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

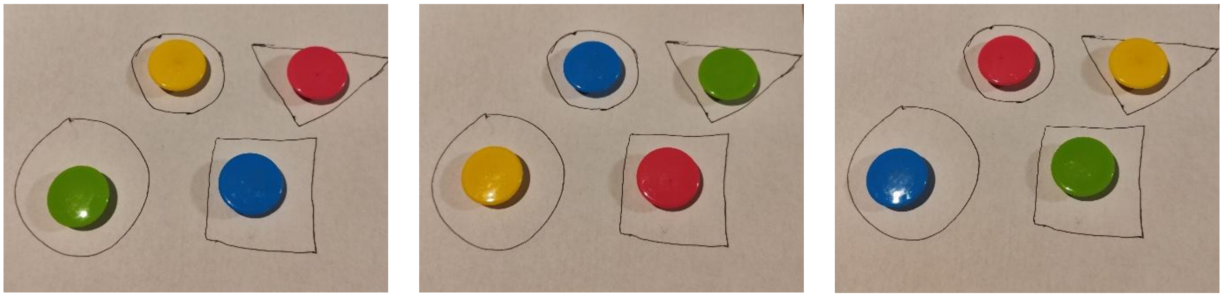
Weekly Assignment 7: Putting Everything Together

Objective

The main objective of this assignment is to evaluate the usage of multiple techniques studied in this course to solve a single task on a small group of images of the same type.

Data

To complete this assignment, you are given the following images:



Input 1 Input 2 Input 3

Figure 1. Images that you will use for this assignment.

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.

Task

The main goal of this assignment is to locate and highlight the closed region/shape that contains the **blue piece** on each one of the given images. The approach should be as automated as possible, but a few assumptions can be made:

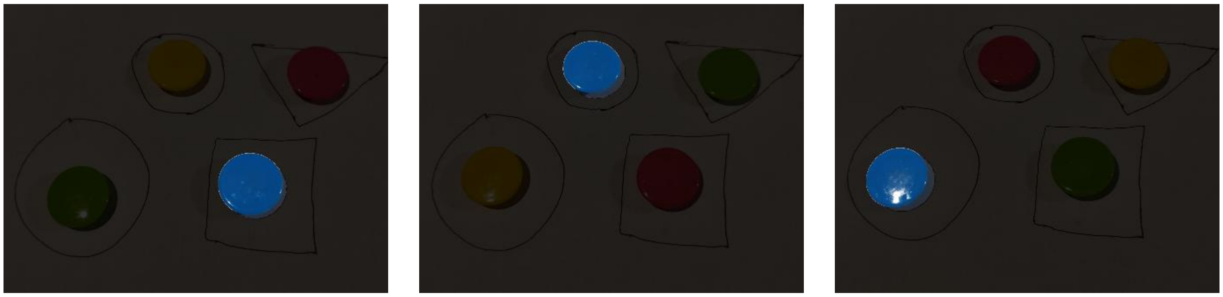
1) The background will always be a piece of white paper
2) There will be 4 plastic pieces per image. One per color: yellow, green, magenta and blue.
3) There are basic shapes (rectangles, circles, or triangles) drawn with dark ink on the paper.
4) Each plastic piece will be "contained" within one of these basic shapes.
5) The light on the scene can cause noise that complicates color-based segmentation a bit.

Part 1 – Locating the Blue Piece

The goal of this part is to **locate the blue piece** on the image, and you can use **any approach** that you want for this purpose, as long as it is something that we studied in this course. **The recommended approach is to use a combination of color-based segmentation and morphological operations**. First, you can convert the image to HSV to locate the bluish pixels. Note that due to the noise introduced by changes in illumination (e.g. Input 3 in Figure 1), it is likely that many pixels of the blue piece will be missing after any color-based segmentation. You should use a large structuring element of any useful shape (disk/ellipse is recommended), and a combination of closing and opening operations to effectively recover the missing pixels. Note that it is also possible that there are noisy pixels on the image that do not belong to the blue piece, and you can remove these by either explicit analysis of the connected components, or by simply using more morphological operations. Keep in mind that the order of the operations affects their outcome.

Expected outputs per input image:

a) Show the binary mask of the blue piece.
b) Visualize the blue piece by lowering the intensity of all other pixels.



Output 1 Output 2 Output 3

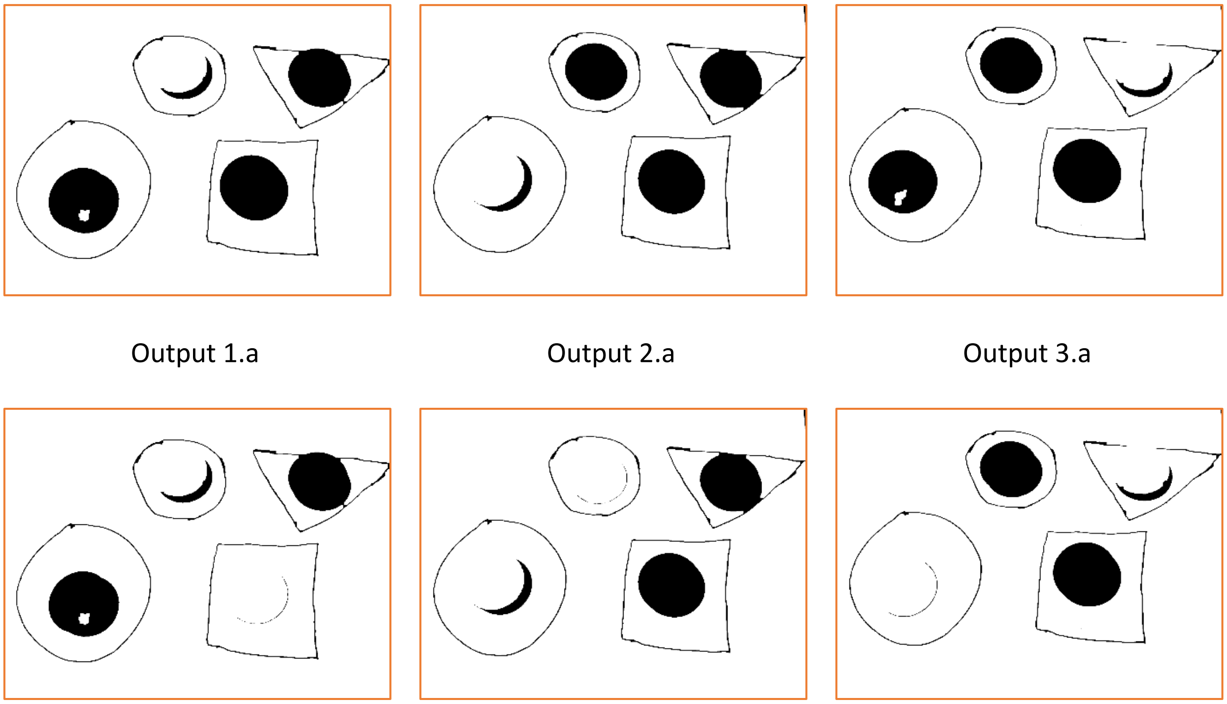
Figure 2. Examples of the expected outputs for Part 1.

Part 2 – Locating the Boundaries of the Shapes

The goal of this part is to **locate the boundaries of the shapes** on the image, and you can use **any approach** that you want for this purpose, as long as it is something that we studied in this course. **The recommended approach is to use a combination of binarization or edge detection with morphological operations**. First, you can binarize the image. You should an automatically selected threshold for this. Then, it is possible that some figures will not be completely closed afterwards, specially if the target piece is not fully contained on the shape and it is overlapping any of its boundaries (e.g. the triangle in all cases). Fortunately, in our case we will assume that the piece of interest (the blue one) will always be fully contain as shown in all examples in Figure 1. This will allow us to use a simpler approach. Still, it is highly recommended to use morphological operations to close the shapes where a small gap might exist (e.g. the rectangle). Next, **you need to remove the only the blue piece from this binary image**. This means that other **dark pieces might show up in the image but that is perfectly okay here**.

Expected outputs per input image:

a) Refined Binary image of the boundaries.
b) Refined Binary Image after removing the pixels of the blue piece.



Output 1.a Output 2.a Output 3.a

Output 1.b Output 2.b Output 3.b

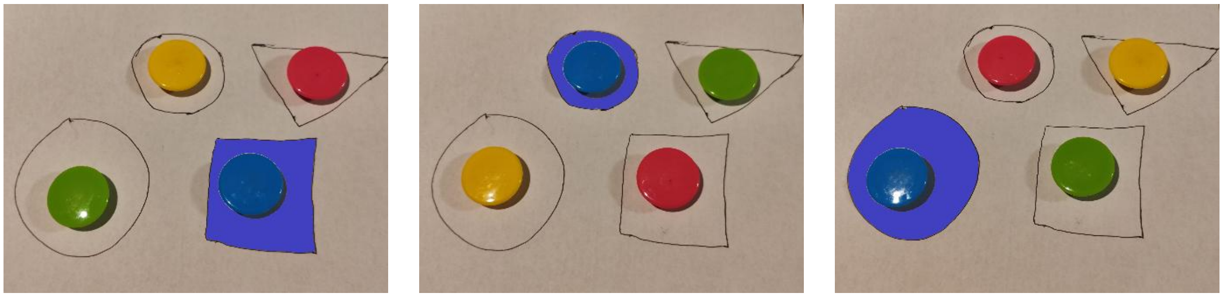
Figure 3. Examples of the expected outputs for Part 2.

Part 3 – Locating the Target Shape

The goal of this part is to **locate the target shape** (the one that contains the blue piece) on the image, and you can use **any approach** that you want for this purpose, as long as it is something that we studied in this course. **The recommended approach is to use a combination of logical operations and connected component analysis**. If you run a connected component (CC) labeling algorithm over the output of **Part 2**, you will be able to detect all white and black regions. Depending on the implementation, it is likely that you will only get connected components for the white regions while other pixels are considered background. The output of CC labeling can be used to create binary masks for each CC. You can iterate through each CC to locate the CC that overlaps with the mask of the blue piece generated in **Part 1**. Once that you have identified this CC, you have identified the containing shape, and you can use it to create a modified version of the input that highlights this region on the original image.

Expected outputs per input image:

c) Show the binary mask of the CC that contains the blue piece.
d) Visualize the shape that contains the blue piece by changing the colors for this region.



Output 1 Output 2 Output 3

Figure 4. Examples of the expected outputs for Part 3.

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).