Homework #1

Deadline: September 2nd, 2025 @23:59

Submissions: (1) PDF version of this file

(2) .ipynb file; template in the link below

***** Only Questions 1 – 3 *****

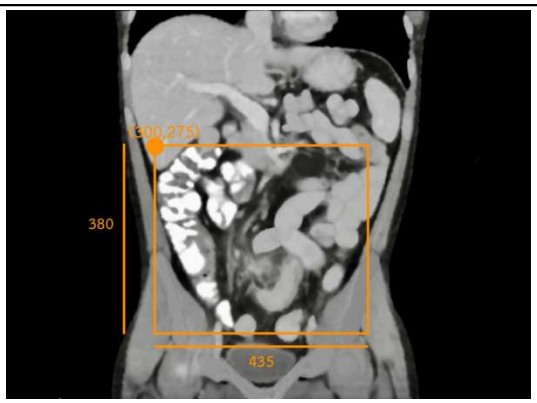
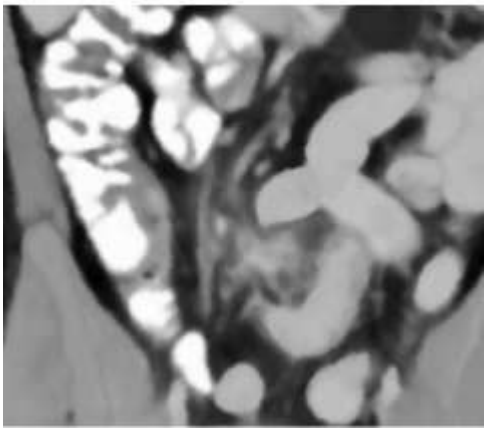
COLAB TEMPLATE:
https://colab.research.google.com/drive/1ugPEpx4NCgl7qLv-IvCaVyp3eQXRczNs?usp=sharing

1. Assume you work in the field of image processing. Your boss has assigned you a task to detect malignant tumors (assuming in this case, in bright intensity) from CT-SCAN images. The pain point is that the doctors saved images from the CT-SCAN, but the output images are incomplete and have Salt and Pepper noise. Please help the doctor remove the noise.

1.1 Apply a filter to remove the noise and select the appropriate size of the kernel. Provide your filtered image into the blank box below. Hint cv2.medianBlur

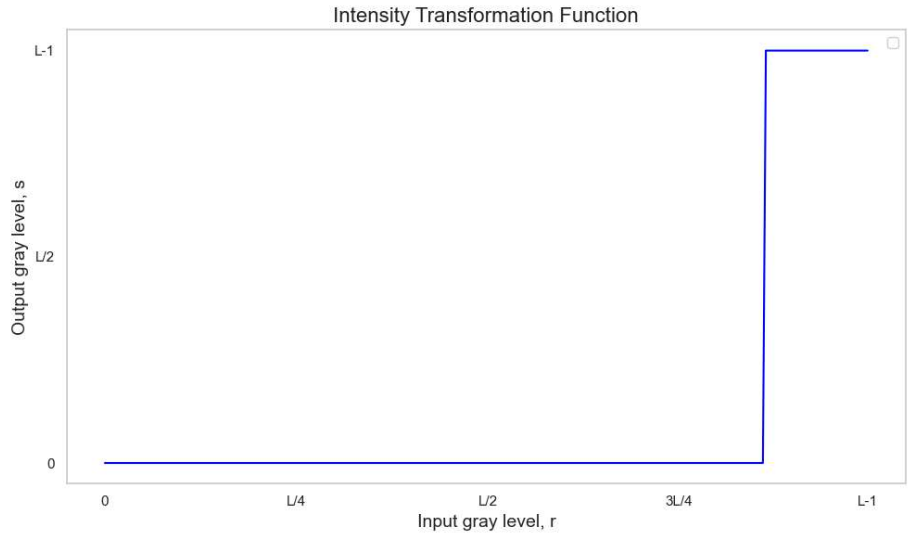
Original	Image result after Filtering
	

1.2 Apply Region of Interest (ROI) with width=380 and height=435 start at x=300, y=275 as shown in the orange rectangle below and provide the ROI image in the blank box below. Hint cv2.rectangle

Hint	Mark ROI Image
	

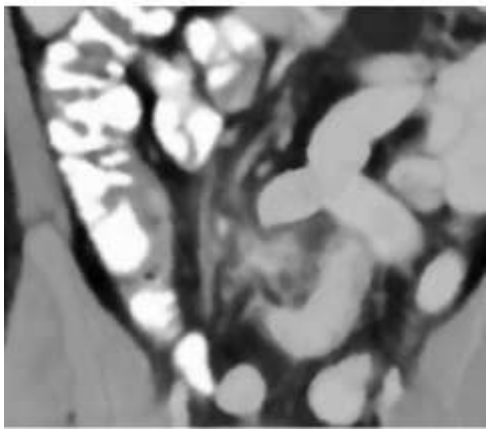
1.3 Apply the transformation function shown in the graph below on the ROI image. This transformation function is used for segmenting malignant tumors (assuming, in this case the higher intensity) and show in a white mask. Provide the final segmented tumors in the blank box below.

Transformation Function



L-1 is the highest intensity level.

You can estimate the stepping point from the graph.

Original (Mark ROI)	Segmentation with Transformation
	

You can write your own code or use the code template below and modify the _____ and put it in `homework1_1()` function in `homework1.ipynb` file

```
# import libraries here
```

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


def homework1_1(rgb_image):
    # input -> rgb_image - type -> np.ndarray, size of - (height, width, 3) with dtype - uint8
    # output -> smoothed_img - type -> np.ndarray, size of - (height, width) with dtype - uint8
    #           mark_ROI - type -> np.ndarray, size of - (height, width) with dtype - uint8
    #           Gray_Level_img - type -> np.ndarray, size of - (height, width) with dtype - uint8

    # TO DO : You can use/modify the code guideline below or write your
own code below here
    # Convert the image to grayscale
    gray_image = cv2._____________(rgb_image,cv2._____________)

    # 1.1 Use the median filter to smooth the image
    smoothed_img = cv2._____________(gray_image, _____________)

    # 1.2 Make ROI with
    # Create an area of interest (ROI) using image slicing
    mark_ROI = smoothed_img[_____________]

    # 1.3 Use Gray Level slicing
    Gray_Level_img = np.zeros_like(mark_ROI)
    Gray_Level_img[(mark_ROI >= _____________)]  = 255

    return smoothed_img, mark_ROI, Gray_Level_img
```

2. Design your own filter on an RGB image. Write your code in **`homework1_2()`** function in **`homework1.ipynb`** file
Provide motivation behind the designed filter. Display it in terms of an RGB image.

Idea / Motivation:

Want to create a filter to make picture into my favorite picture style so in future i can batching process my picture instead of doing one by one.

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

1. Sharpening (`sharpen = original + (original - blur)*amount`)
2. Saturation, Brightness enhancing

Examples of filtered image:

Original	After filtering
Kitty.jpg: 	

Your image:	
	

3. Let $f(x, y)$ and $g(x, y)$ be independent random variables with histograms h_f and h_g. Define new images as:

(a) Define the addition operations:

$$z(x, y) = f(x, y) + g(x, y)$$

prove that the histogram of $z(x, y)$ is given by

$$h_z(k) = \sum_i h_f(i) \cdot h_g(k - i)$$

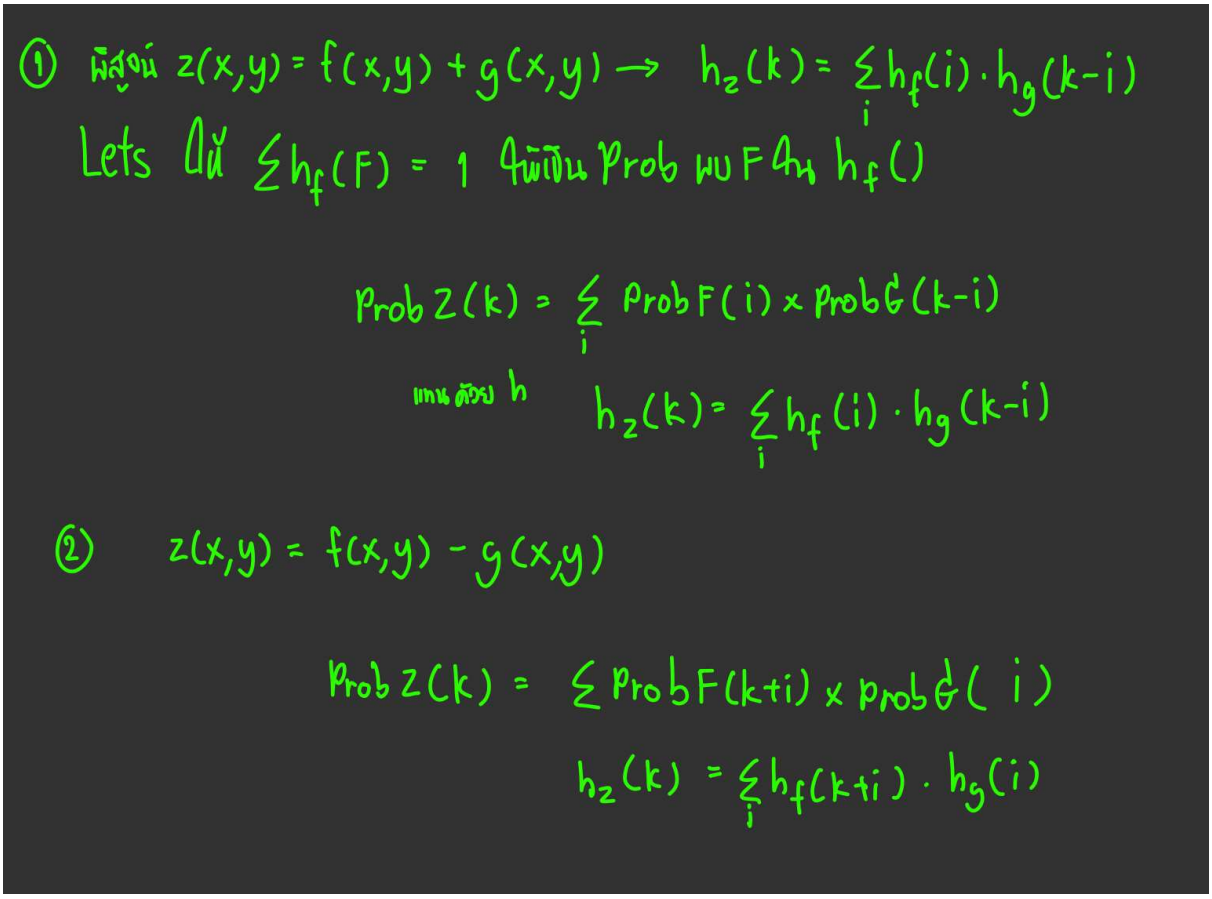
(Hint: This corresponds to the convolution of distributions.)

(b) Now consider the subtraction operation

$$z(x, y) = f(x, y) - g(x, y)$$

By following the same reasoning as in part (a), derive the expression for the histogram of $z(x, y)$ in terms of h_f and h_g.

(Hint: This should lead you to a general form related to correlation.)



(c) Explain the constraint in image context. (For example: intensity values must remain within the range [0,255]; negative values or overflow are clipped or rescaled.)

Intensity range

- สำหรับ 8-bit grayscale/RGB → ค่าอยู่ใน [0,255]
- ถ้าเป็น float image (เช่น normalized) → อยู่ใน [0,1]
- เวลา operation (add, subtract, multiply) มักจะเกินช่วง → ต้อง clip (เกิน 255 ให้เป็น 255, น้อยกว่า 0 เป็น 0)

/rescale (map ค่ากลับเข้าช่วง [0,255] แบบ linear หรือ normalization)