import numpy as np  
import cv2  
import copy  
  
  
def convWithAll(img, imgS, filS):  
 imgTempf = copy.deepcopy(img)  
 imgAfterConv = copy.deepcopy(img)  
 for i in range(imgS[0]): #use i as index of image to determine which pixels need to be filtered  
 for j in range(imgS[1]):  
 imgAfterConv = conv(imgTempf, imgAfterConv, imgS, filS, i, j)  
  
 for i in range(imgS[0]):  
 for j in range(imgS[1]):  
 imgTempf[i][j] = int(imgAfterConv[i][j]) - int(imgTempf[i][j])  
  
 return imgTempf  
  
  
def conv(imgOrignal, imgNew, imgSS, filSS, ii, jj): #imgOrignal - Original Image, imgNew - New Image, ker - filter  
 xCenter = int(filSS[0]/2) # if 3x3 kernel, xCenter = 1 from 0, 1, 2  
 yCenter = int(filSS[1]/2)  
  
 for l in range(filSS[0]): # if 3x3 kernel, l = 0, 1, 2  
 if ii + filSS[0] < imgSS[0]:  
 for m in range(filSS[1]): # m = 0, 1, 2  
 if jj + filSS[1] < imgSS[1]:  
 if imgOrignal[ii + xCenter][jj + yCenter] > 0:  
 imgNew[ii + l][jj + m] = imgOrignal[ii + xCenter][jj + yCenter]  
 else:  
 break  
 else:  
 break  
 return imgNew  
  
  
image = cv2.imread('pic4.png')  
imgGray = cv2.cvtColor(image, cv2.COLOR\_RGB2GRAY)  
imgTemp = copy.deepcopy(imgGray)  
arr = np.array(imgTemp)  
imgSize = arr.shape  
  
print('imSize: ', imgSize)  
  
fNum = 3  
kernel = []  
for i in range(fNum):  
 kernel.append([1] \* fNum)  
  
fil = np.array(kernel)  
filSize = fil.shape  
print('kernel: ', kernel)  
print('filSize: ', filSize)  
  
imgTemp2 = convWithAll(imgTemp, imgSize, filSize)  
  
print('imSize: ', imgSize)  
print('end')  
  
cv2.imshow('Before', imgGray)  
cv2.imshow('After', imgTemp2)  
  
cv2.waitKey(0)