Stop Signs Detection by Image Processing

Salma M. Gira

Cairo University

Faculty of Engineering Cairo University, computer department

salmagira16@gmail.com

Table of Contents

List of Figures

Abstract

In this report, stop signs and object detection using OpenCV and python is discussed. Detecting stop signs in an image is useful for self-driving cars. The main purpose of this project is to detect stop signs without any manual processes to detection. This project is coded to detect only one stop sign, and only the stop sign is detected, while any noise is ignored.

Keywords: OpenCV, object detection, image processing

Abbreviation

OpenCV: Open Source Computer Vision

Stop Signs Detection by Image Processing

This program is written in python 3, using OpenCV 2, os and numpy libraries. The aim is to detect the stop sign from a given image. The code's operation is to get the full image path and its name with its type as an input, then apply the image to be converted from RGB to grayscale. The code then takes this grayscale result and apply threshold on it. Then enters a loop for detecting each octagon in this image, as stop sign has an octagon shape. Then makes a white image with same dimensions of the input image every time an octagon is detected, and drawing that octagon on this new white image. Then for each white image with the octagon made, it flood-fills this octagon with black. Then if the octagon contains filling pixels, it means that the octagon detected is not a dot (noise), hence the code prints word "stop". Then for the big octagon we invert this image of the black octagon by changing black pixels into white and vice versa, then we merge this black and white image with the original image so that all the image is black except the stop sign. Then the output is extracted in the same folder entered.

Method

Assessments and Measures

First of all, to write this program, anaconda navigator and Spyder IDE were used. For programming language, I used python 3.7 with OpenCV 2. Which made it not difficult to read, save or show image and all needed functionalities to apply necessary processing for the image. Also used os library to bind image path with image name, then save the image in the given path.

Steps:

1. Take as an input the full path of the image and the name of the image with its type, e.g. C:/Users/DELL/.spyder-py3/folder/ >> stop.png
2. The program then performs all processes by itself:

a. Takes the original image [figure 1] and convert it from RGB to grayscale [figure 2].
b. Applies the resulting image to threshold using OpenCV. [figure 3]
c. Then the resulting image enters a loop to detect octagons in this image. When detecting an octagon, the detected octagon enters some operations.
d. It first makes another white image having the same dimension of the original one, then takes that octagon and puts it in black on the white image.
e. Then fills the octagon with black pixels using OpenCV floodfill.
f. If the octagon is filled, that means it is a real octagon not a tiny spot detected as octagon [figure 4], and if it is a real octagon it means that this octagon is a stop sign [figure 5].
g. Then prints "stop", then takes the white image with the detected big octagon and inverts black pixels into white and vice versa, so that the octagon become white and the rest of the image is black.
h. After that takes the resulting binary image of the real white octagon and uses it as a mask, that is applied to the original image to get as output the stop sign as it is while the rest of the image is black. [figure 6]

Figure 1: original image (input)

Figure 2: after inverting from RGB to grayscale

Figure 3: after applying threshold, to take off most af the noise and fake octagons

Figure 4: after detecting an octagon and putting it in a separate image

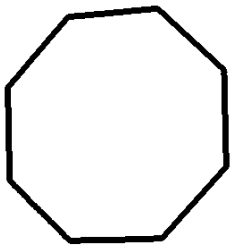

Figure 5: detecting a big octagon

Figure 6: after applying the mask

Results

This program results one image named by “result”, which is all black except the stop sign, which remains the same. [figure 7]

Figure 7: the resulting image

Future Work

This program shows only one stop sign in an image using object detection (geometric shapes detection). Hope to work more on this program so that it can detect more than one stop sign in a single image. Also, for more features, hope to make the program detect more traffic signs rather than stop signs only. That could be achieved by using more detection for other shapes, as well as using OCR for detecting more than one type of traffic signs written in the same geometric shape, to differentiate between words not only the geometric shape.

References

https://opencv.org/