# CGRA352 Assignment 1

20 marks

Core (11 marks)

Basic operations on images

1. **(5 marks)** Read image from the given input image "`Flower.jpg`". Convert the image from the RGB color space to the HSV color space, using an OpenCV color-space converting function. Create a single large image that shows each of its channels of the RGB and HSV color-space as a separate grayscale images.

    Note: When working with these matrices, take a note of their type; the default format for storing a color image in the RGB color space is stored in a matrix of type `CV_8UC3` (unsigned integers in range 0 to 255), which means there are 3 Channels of Unsigned char (8 bits for each channel). A grayscale image with a single channel has a matrix type of `CV_8UC1`.
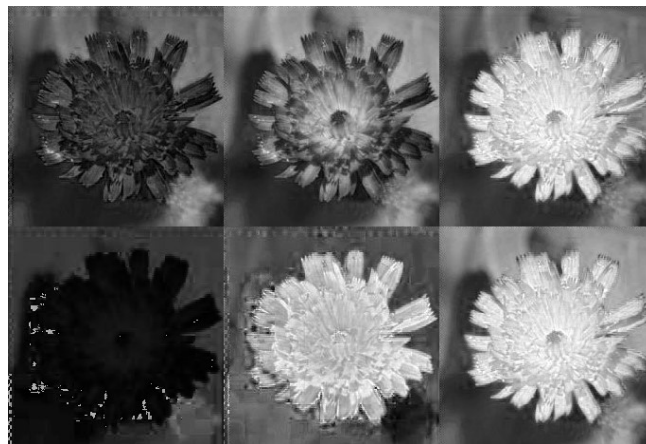
(top-row) B, G, and R channels, (bottom-row) H, S, and V channels.

2. **(3 marks)** Investigate how changing the values of individual channels affects the image in HSV color space. Write a program to modify the image in the HSV color-space by

independently scaling (multiplying) the H, S and V channels by 0, 0.2, 0.4, 0.6, and 0.8, and combine them into one image. In total you should have 15 sub-images, 5 for each independent scaling of H, S and V.

Note: Before we can display the HSV image, we need to convert it back to the RGB color space. This can also be done using the OpenCV function.

(top-row) scaling H channel, (middle-row) scaling S channel, (bottom-row) scaling V channel.

3. **(3 marks)** The color similarity between a pair of pixels can be measured by the distance in color space and we can segment an image if we separate pixel pairs based on their distance. For the given input image, take the pixel located at index (80, 80) and find all the pixels in the image with Euclidean distance of less than 100 in color space (the Euclidean distance can be computed as `sqrt((R1-R2)²+ (G1-G2)²+(B1-B2)²)` or by using the `cv::norm` function). Create a mask image to show all the valid pixel positions where the color difference is less than the threshold of 100. The valid pixels in the mask image should be shown using white (255), the remaining pixels should be set to black (0).
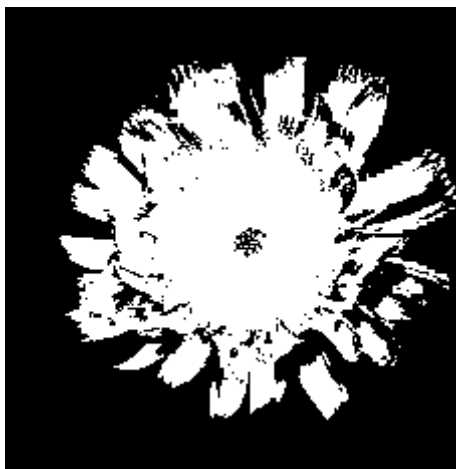
*The expected output of creating a mask.*

Completion (5 marks)

Edge extraction by convolution

Write a program that finds edge intensity in an image. The program should load an image in grayscale mode (using: `cv::imread("work/res/Flower.jpg", 0)`) and generate three images: edge magnitude image by Laplacian filter, and two images which are the edge filter responses in the x and y directions respectively by Sobel filters.

Note: The intensity range you can show as visible images is in range [0, 255], so you must find a way to make negative values visible. You can do this by scaling the result to the range [0, 255] making the most negative edge intensities black (0), the most positive edge intensities white (255) and no intensity gray (127).
You may NOT use the `cv::Sobel` or `cv::Laplacian` functions for this section.
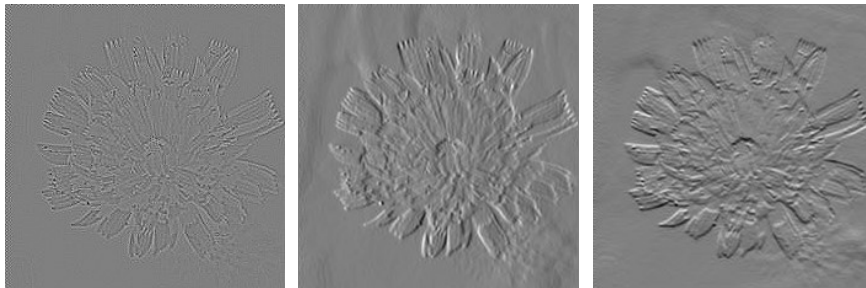
(left) Laplacian, (middle) Sobel x-direction, and (right) Sobel y-direction.

Challenge (2 marks)

Histogram equalization

Given a gray (single channel) image as input, perform histogram equalization and output the result. You are allowed to use the OpenCV built-in function `cv::calcHist` to extract the histogram, where the resulting histogram will be stored in the `hist` output argument and you can retrieve the value for every bin, similar to accessing matrix elements in OpenCV.
You may NOT use the `cv::equalizeHist` (or equivalent) function for this section.

Report (2 marks)

You should submit a 1-2 page PDF document reflecting your experience. The score of your report will be capped at 50% (if you finished CORE), 80% (if you finished COMPLETION), and 100% (if you finished CHALLENGE). In this report, you should include:

- Brief introduction of your functions in your programs.
- How to run your program to perform the functions required by the assignment.
- The results of core, completion, and challenge (depending on how much you have done).

Practical and Assessment Matters

*You are required to submit both your program and any supplementary material including the report. You may complete the assignment in single or multiple programs, as long as it meets the assignment specification. Please zip your program(s) source and other material into a single compressed file to upload.

Please note that when evaluating each component of core, completion, and challenge, 60% of the marks will be allocated based on the extent of completion for that specific aspect, while 40% will be determined by the clarity and depth of comprehension demonstrated through your answers and explanations during the evaluation process.