HW 2: Basic Image Manipulation II

Source Code

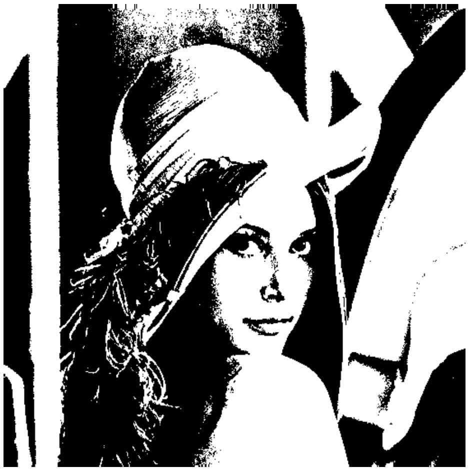
All questions are written in Python code, please refer to the file “main.py”.
All images will be stored in the folder “res” (automatically create a new folder).
In accordance with the **FAQ** of course website:

- All parts of the question are written from scratch, except for plotting image and bar chart

Answer

1. Binary image with threshold of 128
   Algorithm:
   1) Read all pixels in the image
   2) All pixel value less than 128 is converted to 0, and 1 for the rest

```
binaryimg = []
for i in range(height):
    for j in range(width):
        if img[i][j] < 128:
            binaryimg.append(0)
        else:
            binaryimg.append(255)
binaryimg = np.asarray(binaryimg).reshape(height, width)
cv2.imwrite("res/binary_image.bmp", binaryimg)
```

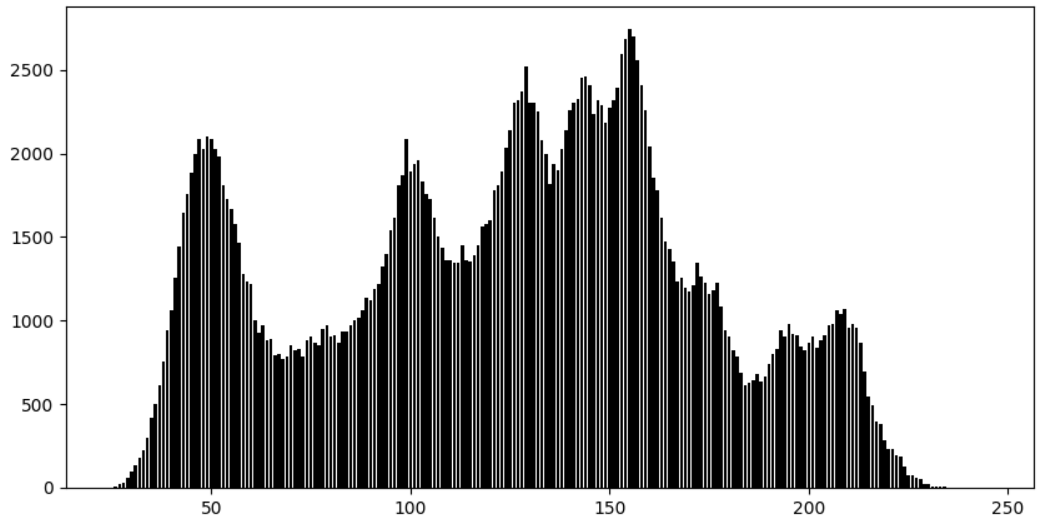


2. <u>Histogram</u>

   Algorithm:

   1) Count pixels of the same value from the original image
   2) Plot the counted pixel using matplotlib bar chart

```
data = {}
for i in range(height):
    for j in range(width):
        if img[i][j] not in data:
            data[img[i][j]] = 0
        data[img[i][j]] += 1
pixel = list(data.keys())
count = list(data.values())
fig = plt.figure(figsize=(10,5))
plt.bar(pixel, count, color='black')
plt.savefig('res/histogram.png')
```



3. <u>Connected Component</u>

   Algorithm (Using Iterative Algorithm):

   1) Assign a unique label to all pixels of the value 255
   2) Using 4-connectedness check its neighbors using top down and bottom up method
   3) Iterate until there are no further changes
   4) Count unique pixel value
   5) Bound pixels of the same value with threshold of pixel value count of 500
      a) Find start point and endpoint
      b) Bound using the counted points and count centroids using method explained in chapter 3

```
start_time = time.time()
temp = np.copy(binaryimg)
value = 1
# First pass
for i in range(height):
    for j in range(width):
        if temp[i][j] == 255:
            temp[i][j] = value
            value += 1
# Iterations
change = True
while change:
    change = False
    # Top Down
    for i in range(height):
        for j in range(width):
            if temp[i][j] > 0:
                # check top
                if (i > 0) and (temp[i-1][j] > 0):
                    if temp[i][j] != temp[i-1][j]:
                        temp[i][j] = min(temp[i][j], temp[i-1][j])
                        change = True
                # check left
                if (j > 0) and (temp[i][j-1] > 0):
                    if temp[i][j] != temp[i][j-1]:
                        temp[i][j] = min(temp[i][j], temp[i][j-1])
                        change = True
                # check bottom
                if (i+1 < height) and (temp[i+1][j] > 0):
                    if temp[i][j] != temp[i+1][j]:
                        temp[i][j] = min(temp[i][j], temp[i+1][j])
                        change = True
                # check right
                if (j+1 < height) and (temp[i][j+1] > 0):
                    if temp[i][j] != temp[i][j+1]:
                        temp[i][j] = min(temp[i][j], temp[i][j+1])
                        change = True
    # Bottom up
    for i in reversed(range(height)):
        for j in reversed(range(width)):
            if temp[i][j] > 0:
                # check top
                # check left
                # check bottom
                # check right
print(str(time.time() - start_time), "seconds")
# Count pixel value
pixel = []
pixelcount = []
unique, counts = np.unique(temp, return_counts=True)
for i in range(len(counts)):
    if counts[i] > 500 and unique[i] != 0:
        pixel.append(unique[i])
        pixelcount.append(counts[i])
# Bounding box
binaryimg = binaryimg.astype('uint8')
```

```
img = cv2.cvtColor(binaryimg, cv2.COLOR_GRAY2BGR)
for x in range(len(pixel)):
    minpoint = [sys.maxsize, sys.maxsize]
    maxpoint = [-1, -1]
    rsum = 0
    csum = 0
    for i in range(height):
        for j in range(width):
            if temp[i][j] == pixel[x]:
                csum += i
                rsum += j
                if i < minpoint[0]:
                    minpoint[0] = i
                if j < minpoint[1]:
                    minpoint[1] = j
                if i > maxpoint[0]:
                    maxpoint[0] = i
                if j > maxpoint[1]:
                    maxpoint[1] = j
    cv2.rectangle(img, (minpoint[1], minpoint[0]), (maxpoint[1],
maxpoint[0]), (255, 0, 0), 2)
    csum /= pixelcount[x]
    rsum /= pixelcount[x]
    xpoint = int(csum)
    ypoint = int(rsum)
    cv2.line(img, (ypoint-10, xpoint), (ypoint+10, xpoint), (0, 0, 255),
2)
    cv2.line(img, (ypoint, xpoint-10), (ypoint, xpoint+10), (0, 0, 255),
2)
cv2.imwrite("res/connected_components.bmp", img)
```