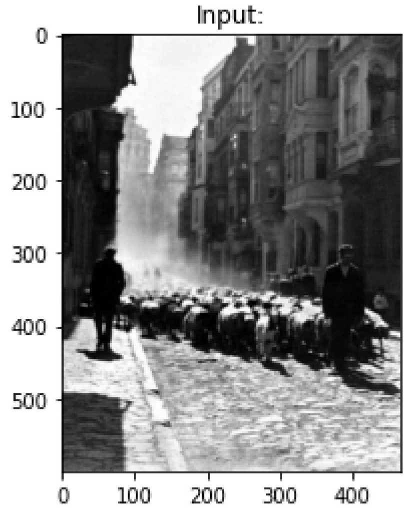
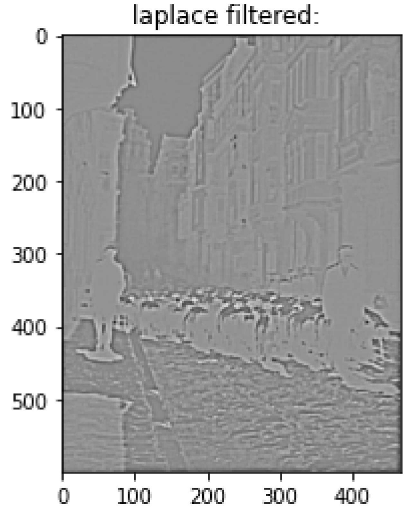
```
# -*- coding: utf-8 -*-
"""
Created on Wed May  4 16:32:09 2022

@author: Asus
"""

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

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("Input:")

plt.show()

print("Enter kernel height: ")

k_h = int(input())

print("Enter kernel weidth: ")

k_w = int(input())

kernel1 = np.zeros((k_h,k_w), np.float32)

kernel2 = np.zeros((k_h,k_w),np.float32)

a = kernel1.shape[0] // 2
b = kernel2.shape[1] // 2

pi = 3.1416

sigma1 = 1.0

sigma2 = 2.5

for x in range(-a,a+1):
    for y in range(-b,b+1):
        s = 2*sigma1*sigma1
        term = (x*x+y*y)
        term/=s
```

```
            r = math.exp(-(term))
            r = r/(s*pi)
            kernel1[a+x][b+y] = r

    for x in range(-a,a+1):
        for y in range(-b,b+1):
            s = 2*sigma2*sigma2
            term = (x*x+y*y)
            term/=s
            r = math.exp(-(term))
            r = r/(s*pi)
            kernel2[a+x][b+y] = r

    kernel2 = kernel2-kernel1

    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]+=kernel2[a+x][b+y]*img[i-x][j-y]
                    else:
                        op[i][j]+=0
    plt.imshow(op,'gray')
    plt.title("laplace filtered: ")
    plt.show()
```



```
Enter kernel height:
13
Enter kernel weidth:
13
```


laplace filtered: