Traffic sign detection

Ivan Bagaric
International Master of Computer Vision

Universidade da Coruna, Spain
bagaric.ivan@udc.es

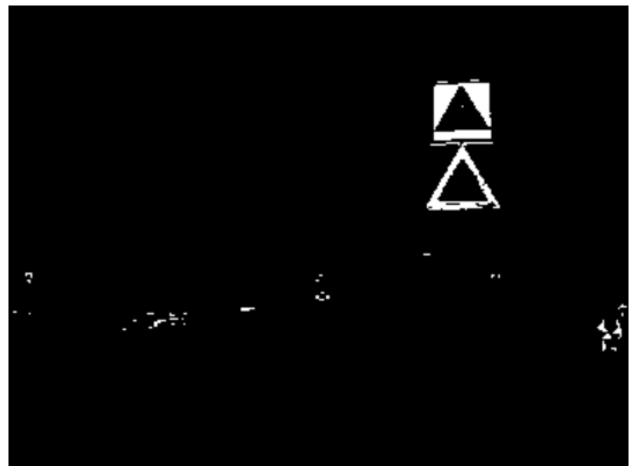
***Abstract*— With the increase in popularity of autonomous vehicles, the search for better and faster algorithms that would overcome the problems that autonomous vehicles face has also increased. In order to speed up the whole process, we are trying to transfer as many operations as possible to image processing instead of using deep learning, which is still slower. In this paper we will deal with the recognition of traffic signs with the help of open cv operations.**

I. Introduction

As we said, we will exclusively use OpenCv operations to detect traffic signs from the images we have. The first step we will set is that we will define the ROI (Region of interest) and we will add it to each of the layers on the HSV image. After that, we use the inRange function to detect the traffic sign in the image with the help of the values obtained through experimentation. After detection, we use Interest point detector SIFT to match sign with one of our templates.. The formatter will need to create these components, incorporating the applicable criteria that follow.

II. inRange

InRange is a thresholding operation that allows us to create a mask from an HSV image by defining RGB minimum and maximum values. But why do we use HSV? HSV (hue, saturation, value) color space is a model to represent the color space similar to the RGB color model. Since the hue channel models the color type, it is very useful in image processing tasks that need to segment objects based on their color. Variation of the saturation goes from unsaturated to represent shades of gray and fully saturated (no white component). Value channel describes the brightness or the intensity of the color. Since colors in the RGB color space are coded using the three channels, it is more difficult to segment an object in the image based on its color. Through experimentation, we came to the conclusion that the minimum RGB values are (55,130,80) and the maximum RGB values are (255,255,255) after which we get mask like in picture1.



Picture1. Example mask after applying inRange

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

After this we can easily get contours (in this case biggest one) with function cv2.findContours. We are sending three parameter:

- first one is source mask image;
- the second is in charge of mode of the contour retrieval algorithm. We will choose cv2.RETR_TREE what means that we will retrieve all of the contours (parents and childs)
- CHAIN_APPROX_NONE - is going to store absolutely all contour points. Now we have to sort list of contours so we are going to set contours as a input in sorted python function, and in the key property we should make clear what type of data we are handling with, at the and reverse the list order.

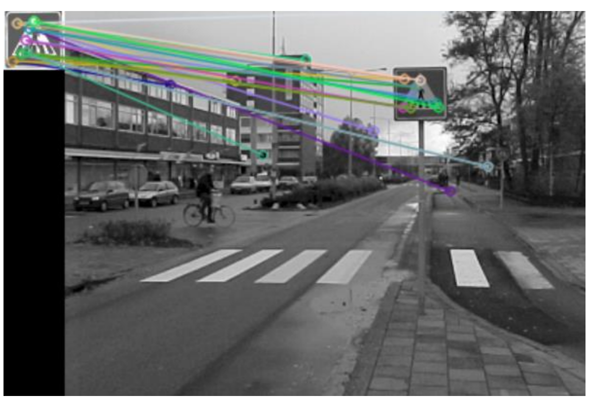
With BoundingReact function we can get start point, height and with for each contour. With that information we can create

ROI for next step and also draw rectangle around given contour to visualize data like on picture2.

Picture 2. Draw rectangle

III. SIFT

SIFT (Scale Invariant Fourier Transform) Detector is used in the detection of interest points on an input image. It allows identification of localized features in images. Before we start to use this function we have to transform our image to grayscale. So, we will look for interest points on both image and then compare them to detect what is the right class of our traffic sign. In order to save only good matches we will use ratio test with 0.75 value what is standard value for similar cases.

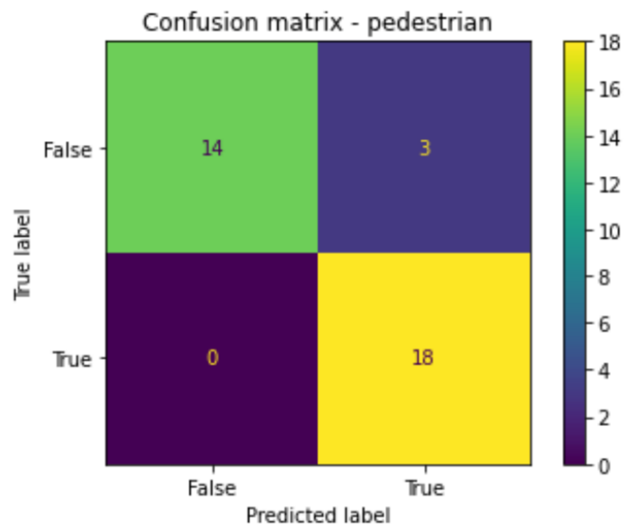


Picture 3. Detect interest points

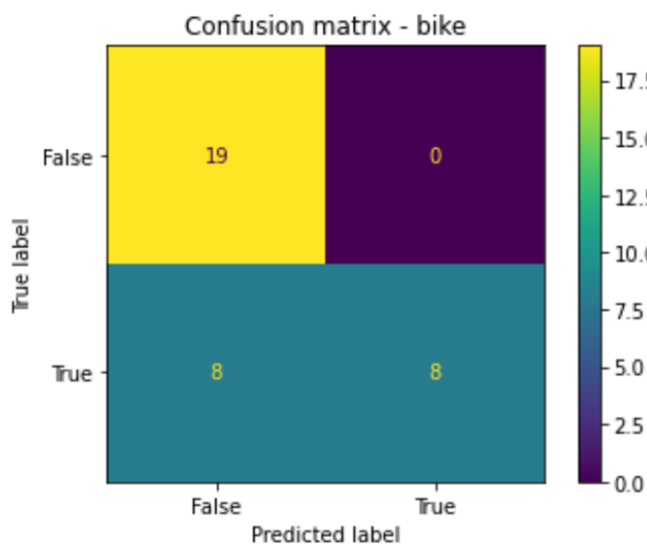
In picture 3. We can see why we should use only region where is sign placed. We can notice that most of interest point are really placed on traffic sign, but we also have some errors. After all program will compare matches of both templates and return the name of one with more descriptors.

IV. RESULTS

We can see in this two confusion matrix, that pedestrian has much better results, what is logic, because graphic of pedestrian sign has much more contrast and robust corners.



Picture 4. Pedestrian confusion matrix



Picture 5. Bike confusion matrix

Picture 6. Bike and pedestrian key points

REFERENCES

[1] https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html

[2] https://docs.opencv.org/3.4/da/d97/tutorial_threshold_inRange.html