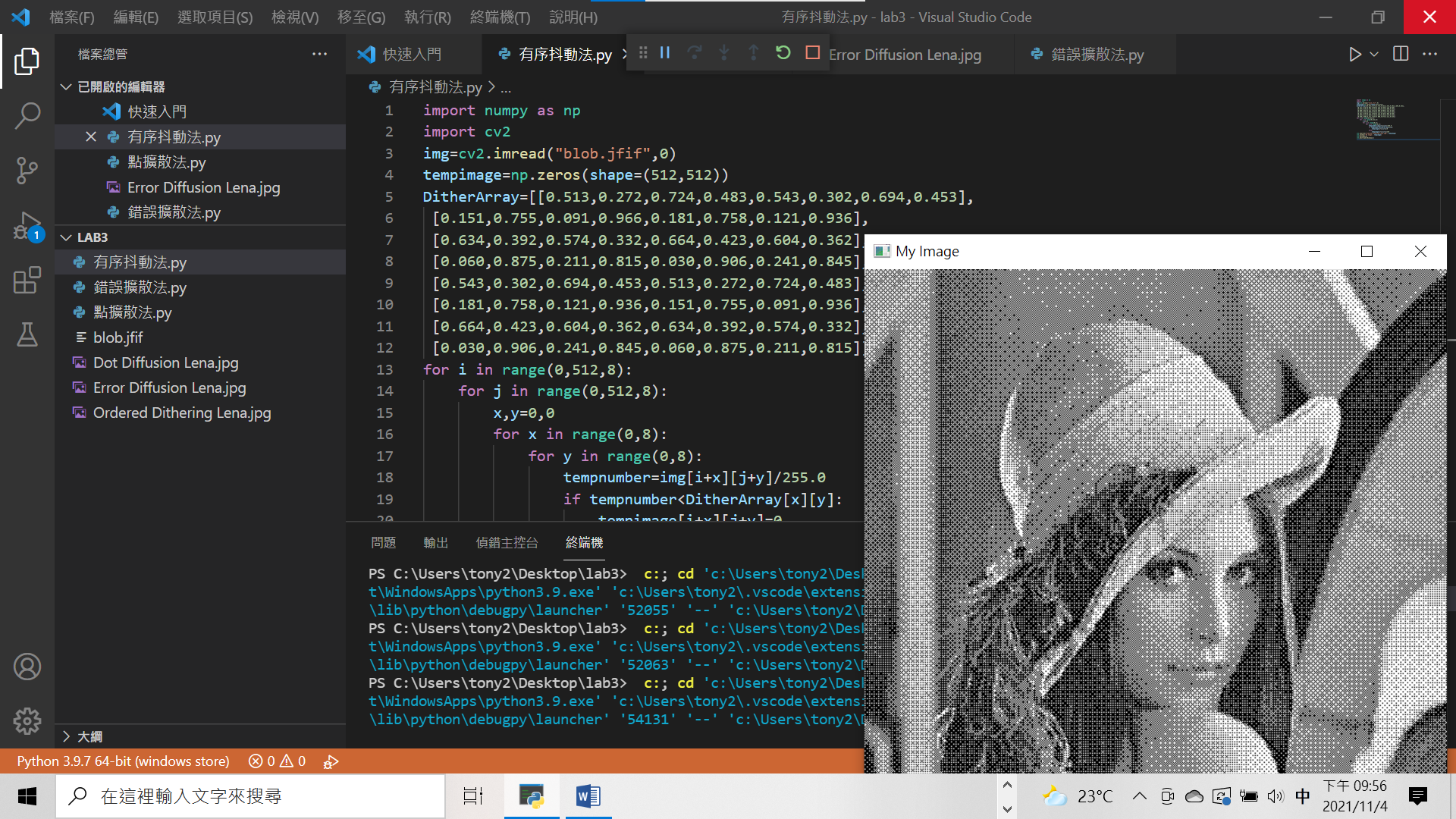
有序擴散法:



Code:

import numpy as np

import cv2

img=cv2.imread("blob.jfif",0)

tempimage=np.zeros(shape=(512,512))

DitherArray=[[0.513,0.272,0.724,0.483,0.543,0.302,0.694,0.453],

 [0.151,0.755,0.091,0.966,0.181,0.758,0.121,0.936],

 [0.634,0.392,0.574,0.332,0.664,0.423,0.604,0.362],

 [0.060,0.875,0.211,0.815,0.030,0.906,0.241,0.845],

 [0.543,0.302,0.694,0.453,0.513,0.272,0.724,0.483],

 [0.181,0.758,0.121,0.936,0.151,0.755,0.091,0.936],

 [0.664,0.423,0.604,0.362,0.634,0.392,0.574,0.332],

 [0.030,0.906,0.241,0.845,0.060,0.875,0.211,0.815]]

for i in range(0,512,8):

    for j in range(0,512,8):

        x,y=0,0

        for x in range(0,8):

            for y in range(0,8):

                tempnumber=img[i+x][j+y]/255.0

                if tempnumber<DitherArray[x][y]:

                    tempimage[i+x][j+y]=0

                else:

                    tempimage[i+x][j+y]=255

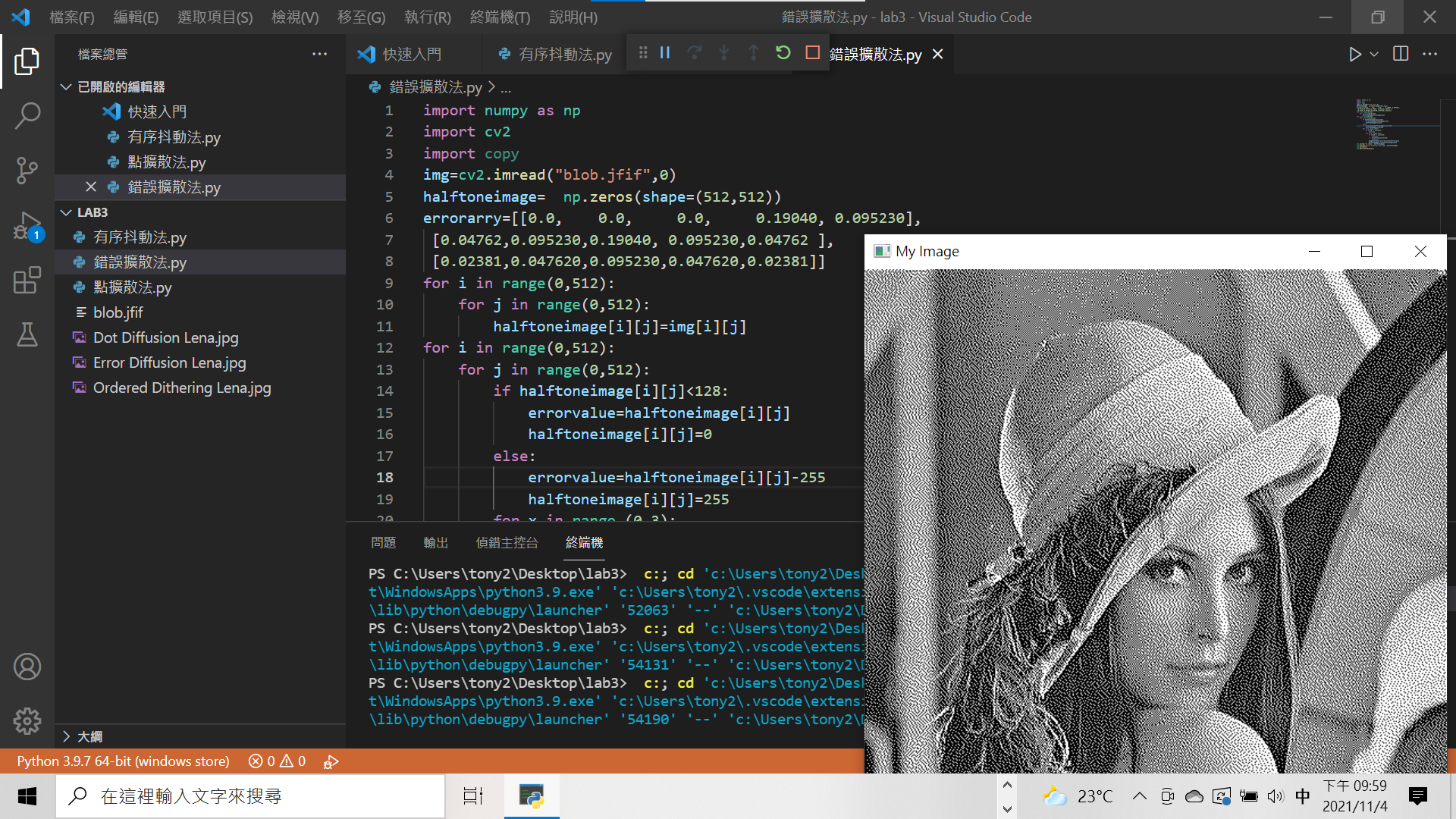
cv2.imwrite("Ordered Dithering Lena.jpg", tempimage)

cv2.imshow('My Image', tempimage)

cv2.waitKey(0)

cv2.destroyAllWindows()

錯誤擴散法:



Code:

import numpy as np

import cv2

import copy

img=cv2.imread("blob.jfif",0)

halftoneimage=  np.zeros(shape=(512,512))

errorarry=[[0.0,    0.0,     0.0,     0.19040, 0.095230],

 [0.04762,0.095230,0.19040, 0.095230,0.04762 ],

 [0.02381,0.047620,0.095230,0.047620,0.02381]]

for i in range(0,512):

    for j in range(0,512):

        halftoneimage[i][j]=img[i][j]

for i in range(0,512):

    for j in range(0,512):

        if halftoneimage[i][j]<128:

            errorvalue=halftoneimage[i][j]

            halftoneimage[i][j]=0

        else:

            errorvalue=halftoneimage[i][j]-255

            halftoneimage[i][j]=255

        for x in range (0,3):

            if i+x<0 or i+x>=512:

                continue

            for y in range(-2,3):

                if j+y<0 or j+y>=512:

                    continue

                if  errorarry[x][y+2]==0:

                    continue

                weightvalue=errorarry[x][y+2]\*errorvalue

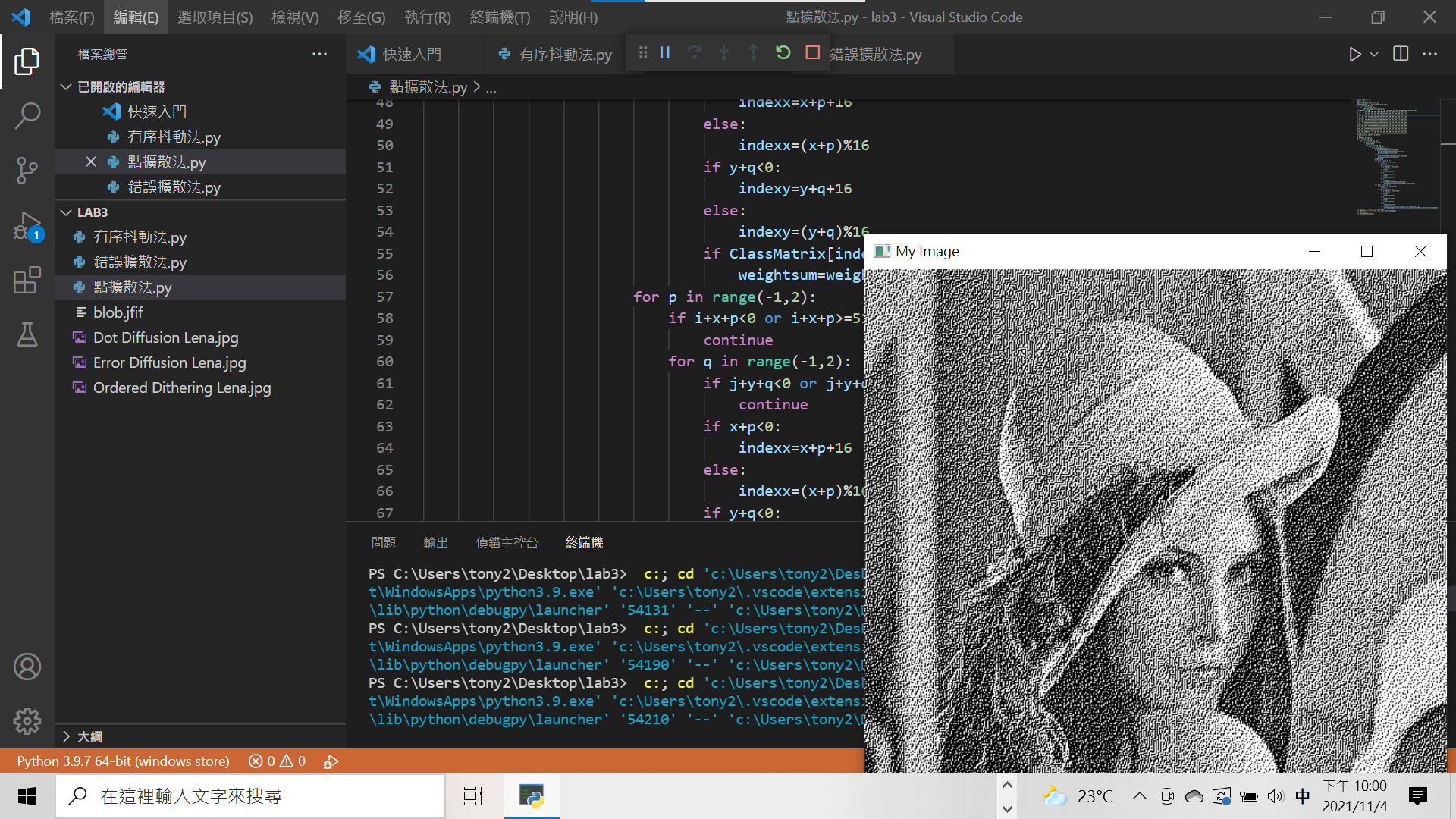
                halftoneimage [i+x][j+y]+=weightvalue

cv2.imshow('My Image', halftoneimage)

cv2.imwrite("Error Diffusion Lena.jpg", halftoneimage)

cv2.waitKey(0)

cv2.destroyAllWindows()

點擴散法:  


Code:

import numpy as np

import cv2

img=cv2.imread("blob.jfif",0)

halftoneimage=  np.zeros(shape=(512,512))

for i in range(0,512):

    for j in range(0,512):

        halftoneimage[i][j]=img[i][j]

ClassMatrix =[[204,  0,  5, 33, 51, 59, 23,118, 54, 69, 40,160,169,110,168,188],

 [  3,  6, 22, 36, 60, 50, 74,115,140, 82,147,164,171,142,220,214],

 [ 14,  7, 42, 16, 63, 52, 94, 56,133,152,158,177,179,208,222,  1],

 [ 15, 26, 43, 75, 79, 84,148, 81,139,136,166,102,217,219,226,  4],

 [ 17, 39, 72, 92,103,108,150,135,157,193,190,100,223,225,227, 13],

 [ 28,111, 99, 87,116,131,155,112,183,196,181,224,232,228, 12, 21],

 [ 47,120, 91,105,125,132,172,180,184,205,175,233,245,  8, 20, 41],

 [ 76, 65,129,137,165,145,178,194,206,170,229,244,246, 19, 24, 49],

 [ 80, 73,106,138,176,182,174,197,218,235,242,249,247, 18, 48, 68],

 [101,107,134,153,185,163,202,173,231,241,248,253, 44, 88, 70, 45],

 [123,141,149, 61,195,200,221,234,240,243,254, 38, 46, 77,104,109],

 [ 85, 96,156,130,203,215,230,250,251,252,255, 53, 62, 93, 86,117],

 [151,167,189,207,201,216,236,239, 25, 31, 34,113, 83, 95,124,114],

 [144,146,191,209,213,237,238, 29, 32, 55, 64, 97,126, 78,128,159],

 [187,192,198,212,  9, 10, 30, 35, 58, 67, 90, 71,122,127,154,161],

 [199,210,211,  2, 11, 27, 37, 57, 66, 89, 98,121,119,143,162,186]]

ErrorArray=[[0.38459,1,0.38459],

[1,0,1],

[0.38459,1,0.38459]]

for k in range(0,256):

    for i in range(0,512,16):

        for j in range(0,512,16):

            findvalue=0

            for x in range(0,16):

                for y in range(0,16):

                    if k==ClassMatrix[x][y]:

                        if halftoneimage[i+x][j+y]<128:

                            errorvalue=halftoneimage[i+x][j+y]

                            halftoneimage[i+x][j+y]=0

                        else:

                            errorvalue=halftoneimage[i+x][j+y]-255

                            halftoneimage[i+x][j+y]=255

                        weightsum=0

                        for p in range(-1,2):

                            if i+x+p<0 or i+x+p>512:

                                continue

                            for q in range(-1,2):

                                if j+y+q<0 or j+y+q>512:

                                    continue

                                if x+p<0:

                                    indexx=x+p+16

                                else:

                                    indexx=(x+p)%16

                                if y+q<0:

                                    indexy=y+q+16

                                else:

                                    indexy=(y+q)%16

                                if ClassMatrix[indexx][indexy]>k:

                                    weightsum=weightsum+ClassMatrix[p+1][q+1]

                        for p in range(-1,2):

                            if i+x+p<0 or i+x+p>=512:

                                continue

                            for q in range(-1,2):

                                if j+y+q<0 or j+y+q>=512:

                                    continue

                                if x+p<0:

                                    indexx=x+p+16

                                else:

                                    indexx=(x+p)%16

                                if y+q<0:

                                    indexy=y+q+16

                                else:

                                    indexy=(y+q)%16

                                if ClassMatrix[indexx][indexy]>k and weightsum!= 0:

                                    halftoneimage[i+x+p][j+y+q]+=errorvalue\*ClassMatrix[p+1][q+1]/weightsum

cv2.imshow('My Image', halftoneimage)

cv2.imwrite("Dot Diffusion Lena.jpg", halftoneimage)

cv2.waitKey(0)

cv2.destroyAllWindows()

實習心得:

這次透過三種方法將影像混色，使我對影像混色的理論即實作更加了解。雖然程式碼很冗長，但成功將影像混色也帶給我很大的成就感。