Computer Vision Hw9

by b06902034 黃柏諭

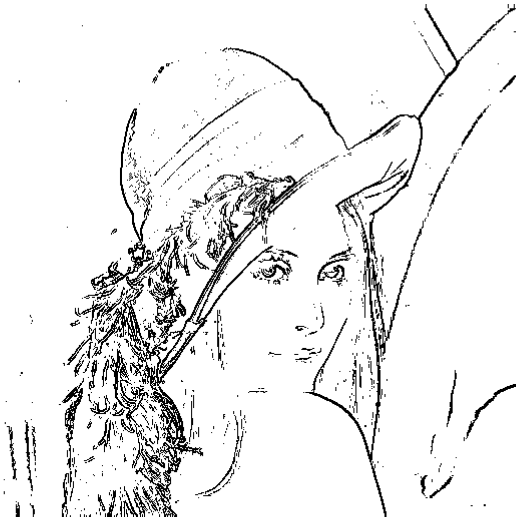
Code

```
import cv2
import numpy as np
from scipy import signal
import math

def crossCorrelation(img, msk):
    pad_width = int(msk.shape[0] / 2)
    return signal.fftconvolve(np.pad(img, (pad_width, pad_width), "edge"), np.flip(msk),
mode = "valid").round(0).astype(int)

def dis(img, msk1, msk2, threshold):
    r1 = crossCorrelation(img, msk1)
    r2 = crossCorrelation(img, msk2)
    r1 = np.multiply(r1, r1)
    r2 = np.multiply(r2, r2)
    r1 = np.add(r1, r2)
    return (r1 <= (threshold * threshold)) * np.full(img.shape, 255)

def maxv(img, msk, threshold):
    rot = np.copy(msk)
    res = np.full(img.shape, -threshold)
    for _ in range(8):
        rot[0, 0], rot[0, 1], rot[0, 2], rot[1, 2], rot[2, 2], rot[2, 1], rot[2, 0],
rot[1, 0] =\
        rot[0, 1], rot[0, 2], rot[1, 2], rot[2, 2], rot[2, 1], rot[2, 0], rot[1, 0],
rot[0, 0]
        res = np.maximum(res, crossCorrelation(img, rot))
    return (res <= threshold) * np.full(img.shape, 255)

def nevatia(img, threshold):
    with open("nevatia_matrix.in", "r") as fr:
        res = np.full(img.shape, -threshold)
        for _ in range(6):
            line = fr.readline()
            lst = []
            for _ in range(5):
                line = fr.readline()
                lst.append([int(num) for num in line.split(' ')])
            print(np.array(lst))
            res = np.maximum(res, crossCorrelation(img, np.array(lst)))
        return (res <= threshold) * np.full(img.shape, 255)

robert1 = np.array([[0, 0, 0], [0, -1, 0], [0, 0, 1]])
robert2 = np.array([[0, 0, 0], [0, 0, -1], [0, 1, 0]])
prewitt = np.array([[-1, -1, -1], [0, 0, 0], [1, 1, 1]])
sobel = np.array([[-1, -2, -1], [0, 0, 0], [1, 2, 1]])
```

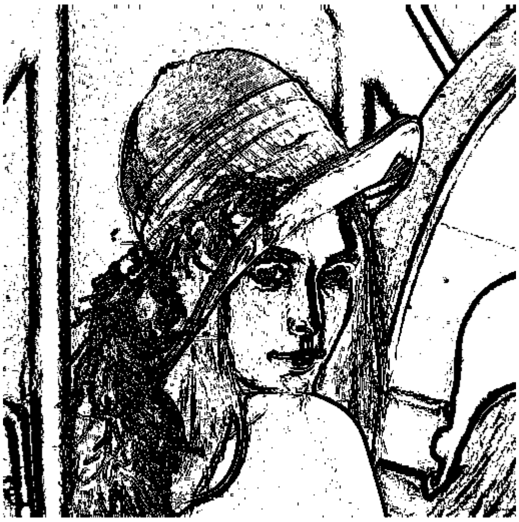
```
frei = np.array([[-1, -math.sqrt(2), -1], [0, 0, 0], [1, math.sqrt(2), 1]])
kirsch = np.array([[-3, -3, 5], [-3, 0, 5], [-3, -3, 5]])
robinson = np.array([[-1, 0, 1], [-2, 0, 2], [-1, 0, 1]])

image = cv2.imread("lena.bmp", cv2.IMREAD_GRAYSCALE)
cv2.imwrite("1_robert_30.bmp", dis(image, robert1, robert2, 30))
cv2.imwrite("2_prewitt_24.bmp", dis(image, prewitt, prewitt.T, 24))
cv2.imwrite("3_sobel_38.bmp", dis(image, sobel, sobel.T, 38))
cv2.imwrite("4_frei_30.bmp", dis(image, frei, frei.T, 30))
cv2.imwrite("5_kirsch_135.bmp", maxv(image, kirsch, 135))
cv2.imwrite("6_robinson_43.bmp", maxv(image, robinson, 43))
cv2.imwrite("7_nevatia_12500.bmp", nevatia(image, 12500))
```

Result

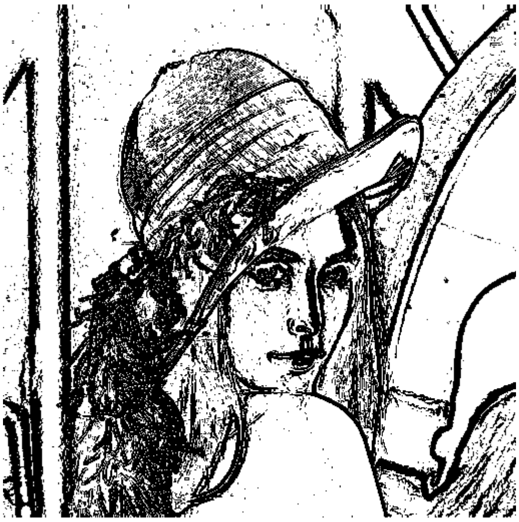
Roberts with thresholds 30



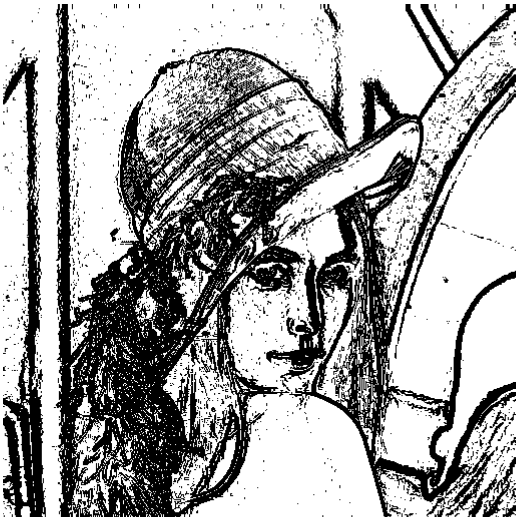
Prewitt with thresholds 24



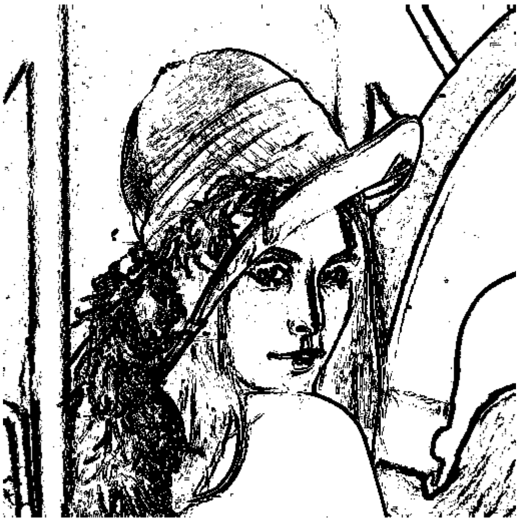
Sobel with thresholds 38



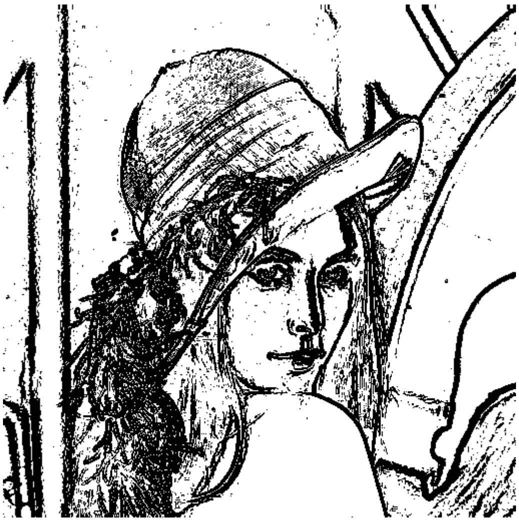
Frei and Chen gradient with thresholds 30



Kirsch compass with thresholds 135



Robinson compass with threshold 43



Nevatia-Babu 5×5 operator with threshold 12500