VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL PROJECT**

**INTRODUCTION TO DIGITAL IMAGE PROCESSING**

**TRAFFIC SIGN RECOGNITION**

*Instructing Lecturer*: **TRỊNH HÙNG CƯỜNG**

*Student’s name*: **NGUYỄN ANH KHOA- 519H0303**

**LÊ MINH – 519H0315**

**HO CHI MINH CITY, 2024**

VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL PROJECT**

**INTRODUCTION TO DIGITAL IMAGE PROCESSING**

**TRAFFIC SIGN RECOGNITION**

*Instructing Lecturer*: **TRỊNH HÙNG CƯỜNG**

*Student’s name*: **NGUYỄN ANH KHOA- 519H0303**

**LÊ MINH – 519H0315**

**HO CHI MINH CITY, 2024**

**THESIS AND DISSERTATION ACKNOWLEDGEMENTS**

After a semester of studying at Ton Duc Thang University, we want to give our sincere thanks to our teachers and Faculty of Information Technology for bringing a desired condition for students able to complete their course by giving many topics that can be applied in real-life.

We want to express our gratitude for our lecturer – Trinh Hung Cuong. He helped us a lot on this course, giving us information about Machine Learning so that we can improve our knowledge. Also, we want to give thanks to each other member in this group. All of us are busy because everyone has other classes, but we still reply to each other when we need to. Once again, we are truly grateful for everyone that helped us to do this project. With challenging work and effort, we have successfully completed this report. Maybe this report has some mistakes because this is our first report on this course, so we really want to take the comments of the teacher and lecturer for our improvement in future projects.

With sincere thanks.

REPORT COMPLETED AT

TON DUC THANG UNIVERSITY

I hereby declare that this is my own project and is under the guidance of lecturer Trinh Hung Cuong. The research contents and results in this topic are honest and have not been published in any publication before. The data in the tables for analysis, comments and evaluation are collected by the author himself from different sources, clearly stated in the reference section.

In addition, the project also uses a number of comments, assessments as well as data of other authors, other agencies and organizations, with citations and source annotations.

If I find any fraud, I will take full responsibility for the content of my project. Ton Duc Thang University is not related to copyright and copyright violations caused by me (if any).

*Ho Chi Minh, June 4, 2024*

*Author*

*Nguyễn Anh Khoa*

*Lê Minh*

EVALUATION OF INSTRUCTING LECTURER

**Confirmation of the instructor**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ho Chi Minh City, 2024

(sign and write full name)

**The assessment of the teacher marked**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ho Chi Minh City, 2023

(sign and write full name)

ABSTRACT

This project focuses on using straightforward digital image processing techniques from OpenCV, rather than implementing complex deep learning or machine learning models. The goal is to recognize traffic signs using these simpler OpenCV methods.

The process begins with the input images, which contain various types of traffic signs. These images first undergo some preprocessing steps, such as noise reduction, contrast enhancement, and resizing, to prepare them for further analysis.

Next, the preprocessed images go through the following steps:

1. Canny Edge Detection: The Canny edge detection algorithm is applied to the images to identify the edges and outlines of the objects.
2. Color-based Segmentation: The key colors of traffic signs (red, blue, and black) are used to segment the images and isolate the color regions of interest.
3. Shape Detection: Two main shapes are detected - circles and rectangles. The Hough Circle Transform is used for circle detection, and the Douglas-Peucker contour approximation is used for rectangle detection.
4. Contour Detection and Ranking: Contour detection is applied to both the edge detection and color-based segmentation outputs. The largest contours, ranked by their area using cv2.contourArea(), are likely to correspond to the actual traffic signs.

The final output of this process is a rectangle box surrounding the detected traffic sign objects in the input images. This approach, using standard OpenCV techniques without relying on complex machine learning models, demonstrates a straightforward method for traffic sign recognition.

CONTENT

[REPORT COMPLETED AT iv](#_Toc168437843)

[TON DUC THANG UNIVERSITY iv](#_Toc168437844)

[EVALUATION OF INSTRUCTING LECTURER v](#_Toc168437845)

[ABSTRACT vi](#_Toc168437846)

[CONTENT vii](#_Toc168437847)

[CHAPTER 1 – METHODOLOGY OF SOLVING TASKS 1](#_Toc168437848)

[1.1 Image Preprocessing 1](#_Toc168437849)

[Contrast Enhancement 1](#_Toc168437850)

[1.2 Canny Edge Detection 2](#_Toc168437851)

[1.3 Color based segmentation 3](#_Toc168437852)

[1.3.1 Define HSV Color Ranges 3](#_Toc168437853)

[1.3.2 Merge Color Masks 3](#_Toc168437854)

[1.3.3 Morphological Operations 3](#_Toc168437855)

[1.4 Shapes Detection 4](#_Toc168437856)

[1.5 Contour detection and filtering 4](#_Toc168437857)

[CHAPTER 2: EXPERIMENTAL STEPS AND RESULTS 6](#_Toc168437858)

[2.1 Load Images 6](#_Toc168437859)

[2.2 Denoise and enhance contrast 6](#_Toc168437860)

[2.3 Resize the images 7](#_Toc168437861)

[2.4 Change to Grayscale 9](#_Toc168437862)

[2.5 Canny Edge Detection 10](#_Toc168437863)

[2.6 Color Segmentation 11](#_Toc168437864)

[2.7 Shape Detection 12](#_Toc168437865)

[2.8 Combine outputs of edge detector and color based segmentation by shape detection 12](#_Toc168437866)

[2.9 Draw rectangular frames with different color 14](#_Toc168437867)

[2.10 Others results 14](#_Toc168437868)

[REFERENCE 18](#_Toc168437869)

CHAPTER 1 – METHODOLOGY OF SOLVING TASKS

1.1 Image Preprocessing

In this stage, there are several methods used to process the raw input images:

* **Denoising and Blurring**: A 3x3 median filter is applied to the input images to remove noise while preserving edges.
* **Contrast Enhancement**: The low contrast images are identified using the following expression (1)
* **Image Resizing**: The input images are resized to a width of 200 pixels, using the cv2.INTER\_AREA interpolation method.

1. We using *skimage.exposure import is\_low\_contrast* function to detect the contrast level of images that are inadequate with following expression:

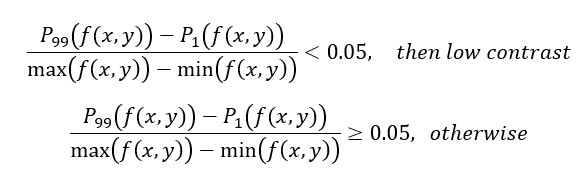


Figure 1.1. Expression for detecting low level of contrast.

f(x,y) : pixel intensity.

P99 : 99th percentile of the pixel value.

P 1 : 1th percentile of the pixel value.

0.05 is the threshold.

Contrast Enhancement

Proccessing steps and methods:

**Color Space Conversion**: The input image, which is in the BGR (Blue, Green, Red) color space, is converted to the Lab color space.

* L\* represents the lightness of the color.
* a\* represents the red/green value.
* b\* represents the blue/yellow value.

**Channel Splitting and Histogram Equalization**: The Lab image is split into its component channels (L, a, and b).

Histogram equalization is then applied to the L (lightness) channel to enhance the overall brightness and contrast of the image.

**Channel Merging**: The modified L channel is then merged back with the original a and b channels using the cv2.merge() function.

**Color Space Conversion (Back to BGR)**:The processed image in the Lab color space is then converted back to the original BGR color space.

1.2 Canny Edge Detection

**Canny Edge Detection**: After the initial preprocessing steps, the images are processed using the Canny edge detection algorithm from the OpenCV library. The Canny function cv2.Canny() is called with the following parameters:

* input\_img: the input image
* lower\_bound: the lower threshold value for edge detection
* upper\_bound: the upper threshold value for edge detection

**Automated Thresholding for Canny**: To determine the optimal threshold values for the Canny edge detection, an automated thresholding approach is used. The lower and upper threshold values are calculated as follows:

* lowTh = (1 - sigma) \* Thresh
* highTh = (1 + sigma) \* Thresh

Where Thresh is the overall threshold value, and sigma is a default value set to 0.33, which corresponds to approximately 1/3 of the range around the threshold bounds.

**Otsu's Method**: To find the threshold value more easily, Otsu's method is applied. Otsu's method is an algorithm that automatically selects the optimal threshold value by minimizing the intra-class variance of the black and white pixels.

1.3 Color based segmentation

1.3.1 ****Define HSV Color Ranges****

The key colors for traffic signs are identified as red, blue, and black. For each of these colors, the lower and upper bounds of the HSV (Hue, Saturation, Value) color space are defined. This is done using the cv2.inRange() function, which creates a binary mask for the specified color range.

1.3.2 ****Merge Color Masks****

The output of the cv2.inRange() function for each color (red, blue, and black) results in separate color masks. These individual color masks are then combined using the bitwise OR operator to create a single, unified mask that encompasses all the relevant color regions.

1.3.3 ****Morphological Operations****

To improve the quality of the segmented mask, morphological operations are applied. Specifically:

* **Opening**: This operation is used to reduce noise and remove small isolated regions that may not be part of the desired objects.
* **Closing**: This operation is used to close small holes inside the objects or connect nearby components that may have been separated.

These morphological operations help to refine the color-based segmentation, ensuring that the resulting mask accurately represents the desired traffic sign objects while minimizing noise and unwanted artifacts.

The final outcome of this process is a clean, segmented mask that separates the color objects (traffic signs) from the background, which can then be used for further analysis or processing.

1.4 Shapes Detection

There are two main shapes we need to detect: circles and rectangle

* The Hough Circle Transform and the Douglas-Peucker contour approximation are two complementary techniques used to detect the two main shapes of interest: circles and rectangles.
* The Hough Circle Transform is well-suited for identifying circular shapes, as it can accurately locate the center and radius of the circles. On the other hand, the Douglas-Peucker contour approximation is effective in detecting rectangular shapes by simplifying the contours and identifying those that approximate a rectangular geometry.

1.5 Contour detection and filtering

**Contour Detection**:

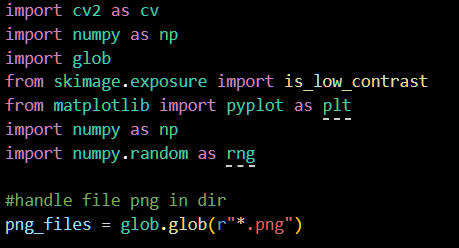
* Contour detection is applied to the outputs of both the edge detection (Canny method) and the color-based segmentation.
* This results in two sets of contours: one from the edge detection and another from the color-based segmentation.

**Traffic Sign Identification**:

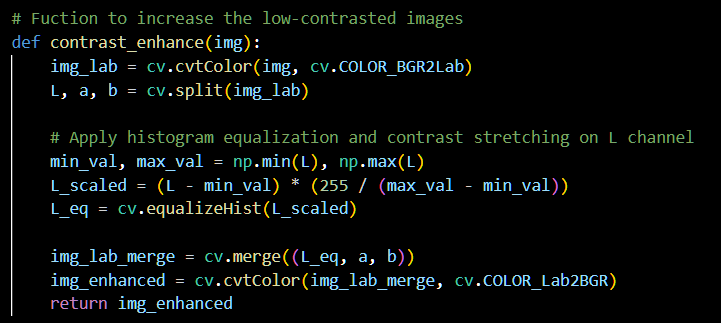
* To detect the exact traffic signs, the system focuses on the largest contours from the two sets.
* The contours are ranked by their area using the cv2.contourArea() function.
* The largest contours are likely to correspond to the traffic signs, as they are the most prominent and significant objects in the image.

CHAPTER 2: EXPERIMENTAL STEPS AND RESULTS

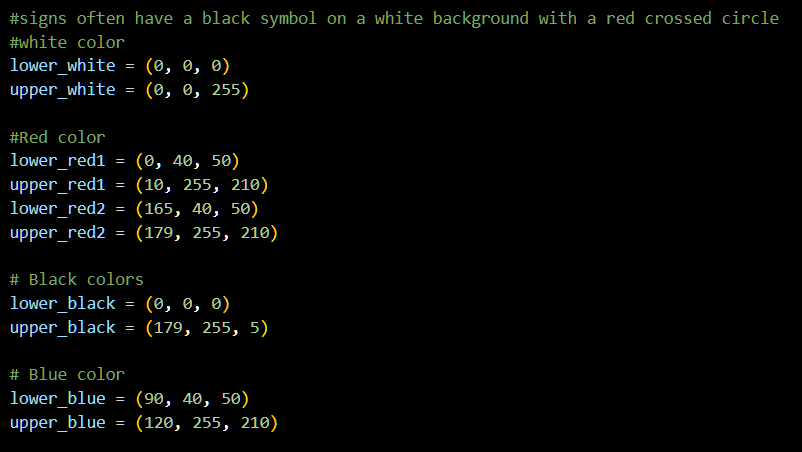
2.1 Import and load Images:

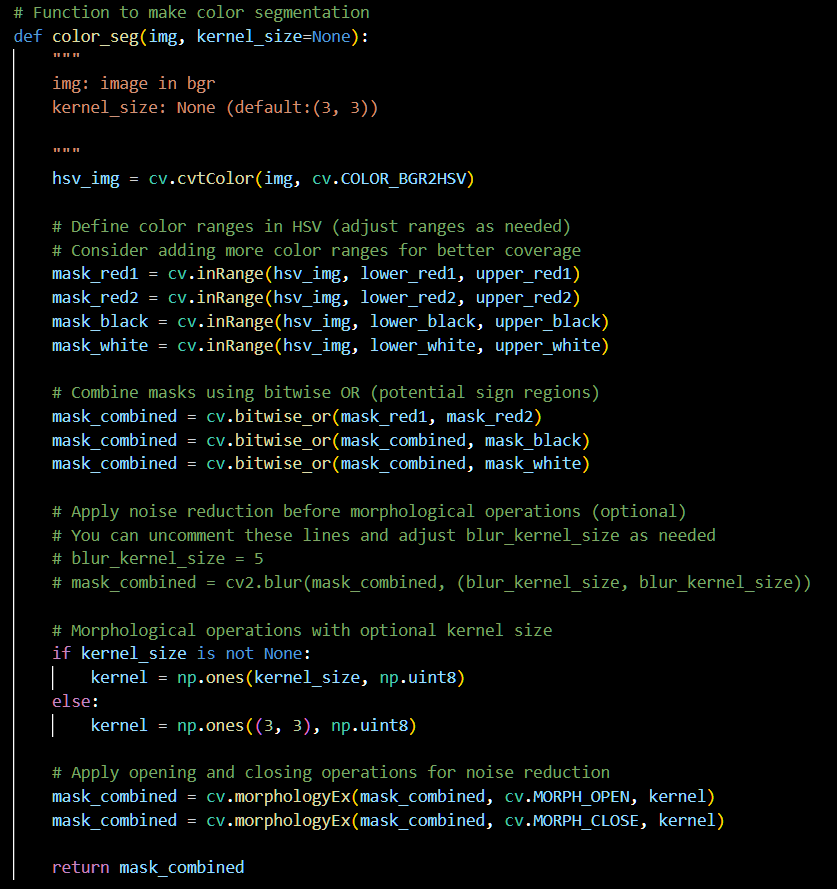


2.2 Increase the low-contrasted images

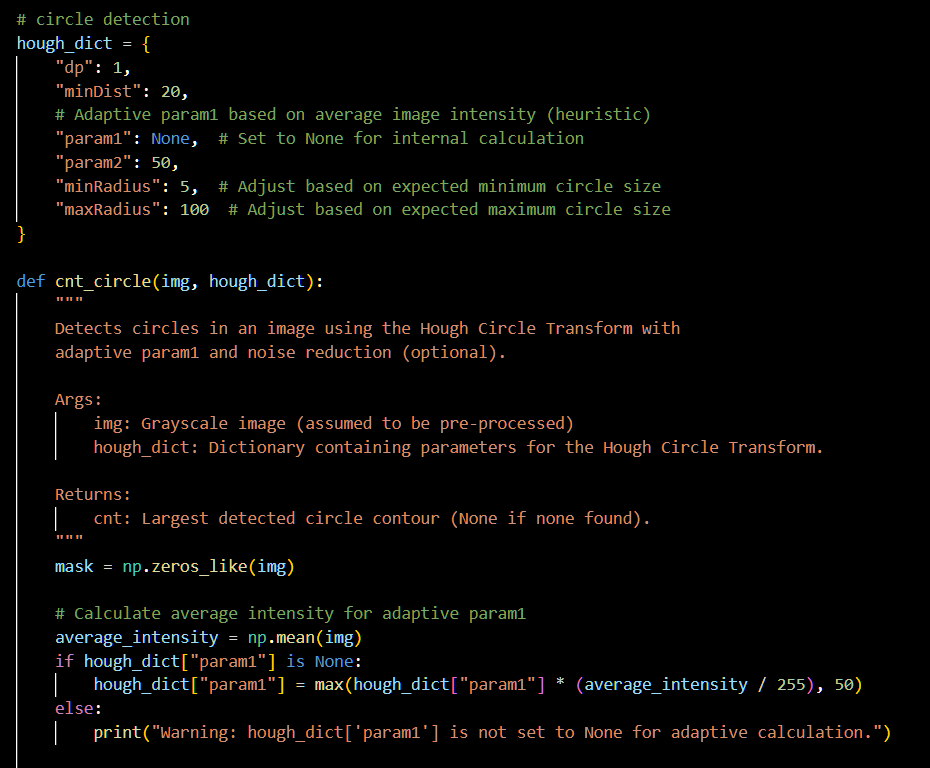


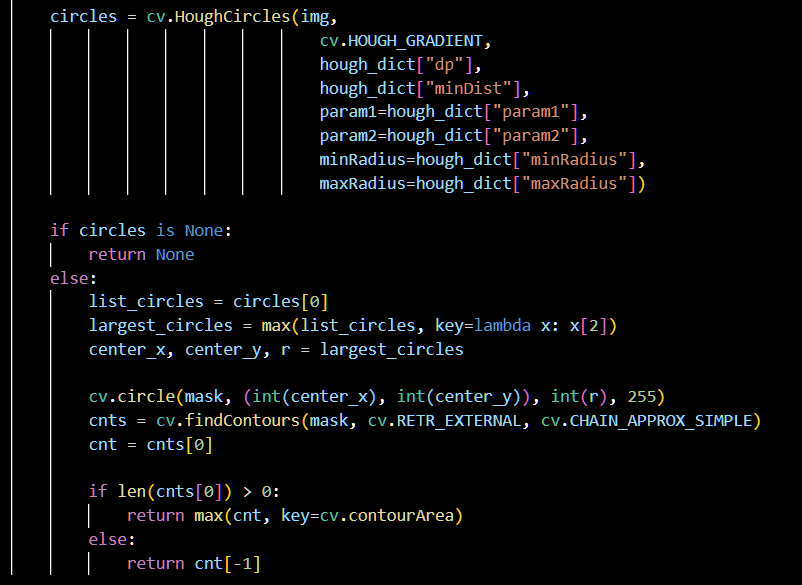
2.3 Color segmentation and handle color background of signs





2.4 Circle detection

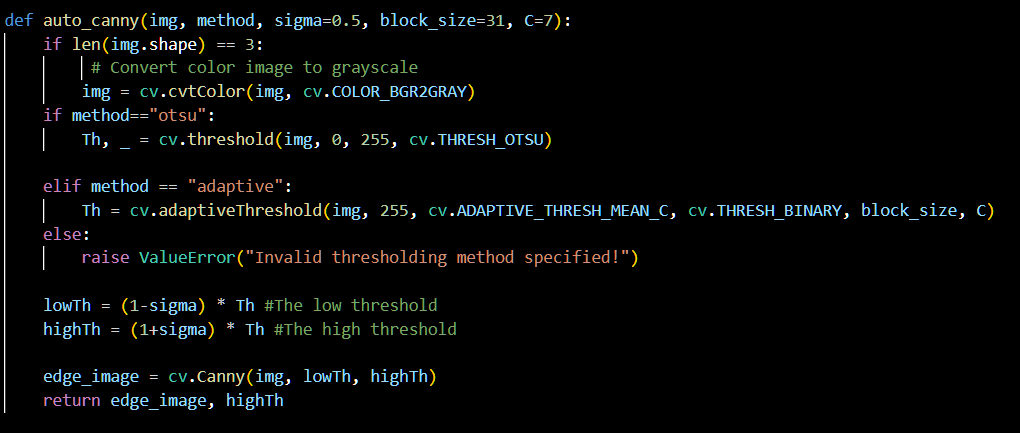




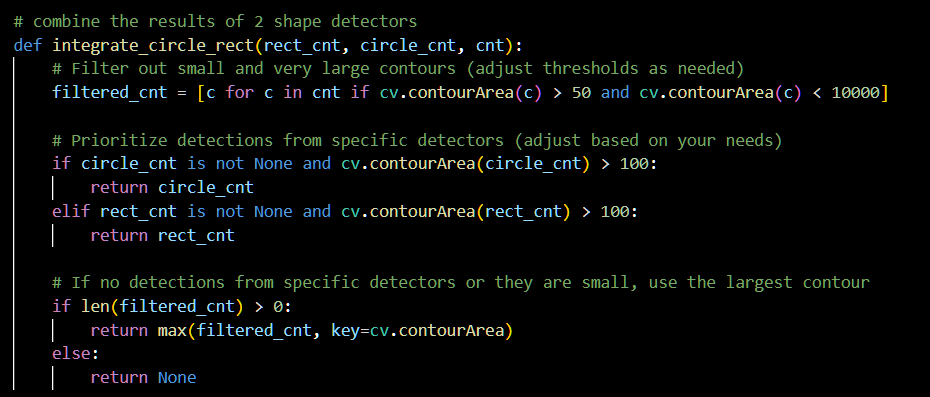
2.5 Reactangle Detection



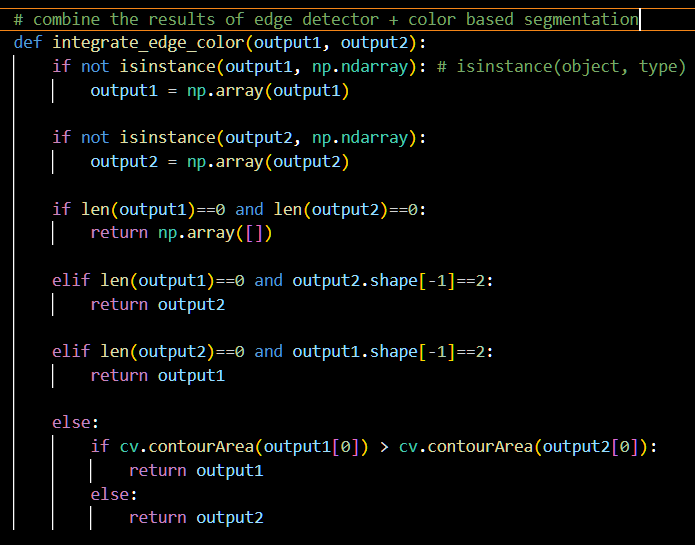
2.6 Auto canny



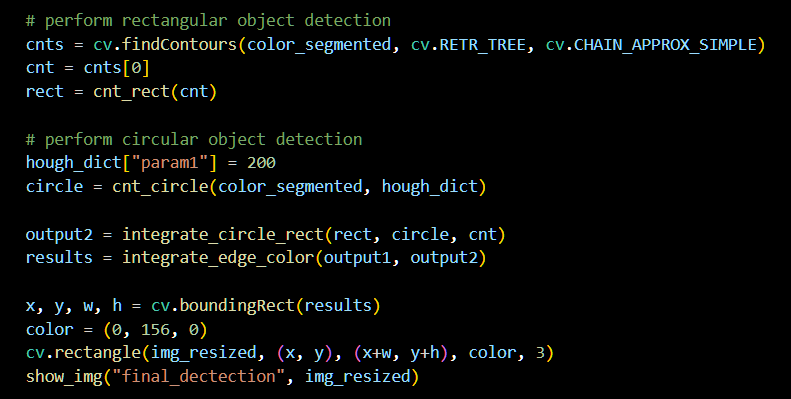
2.7 Combined Shape Detection



2.8 Combine outputs of edge detector and color based segmentation by shape detection



2.9 Draw rectangular frames with different color



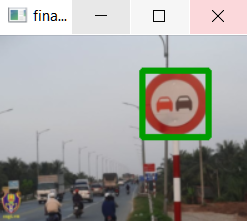
2.10 Different of results

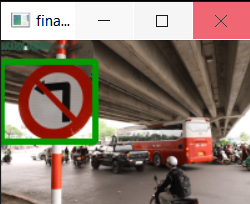


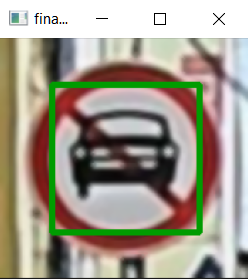


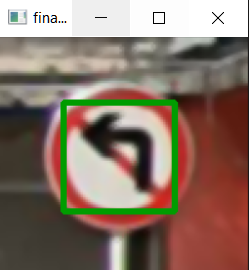














REFERENCE

1. [***https://docs.opencv.org/4.x/dd/d49/tutorial\_py\_contour\_features.html***](https://docs.opencv.org/4.x/dd/d49/tutorial_py_contour_features.html)
2. [***https://www.geeksforgeeks.org/finding-minimum-enclosing-rectangle-in-opencv-python/***](https://www.geeksforgeeks.org/finding-minimum-enclosing-rectangle-in-opencv-python/)
3. [***https://www.tutorialspoint.com/how-to-detect-a-rectangle-and-square-in-an-image-using-opencv-python***](https://www.tutorialspoint.com/how-to-detect-a-rectangle-and-square-in-an-image-using-opencv-python)