รหัสวิชา: 010113337 ชื่อวิชา: ปฏิบัติการระบบโทรคมนาคม
ชื่อ: _____รหัสนักศึกษา: _____
ภาคการศึกษาที่: ___ ปีการศึกษา: _____
วันที่และเวลาที่ทำการทดลอง: _____
อาจารย์ผู้สอน: ผู้ช่วยศาสตราจารย์ ดร. ไกรสร ไชยซาววงค์

วัตถุประสงค์

1. ทำความรู้จักกับคำสั่งพื้นฐานในโปรแกรม Python ที่เกี่ยวข้องกับการประมวลผลรูปภาพดิจิทัล
2. การสร้างตัวอย่างย่อย (subsampling) ของรูปภาพดิจิทัล
3. การหาขอบในรูปภาพดิจิทัล (Edge detection)
4. เวลาที่ใช้ในการประมวลผลรูปภาพดิจิทัลขนาดต่างๆ

ขั้นตอนการทดลอง

<u>การทดลองที่ 2.1</u>

1. ให้นักศึกษาเขียนข้อมูลภาพใหม่ จากรูปภาพใบหน้าของตนเองจากปฏิบัติการที่ 1 โดยตั้งชื่อไฟล์ว่า **MyCompressedFace.jpeg** โดยใช้มาตรฐานการบีบอัดรูปภาพดิจิทัลแบบ jpeg ให้เหลือคุณภาพเพียง 33%
2. จงนำเสนอรูปต้นแบบและรูปที่ถูกบีบอัดบนหน้าต่างเดียวกันพร้อม โดยกำหนดให้ title แสดงชื่อของนักศึกษา พร้อมคุณลักษณ์ขนาดของแต่ละภาพ (size และ dimension) แสดงบน title ด้วย
3. ให้นักศึกษาแสดงฮิสโตแกรมของรูป **fullSizeImage** และรูป **MyCompressedFace.jpeg** หน้าต่างเดียวกัน

<u>การทดลองที่ 2.2</u>

1. ให้นักศึกษาทำการเปลี่ยนขนาดรูป `'cameraman.jpg'` โดยให้มีขนาดเล็กลง 10 เท่า แล้วแสดงรูป **fullSizeImage** และรูป **subsampledImage** ทั้งสองรูปใน graphic window เดียวกันโดยปรับขนาดให้เห็นผลการ subsample รูป
2. ให้นักศึกษาแสดงฮิสโตแกรมของรูป **fullSizeImage** และรูป **subsampledImage** โดยให้ความยาวของ grayscale bar เท่ากับ 64 bins หน้าต่างเดียวกัน

การทดลองที่ 2.3

1. ให้นักศึกษาดำเนินคำสั่งต่อไปนี้

```
import cv2
from matplotlib import pyplot as plt
img = cv2.imread('cameraman.jpg' , 0)

# set gaussian filter SD and factor of threshold
sigma = 2
factor = 0.75

# blur the image first with gaussian blur using GaussianBlur() function of cv2,
with kernel size of (7,7) and set sigmaX to our sigma value
smoothedInput = cv2.GaussianBlur(img,(7,7),sigmaX=sigma)

# here use threshold() function of cv2 to find optimal threshold for image using
Otsu’s binarization, thresholding algorithm can be specified by passing these flags
to the function “cv.THRESH_BINARY + cv.THRESH_OTSU” for Otsu’s binarization, this
function will return 2 values which are the calculated threshold and a thresholded
input image
ret, otsu = cv2.threshold(smoothedInput, 0, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)

# Canny edge detection with Canny() function of cv2, setting lower and higher
threshold, use lower threshold of 0.4 times
edges = cv2.Canny(smoothedInput, ret * 0.4 * factor, ret * factor)

plt.subplot(121)
plt.imshow(img , cmap = 'gray')
plt.title('Original Image')
plt.subplot(122)
plt.imshow(edges , cmap = 'gray')
plt.title('Edges')
plt.show()
```

2. ให้นักศึกษาใช้คำสั่งที่เหมาะสม**จับเวลา**การประมวลผลการหาขอบในข้อ 1.
3. ให้นักศึกษา subsample รูปภาพจากข้อ 1. แล้วหาขอบด้วยวิธี Canny เหมือนข้อ 1. โดยกำหนดให้ **subsampleRate = 5** พร้อมจับเวลาการประมวลผลการหาขอบ
4. ให้นักศึกษานำรูปใบหน้าของตนเองจากการทดลองที่ 2.1 ทั้งสองรูปมาหาขอบและจับเวลา