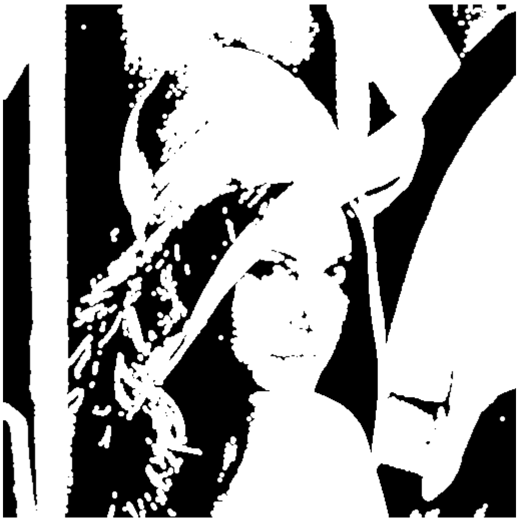
Computer Vision HW3

by b06902034 黃柏諭

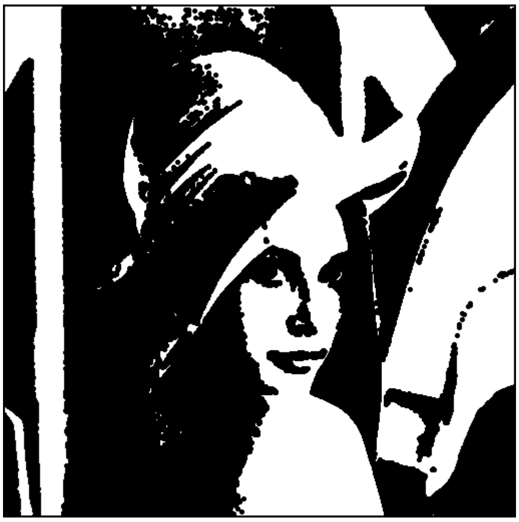
Code

```
import cv2
import numpy as np

def add(x, y):
    res = set()
    for v1 in x:
        for v2 in y:
            res.add(tuple(v1 + np.array(v2)))
    return list(res)

def inn(x, y):
    return x[0] >= 0 and x[1] >= 0 and x[0] < y.shape[0] and x[1] < y.shape[1]

def dilation(origin, kernel):
    lst = []
    for i, j in np.ndindex(origin.shape):
        if origin[i, j] == 255:
            lst.append(np.array([i, j]))
    res = np.zeros(origin.shape)
    for i, j in add(lst, kernel):
        if inn((i, j), res):
            res[i, j] = 255
    return res

def erosion(origin, kernel):
    res = np.zeros(origin.shape)
    for i, j in np.ndindex(origin.shape):
        res[i, j] = 255
        for r, c in add([np.array([i, j])], kernel):
            if not inn((r, c), res) or origin[r, c] == 0:
                res[i, j] = 0
                break
    return res

def hitMiss(origin, kernel_1, kernel_2):
    origin_prime = (origin < 128) * np.full(origin.shape, 255)
    origin = erosion(origin, kernel_1)
    origin_prime = erosion(origin_prime, kernel_2) / 255
    return origin * origin_prime


image = cv2.imread("lena.bmp", cv2.IMREAD_GRAYSCALE)
binary = (image >= 128) * np.full(image.shape, 255)
_35553 =\
[[-2, -1], [-2, 0], [-2, 1],\
 [-1, -2], [-1, -1], [-1, 0], [-1, 1], [-1, 2],\
  [0, -2], [0, -1], [0, 0], [0, 1], [0, 2],\
  [1, -2], [1, -1], [1, 0], [1, 1], [1, 2],\
  [2, -1], [2, 0], [2, 1]]
_J = [[0, -1], [0, 0], [1, 0]]
_K = [[-1, 0], [-1, 1], [0, 1]]

dil = dilation(binary, _35553)
ero = erosion(binary, _35553)
cv2.imwrite("1_dilation.bmp", dil)
cv2.imwrite("2_erosion.bmp", ero)
cv2.imwrite("3_opening.bmp", dilation(ero, _35553))
cv2.imwrite("4_closing.bmp", erosion(dil, _35553))
cv2.imwrite("5_hitmiss.bmp", hitMiss(binary, _J, _K))
```

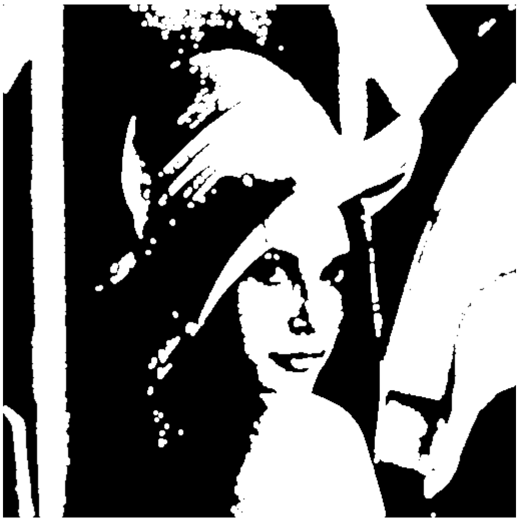
Dilation



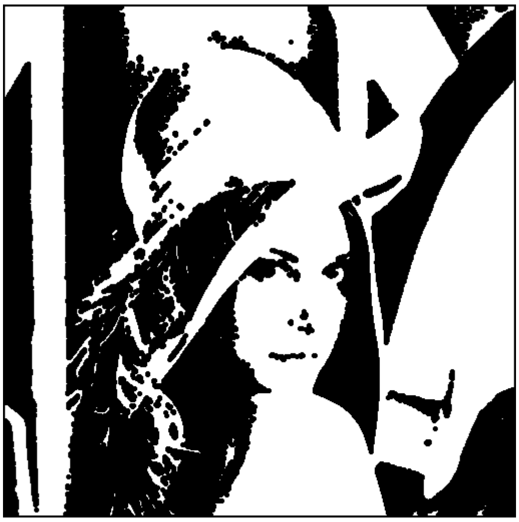
Erosion



Opening



Closing



Hit and Miss