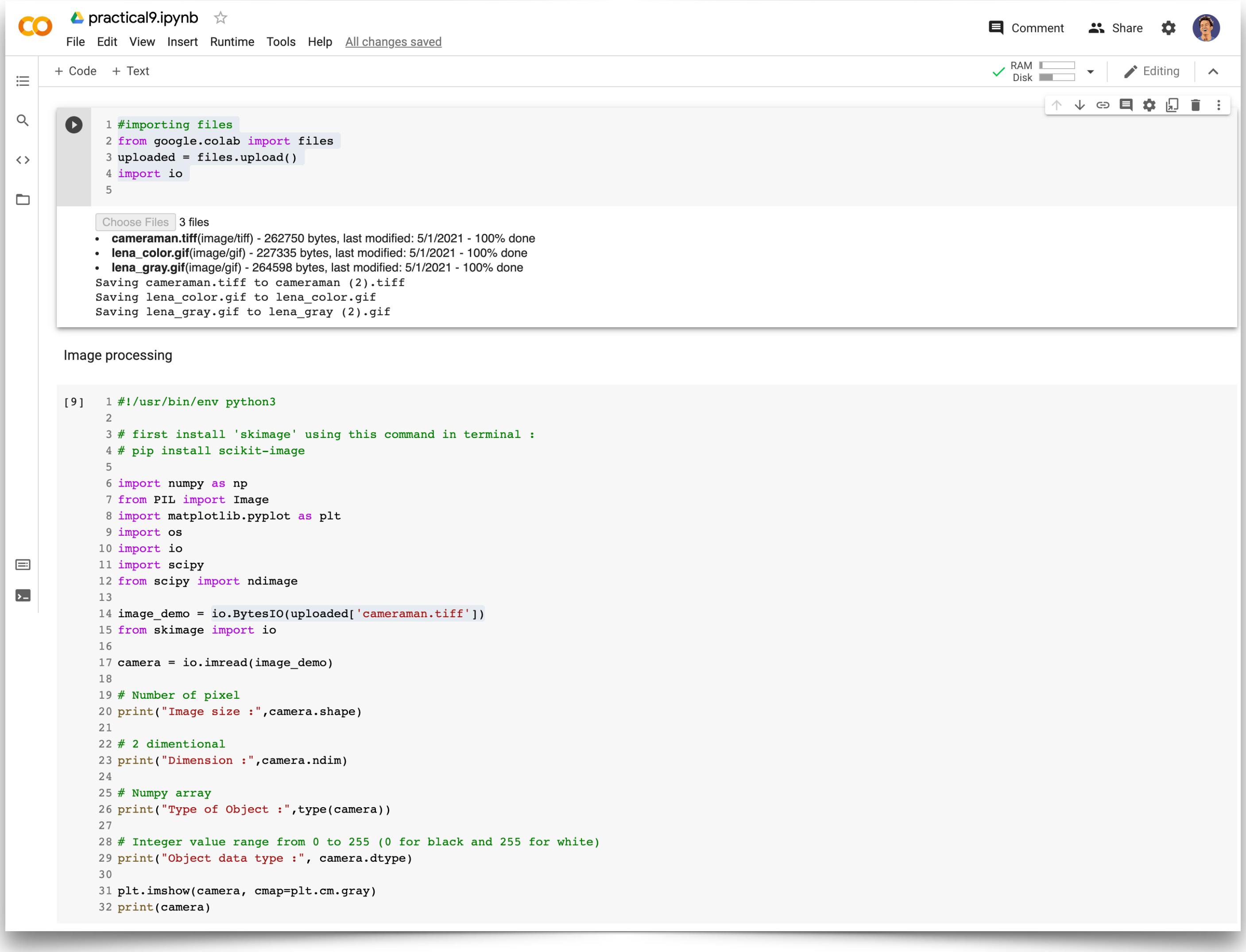
Aayush Shah
D1 - 19BCE245
30 April 2021

Practical 9

Image Processing

Develop a python program that reads the image, display matrix representation of an image creates a histogram of the image and apply the smoothing effect on an image.

8a Code and Output :

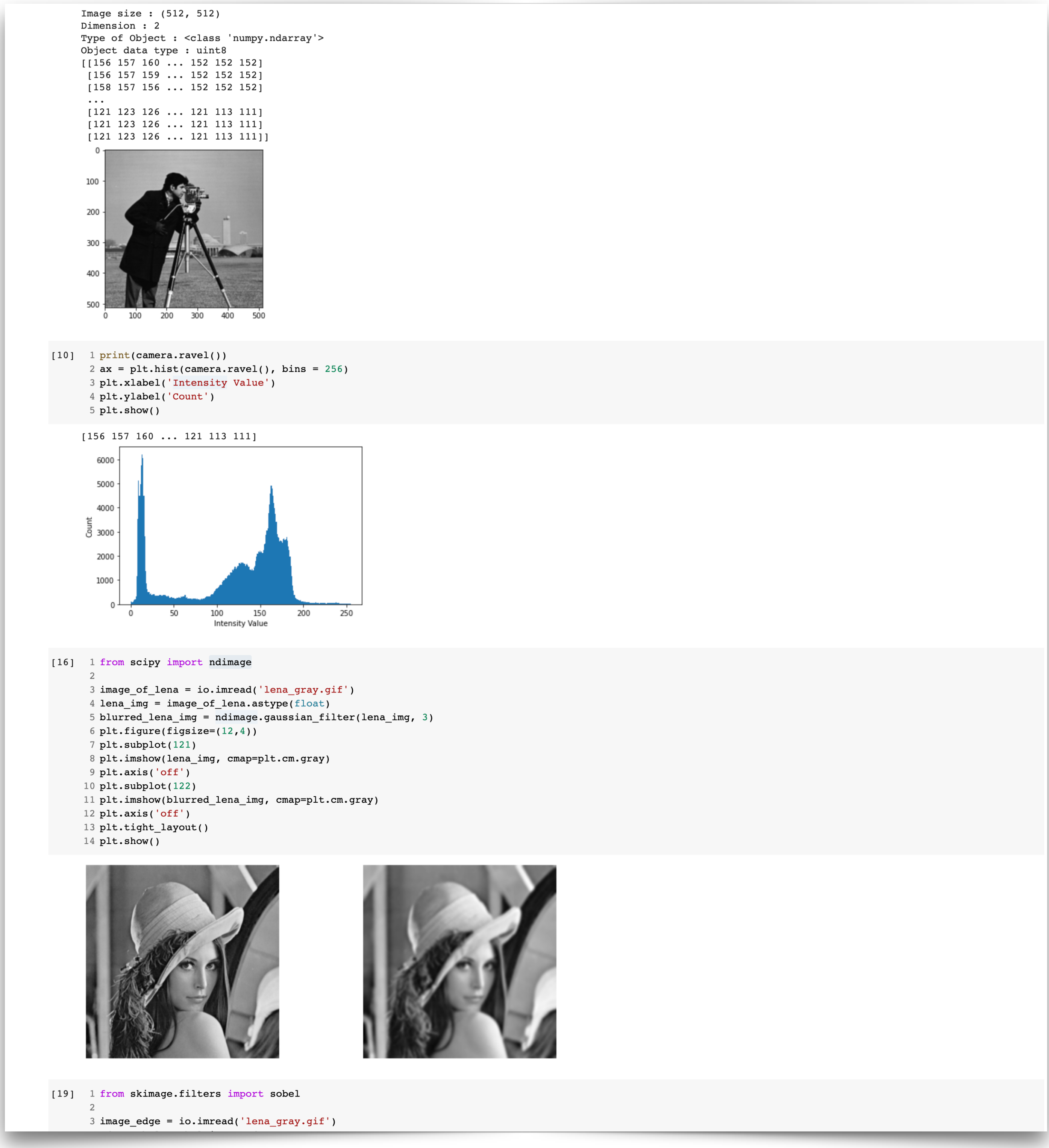


```
#importing files
from google.colab import files
uploaded = files.upload()
import io

```

```
Choose Files 3 files
• cameraman.tiff(image/tiff) - 262750 bytes, last modified: 5/1/2021 - 100% done
• lena_color.gif(image/gif) - 227335 bytes, last modified: 5/1/2021 - 100% done
• lena_gray.gif(image/gif) - 264598 bytes, last modified: 5/1/2021 - 100% done
Saving cameraman.tiff to cameraman (2).tiff
Saving lena_color.gif to lena_color.gif
Saving lena_gray.gif to lena_gray (2).gif
```

Image processing

```
[9]
#!/usr/bin/env python3

# first install 'skimage' using this command in terminal :
# pip install scikit-image

import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
import os
import io
import scipy
from scipy import ndimage

image_demo = io.BytesIO(uploaded['cameraman.tiff'])
from skimage import io

camera = io.imread(image_demo)

# Number of pixel
print("Image size :",camera.shape)

# 2 dimentional
print("Dimension :",camera.ndim)

# Numpy array
print("Type of Object :",type(camera))

# Integer value range from 0 to 255 (0 for black and 255 for white)
print("Object data type :", camera.dtype)

plt.imshow(camera, cmap=plt.cm.gray)
print(camera)
```



```
Image size : (512, 512)
Dimension : 2
Type of Object : <class 'numpy.ndarray'>
Object data type : uint8
[[156 157 160 ... 152 152 152]
 [156 157 159 ... 152 152 152]
 [158 157 156 ... 152 152 152]
 ...
 [121 123 126 ... 121 113 111]
 [121 123 126 ... 121 113 111]
 [121 123 126 ... 121 113 111]]
```

[10]

```python
print(camera.ravel())
ax = plt.hist(camera.ravel(), bins = 256)
plt.xlabel('Intensity Value')
plt.ylabel('Count')
plt.show()
```

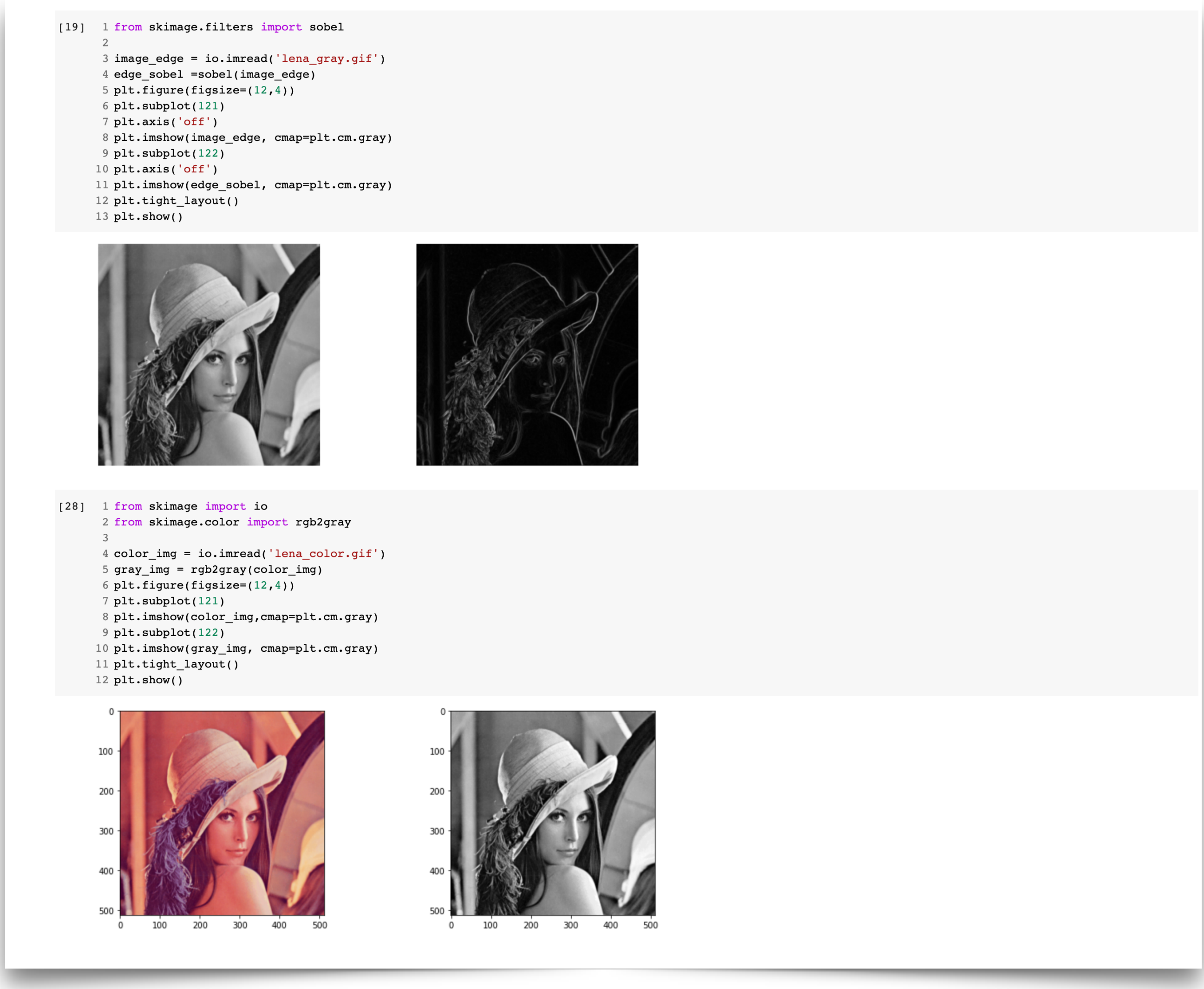
```
[156 157 160 ... 121 113 111]
```

[16]

```python
from scipy import ndimage

image_of_lena = io.imread('lena_gray.gif')
lena_img = image_of_lena.astype(float)
blurred_lena_img = ndimage.gaussian_filter(lena_img, 3)
plt.figure(figsize=(12,4))
plt.subplot(121)
plt.imshow(lena_img, cmap=plt.cm.gray)
plt.axis('off')
plt.subplot(122)
plt.imshow(blurred_lena_img, cmap=plt.cm.gray)
plt.axis('off')
plt.tight_layout()
plt.show()
```

[19]

```python
from skimage.filters import sobel

image_edge = io.imread('lena_gray.gif')
```



[19]

```
from skimage.filters import sobel

image_edge = io.imread('lena_gray.gif')
edge_sobel =sobel(image_edge)
plt.figure(figsize=(12,4))
plt.subplot(121)
plt.axis('off')
plt.imshow(image_edge, cmap=plt.cm.gray)
plt.subplot(122)
plt.axis('off')
plt.imshow(edge_sobel, cmap=plt.cm.gray)
plt.tight_layout()
plt.show()
```

[28]

```
from skimage import io
from skimage.color import rgb2gray

color_img = io.imread('lena_color.gif')
gray_img = rgb2gray(color_img)
plt.figure(figsize=(12,4))
plt.subplot(121)
plt.imshow(color_img,cmap=plt.cm.gray)
plt.subplot(122)
plt.imshow(gray_img, cmap=plt.cm.gray)
plt.tight_layout()
plt.show()
```

Conclusion :

Here we have learned about Image processing with libraries like PIL, Matplotlib, idimage, rgb2gray and sobel.

- **PIL** : PIL is the Python Imaging Library which provides the python interpreter with image editing capabilities. The image module provides a class with the same name which is used to represent a PIL image. The module also provides a number of factory functions, including functions to load images from files, and to create new images.