

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
วิชา Image Processing Laboratory

**การทดลองที่ 1 : พื้นฐานการอ่านเขียน แสดงผล และทดสอบการดำเนินการตัดวัตถุที่สนใจด้วย  
คุณสมบัติสีอย่างง่าย และการแทนค่าสีใหม่กับรูปภาพ**

**วัตถุประสงค์**

1. เพื่อศึกษาและทดลองการอ่านและเขียนไฟล์รูปภาพ
2. เพื่อศึกษาโครงสร้างข้อมูลภาพ
3. เพื่อศึกษาและทดลองการแสดงผลรูปภาพบนแกนสองมิติและสามมิติ
4. เพื่อศึกษาและทดลองกำหนดตารางสีเพื่อแสดงสีของรูปภาพตามค่าสีที่กำหนด
5. เพื่อศึกษาและแสดงผลของการแสดงรูปภาพในองค์ประกอบสีแบบต่างๆ
6. เพื่อศึกษาและทดลองสร้างฟังก์ชันในการปรับระดับความเข้มแสงของภาพตามระดับที่กำหนด

**อุปกรณ์ และเครื่องมือที่ใช้ในการทดลอง**

1. โปรแกรม python

**ข้อกำหนดในการตรวจการทดลอง**

1. แสดงโค้ดและภาพผลการทดลองที่ทำพร้อมอธิบาย
2. นศ.ที่ได้รับการตรวจจากอาจารย์เรียบร้อยแล้ว อาจารย์จะเช็คส่งงานในระบบ
3. นศ.ทุกคนให้ตอบคำถามท้ายการทดลองใน <https://forms.gle/L4gNFsQMN1boFtd28>
4. ให้นศ. นำภาพ figure ที่ให้แสดงทุกภาพ โฟสลง facebook group พร้อมชื่อกลุ่ม รหัสสนศ.และชื่อสมาชิกในกลุ่ม ถ้าการทดลองใดมีการปรับค่า ให้แสดงค่าที่เลือกใช้สำหรับผลลัพธ์ภาพนั้นๆ ด้วย รวมทั้งวางภาพผลลัพธ์ลงในเอกสารการทดลองใน google classroom ตามหัวข้อย่อยที่ให้แสดงภาพผลลัพธ์ส่งภายในวันที่ 25 สิงหาคม 2562 เวลา 18.00 น.

## ตอนที่ 1: การทดลองอ่านไฟล์ภาพ การปรับเปลี่ยนภาพสีเป็นภาพเฉดเทา (Grayscale) และแสดงภาพ

1.1 Import Lib (cv2, numpy, math)

1.2 ฟังก์ชันใช้งาน

`mpl_toolkits.mplot3d import Axes3D`

`matplotlib import cm`

`matplotlib import pyplot`

`matplotlib.ticker import LinearLocator, FormatStrFormatter`

1.3 อ่านไฟล์ภาพที่เตรียมมา

1.4 ปรับภาพเป็นภาพ Grayscale โดยใช้ฟังก์ชัน `cvtColor()` ใน `cv2 lib`

1.5 แสดงค่าข้อมูลภาพ ได้แก่

Image dimension (size: height x width)

image data type (uint8, uint16, int8, int16, double, single)

Image statistics (separable histogram R, G, B)

1.6 สร้างภาพผลลัพธ์จากการลดจำนวนบิตต่อพิกเซล (Bit depth) ด้วยเทคนิค quantization ของภาพ Grayscale ในข้อ 1.4 ตามฟังก์ชันคณิตศาสตร์ด้านล่าง จาก Bit depth จากไฟล์ภาพ ให้เหลือ  
Bit depth = 4

$$Q = \text{floor} \left( \frac{\text{input} - \text{MIN}}{Q\text{step}} \right)$$
$$Q\text{step} = \frac{\text{MAX} - \text{MIN}}{Q\text{level}}$$
$$Q\text{level} = 2^n; n = Q\text{bits} = \text{Bit Depth}$$

1.7 แสดงผลภาพ Figure 1

1.7.1 แสดงภาพอินพุทจากข้อ 1.3 ด้วยคำสั่ง `imshow()` ในตำแหน่ง แถวที่ 1 คอลัมน์ที่ 1 ด้วยคำสั่ง `subplot()`

1.7.2 แสดงภาพอินพุทจากข้อ 1.4 ด้วยคำสั่ง `imshow()` ในตำแหน่ง แถวที่ 1 คอลัมน์ที่ 2 ด้วยคำสั่ง `subplot()`

1.8 แสดงผลภาพ Figure 2

1.8.1 แสดงภาพจากข้อ 1.4 ในตำแหน่ง แถวที่ 1 คอลัมน์ที่ 1 พร้อมแสดงตารางสี ด้วยคำสั่ง `colorbar()`

1.8.2 แสดงภาพจากข้อ 1.4 อีกครั้ง ด้วย `colormap` ที่ให้สีที่แตกต่างกับรูปก่อนหน้า ในตำแหน่ง แถวที่ 1 คอลัมน์ที่ 2 พร้อมแสดงตารางสี

1.8.3 แสดงภาพจากข้อ 1.6 ในตำแหน่ง แถวที่ 2 คอลัมน์ที่ 1 พร้อมแสดงตารางสี

1.8.4 แสดงภาพจากข้อ 1.6 อีกครั้ง ด้วย `colormap` ที่ให้สีเดียวกับข้อ 1.6.2 ในตำแหน่ง แถวที่ 2 คอลัมน์ที่ 2 พร้อมแสดงตารางสี

- 1.9 แสดงผลภาพ Figure 3 เพื่อแสดง surface ของข้อมูล 3 มิติ ของภาพ Grayscale จากข้อ 1.4
  - 1.9.1 กำหนดสเกลของแกน x, y ตาม dimension ของภาพที่ต้องการแสดง ด้วยคำสั่ง `mgrid()`
  - 1.9.2 กำหนดค่าแกน z เป็นค่า intensity ของภาพที่ต้องการแสดง
  - 1.9.3 แสดงภาพ 3D surface ของภาพที่ต้องการแสดง ด้วยคำสั่ง `plot_surface()` โดยเลือกสีให้เห็นข้อมูลภาพ 3D ชัดเจนใกล้เคียงกับการมองเห็นภาพ 2D ของภาพที่ต้องการแสดง
- 1.10 โปสภาพผลลัพธ์ Figure 1-3 ลงใน Facebook Group ตรวจสอบลำดับ

Note: เพื่อเปิด window แสดงภาพขึ้นมา ต้องใส่ `cv2.waitKey(0)` ด้วย เพื่อไม่ให้ python เปิดมาแสดงภาพแล้วปิดตัวไปทันที โดยผู้ใช้งานยังไม่ได้เห็นภาพ

---

(อาจารย์ตรวจผลการทดลอง)

## ตอนที่ 2: การทดลองกำหนดสีให้กับวัตถุที่มีเงื่อนไขตามที่กำหนดและกำหนดเฉดสีเทาให้กับพื้นหลัง

- 2.1 แสดงภาพ Figure 4 จากข้อ 1.3 ในตำแหน่ง แฉกที่ 1 คอลัมน์ที่ 1
- 2.2 แสดงภาพวัตถุที่สนใจ (ROI: Region-Of-Interest) ในตำแหน่ง แฉกที่ 1 คอลัมน์ที่ 2 โดยกำหนดให้ทำการแทนสีภาพวัตถุที่สนใจที่อยู่ในช่วงของสีที่ต้องการ ด้วยสีเดียวกันทุกตำแหน่งภายในวัตถุ โดยให้กำหนดสีตัวแทนวัตถุเป็นสีที่โดดเด่นชัดเจน เพื่อให้ทราบว่าเป็นตำแหน่งสนใจ และตำแหน่งที่ไม่ใช่วัตถุสนใจให้เป็นสีเฉดเทา
- 2.3 ทดลองเปลี่ยนภาพสี RGB เป็นภาพสีโมเดล YCbCr แสดงภาพที่ได้ ในตำแหน่ง แฉกที่ 2 คอลัมน์ที่ 1
- 2.4 ทดลองเปลี่ยนภาพสี RGB เป็นภาพสีโมเดล HSV แสดงภาพที่ได้ ในตำแหน่ง แฉกที่ 2 คอลัมน์ที่ 2
- 2.5 save ภาพผลลัพธ์ โดยใช้ฟังก์ชัน `imsave` หรือ `imwrite` Figure 4 เป็นไฟล์ .jpg และ post ใน Facebook group พร้อม แสดงช่วงสีของวัตถุสนใจที่เลือก

---

(อาจารย์ตรวจผลการทดลอง)

## ตอนที่ 3 ทำคำถามท้ายการทดลองใน google form (<https://forms.gle/L4gNFsQMN1boFtd28>)

## Tutorial

1) <https://www.anaconda.com/download/>

2) <https://pypi.org/project/opencv-python/>

3) [https://matplotlib.org/users/image\\_tutorial.html](https://matplotlib.org/users/image_tutorial.html)

4) [http://opencv-python-](http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_histograms/py_histogram_begins/py_histogram_begins.html#histograms-getting-started)

[tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_histograms/py\\_histogram\\_begins/py\\_histogram\\_begins.html#histograms-getting-started](http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_histograms/py_histogram_begins/py_histogram_begins.html#histograms-getting-started)

5) [https://matplotlib.org/examples/pylab\\_examples/subplot\\_toolbar.html](https://matplotlib.org/examples/pylab_examples/subplot_toolbar.html)

6) [https://docs.opencv.org/3.4.0/d7/d4d/tutorial\\_py\\_thresholding.html](https://docs.opencv.org/3.4.0/d7/d4d/tutorial_py_thresholding.html)