CSC 481 – Intro. To Image Processing
2024-25 – Winter Term
Instructor: Kenny Davila Castellanos

Weekly Assignment 5: Color

Objective

The main objective of this assignment is to help student get familiar with multiple color-based operations that were studied in class. In addition, the homework provides an opportunity to learn more about image segmentation using color.

Data

To complete this assignment, please pick 3 images of your liking. You should use:

1) A picture of a very colorful object with a contrasting background. This must be a picture taken by you.
2) A picture of multiple colorful objects, each one with different dominant colors (e.g. flowers, fruits, vegetables, etc.) with a contrasting background. This can be a picture taken by you or from the internet. Please clearly indicate the source on your report.
3) A random picture of your liking different from the first two. Ideally, look for a colorful image with contrasting background.

   You will have to run your code with each of these images, and then you must display and briefly discuss the results in your report.

Part 1 – Edge Detection with Color

For each image, do the following main steps:

a) Read the image and display it.
b) Convert to grayscale using a **simple average** of all color channels and display it.
c) Use any method for edge detection on this grayscale version, and display the results.
d) Convert the original image to HSI.
e) Use the same edge detection method to find edges on the I channel, and display the results.
f) Use the same edge detection method to find edges on the H channel, and display the results.
g) In the **report**, compare the three edge images, and discuss their similarities and/or differences.
h) In the **report**, mention which of these gave you the best results PER INPUT image, and explain why you think so.

For **every input image** (you have 3 inputs, see **Data**), you must show: the input image, the grayscale image, edge detection results for grayscale, edge detection results for I channel of HSI image, edge detection results for H channel of HSI image. Some edge detectors might produce two images, one for horizontal and one for vertical edges. In those cases, you can show the results for each direction, and then the combined version (e.g. a binary OR).

Part 2 – Color Segmentation

A natural cue to use in segmenting objects from their surroundings in images is color. Here, you will have a chance to try color-based segmentation on the RGB color space and compare it against segmentation on the HSI/HSV color space. For this comparison to be meaningful, your color images should be carefully chosen to have variety of contrasting colors. For **each input image**, you need to complete the following tasks:

Part 2 - Task 1 – Grayscale Segmentation

a) Convert the image to grayscale using a **simple average** of channel intensities.
b) Segment the image into **objects** and **background** using a threshold on the intensity of the pixels. The result is a binary image.
c) Demonstrate the segmentation results by making a copy of the original color image, and replacing the background pixels with a visually distinct color.

Part 2 - Task 2 – RGB Segmentation

a) Segment the image using a manually set threshold in the **R color band** of the RGB image. Try identifying a good threshold that isolates your object of interest. Show the resulting **binary mask** for the **best threshold**, and also display a modified copy of the original image where the background pixels are replaced by a fixed color.
b) Repeat Part2.Task2.a, but using the **G color band** of RGB.
c) Repeat Part2.Task2.a, but using the **B color band** of RGB.
d) Segment the image using an automatically set threshold in the **R color band** of the RGB image. You can try with techniques such as OTSU's method. Show the resulting **binary mask** for the **best threshold**, and also display a modified copy of the original image where the background pixels are replaced by a fixed color.
e) Repeat Part2.Task2.d, but using the **G color band** of RGB.
f) Repeat Part2.Task2.d, but using the **B color band** of RGB.

Part 2 - Task 3 – HSI/HSV Segmentation

a) Convert the original RGB image to HSI/HSV color space.
b) Segment the image using manually set thresholds in the **H color band** of the HSV/HSI color space. Note that different libraries/tools might use different standards. In the particular case of matlab's

HSV space, the hue (first component) of a pixel ranges from 0.0 (**red**) to 1.0 (**red** again), passing through **orange**, **yellow**, **green**, **cyan**, **blue**, **purple**, and **magenta** along the way. The second component, saturation, varies from 0.0 (grayscale) to 1.0 (completely saturated -- no white at all). The final component, intensity (or "value"), also ranges from 0.0 (no intensity) to 1.0 (max intensity).

A threshold of the hue component of pixels **must be an interval**, because the **hue** actually wraps around and is best envisioned as a circle. Thus, to segment a blue region, you need to accept only hues around 2/3 (0 = red, 1/3 = green, 2/3 = blue).

c) Try identifying good thresholds that isolates your object of interest. Show the resulting combined **binary mask** for the **best thresholds**, and also display a modified copy of the original image where the background pixels are replaced by a fixed color.

In the **report**, analyze how your segmentation based on hue differs from your segmentations based on RGB and grayscale. Explain which one seems to work better per image, also elaborate on your reasons for thinking this way.

For **every input image** (you have 3 inputs, see **Data**), you must show: the input image, the grayscale version, and then [binary masks] + [background removed version] for each type of segmentation (based on intensities; manual thresholds on R, G and B; automatically set thresholds on R, G, and B; and hue-based thresholds).

The Program

In this assignment, you will write a program that handles multiple operations. You can use any programming language of your preference, but Python is highly recommended. Your code should be properly organized and well-commented to clearly identify the correspondence between portions of the code and each part of this assignment. The code should compile and run on any properly configured programming environment without the need of making significant changes to it. In this sense, it is highly recommended to avoid the usage of absolute paths.

The report

You are asked to submit a written report showing the results for each part of the assignment. Use proper section headings and descriptions to clearly identify the results of each part. If I cannot easily find a match in the report for a required result, I might assume that this portion was not completed, and a score of 0 will be given by default.

For students working with Python, you are allowed to use Jupyter Notebooks. These allow you to combine code with the report in a single deliverable, which is perfectly acceptable for this course. Besides the original Jupyter notebook file, please also submit an export to PDF.

Delivery Instructions

You are given the freedom to use any programming language and IDE of your preference. You are also required to provide your own images, and to produce a written report. When done, **you should submit everything using a single Zip file**.

File names. The zip file that you submit should use "[Last Name(s)], [Given Name(s)].zip" as it appears in D2L. For example, "Kenny Davila Castellanos" (Davila Castellanos is two last names), would have to submit the homework with the name "Davila Castellanos, Kenny.zip". Another student named "Kenny Mauricio Davila" (Mauricio is a middle name), would have to submit the homework as "Davila, Kenny Mauricio.zip". Not following these instructions might lead to a penalty.

Policies

1. All general policies about Plagiarism and Cheating apply to this homework. If you plagiarize or receive code from other people, you will be caught and you will receive a score of 0, and a report of the academic integrity violation will be filed.
2. Please limit the usage of Chat-GPT and other code generators to ethical usage only. Submitting code that was directly generated by these tools is considered a form of plagiarism.
3. Do not post your solutions online and do not share them with anyone. It is your responsibility to safeguard your private data.
4. **Code that does not compile due to syntax and/or semantic errors will automatically receive a score of 0.** It is hard to assign partial credit when I cannot even run your code.
5. You can use any programming language of your choosing.
6. You must follow the delivery instructions.
7. Very late homework's without justification will receive a score of 0.
8. The homework description outlines very specific requirements. You are welcome to try other things and report your results. However, no amount of extra work can be used as a substitution of the actual requirements.
9. Do ask for help if anything is unclear, but do it in a timely manner (e.g., by e-mail, Discord or during the Office Hours).