

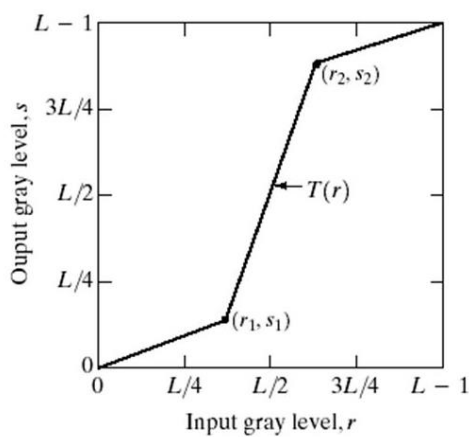
Homework #1

Deadline: **September 7, 2021 @23:59**

Submissions: (1) PDF version of this file



(2) Python files (homework1_1_<ID>.py, homework1_2_<ID>.py, homework1_3_<ID>.py)

1. Write a program in python into `homework1_1()` function in `homework1_1_<ID>.py` file to implement contrast stretching follow the transformation in the graph below.

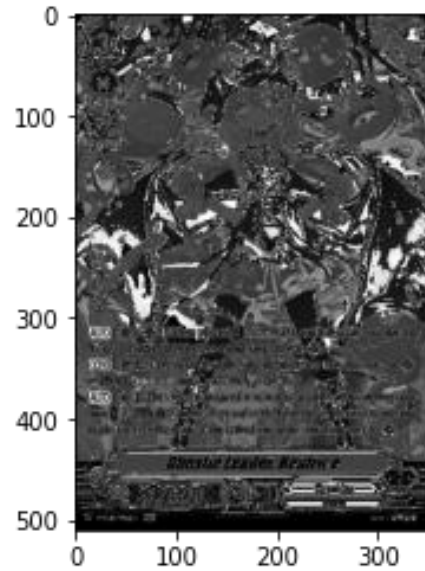


Test your program using kitty.jpg and your own image and display your results in the blank below.

Results of the processed images:

Original	After filtering
	

Your image:



Your code:

homework1_1_<ID>.py

```
# import library here
import cv2
import matplotlib.pyplot as plt

def homework1_1(image_grayscale):
    # input -> image_grayscale - type -> np.ndarray, size of - (height,
    width)
    # output -> image_grayscale - type -> np.ndarray, size of - (height,
    width)

    # TO DO - Implement transformation based on the contrast stretching
```

```

graph
Image = image_grayscale.copy();
r1 = 85.3;
s1 = 32;
r2 = 160;
s2 = 224;
for i in range(image.shape[0]):
    for j in range(image.shape[1]):
        x = image[i][j];
        image[i][j] = (x<=r1)*(x*s1/r1) + (x>r1 and x<=r2)*(x*(s2-
s1)/(r2-r1)) + (x > r2)*(x*(255-s2)/(255-r2))
    filtered_image = image;

    return filtered_image

```

2. Design your own filter on an RGB image. Write your code in **homework1_2()** function in **homework1_2.py** file Provide motivation behind the designed filter. Display it in terms of an RGB image.

Idea / Motivation:

อยากลองทำ filter ที่ทำให้สีทั้งหมด เหลือเพียงเจ็ดสี หลัก คือ R, G, B, R+G, R+B, G+B, R+G+B เพราะอยากเห็นภาพที่ดูมีความ retro ผสมความเรียบง่าย อีกทั้งยังอยากเห็นความสำคัญของ level ของแต่ละสี ที่สามารถมีค่าได้ตั้งแต่ 0-255 ว่ามีความสำคัญเพียงใด

Your filter design (at least two equations and/or conditions):

สมมติให้ output level เป็น s และ input level เป็น r จะได้ว่า

$s(\text{สีน้ำเงิน}) = 255$ เมื่อ $x(\text{สีน้ำเงิน})$ มีค่า ≥ 128

$= 0$ เมื่อ $x(\text{สีน้ำเงิน})$ มีค่า < 128



$s(\text{สีเขียว}) = 255$ เมื่อ $x(\text{สีเขียว})$ มีค่า ≥ 128

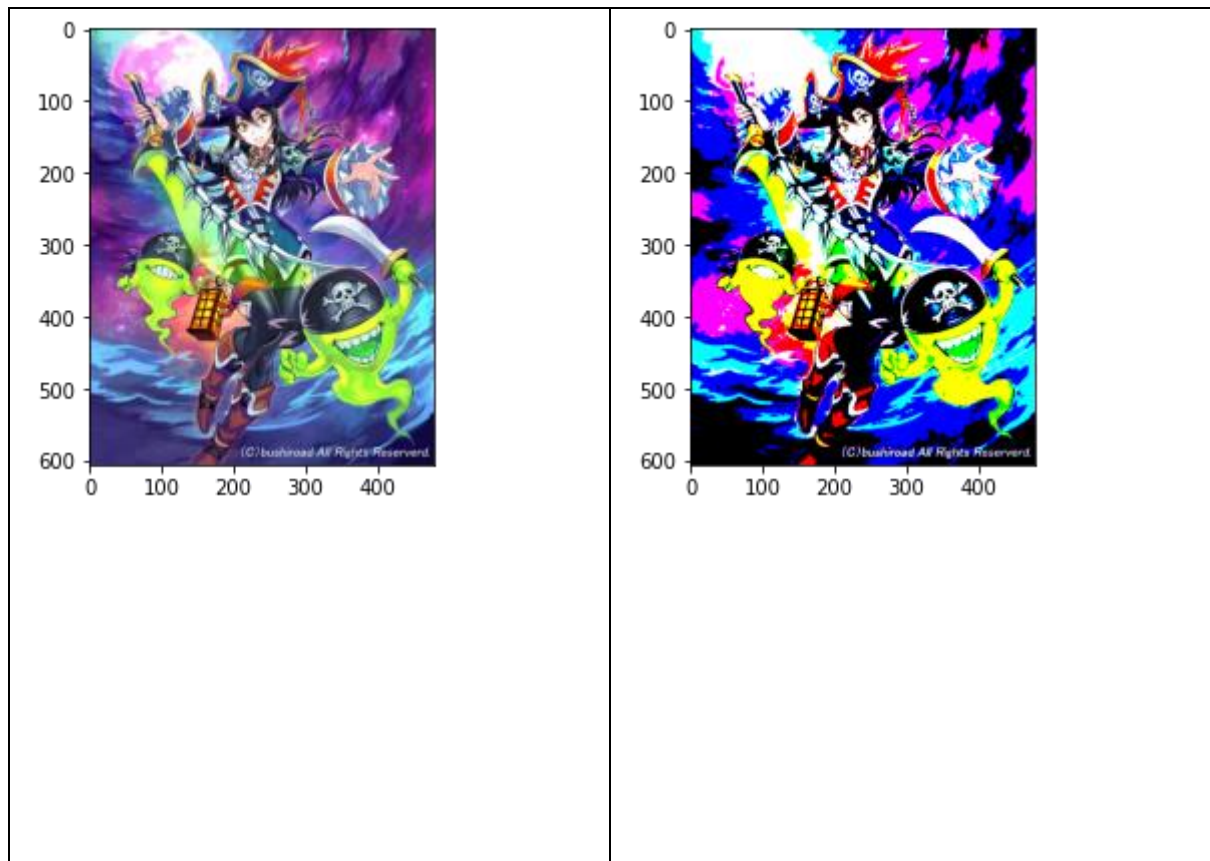
$= 0$ เมื่อ $x(\text{สีเขียว})$ มีค่า < 128

$s(\text{สีแดง}) = 255$ เมื่อ $x(\text{สีแดง})$ มีค่า ≥ 128

$= 0$ เมื่อ $x(\text{สีแดง})$ มีค่า < 128

Examples of filtered image:

Original	After filtering
	



Your code:

homework1_2_<ID>.py

```
# import library here
import cv2
import matplotlib.pyplot as plt
def homework1_2(rgbimage):
    # input -> rgbimage - type -> np.ndarray, size of - (height,
width, 3)
    # output -> filtered_image - type -> np.ndarray, size of -
(height, width, 3)

    # TO DO - Design your own filter
    threshold = 128;
    for i in range(rgbimage.shape[0]):
        for j in range(rgbimage.shape[1]):
            x = rgbimage[i][j];
            rgbimage[i][j][0] = (x[0]>=threshold)*255;
            rgbimage[i][j][1] = (x[1]>=threshold)*255;
            rgbimage[i][j][2] = (x[2]>=threshold)*255;
```

```
filtered_image = rgbimage;  
return filtered_image;
```

3. Two images, $f(x, y)$ and $g(x, y)$, have histograms h_f and h_g . Write a program in `homework1_3_<ID>.py` to display the histograms h_f and h_g . Then implement the operations below and display the new histogram of the output of each operation. Determine the new histograms in terms of h_f and h_g and explain how to obtain the histogram in each case.

(a) $f(x, y) + g(x, y)$

(b) $f(x, y) - g(x, y)$

(c) $f(x, y) \times g(x, y)$

(d) $f(x, y) \div g(x, y)$

Hint: design one of the images very simple.

homework1_3_<ID>.py is free style. Students have to make sure the program runs properly.

ใช้ image 1 จาก รูป kitty.jpg จากข้อ 1 ส่วน image 2 คือการนำ image 1 ไปเพิ่มความเข้ม และเปลี่ยนค่าสีของ น้ำเงินเป็น 5 เขียวเป็น 10 และ แดงเป็น 20 ตามลำดับ จากโจทย์ ทำการบวก ลบ คูณหาร histogram ของภาพทั้งสอง โดยใช้ function `cv2.add()`, `cv2.subtract()`, `cv2.multiply()`, `cv2.divide()`