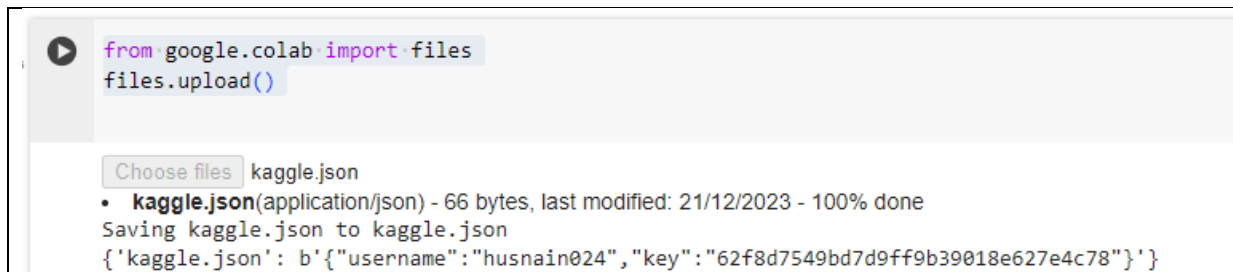


LAB 10

Task 1:

```
from google.colab import files
files.upload()
```

Output:

A screenshot of the Google Colab interface. At the top, there is a code editor with the following code:

```
from google.colab import files
files.upload()
```

 Below the code editor, there is a file upload section. It shows a button labeled "Choose files" followed by the filename "kaggle.json". Below this, there is a list of files being uploaded:

- **kaggle.json**(application/json) - 66 bytes, last modified: 21/12/2023 - 100% done

 Below the list, it says "Saving kaggle.json to kaggle.json" and then shows the JSON content:

```
{'kaggle.json': b'{"username": "husnain024", "key": "62f8d7549bd7d9ff9b39018e627e4c78"}'}
```

Task 2:

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

Output:

```
mkdir: cannot create directory '/root/.kaggle': File exists
Downloading fruits.zip to /content
 99% 1.27G/1.28G [00:09<00:00, 136MB/s]
100% 1.28G/1.28G [00:09<00:00, 146MB/s]
```

Task 3:

```
!unzip fruits.zip
```

Output:



```
inflating: fruits-360_dataset/fruits-360/Training/Walnut/r_90_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Walnut/r_91_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Walnut/r_92_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Walnut/r_93_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Walnut/r_94_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Walnut/r_95_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Walnut/r_96_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Walnut/r_97_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Walnut/r_98_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Walnut/r_99_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Watermelon/0_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Watermelon/100_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Watermelon/101_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Watermelon/102_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Watermelon/103_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Watermelon/104_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Watermelon/105_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Watermelon/106_100.jpg
```

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
)
```

Output:

```
704/2262 [=====>.....] - ETA: 2:31:47 - loss: 94307568.0000 - accuracy: 0.0085
```

The accuracy increases as the code running.....