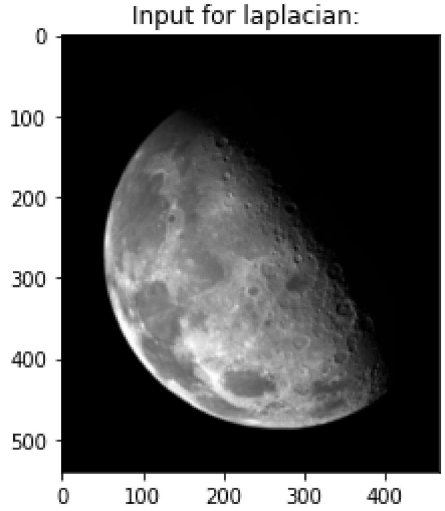
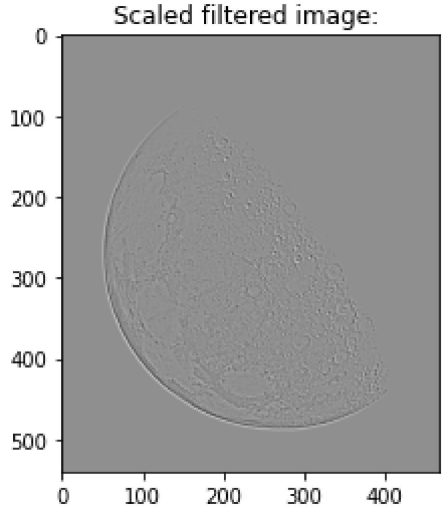
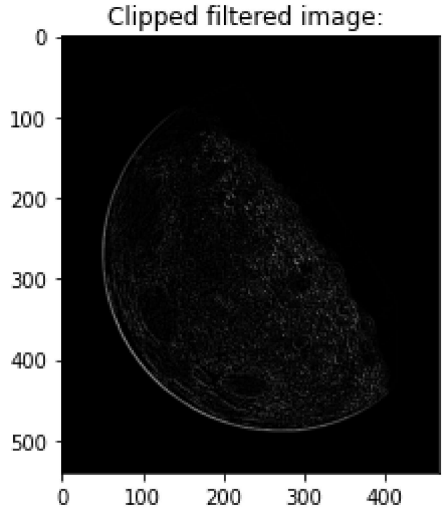
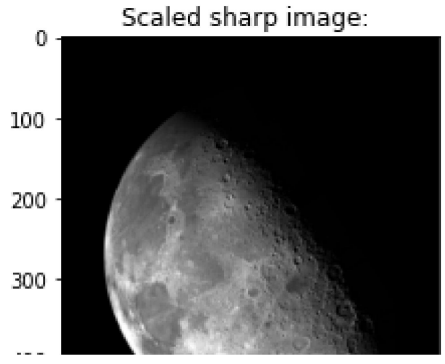
```
# -*- coding: utf-8 -*-
"""
Created on Mon Apr 25 14:59:18 2022

@author: Asus
"""

import cv2 as cv
import matplotlib.pyplot as plt
import numpy as np
import math

def scaling(img):
    g_m = img - img.min()
    g_s = 255*(g_m/g_m.max())
    return g_s.astype(np.float32)

def clipping(img):
    m = img.shape[0]
    n = img.shape[1]

    for i in range(m):
        for j in range(n):
            if(img[i][j] > 255):
                img[i][j] = 255
            if(img[i][j] < 0):
                img[i][j] = 0
    return img.astype(np.float32)

path·=·"/content/drive/MyDrive/vision/New·folder/laplace.jpg"

img  = cv.imread(path)

img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)

plt.imshow(img,"gray")

plt.title("Input for laplacian: ")

plt.show()

kernel = np.array(([0,1,0],[1,-4,1],[0,1,0]), np.float32)

a = kernel.shape[0] // 2
b = kernel.shape[1] // 2
m = img.shape[0]
n = img.shape[1]

op = np.zeros((m,n), np.float32)
```

```
    for i in range(m):
        for j in range(n):
            for x in range(-a,a+1):
                for y in range(-b,b+1):
                    if i-x>=0 and i-x<m and j-y>=0 and j-y<n:
                        op[i][j] += kernel[a+x][b+y]*img[i-x][j-y]
                    else:
                        op[i][j] += 0

    scaled_op = scaling(op)
    clipped_op = clipping(op)

    plt.imshow(scaled_op, "gray")
    plt.title("Scaled filtered image: ")
    plt.show()

    plt.imshow(clipped_op, "gray")
    plt.title("Clipped filtered image: ")
    plt.show()

    sharp = img-op #as the center of the kernel is negative

    sharp_clipped = clipping(sharp)
    sharp_scaled = scaling(sharp)

    plt.imshow(sharp_scaled, "gray")
    plt.title("Scaled sharp image: ")
    plt.show()

    plt.imshow(sharp_clipped, "gray")
    plt.title("Clipped sharp image: ")
    plt.show()
```


Input for laplacian:
0
100
200
300
400
500
0
100
200
300
400

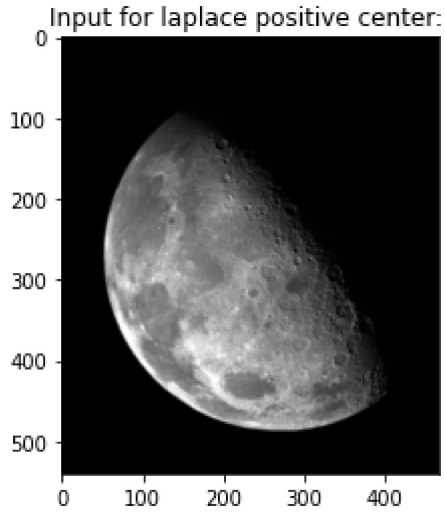

Scaled filtered image:
0
100
200
300
400
500
0
100
200
300
400

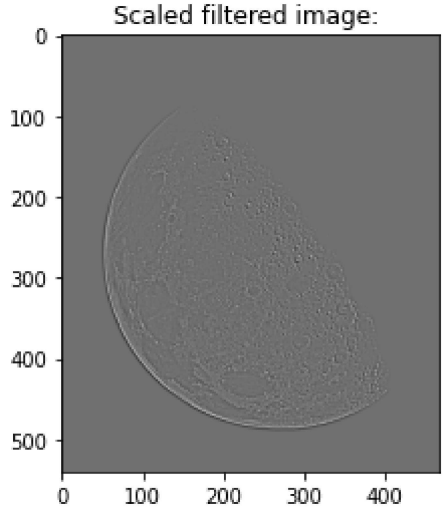

Clipped filtered image:
0
100
200
300
400
500
0
100
200
300
400

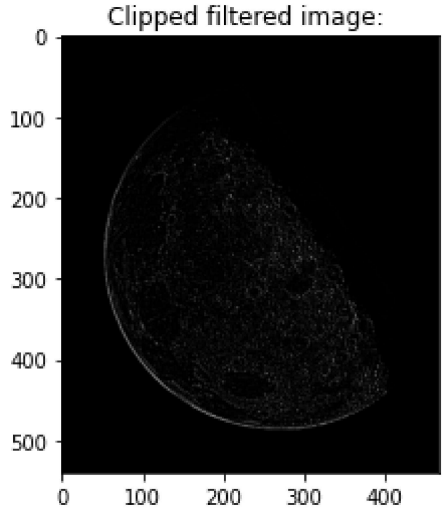

Scaled sharp image:
0
100
200
300

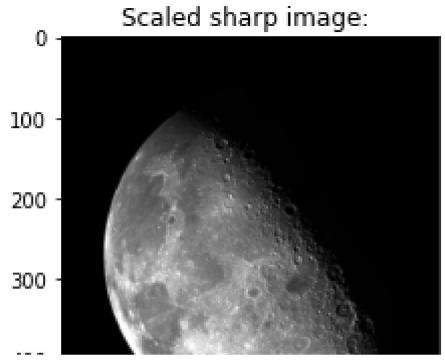
```
# -*- coding: utf-8 -*-
"""
Created on Mon Apr 25 16:08:16 2022

@author: Asus
"""

import cv2 as cv
import matplotlib.pyplot as plt
import numpy as np
import math

def scale(img):
    g_m = img-img.min()
    g_s = 255*(g_m/g_m.max())
    return g_s.astype(np.float32)

def clip(img):
    m = img.shape[0]
    n = img.shape[1]
    for i in range(m):
        for j in range(n):
            if(img[i][j] > 255):
                img[i][j] = 255
            if(img[i][j] < 0):
                img[i][j] = 0
    return img.astype(np.float32)

path = "/content/drive/MyDrive/vision/New folder/laplace.jpg"

img = cv.imread(path)

img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)

plt.imshow(img, "gray")

plt.title("Input for laplace positive center: ")

plt.show()

kernel = np.array(([0,-1,0],[-1,4,-1],[0,-1,0]), np.float32)

a = kernel.shape[0] // 2
b = kernel.shape[1] // 2
m = img.shape[0]
n = img.shape[1]

op = np.zeros((m,n), np.float32)

for i in range(m):
    for j in range(n):
```

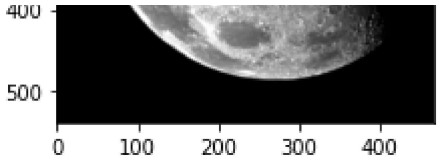
```
            for x in range(-a,a+1):
                for y in range(-b,b+1):
                    if i-x >= 0 and i-x<m and j-y>=0 and j-y<n:
                        op[i][j]+=kernel[a+x][b+y]*img[i-x][j-y]
                    else:
                        op[i][j]+=0
    scaled_op = scale(op)
    clipped_op = clip(op)

    plt.imshow(scaled_op, "gray")
    plt.title("Scaled filtered image: ")
    plt.show()

    plt.imshow(clipped_op, "gray")
    plt.title("Clipped filtered image: ")
    plt.show()

    sharp = img+op #as the center of the kernel is positive

    sharp_clipped = clip(sharp)
    sharp_scaled = scale(sharp)

    plt.imshow(sharp_scaled, "gray")
    plt.title("Scaled sharp image: ")
    plt.show()

    plt.imshow(sharp_clipped, "gray")
    plt.title("Clipped sharp image: ")
    plt.show()
```

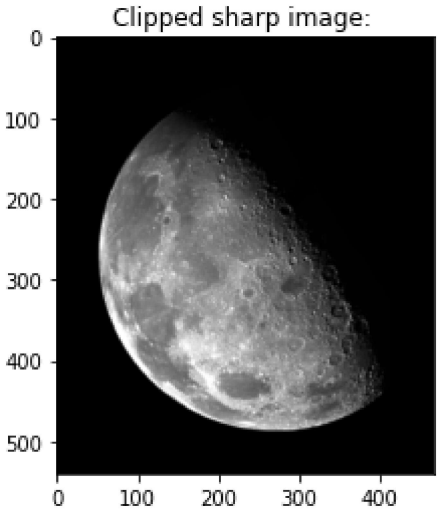

Input for laplace positive center:
0
100
200
300
400
500
0
100
200
300
400


Scaled filtered image:
0
100
200
300
400
500
0
100
200
300
400


Clipped filtered image:
0
100
200
300
400
500
0
100
200
300
400


Scaled sharp image:
0
100
200
300


400
500
0
100
200
300
400


Clipped sharp image:
0
100
200
300
400
500
0
100
200
300
400