S2 6201011631188 โสภณ สุขสมบูรณ์

9 กุมภาพันธ์ 2565

ปฏิบัติการที่ 4 Otsu and Supervised Range-Constrained Thresholding

การทดลองที่ 1 การแยกบริเวณวัตถุที่สนใจด้วยวิธี Otsu thresholding

การทดลองที่ 1.1

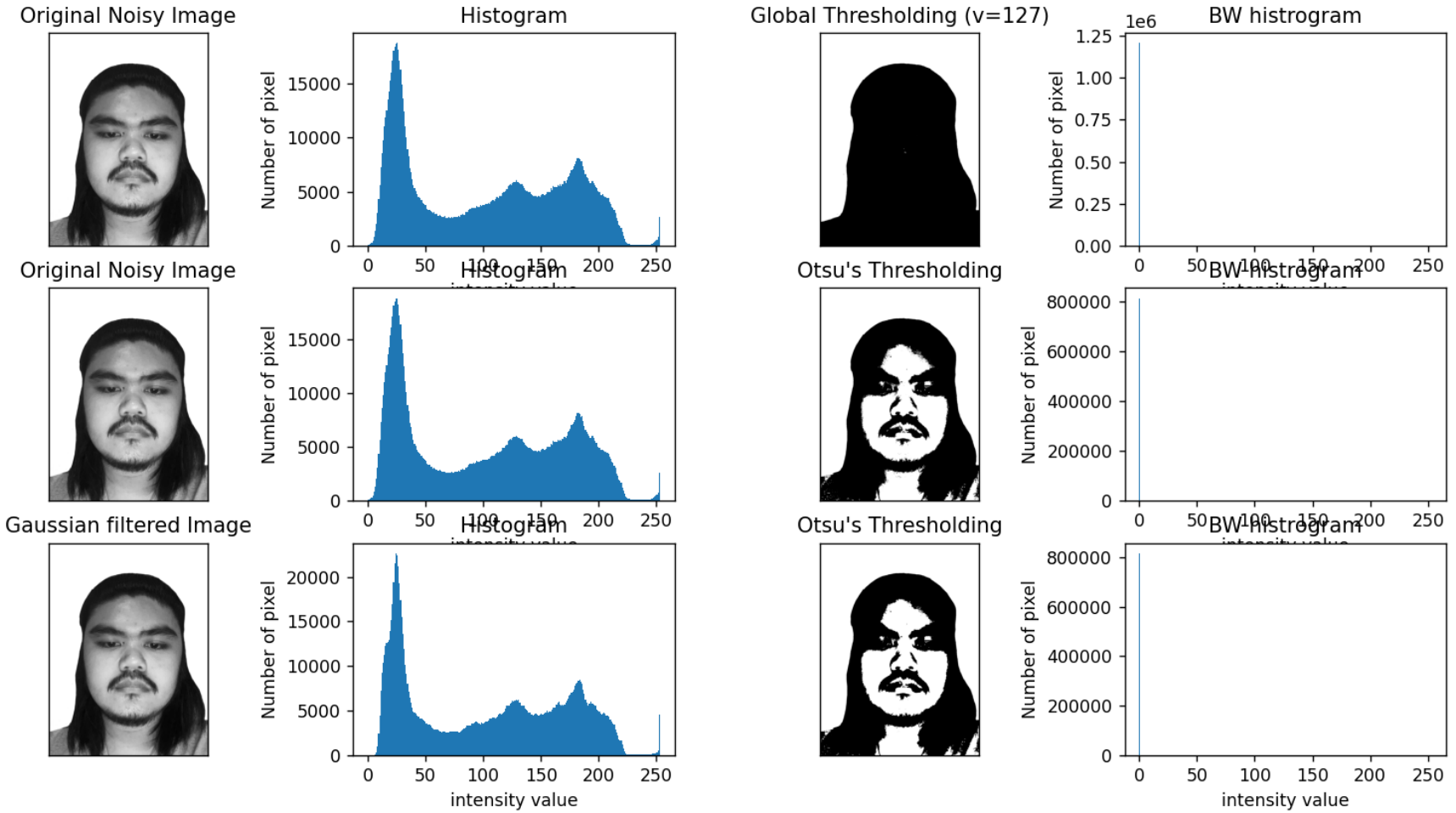
1. ให้นักศึกษาใช้รูปใบหน้าตนเอง “MyFacePic1.jpg” ดำเนินคำสั่งต่อไปนี้โดยทำการกำหนด x เป็นค่า threshold ที่สามารถแยกส่วนของใบหน้าออกจากพื้นหลังได้และสังเกตผลการทดลองที่ปรากฏ

คำสั่ง

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
img = cv.imread('D:\Telecom_Lab\Lab4\pic\MyFacePic1.jpg',0)
ret1,th1 = cv.threshold(img,236,255,cv.THRESH_BINARY)
ret2,th2 = cv.threshold(img,0,255,cv.THRESH_BINARY+cv.THRESH_OTSU)
blur = cv.GaussianBlur(img,(5,5),0)
ret3,th3 = cv.threshold(blur,0,255,cv.THRESH_BINARY+cv.THRESH_OTSU)
images = [img, 0, th1,img, 0, th2,blur, 0, th3]
titles = ['Original Noisy Image','Histogram','Global Thresholding (v=127)','Original Noisy
Image','Histogram',"Otsu's Thresholding",'Gaussian filtered Image','Histogram',"Otsu's Thresholding"]
for i in range(3):
 plt.subplot(3,4,i*4+1),plt.imshow(images[i*3],'gray')
 plt.title(titles[i*3]), plt.xticks([]), plt.yticks([])
 plt.subplot(3,4,i*4+2),plt.hist(images[i*3].ravel(),253,[0,253])
```

```
 plt.title(titles[i*3+1]),plt.ylabel('Number of pixel'),plt.xlabel('intensity value')
 plt.subplot(3,4,i*4+3),plt.imshow(images[i*3+2],'gray')
 plt.title(titles[i*3+2]), plt.xticks([]), plt.yticks([])
 plt.subplot(3,4,i*4+4),plt.hist(images[i*3+2].ravel(),253,[0,253])
 plt.title('BW histrogram'), plt.ylabel('Number of pixel'),plt.xlabel('intensity value')
plt.show()
```

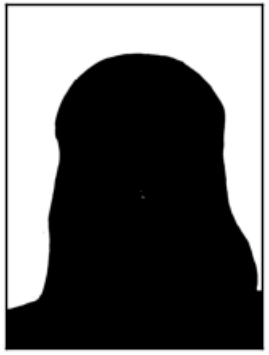
การแสดงผล



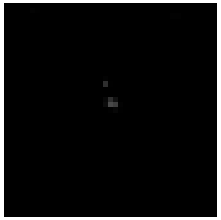
อภิปราย

การเลือกค่า x หรือค่า Threshold นั้นต้องเลือกค่าที่แบ่งระหว่างส่วนภาพที่สนใจและส่วนพื้นหลัง ในที่นี้ ภาพบุคคลมีค่าความเข้มมากที่สุดคือ 236 วิธีการหาค่า พิจารณาจากค่าbinsจาก Histogram ของภาพนั้น หรือทำการ Ramdom ค่าจากการทำ Global Thresholding แต่วิธีการนี้ต้องทำการพิจาณาค่าความเข้มจากภาพดั้งเดิมด้วย เพื่อทำการพิจารณาว่าค่าบริเวณใดหน้าใดเป็นค่ามากที่สุดที่จะเป็นตัวแบ่งใบหน้ากับพื้นหลังนั่นเอง ดังที่จะแสดงต่อไปนี้

Global Thresholding (v=127)



จากภาพ **กำหนดค่า x ที่ 235** จะสังเกตว่าช่วงจมูกมีจุดขาว ซึ่งสีขาวเรากำหนดให้เป็นส่วนของพื้นหลัง ดังนั้นแปลว่าการกำหนดค่าที่ 235 จะไม่ครอบคลุมทุกช่วงบนใบหน้า มองเผินๆอาจจะไม่มีผลกระทบอะไร แต่ในทางปฏิบัติจะส่งผลอย่างมากสำหรับการนำไปใช้กับซอฟแวร์ในระบบคอมพิวเตอร์



Global Thresholding (v=127)

ดังนั้น เราจึง**กำหนดค่า x ที่มากกว่า 235** เท่าไหร่ก็ได้ แต่ไม่ควรใกล้กับค่าความเข้มของพื้นหลัง เพื่อเป็นตัวแบ่งระหว่างใบหน้ากับพื้นหลัง ทั้งนี้เราเลือกค่า 236 เพื่อเป็นการลดการระบุส่วนที่ไม่จำเป็นที่ไม่ใช่ใบหน้านั่นเอง

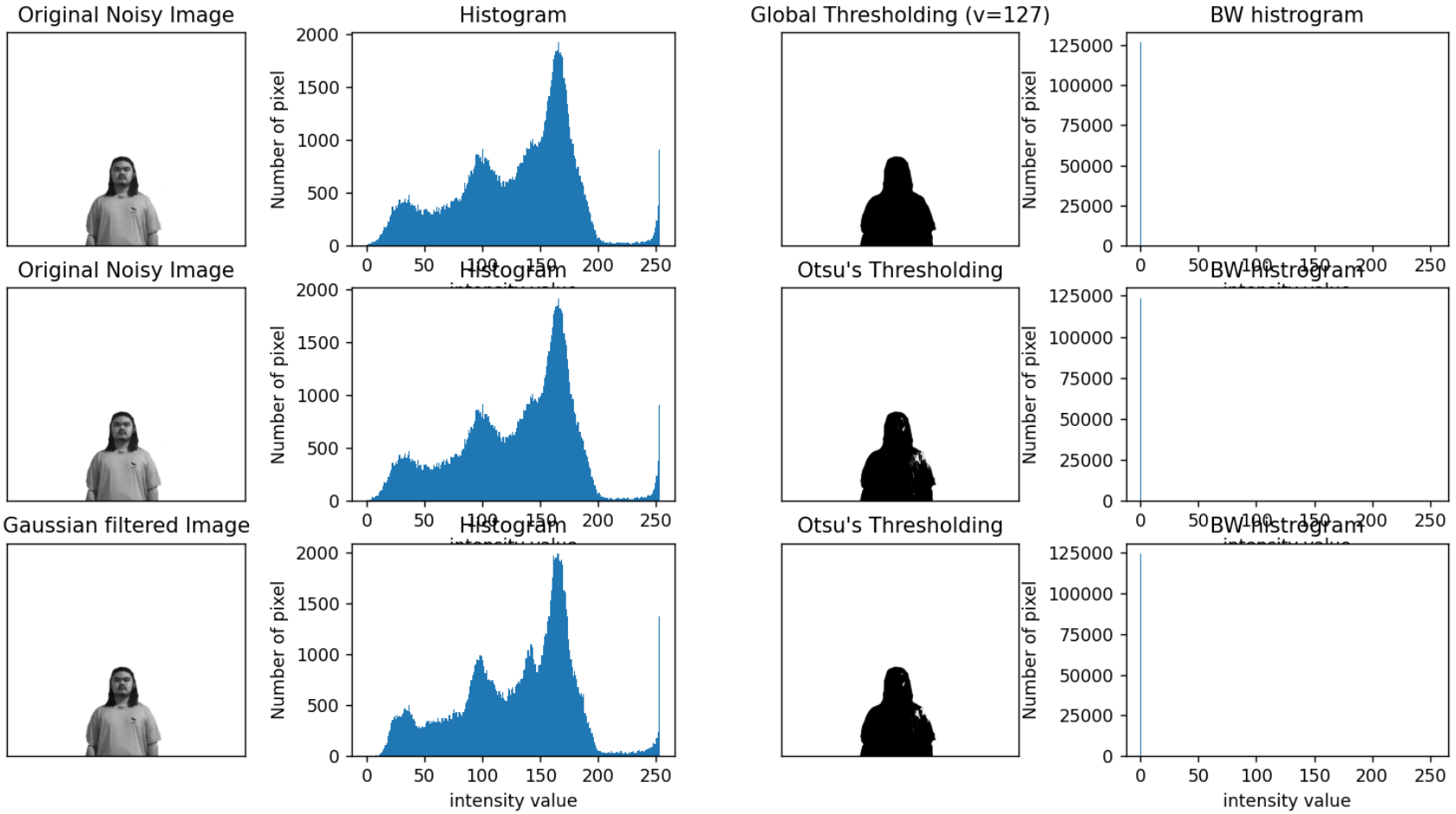
การทดลองที่ 1.2

1. ให้นักศึกษาถ่ายรูปครึ่งตัวของตนเอง “MyFacePic2.jpg”โดยให้นักศึกษาถ่ายในระยะห่าง 2-3 เมตรหรือให้ตัวของนักศึกษาเล็กกว่าพื้นหลัง 4-5 เท่า

คำสั่ง

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
img = cv.imread('D:\Telecom_Lab\Lab4\pic\MyFacePic2.jpg',0)
ret1,th1 = cv.threshold(img,211,255,cv.THRESH_BINARY)
ret2,th2 = cv.threshold(img,0,255,cv.THRESH_BINARY+cv.THRESH_OTSU)
blur = cv.GaussianBlur(img,(5,5),0)
ret3,th3 = cv.threshold(blur,0,255,cv.THRESH_BINARY+cv.THRESH_OTSU)
images = [img, 0, th1,img, 0, th2,blur, 0, th3]
titles = ['Original Noisy Image','Histogram','Global Thresholding (v=127)','Original Noisy
Image','Histogram',"Otsu's Thresholding",'Gaussian filtered Image','Histogram',"Otsu's
Thresholding"]
for i in range(3):
 plt.subplot(3,4,i*4+1),plt.imshow(images[i*3],'gray')
 plt.title(titles[i*3]), plt.xticks([]), plt.yticks([])
 plt.subplot(3,4,i*4+2),plt.hist(images[i*3].ravel(),253,[0,253])
 plt.title(titles[i*3+1]),plt.ylabel('Number of pixel'),plt.xlabel('intensity value')
 plt.subplot(3,4,i*4+3),plt.imshow(images[i*3+2],'gray')
 plt.title(titles[i*3+2]), plt.xticks([]), plt.yticks([])
 plt.subplot(3,4,i*4+4),plt.hist(images[i*3+2].ravel(),253,[0,253])
 plt.title('BW histrogram'), plt.ylabel('Number of pixel'),plt.xlabel('intensity value')
plt.show()
```

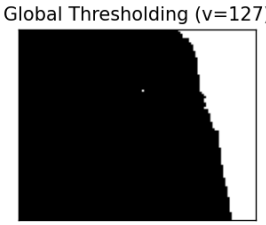
การแสดงผล


Original Noisy Image
Histogram
Global Thresholding (v=127)
BW histrogram
Original Noisy Image
Histogram
Otsu's Thresholding
BW histrogram
Gaussian filtered Image
Histogram
Otsu's Thresholding
BW histrogram
Number of pixel
intensity value

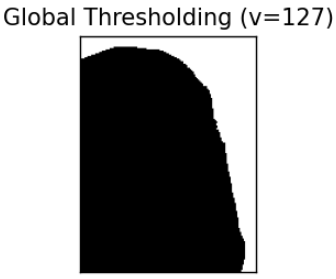
อภิปราย

การเลือกค่า x ยังคงใช้เหตุผลเดียวกับข้อ 1.1 นั่นคือ ต้องครอบคลุมทั้งใบหน้า และเป็นตัวแบ่งระหว่างพื้นหลังกับรูปบุคคล ทั้งนี้เราเลือกค่า x คือ 211 เหตุผลที่เลือกค่านี้ จะอธิบายต่อจากนี้

ค่า **x ที่กำหนดคือ 210** มองเผินๆจะไม่มีอะไร แต่ถ้าทำการซูมภาพจะพบว่ามีจุดขาวขึ้นในภาพดังที่แสดง



ค่า **x ที่กำหนดคือ 211** เราจะทำการซูมภาพ ณ ตำแหน่งเดิม เพื่อทำการพิจาณาและทำการเปรียบเทียบภาพทั้ง2



จากภาพที่ทำการ Crop มาจะเห็นว่า การเปลี่ยนค่าความเข้มเพียงค่าเดียว ทำให้จุดขาวนั้นหายไป แปลว่า ค่า x หรือ ค่า Threshold ที่เรากำหนดนั้นครอบคลุมทั้งใบหน้าของเรา

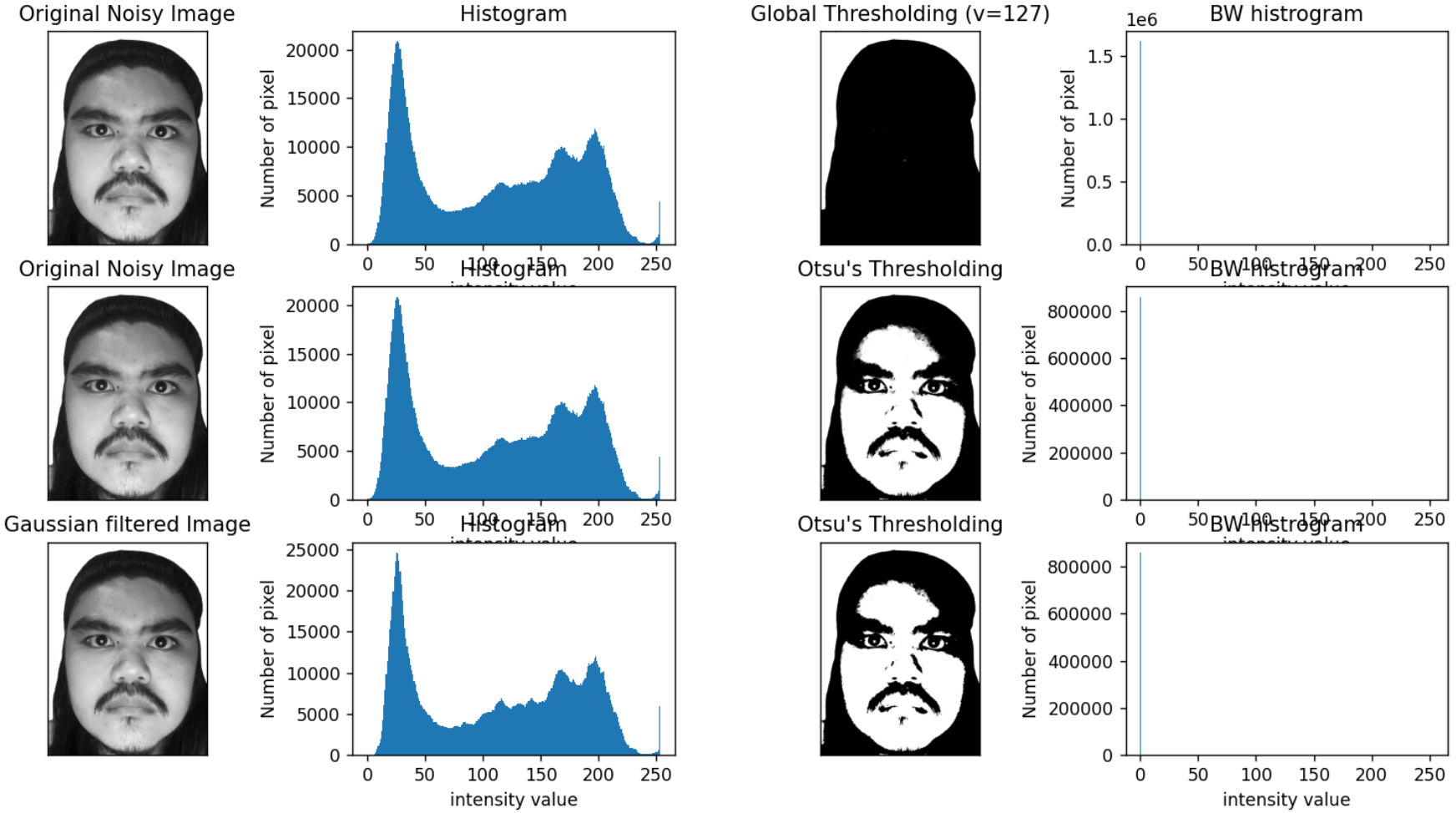
การทดลองที่ 1.3

1. ให้นักศึกษาถ่ายภาพใบหน้าของตัวเองโดยพยายามไม่ให้มีบริเวณพื้นหลังภายในภาพ “MyFacePic3.jpg”

คำสั่ง

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
img = cv.imread('D:\Telecom_Lab\Lab4\pic\MyFacePic3.jpg',0)
ret1,th1 = cv.threshold(img,240,255,cv.THRESH_BINARY)
ret2,th2 = cv.threshold(img,0,255,cv.THRESH_BINARY+cv.THRESH_OTSU)
blur = cv.GaussianBlur(img,(5,5),0)
ret3,th3 = cv.threshold(blur,0,255,cv.THRESH_BINARY+cv.THRESH_OTSU)
images = [img, 0, th1,img, 0, th2,blur, 0, th3]
titles = ['Original Noisy Image','Histogram','Global Thresholding (v=127)','Original Noisy
Image','Histogram',"Otsu's Thresholding",'Gaussian filtered Image','Histogram',"Otsu's
Thresholding"]
for i in range(3):
 plt.subplot(3,4,i*4+1),plt.imshow(images[i*3],'gray')
 plt.title(titles[i*3]), plt.xticks([]), plt.yticks([])
 plt.subplot(3,4,i*4+2),plt.hist(images[i*3].ravel(),253,[0,253])
 plt.title(titles[i*3+1]),plt.ylabel('Number of pixel'),plt.xlabel('intensity value')
 plt.subplot(3,4,i*4+3),plt.imshow(images[i*3+2],'gray')
 plt.title(titles[i*3+2]), plt.xticks([]), plt.yticks([])
 plt.subplot(3,4,i*4+4),plt.hist(images[i*3+2].ravel()253,[0,253])
 plt.title('BW histrogram'), plt.ylabel('Number of pixel'),plt.xlabel('intensity value')
plt.show()
```

การแสดงผล


Original Noisy Image
Histogram
Number of pixel
Global Thresholding (v=127)
BW histrogram
1e6
Original Noisy Image
Histogram
Otsu's Thresholding
BW histrogram
Gaussian filtered Image
Histogram
Otsu's Thresholding
BW histrogram
intensity value

อภิปราย

เฉกเช่นเดียวกับข้อ 1.1 และ 1.2 แต่ในข้อนี้สิ่งที่แตกต่างกับข้อก่อนหน้าคือการเก็บรายละเอียดของใบหน้าที่ทำด้วยวิธี Otsu’s Threshold นั้นมีความละเอียดมากกว่าข้อก่อนๆนั่นเอง และการเลือกค่า x เราเลือกค่า 240 เนื่องจาก ค่ามากที่สุดบริเวณใบหน้าคือ 239 ถ้าถามว่าเราทราบได้อย่างไร เราใช้วิธี Random ค่า จงกระทั่ง จุดขาวของภาพ ที่ผ่านการ Global Thesholding ในส่วนบริเวณใบหน้าไม่เกิดขึ้นอีก

Global Thresholding (v=127)

จากภาพ **ค่า x เท่ากับ 239** จะเห็นว่าบริเวณใบหน้ามีจุดขาวเล็กๆ ซึ่งถ้าไม่สังเกตดีๆ เราแทบจะมองไม่เห็นเลย แต่เวลาเราทำผ่านคอมพิวเตอร์ แน่นอนว่าไม่รอดพ้นสายตาของคอมพิวเตอร์แน่นอน ดังนั้นเราจึงกำหนดค่า xให้เกิน 239นั่นเอง

Global Thresholding (v=127)

จากภาพ **ค่า x เท่ากับ 240** เท่านี้ภาพเราก็จะแบ่งระหว่างพื้นหลังและรูปใบหน้าออกจากกันอย่างชัดเจนแล้ว

จากการทดลองทั้ง 3 แบบให้นักศึกษาทำการ

1. อธิบายการทำงานของคำสั่ง cv.threshold()

จากที่เราได้เรียนรู้มา การทำ Thresholding คือการแยกบริเวณที่เราสนใจออกจากพื้นหลัง ซึ่งหลังจากที่เราทำการทดลองในข้อ 1 นั้นทำให้เราสามารถอธิบายได้ว่า cv.threshold() เป็นคำสั่งที่ทำการแบ่งจุดที่สนใจออกจากพื้นหลัง โดยค่าต่ำกว่าค่า Threshold จะถูกกำหนดให้เป็นสีดำ และค่าที่มากกว่าจะถูกกำหนดให้เป็นสีขาว และตัวแปรภายในของคำสั่ง(Argument)นั้น ได้แก่

1.1 img รูปภาพที่เราต้องการทำ Thresholding

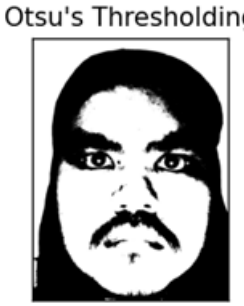
1.2 x หรือ Threshold Value เป็นค่าที่เราต้องกำหนดค่าbins หรือค่าความเข้ม ซึ่งเป็นเส้นแบ่งเขตแดนระหว่างจุดที่สนใจและพื้นหลัง

1.3 255 หรือ Max Value เป็นค่าที่เราต้องกำหนดซึ่งเป็นค่าสูงสุดที่ต้องการจะแสดง

1.4 cv.THRESH_BINARY หรือ Thresholding Technique เป็นการกำหนดว่าจะทำใช้เทคนิคไหนในการทำ Thresholding

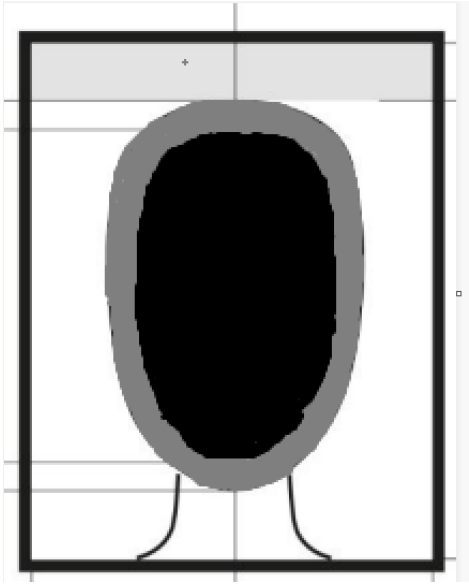
2.สรุปและเปรียบเทียบผลลัพธ์ที่ได้ว่ารูปภาพแบบไหนที่ใช้วิธีการตรวจจับวัตถุแบบ Otsu thresholding แล้วได้ผลลัพธ์ดีที่สุด

จากการทดลองที่ผ่านมา เราจะสังเกตว่า Otsu’s thresholding method นั้นจะเป็นการเก็บรายละเอียดของส่วนใบหน้าได้ดีกว่า Global thresholding method และภาพที่เก็บรายละเอียดของใบหน้าได้ดีที่สุดคือ ภาพของการทดลองที่ 1.3 จากภาพด้านล่างนี้จะเห็นว่าภาพที่เห็นใบหน้าชัดเจน(การทดลองที่1.3) การทำThresholdด้วยวิธี Otsu สามารถเห็นได้กระทั่งดวงตาอย่างชัดเจน ในขณะที่ภาพอื่นๆเห็นเพียงรางๆ และภาพที่เห็นได้น้อยที่สุดเลยคือ ภาพที่เกิดจากการที่1.2



การทดลองที่ 2 การตรวจจับวัตถุด้วยวิธี Supervised range-constrained thresholding

1. ให้นักศึกษาหาอัตราส่วนสูงสุดและต่ำสุดของใบหน้าต่อบริเวณรูปตามกรอบกำหนดมาตรฐานรูปภาพใบหน้า (รูปที่ 2) โดยที่ค่าแรก FacePortion_UpperBound เป็นค่าของขนาดใบหน้าที่สามารถมี สัดส่วนใหญ่ที่สุดตามแนวเส้นหนาดำและค่าที่สอง FacePortion_LowerBound เป็นค่าของขนาด ใบหน้าที่สามารถมีสัดส่วนในรูปน้อยที่สุดตามแนวเส้นหนาประ เช่นนำรูปที่ 2 นี้ระบายสีด้วยโปรแกรม ช่วย (ตัวอย่างเช่น Microsoft Paint) โดยแบ่งพื้นหลังเป็นสีขาว ขอบใบหน้าตามแนวเส้นหนาดำถึงเส้น หนาประเป็นสีเทา และภายในขอบใบหน้าตามแนวเส้นหนาประเป็นสีดำ เป็นต้น



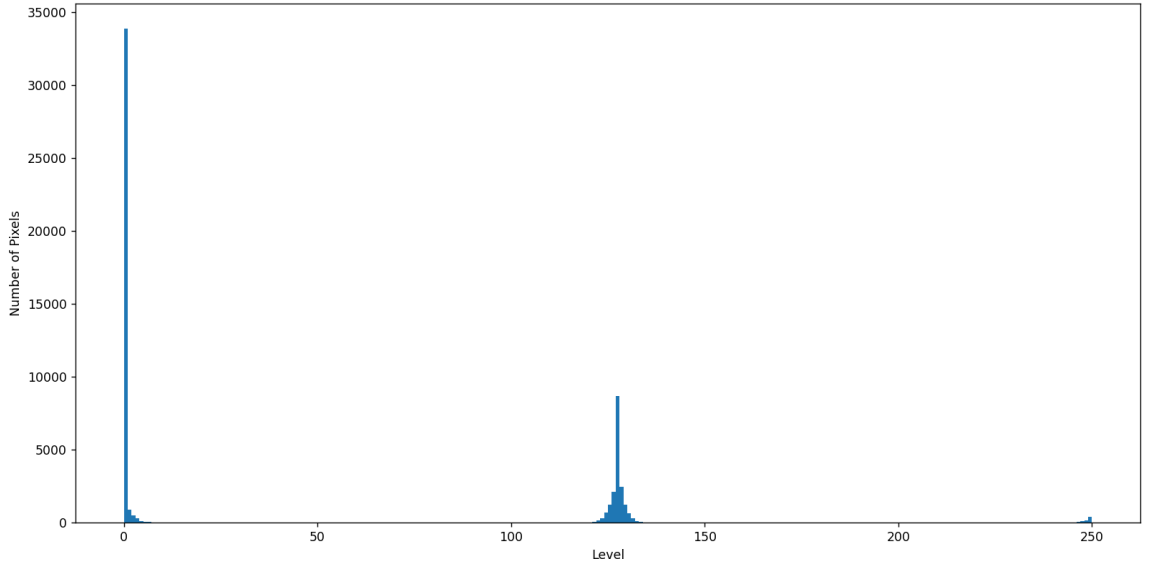
รูปภาพขนาด มี Pixel ประมาณ 89512 px

ค่า FacePortion_UpperBound (สีเทา) มีค่า Pixel โดยประมาณ 8,664.3+33,887=42,551.3 px

(RGB value = 127) ซึ่งคิดเป็น 47.53% ของรูปภาพ

ค่า FacePortion LowerBound (สีดำ) มีค่า Pixel โดยประมาณ 33,887 px (RGB value = 0)

ซึ่งเป็น 37.86% ของรูปภาพ สามารถดูได้จากHistogramของรูปภาพ (ในที่นี้ไม่ได้พิจารณาพื้นหลัง)



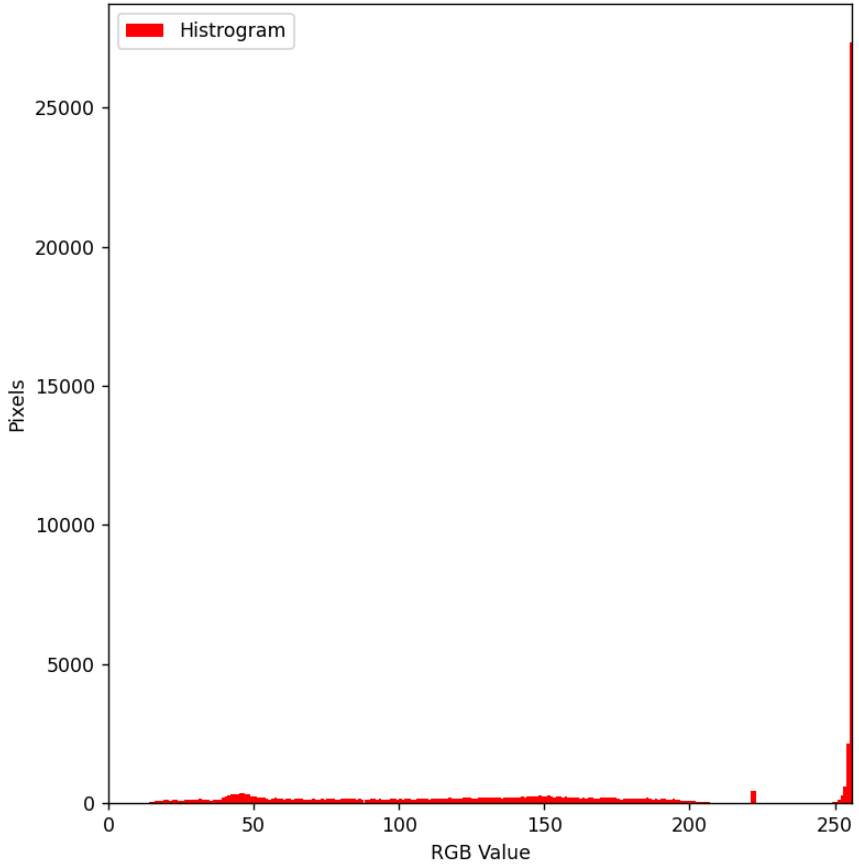
2. ให้นักศึกษาทำการถ่ายรูปของนักศึกษาหน้าตรง สีพื้นหลังสีขาวโทนสีเดียว ลักษณะแสงที่ตกกระทบ ใบหน้าเท่ากันทั่วบริเวณใบหน้าเพื่อให้ใบหน้าตัดกับพื้นหลังอย่างชัดเจน ทำการ resize ภาพให้มีขนาด ที่เหมาะสมในการ crop ภาพโดยจะต้องมีสัดส่วนใบหน้าเหมือนเดิม ไม่เพี้ยนไม่เบี้ยวไปจากเดิม แล้ว ทำการ crop รูปใบหน้าของตัวเอง “MyFacePicSVRC.jpg” โดยให้มีขนาดและสัดส่วนของใบหน้าต่อ บริเวณรูปภาพดังกรอบกำหนดมาตรฐาน (รูปที่ 2)

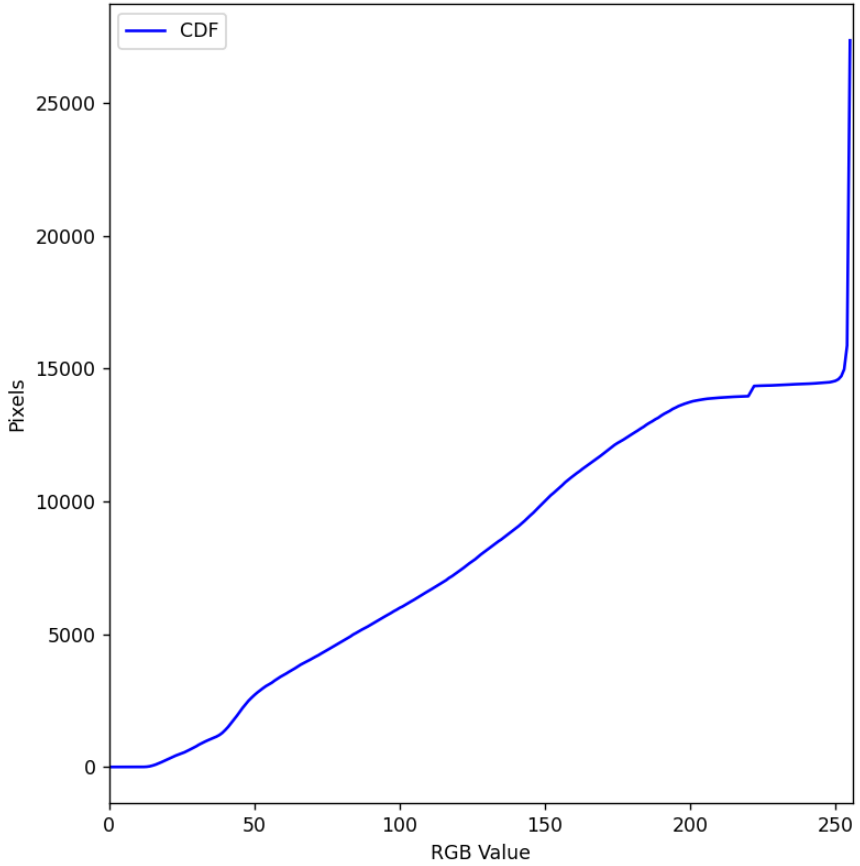
3. สร้าง normalized cumulative distribution function ของรูป “MyFacePicSVRC.jpg” จากฮิสโตแกรม แล้วระบุค่าความเข้ม Threshold_LowerBound และ Threshold_UpperBound ที่สอดคล้องกับค่า FacePortion_LowerBound และ FacePortion_UpperBound ที่ได้จากข้อที่แล้ว ลงบนกราฟ normalized cumulative distribution function

คำสั่ง

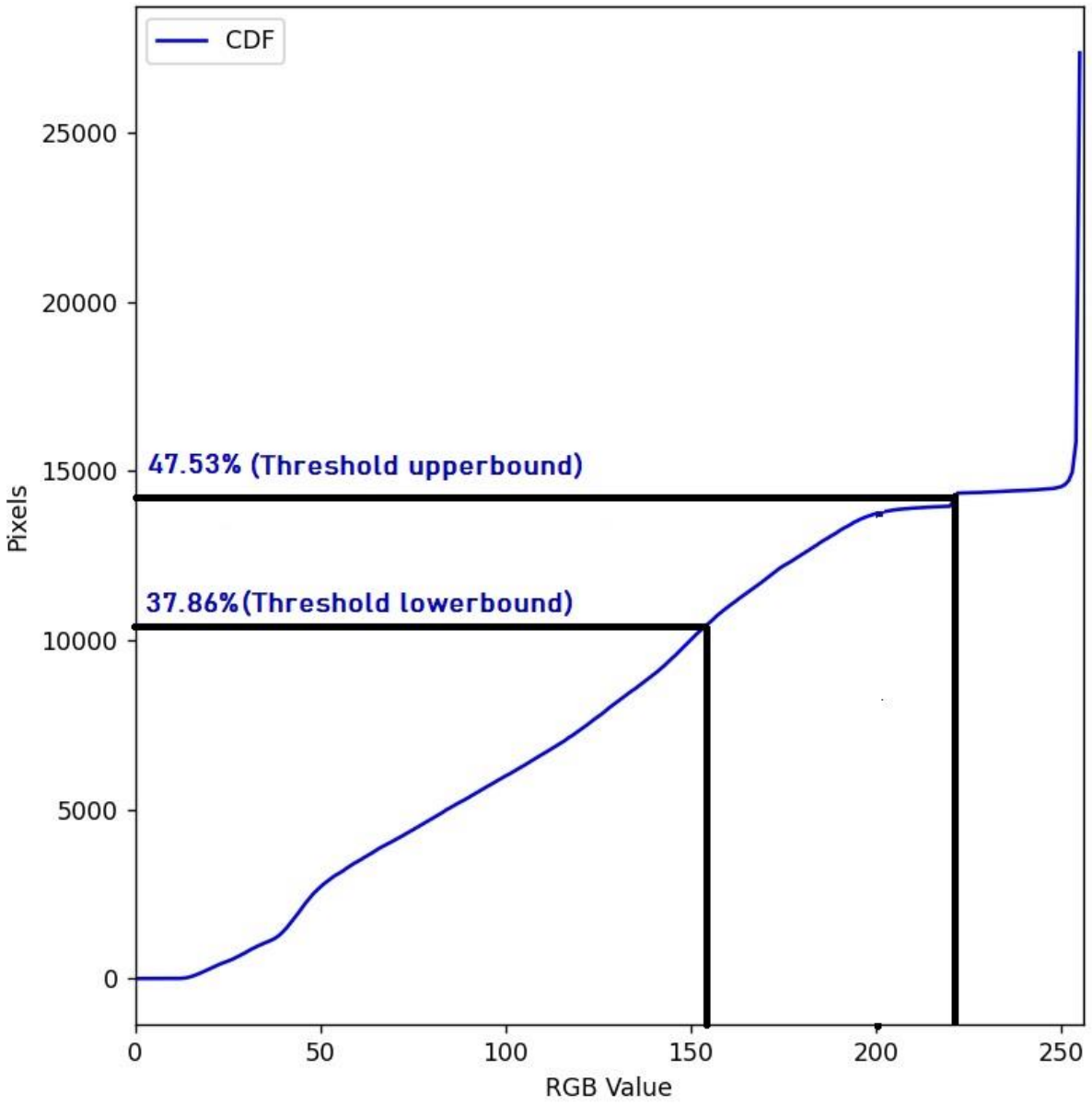
```
import numpy as np
import cv2
import matplotlib.pyplot as plt

img=cv2.imread('D:\Telecom_Lab\Lab4\pic\MyFacePicSVRC.jpg',0)

hist,bins=np.histogram(img.ravel(),256,[0,256])
cdf=hist.cumsum()
cdf_normalized=cdf*float(hist.max())/cdf.max()
plt.figure()
plt.subplot(121)
plt.hist(img.ravel(),256,[0,256],color='r')
plt.xlim([0,256])
plt.legend(('Histrogram','histogram'),loc='upper left')
plt.xlabel('RGB Value')
plt.ylabel('Pixels')

plt.subplot(122)
plt.plot(cdf_normalized,color='b')
plt.xlim([0,256])
plt.legend(('CDF','histogram'),loc='upper left')
plt.xlabel('RGB Value')
plt.ylabel('Pixels')

plt.show()
```

การแสดงผล


Histrogram
25000
20000
15000
10000
5000
0
Pixels
0
50
100
150
200
250
RGB Value


CDF
25000
20000
15000
10000
5000
0
Pixels
0
50
100
150
200
250
RGB Value

พิจารณาหาค่า Threshold_lowerbound และ Threshold_Upperbound ที่สอดคล้องกับ FacePortion_lowerBound และ FacePortion_UpperBound จากกราฟ cumulative distribution function

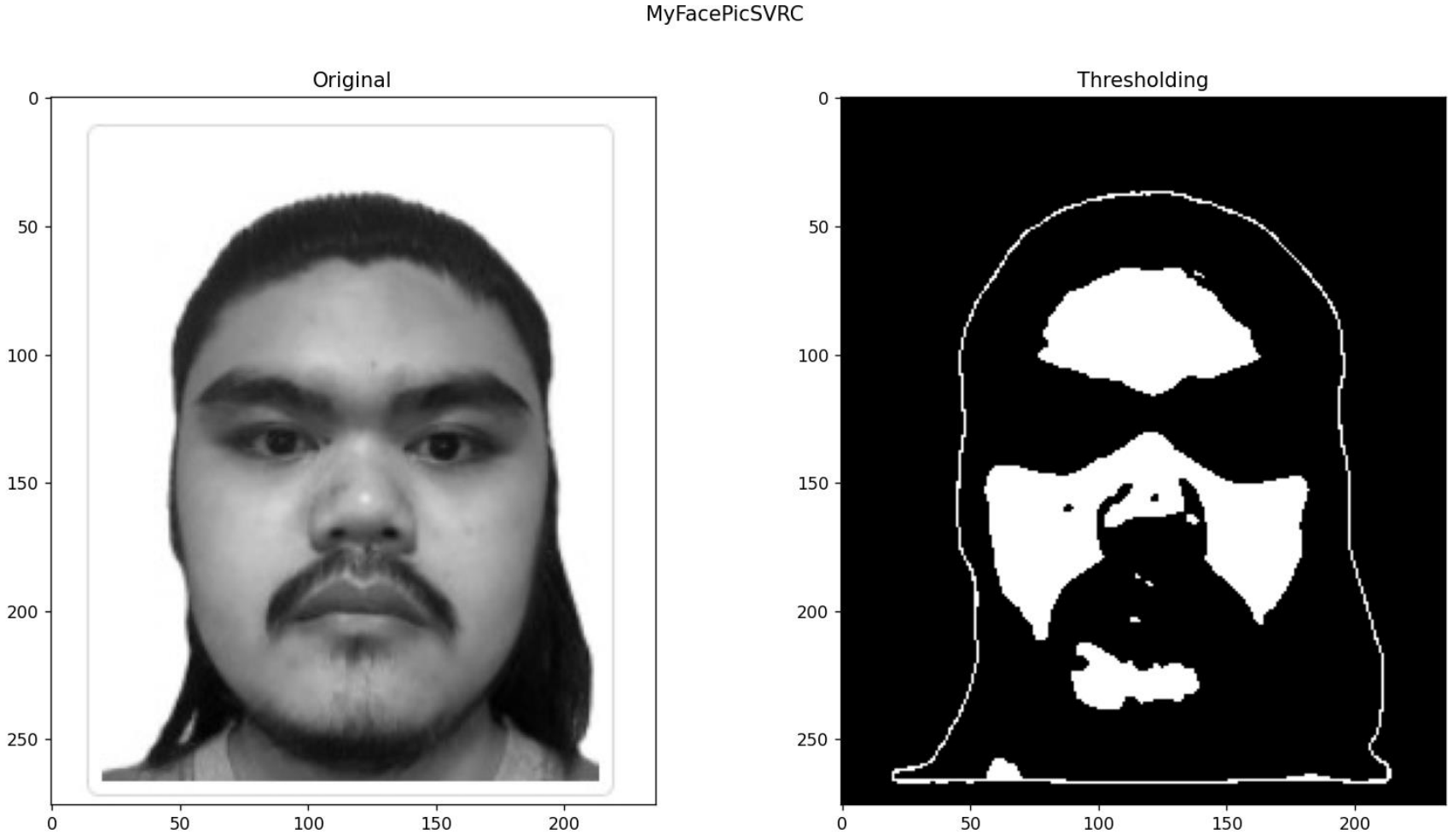


4. หลังจากที่ได้ค่า Threshold_LowerBound และ Threshold_UpperBound ให้ทำการใช้คำสั่งหาค่า Threshold สำหรับการแยกใบหน้าออกจากพื้นหลัง

คำสั่ง

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
img=cv2.imread('D:\Telecom_Lab\Lab4\pic\MyFacePicSVRC.jpg',0)
plt.suptitle('MyFacePicSVRC')
plt.subplot(121)
plt.title('Original')
plt.imshow(img,cmap='gray')

blur=cv2.GaussianBlur(img,(5,5),0)
#up 220 low 152
mask=cv2.inRange(blur,152,212)
ret,thresh=cv2.threshold(mask,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
plt.subplot(122)
plt.title('Thresholding')
plt.imshow(thresh,cmap='gray')
plt.show()
```

การแสดงผล



5. สังเกตผลเปรียบเทียบระหว่างผลการทดลองที่ 1 และ 2 บันทึกผลการทดลอง

จากการทดลองที่1 จะพบว่าภาพที่สามารถบอกรายละเอียดที่ดีที่สุดที่ผ่าน Otsu's Thresholding คือภาพที่เกิดจากการถ่ายระยะใกล้ (ภาพที่พื้นหลังน้อยที่สุด) จากภาพจะเห็นว่าสามารถระบุได้ละเอียดมาก ซึ่งแตกต่างจากการทำด้วยวิธี Global Thresholding ที่แยกได้แค่รอยแยกระหว่างจุดที่สนใจกับพื้นหลังเท่านั้น ไม่สามารถบอกรายละเอียดส่วนต่างๆของจุดที่สนใจได้ ทั้งนี้ในการทดลองที่ 2 จะสังเกตเห็นว่าภาพที่ได้มีความละเอียดใกล้เคียงกับภาพที่ 3 จากการทดลองที่1 อาจจะมีบางส่วนที่หายไปบ้างเป็นเพราะเกิดจากขั้นกำหนดค่า threshold upperbound และ threshold lowerbound ที่ไม่ครอบคลุมทุกส่วนบนใบหน้า ซึ่งต่างจากการทดลองที่1ที่ไม่ได้กำหนดค่าให้กับวิธี Otsu แต่เนื่องจากการทดลองที่2 เป็นการทดลองที่นำไปใช้งานจริงได้ สำหรับผู้ที่ต้องการเดินทางระหว่างประเทศที่จำเป็นต้อง VISA และ Passport สำหรับเดินทาง จะพบว่าวิธีOtsuสามารถแยกรายละเอียดของใบหน้าได้ดี แม้จะไม่ครบทุกส่วน อันเป็นผลมาจากการถ่ายภาพหรือแสงที่ตกกระทบใบหน้าอาจทำให้ผิดเพี้ยนไป รวมทั้งการกำหนดค่า FacePortion ที่อาจจะเกิดค่าความคาดเคลื่อน และส่งผลให้เกิดค่า Absolute Errors ได้ภายหลัง ดังนั้นในการทดลองที่2 ภาพที่ได้แม้จะแยกรายละเอียดได้น้อยกว่าภาพจากการทดลองที่1.3 แต่ก็มีความละเอียดใกล้เคียงกัน สามารถแยกดวงตา ปาก จมูก รวมทั้งผม ของผู้ทดลองได้

ทั้งนี้การถ่ายภาพให้ตรงตามเงื่อนไขของสถานทูตเป็นสิ่งจำเป็นและสมควรปฏิบัติตาม แต่เนื่องจากผู้ทำการทดลองไม่ได้ชำนาญการ อีกทั้งยังเป็นเพียงการทดลองเพื่อให้เห็นถึงความแตกต่าง ข้อผิดพลาด และการแก้ไขในอนาคตต่อไป