

```
# -*- coding: utf-8 -*-
```

```
"""
```

Created on Tue May 10 18:05:25 2022

```
@author: Asus
```

```
"""
```

```
import cv2 as cv
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
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)
```

```
path = "C:/Users/Asus/imagelab/Image-Processing-and-Computer-Vision-Lab/Lab 2/Prewitt/input.jpg"
```

```
img = cv.imread(path)
```

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

```
plt.imshow(img,'gray')
```

```
plt.title("Input for Robert: ")
```

```
plt.show()
```

```
kernel_v = np.array([[1,0],[0,-1]],np.float32)
```

```
kernel_h = np.array([[0,1],[-1,0]],np.float32)
```

```
k_s = kernel_h.shape[0]
```

```
a = kernel_h.shape[0] // 2
```

```
b = kernel_h.shape[1] // 2
```

```
m = img.shape[0]
```

```
n = img.shape[1]
```

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

```
op_h = 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 and x+a>=0 and x+a<k_s and y+b>=0 and y+b<k_s:
```

```
                    op_v[i][j]+=kernel_v[x+a][y+b]*img[i-x][j-y]
```

```
else:
```

```
    op_v[i][j]+=0
```

```
plt.imshow(op_v,'gray')
```

```
plt.title("Vertical raw output:")
```

```
plt.show()
```

```
op_v = clip(op_v)
```

```
plt.imshow(op_v,'gray')
```

```
plt.title("Vertical clipped output:")
```

```
plt.show()
```

```
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 and x+a>=0 and x+a<k_s and y+b>=0 and y+b<k_s:
```

```
                    op_h[i][j]+=kernel_h[x+a][y+b]*img[i-x][j-y]
```

```
                else:
```

```
                    op_v[i][j]+=0
```

```
plt.imshow(op_h,'gray')
```

```
plt.title("Horizontal raw output:")
```

```
plt.show()
```

```
op_h = clip(op_h)
```

```
plt.imshow(op_h,'gray')
```

```
plt.title("Horizontal clipped output:")  
plt.show()
```

```
op_v = op_v+op_h
```

```
op_v = clip(op_v)
```

```
plt.imshow(op_v,'gray')  
plt.title("Added clipped output:")  
plt.show()
```

```
img = img+op_v
```

```
img = clip(img)
```

```
plt.imshow(img,'gray')  
plt.title("Enhanced output:")  
plt.show()
```