รหัสนักศึกษา 1624041400008-2 ชื่อ-นามสกุล นายโชคชัย แจ่มน้อย

กำหนดส่ง 12/09/2564

Lab 5 Image Binarization with Local Threshold

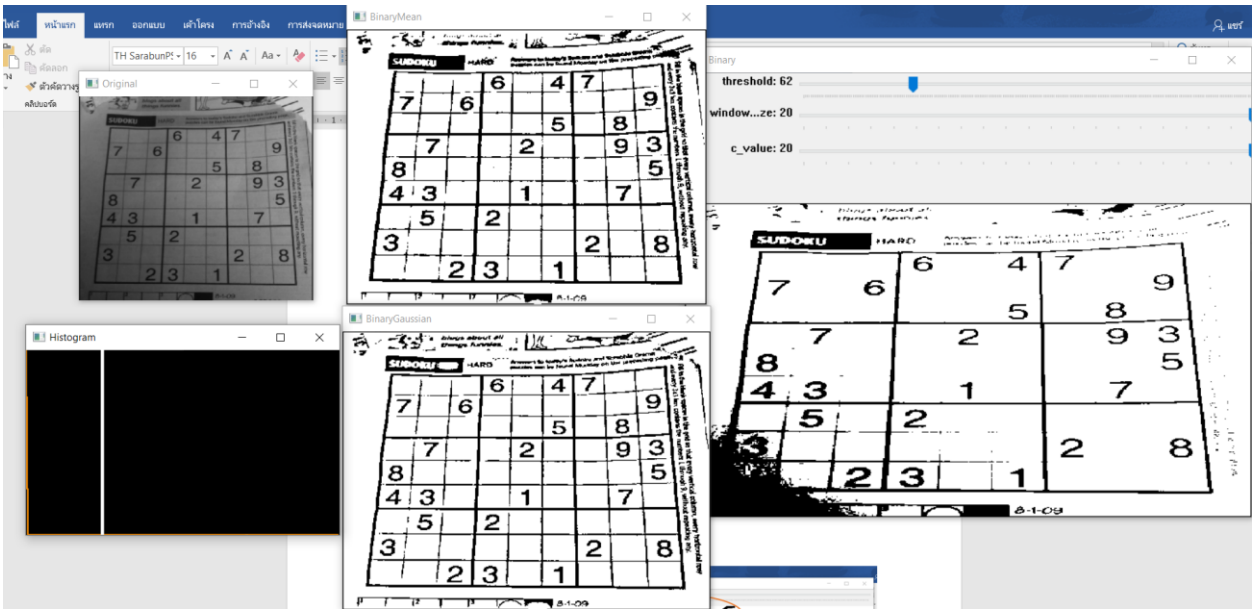
1. Source Code

```python
import cv2 as cv
import numpy as np
import sys
from matplotlib import pyplot as plt
# Global Variable
threshold_value = 180
window_size =  1 #window_size *2 +1
c_value = 10
source_img = np.zeros((10,10,3), dtype=np.uint8)
adjusted_img = np.zeros((10,10,3), dtype=np.uint8) # * Global threshold
adjusted_mean_img = np.zeros((10,10,3), dtype=np.uint8) # * Local threshold - Mean-c Algorithm
adjusted_gaussian_img = np.zeros((10,10,3), dtype=np.uint8)  #* Global Threshold GaussianWeightSum-c algorithm
hist_img = np.zeros((10,10,3), dtype=np.uint8)

def handler_adjustThreshold(x):
    global threshold_value,window_size,c_value
    global source_img,adjusted_img,hist_img,adjusted_mean_img,adjusted_gaussian_img
    threshold_value = cv.getTrackbarPos('threshold','Binary')
    #* c_value
    c_value = cv.getTrackbarPos('c_value','Binary') - 10
    #* box size or window_size
    window_size = (cv.getTrackbarPos('window_size','Binary') * 2 ) + 1
    print(f"Global Threshold = {threshold_value}")
    print(f"window_size = {window_size}")
    print(f"c_value = {c_value}")
    # ! อธิบาย
    # ----------------------------
    # _, adjusted_img = cv.adaptiveThreshold(source_img,255,cv.ADAPTIVE_THRESH_GAUSSIAN_C,\
    #         cv.THRESH_BINARY,11,2)
    _, adjusted_img = cv.threshold(source_img, threshold_value, 255, cv.THRESH_BINARY)
    adjusted_mean_img = cv.adaptiveThreshold(source_img,255,cv.ADAPTIVE_THRESH_MEAN_C,cv.THRESH_BINARY,window_size,c_value)
    adjusted_gaussian_img = cv.adaptiveThreshold(source_img,255,cv.ADAPTIVE_THRESH_GAUSSIAN_C,cv.THRESH_BINARY,window_size,c_value)
    # ! เปลี่ยนเป็น cv.adaptivethreshold
    # ! อธิบายแค่ค่าตั้ง threshold #ค่า max 255 เป็น ขาว 0 เป็นขาว อัลกอริทึมเราใช้เป็น THRESH_BINARY
    # Update histogram
    histSize = 256
    histRange = (0, 256) # the upper boundary is exclusive
    accumulate = False
    gray_hist = cv.calcHist(source_img, [0], None, [histSize], histRange, accumulate=accumulate) # ----------------------------
    hist_w = 512
    hist_h = 400
    bin_w = int(round( hist_w/histSize ))
    hist_img = np.zeros((hist_h, hist_w, 3), dtype=np.uint8)
    cv.normalize(gray_hist, gray_hist, alpha=0, beta=hist_h, norm_type=cv.NORM_MINMAX)
    for i in range(1, histSize):
        cv.line(hist_img, ( bin_w*(i-1), hist_h - int(gray_hist[i-1]) ),
                ( bin_w*(i), hist_h - int(gray_hist[i]) ),
                ( 0, 144, 255), thickness=2)
                                        #histogram 1 channel จาก 3 channel
    cv.line(hist_img,(threshold_value*2,0),(threshold_value*2,hist_h-1),(255,255,255),3)
def main():
    global threshold_value,window_size,c_value,adjusted_mean_img,adjusted_gaussian_img
    global source_img,adjusted_img,hist_img
    if(len(sys.argv)>=2):
        source_img = cv.imread(str(sys.argv[1]))
    else :
        source_img = cv.imread("./sudoku.png", 1)

    source_img = cv.cvtColor(source_img,cv.COLOR_BGR2GRAY) # convert to GrayScale #เปิดใช่ GrayScale แปลง จาก BGR เป็น GRAY  #อธิบายการทำงาน
    #อธิบายภาพแล้วทิ้งไม่ต้องยาวมาก
    #named windows
    cv.namedWindow("Original", cv.WINDOW_NORMAL)
    cv.namedWindow("Binary", cv.WINDOW_NORMAL)
    cv.namedWindow("BinaryMean", cv.WINDOW_NORMAL)
    cv.namedWindow("BinaryGaussian", cv.WINDOW_NORMAL)
    cv.namedWindow("Histogram", cv.WINDOW_NORMAL)

    #create trackbar
    cv.createTrackbar('threshold', 'Binary', threshold_value, 255, handler_adjustThreshold)
    cv.createTrackbar('window_size', 'Binary', window_size, 20, handler_adjustThreshold)
    cv.createTrackbar('c_value', 'Binary', c_value, 20, handler_adjustThreshold)

    adjusted_img  = source_img.copy()
    handler_adjustThreshold(-1);
    while(True):
        cv.imshow("Original",source_img)
        cv.imshow("Binary",adjusted_img)
        cv.imshow("BinaryMean",adjusted_mean_img)
        cv.imshow("BinaryGaussian",adjusted_gaussian_img)
        cv.imshow("Histogram",hist_img)
        key = cv.waitKey(100)
        if(key==27): #ESC = Exit Program
            break

    cv.destroyAllWindows()

if __name__ == "__main__":
    main()
```

รหัสนักศึกษา 1624041400008-2 ชื่อ-นามสกุล นายโชคชัย แจ่มน้อย

กำหนดส่ง 12/09/2564

2. ทดลองปรับค่า WindowSize และ C



อธิบายผลลัพธ์ : ผลลัพธ์ที่เกิดขึ้นคือภาพที่จากการทำ AdaptiveThreshold โดยใช้ทั้งวิธีการของ MEAN และ Weighted-sum of Gaussian มีรายละเอียดเกือบที่จะครบถ้วน 100% เมื่อเทียบกับภาพต้นฉบับ และดีกว่าภาพที่ได้จาก global threshold เนื่องจากภาพ ภาพที่ได้จาก global threshold เป็นการกำหนดค่า threshold เพียงค่าเดียวแต่นำไปใช้ทั้งภาพ แต่ AdaptiveThreshold มีการกำหนดกล่อง(window_size)ขึ้นมาเพื่อหาค่าที่เหมาะสมสำหรับภาพในกล่องนั้น ๆ ทำให้แก้ปัญหาการสูญเสียรายละเอียดของภาพได้

โดยในภาพนี้กำหนดค่า window_size = 20 pixel และค่า c = 20