

[Computer Vision I] Homework 2

學號: R07943087 姓名: 林啟源

Write a program to generate:

```
content = load_image(config.init_pict)           # 讀圖片
width, height = content.size                     # 紀錄圖片長寬
content_np = np.asarray(content).copy()          # 複製圖片至 2D numpy array
content_np_new = np.zeros(shape=(width,height))  # 建立一個空的 2D numpy
                                                  # array，以供之後 function 使用
```

(a) a binary image (threshold at 128)

```
# (a) a binary image (threshold at 128)
def Binarize(content_np, content_np_new, width, height, config):
    for x in range(width):
        for y in range(height):
            if content_np[x][y]<128:
                content_np_new[x][y] = 0
            else:
                content_np_new[x][y] = 255
    Image.fromarray(np.uint8(content_np_new)).save(config.bin)
    print("Binary Image Finish!")
```

- 使用 2 個 for 迴圈控制 lena.bmp 的 2D numpy array。判斷每個 pixel ($\text{content_np}[x][y]$) 是否大於 128，若小於 128 則紀錄此 pixel 為 0 於 $\text{content_np_new}[x][y]$ ；反之，紀錄為 255。
- `Image.fromarray(np.uint8(content_np_new)).save(config.bin)` 用來儲存圖片。



binarize lena.bmp

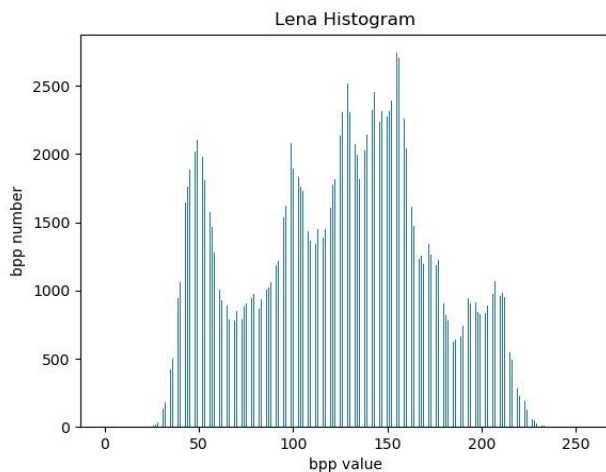
(b) a histogram

```
# (b) a histogram
def Histogram(content_np, width, height, config):
    histogram_np = np.zeros(shape=(256))

    for x in range(width):
        for y in range(height):
            histogram_np[content_np[x][y]] += 1

    x = np.arange(256)
    w = 0.3
    plt.bar(x, histogram_np, w)
    plt.title('Lena Histogram')
    plt.xlabel("bpp value")
    plt.ylabel("bpp number")
    plt.savefig(config.his)
    #plt.show()
    print("Histogram Image Finish!")
```

- 使用 2 個 for 迴圈控制 lena.bmp 的 2D numpy array。建立 1D numpy array “histogram_np” 紀錄 histogram 0~255 的值，例如: pixel[10][10]=125, 則 histogram[125] += 1。
- plt.bar(x, histogram_np, w) # 建立長條圖
- plt.title('Lena Histogram') # 圖名
- plt.xlabel("bpp value") # x 軸名稱
- plt.ylabel("bpp number") # y 軸名稱
- plt.savefig(config.his) # 儲存圖片



histogram lena.jpg

(c) connected components (regions with + at centroid, bounding box)

```
# (c) connected components
def con_com(content_np_new, width, height, config):
    # Initial Binarize Img
    img = cv2.imread(config.init_pict)
    ret, img2 = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY)

    # Labeled Img
    filter9 = np.ones((3, 3), dtype=np.int)
    filter9[0][0], filter9[0][2], filter9[2][0], filter9[2][2] = 0, 0, 0, 0
    label_np, ncomponents = label(content_np_new, filter9)
    dig_num, counts = np.unique(label_np, return_counts=True)
    img_dict = dict(zip(dig_num, counts))
    label_np2 = label_np.copy()
    img_num = []
```

- step1. 讀圖片並轉成灰階
- step2. 使用 3*3 的 filter (8 連通)或 3*3 的 filter4 角為 0 (4 連通)對圖片做 label (label_np, ncomponents = label(content_np_new, filter9))。此次作業我是用 4 連通 filter 對圖片做 label。並儲存 label 後的圖於 label_np。

```
# connected components more than 500 pixels
for l in range(len(img_dict)):
    if (img_dict[l]>500):
        img_num.append(l)
img_num.remove(0)

rect_info = np.zeros(shape=(len(img_num),6), dtype=int) # x, y, w, h, cx, cy

for p in range(len(img_num)):
    for x in range(width):
        for y in range(height):
            if label_np[x][y] == img_num[p]:
                label_np[x][y] = 255
            else:
                label_np[x][y] = 0

    rect_info[p][:4] = cv2.boundingRect(label_np.astype('uint8')) # rectangle
    centroid = cv2.moments(label_np.astype('uint8')) # line
    rect_info[p][4] = int(centroid['m10']/centroid['m00'])
    rect_info[p][5] = int(centroid['m01']/centroid['m00'])
    label_np = label_np2.copy()
```

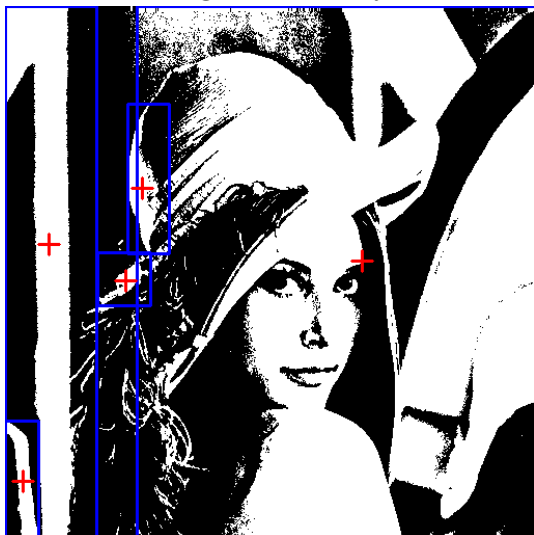
- step3. 篩選“面積 500 以上的連通數值”，儲存於 img_num[] 中。
- step4. 對超過面積 500 以上的連通數值尋找輪廓及 centroid 座標，將數值儲存於 rect_info[] 中，x=輪廓矩形的左上 x 座標, y=輪廓矩形的左上 y 座標, w=輪廓矩形長度, h=輪廓矩形寬度, cx=centroid 座標 x, cy=centroid 座標 y。
- 在尋找輪廓時，會先將目標連通數值以外的 pixel 設為 0，目標連通數值設為 255，並使用 cv2.boundingRect()找出 x, y, w, h。
- 找 centroid 時，使用 cv2.moments() function。

```
for draw in range(len(rect_info)):
    x, y, w, h, cx, cy = rect_info[draw, :]
    label_np = cv2.rectangle(img2, (x, y), (x+w, y+h), (0, 0, 255), 2) # rectangle
    label_np = cv2.line(img2, (cx-10, cy), (cx+10, cy), (255, 0, 0), 2) # line
    label_np = cv2.line(img2, (cx, cy-10), (cx, cy+10), (255, 0, 0), 2) # line

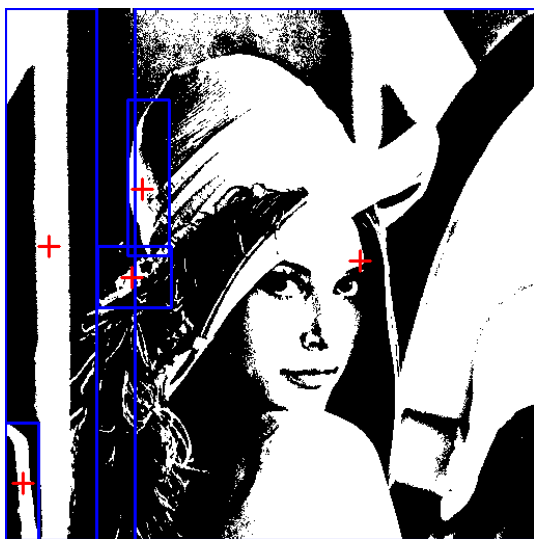
Image.fromarray(np.uint8(label_np)).save(config.con)
print("Connected_Components Image Finish!")
```

- 因為有座標點，因此使用 cv2 的畫線 function 畫出矩形和十字

➤ 使用 `Image.fromarray(np.uint8(label_np)).save(config.con)` 儲存圖片。



4 連通 connected components lena_4.bmp



8 連通 connected components lena_8.bmp