

```
# -*- coding: utf-8 -*-
"""
Created on Wed May  4 12:52:51 2022

@author: Asus
"""

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

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)

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

path = "/content/drive/MyDrive/vision/New folder/ancient image.png"

img = cv.imread(path)

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

plt.imshow(img, 'gray')

plt.title("Inpur for laplace raw")
```

```
plt.show()

print("Enter kernel height: ")

k_h = int(input())

print("Enter kernel weidth: ")

k_w = int(input())

sigma = 1.0

s = sigma*sigma

pi = 3.1416

kernel = np.zeros((k_h,k_w), 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 x in range(-a,a+1):
    for y in range(-b,b+1):
        sqsum = (x*x+y*y)
        term = (sqsum)/(2*s)
        r = math.exp(-(term))
        term = 1-term
        r = r*term*(-1/(pi*s*s))
        kernel[a+x][b+y] = r

for i in range(m):
    for j in range(n):
        for x in range(-a,a+1):
            for y in range(-b,b+1):
```

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

plt.imshow(op,'gray')

plt.title("Laplace filtered: ")

plt.show()

op_c = clip(op)

plt.imshow(op_c,'gray')

plt.title("Clipped: ")

plt.show()

op_s = scale(op)

plt.imshow(op_s,'gray')

plt.title("Scaled: ")

plt.show()

img = img-op_c #center of kernel is negative

img = clip(img)

plt.imshow(img,'gray')

plt.title("After enhancing: ")

plt.show()
```



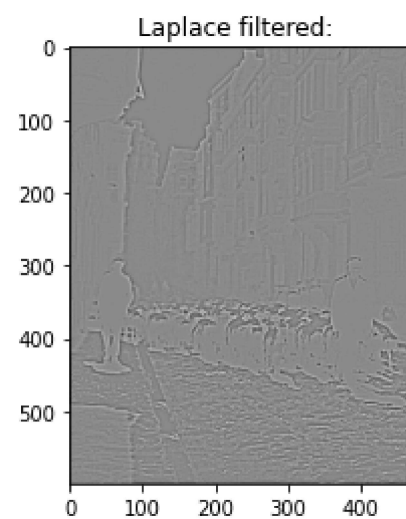


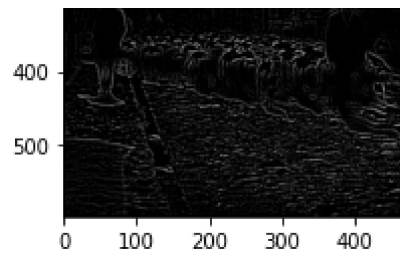
Enter kernel height:

9

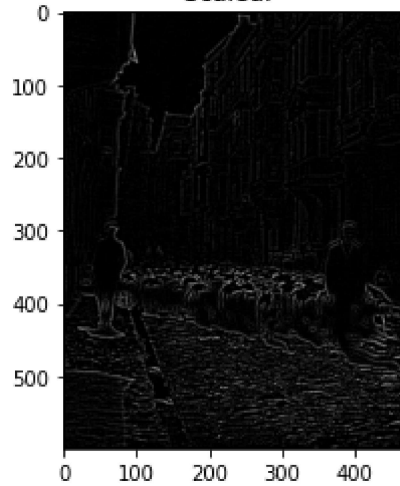
Enter kernel weidth:

9





Scaled:



After enhancing:

