Lab6

April 24, 2022

0.1 Lab6

0.1.1 Submitted By: Manav Doda

0.1.2 Roll No.: 195057

0.2 Importing Necessary modules

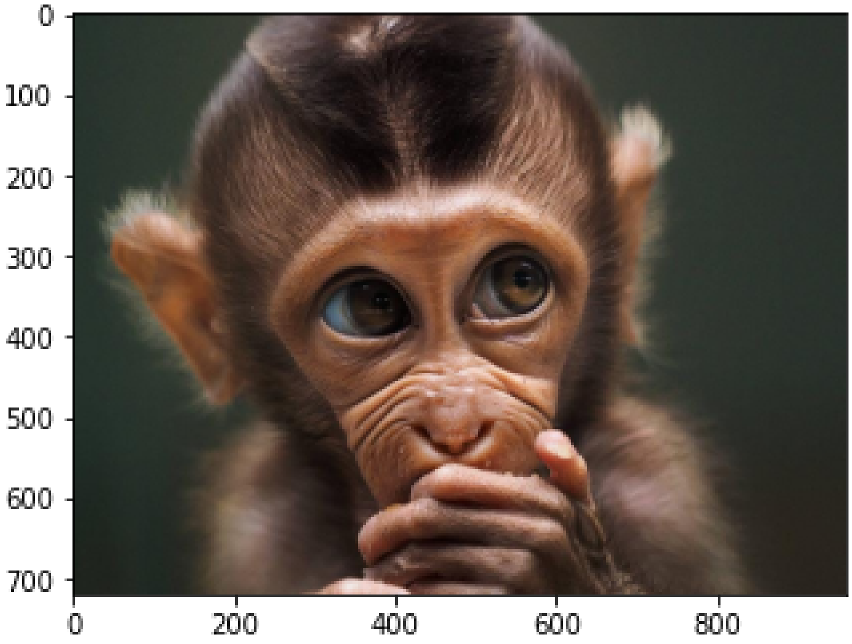
[1]:

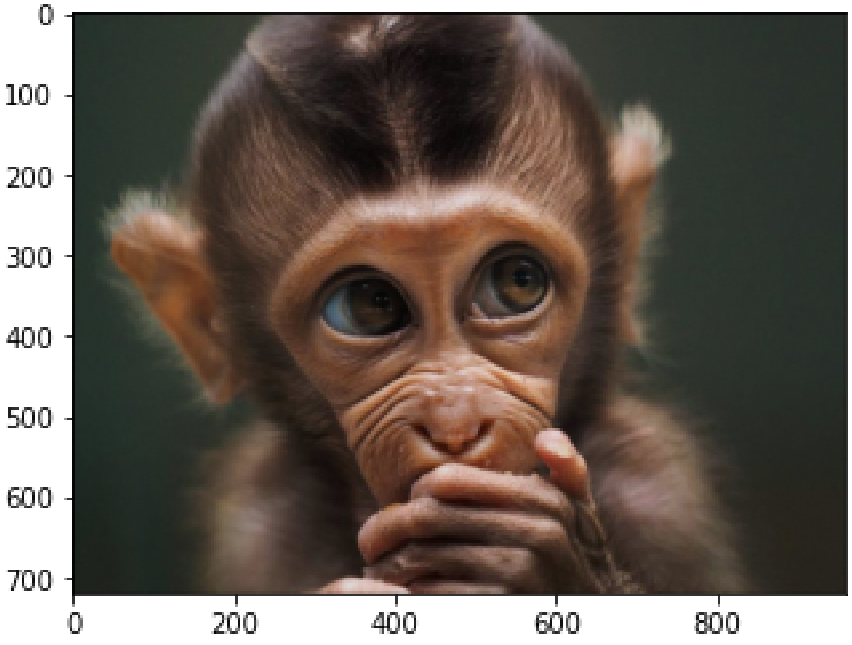
```
from PIL import Image
import matplotlib.pyplot as plt
import numpy as np
import cv2
from scipy import stats
```

0.2.1 Objective 1: To understand and implement the guassian filter on a given Image

[2]:

```
img = cv2.imread('testImage.jpeg')
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.show()
shape = img.shape
rows = shape[0]
cols = shape[1]
for i in range(rows):
    for j in range(cols):
        sumA = int(img[i][j][0])*4
        sumB = int(img[i][j][1])*4
        sumC = int(img[i][j][2])*4
        if i>0 and j>0:
            sumA+=img[i-1][j-1][0]
            sumB+=img[i-1][j-1][1]
            sumC+=img[i-1][j-1][2]
        if i>0:
            sumA+=int(img[i-1][j][0])*2
            sumB+=int(img[i-1][j][1])*2
            sumC+=int(img[i-1][j][2])*2
        if i>0 and j<cols-1:
            sumA+=img[i-1][1][0]
            sumB+=img[i-1][1][1]
            sumC+=img[i-1][1][2]
        if j>0:
            sumA+=img[i][j-1][0]
            sumB+=img[i][j-1][1]
            sumC+=img[i][j-1][2]
        if j<cols-1:
            sumA+=int(img[i][j+1][0])*2
            sumB+=int(img[i][j+1][1])*2
            sumC+=int(img[i][j+1][2])*2
        if i<rows-1 and j>0:
            sumA+=int(img[i+1][j-1][0])*2
            sumB+=int(img[i+1][j-1][1])*2
            sumC+=int(img[i+1][j-1][2])*2
        if i<rows-1:
            sumA+=img[i+1][j][0]
            sumB+=img[i+1][j][1]
            sumC+=img[i+1][j][2]
        if i<rows-1 and j<cols-1:
            sumA+=int(img[i+1][j+1][0])*2
            sumB+=int(img[i+1][j+1][1])*2
            sumC+=int(img[i+1][j+1][2])*2
        img[i][j] = [int(sumA/16), int(sumB/16), int(sumC/16)]
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.show()
```





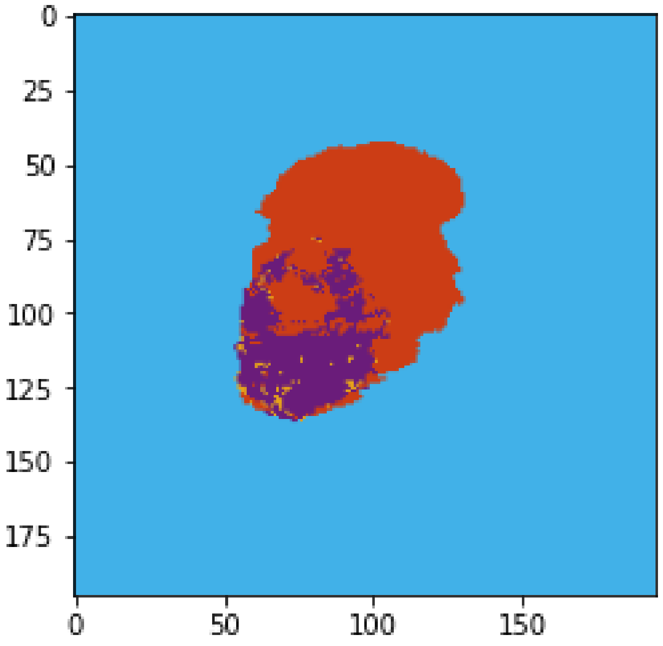
0.3 Objective 2

0.3.1 To understand explain and implement the 2-D and 3-D mode filter on the given image

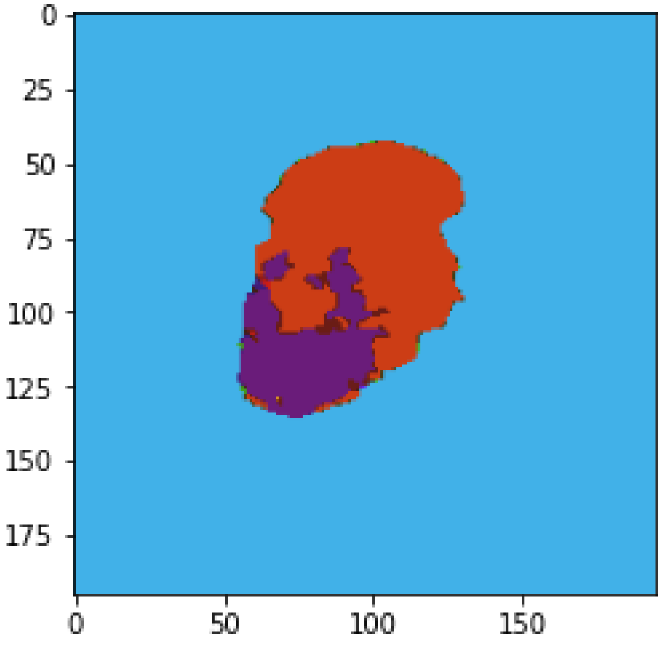
[3]:

```
img = cv2.imread('modeFiltering.png')
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.show()
shape = img.shape
rows = shape[0]
cols = shape[1]
for i in range(rows):
    for j in range(cols):
        A = [img[i][j][0]]
        B = [img[i][j][1]]
        C = [img[i][j][2]]
        if i>0 and j>0:
            A.append(img[i-1][j-1][0])
            B.append(img[i-1][j-1][1])
            C.append(img[i-1][j-1][2])
        if i>0:
            A.append(img[i-1][j][0])
            B.append(img[i-1][j][1])
            C.append(img[i-1][j][2])
        if i>0 and j<cols-1:
            A.append(img[i-1][1][0])
            B.append(img[i-1][1][1])
            C.append(img[i-1][1][2])
        if j>0:
            A.append(img[i][j-1][0])
            B.append(img[i][j-1][1])
            C.append(img[i][j-1][2])
        if j<cols-1:
            A.append(img[i][j+1][0])
            B.append(img[i][j+1][1])
            C.append(img[i][j+1][2])
        if i<rows-1 and j>0:
            A.append(img[i+1][j-1][0])
            B.append(img[i+1][j-1][1])
            C.append(img[i+1][j-1][2])
        if i<rows-1:
            A.append(img[i+1][j][0])
            B.append(img[i+1][j][1])
            C.append(img[i+1][j][2])
        if i<rows-1 and j<cols-1:
            A.append(img[i+1][j+1][0])
            B.append(img[i+1][j+1][1])
            C.append(img[i+1][j+1][2])
        a = stats.mode(A)[0]
        b = stats.mode(B)[0]
        c = stats.mode(C)[0]

        img[i][j] = [a, b, c]
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.show()
```



```
/var/folders/6r/c_0pyh_s5sl09dgwcds_qyk40000gn/T/ipykernel_30317/672425998.py:48
: DeprecationWarning: setting an array element with a sequence. This was
supported in some cases where the elements are arrays with a single element. For
example `np.array([1, np.array([2])], dtype=int)`. In the future this will raise
the same ValueError as `np.array([1, [2]], dtype=int)`.
  img[i][j] = [a, b, c]
```



[ ]:

```
```