)),      tf.keras.layers.MaxPooling2D(2, 2),      # Add more layers as needed      tf.keras.layers.Flatten(),      tf.keras.layers.Dense(512, activation='relu'),      tf.keras.layers.Dense(101, activation='softmax') # 101 for 101 food categories  ])  model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy']) |

**Task 7:**

|  |
| --- |
| history = model.fit(      train\_generator,      steps\_per\_epoch=train\_generator.n//train\_generator.batch\_size,      validation\_data=validation\_generator,      validation\_steps=validation\_generator.n//validation\_generator.batch\_size,      epochs=1  ) |