Traffic sign detection

Ivo Saavedra
up201707093@edu.fe.up.pt
Telmo Baptista
up201806554@edu.fe.up.pt
Tiago Silva
up201806516@edu.fe.up.pt

Faculty of Engineering
University of Porto
Porto, PT

Abstract

In this article we tackle the problem of traffic sign detection, handling noise and tough lighting conditions. The images used in the experimental phase were retrieved from a Kaggle dataset for road sign detection. The traffic signs are classified based on the colour segmentation of blues and reds, and shape detection via edge detection and analysis of contours and their bounding boxes. Our solution detects multiple signs per image, and is effective even on slanted or partially occluded signs. The detected types go beyond the scope of the dataset, as we classify them into 'stop', 'mandatory', 'prohibitory', 'information', and 'priority'.

1 Introduction

The first step to identify traffic signs is to detect them. Over the years most countries have followed a common standard regarding their signage, using simple shapes and bright colours, uncommon in the environment, to ensure they can be easily seen by drivers. This gives two main avenues to the detection of traffic signs: colour segmentation and shape detection. Much research has already been done on this topic, so we plan to combine techniques described by various authors and apply a combination of colour segmentation and shape detection to complete the detection step.

The identification of traffic signs are highly subject to various types of defects, such as different light conditions affecting the image, which are difficult to control and treat. The lighting can vary depending on time of the day, season, weather conditions, defects of the camera used to take the picture, among others. For example, an image lit by a bright sky in the middle of the day can cause specular reflections to signs or tint the image in blue tones, making the segmentation much harder. While not prevalent in our testing dataset, nighttime lowers the images' brightness and contrast, hindering the identification of relevant features. Weather conditions such as rain or fog also affects images, creating noise and lowering the contrast and saturation of the image. Another common defect that makes the identification of traffic signs difficult is the presence of objects that may produce partial occlusions and shadows on the traffic signs.

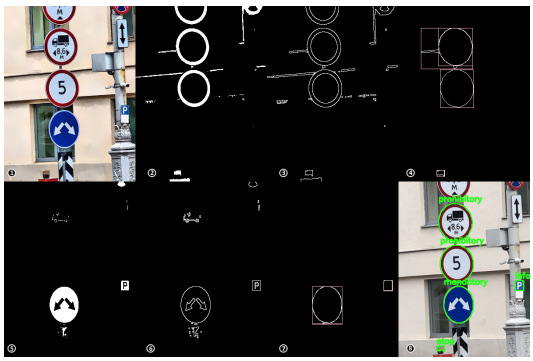


Figure 1: Traffic sign detection

2 Image processing

In order to reduce the aforementioned potential defects that make the detection of traffic signs more difficult, a wide variety of techniques were experimented. There are several defects on the images of the dataset that were studied and then attempted to be solved using one or multiple of these techniques. The final objective of our processing was to build a processing sequence that was able to attenuate or solve the more generalized problems of the images from the dataset while being careful not to introduce new undesired side effects.

2.1 Automatic Brightness and Contrast Correction

Images taken in bad weather conditions often suffer defects, such as low contrast, non-uniform illumination and colour degradations. On the opposite spectrum, images captured with intense sources of light, such as, direct sunlight on the camera, can cause colours to desaturate if the light is directly hitting the object, or can produce shadows causing the loss of intensity of the colours, if the source of light is behind the object.

In an attempt to solve both of these problems, a brightness and contrast correction method was applied to the image[13]. The brightness and contrast of a gray image can be adjusted using α and β parameters, often called the *gain* and *bias* parameters, using the following equation, where $g(i,j)$ is the output pixel and $f(i,j)$ is the input pixel:

$$g(i,j) = \alpha \cdot f(i,j) + \beta \tag{1}$$

If we look at the histogram of the image, the parameter β has the effect of shifting the histogram in the direction of its sign, thus increasing or decreasing the brightness of the image. While the α parameter controls how the levels are spread. If $\alpha > 1$, the histogram will be dilated, resulting in an image with higher contrast, and for $\alpha < 1$, it has the opposite effect, obtaining a lower contrast image by compressing the histogram.

(a) Using $\beta = 80$ (b) Using $\alpha < 0$

Figure 2: Effects of α and β (black) on the histograms of an image (light gray)

However, this process isn't automatic, as appropriate values for α and β change depending on the image being processed. To tackle this problem, we can calculate the values for the parameters by using the histogram of the image. By calculating the cumulative distribution of the histogram, and then determining the cut location on both the left and right side of the histogram, based on the colour frequency being lower than a given threshold, p, which denotes the percentage of the histogram to cut, the values for α and β can be calculated so that the output range is $I \in \{0,\ldots,255\}$ from the following equations, where C is the cumulative distribution of the histogram, and N the size of the histogram:

$$MIN = \frac{C_N \cdot p}{2} \qquad MAX = C_N - MIN$$

$$I_{min} = \arg\min_i C_i \wedge C_i >= MIN \qquad I_{max} = \arg\max_i C_i \wedge C_i < MAX$$

Let $g(i,j) = 0$ and $f(i,j) = I_{min}$ then

$$\alpha = \frac{255}{I_{max} - I_{min}} \qquad 0 = \alpha \cdot I_{min} + \beta \implies \beta = -I_{min} \cdot \alpha$$

With α and β calculated, we can apply the equation (1) and obtain the adjusted image.

2.2 Attenuation of Blue Lighting Effects

In the test dataset, multiple images that have a *"blue filter"* effect can be observed, which makes all pixels in the image appear more blue than

their true colour. In order to try to attenuate or remove this effect, a multitude of techniques were attempted, with the one working best being the equalization of the *RGB* channels, separately. However, this approach is dangerous, as it distorts the colour balances on any image that doesn't have that effect. To prevent this as much as possible, the images this effect was taking place more intensely were studied. It was observed that the majority of those images had a bright blue sky in the background. With that knowledge, we applied this correction only when it was detected a bright sky in the image. To make that detection, we make use of colour segmentation to segment the sky and then calculate the percentage of the space that the sky is occupying. If that percentage is above a given threshold, then we consider that a sky was detected in the image. However, this wasn't enough, as there were multiple images where a sky was detected but weren't suffering from this blue lighting effect, which resulted in a bad processed image if the equalization was performed. So, to restrict the set of images to which this correction was applied, we also verify the percentage of pixels in which the blue channel is maximum. If that percentage is above a given threshold, in addition to the aforementioned criteria, then this correction is applied, otherwise it is skipped. An example can be found in the appendix (see Fig. 4).

2.3 Improving Nighttime Images

Nighttime images can make the detection of traffic signs difficult due to poor lighting of the objects. To enhance this type of images, a method presented by Shi, Z., Zhu, M.m., Guo, B. *et al*. in their paper [12] was explored and implemented. First, the bright and dark channels of the image are obtained, which are the maximum and minimum pixel values in the local patch of the image, respectively, using a sliding convolutional window. Secondly, the global atmosphere lighting is computed using the bright channel obtained above by taking the mean intensity of the top ten percent brightest pixels, this is to ensure small anomalies do not affect the global atmosphere lighting that heavily. Afterwards, the channels and the atmosphere light are used to obtain a transmission map, which describes the portion of light that is not scattered and reaches the camera. An initial transmission map is calculated from the bright channel, I^{bright} and the maximum atmosphere light of a local patch, A^c, using the following equation:

$$t^{bright}(x) = \frac{I^{bright}(x) - A^c}{255 - A^c} \tag{2}$$

To attenuate the problem regarding an intense light source covering a significant portion of the image, which results in over-exposition, a lookup table is used to interpolate the transmission map into a new range, reducing the effect of high exposure. Following this step, a second transmission map is calculated from the dark channel, along with a third channel that results from the difference of the bright and dark channels, I^{diff}, in order to correct potentially unreliable transmission estimates. The transmission of the pixel will be considered unreliable if the difference between the bright and dark channels are below a given threshold, α, which was empirically determined to be $\alpha = 0.4$ in the paper aforementioned [12], this is, if $I^{diff} < \alpha$, in which case, the transmission of the pixel is attenuated using the transmission map of the dark channel, by using the following equation:

$$t^{corrected}(x) = t^{dark}(x) \cdot t^{bright}(x) \tag{3}$$

The resulting corrected transmission filter is smoothed using a guided filter, implemented by Daisuke [4], in order to obtain a refined transmission filter, and finally obtain the improved image, J, by the following equation:

$$J(x) = \frac{I(x) - A}{t(x)} + A \tag{4}$$

2.4 Chosen Preprocessing Techniques

To sum up, in this phase, the image will be treated in an attempt to reduce possible noise from environment and/or weather. This will be achieved by applying CLAHE equalization on the image, followed by an automatic brightness and contrast correction, and finishing with the initial step of meanShift. This final step is the filtering stage of the meanShift segmentation that flattens colour gradients and fine-grain textures of the image (see [2] *pyrMeanShiftFiltering*). An additional step before CLAHE equalization is performed if the image is detected as having a *"blue filter"* (see Section 2.2). The technique used to improve nighttime images wasn't included in the pipeline, as there wasn't a significant number of nighttime images to justify it. However, it could be added to the pipeline in the future for better generalization of the processing pipeline. Similarly to the blue filter attenuation, this correction could be applied only if the image was detected as having been taken during the night.

3 Segmentation

After the processing, with a better defined image, the colour segmentation was performed to extract red and blue objects from the processed image.

Initially, a simple colour segmentation technique was applied to the HSV image by combining blue and red masks to the processed image and then creating a binary image by applying Otsu's thresholding algorithm. Whilst this solution was somewhat successful, the range of reds and blues that were being accepted was questionable. The thresholding algorithm also eliminated some relevant pixels due to its intensity on the greyscale image not being high enough. To address this, we experimented performing a custom greyscale conversion, by assigning higher weight to the relevant channel, instead of using the luminance of the pixel. This proved to help, but it was still segmenting areas incorrectly.

To improve the segmentation, the group tried to implement the solution proposed by C.F. Paulo and P.L. Correia[11], which takes advantage of the HSV colour space and evaluates each pixel of the original image through a hue-based detection of the blue and red colours. The results were not optimal, as the parameters needed a lot of fine tunings to produce good results for the given dataset. After this failed attempt, we tried implementing the other method in the aforementioned paper[11], that uses the RGB colour space instead, but the results weren't great either. Other avenues were also explored, such as the method to segment red colour under various light conditions without being affected by loss of contrast, described by El Baz *et al*. in their paper [5], or a method to segment traffic signs on poor light conditions described by Hasan Fleyeh[6].

However, the final adopted technique was a compromise between the initial approach and the one proposed by C.F. Paulo and P.L. Correia[11]. Three separate empty images were defined, one for the reds, another for the blues and the last for the whites (used for ROI extraction in Section. 4). For each of the segmentations, a set of thresholds was defined according to the hue, saturation and values of the colour in question, which were then applied to the original image. This resulted in three binary images with the segmented areas for each colour.

To remove the noise that came from the segmentation phase, especially on the blue segmented image, the group tried using OpenCV's [2] *connectedComponentsWithStats* to remove small components, but this was not included in the final state of the application due to the lack of testing and understanding of the method's effects.

4 Region of Interest Extraction

The first attempt at region of interest extraction was to apply OpenCV's [2] implementation of the Maximally Stable Extremal Region Extractor, following the approach described in Hendra Maulana's article [9]. However, this idea was discarded as the results unexpected, so it was decided to try a simpler approach for validation, and leave feature matching approaches for future work. The approach that followed was to employ a contour based ROI extraction technique, but this would require the edges of the segmented images to be detected.

4.1 Edge Detection

Several algorithms were tested with different configurations, such as the Canny Edge Detector, the Laplacian of Gaussian (LoG), and the Adaptative Thresholding.

The first approach was to only use the Canny edge detector on the segmented images, which worked for most images. Even so, in cases where the signs were placed close together, it usually joined those edges into a single shape. To try and remedy this, the group resorted to morphology operations such as small erosions and closings on the segmented images, which would in theory help separate the signals. This worked in some

cases, but was too unpredictable to include on the whole pipeline. Having these problems into consideration, the group tried other edge detection algorithms.

After a failed attempt at the LoG algorithm, the Adaptative Threshold was tested and tuned empirically, which led to slightly better overall scores, and thus, was the chosen technique for the edge detection phase.

4.2 Contour Based ROI Extraction

A contour detection algorithm was applied for each of the edge images that were acquired from the previous section. At initial phases of the project, the convex hulls of the detected contours were obtained and used for the following stages in an attempt to close clipped traffic signs, however it also led to inexplicable large areas due to close contours, see Fig. 5, and thus this operation was not included in the final state of the application. Afterwards, an approximation of those contours was performed, to simplify their shapes, in an attempt to reduce the complexity of the shape detection phase, described in Section. 5.

To extract the regions of interest, bounding boxes were drawn for each of the obtained contours and their aspect ratios were evaluated to guarantee they were inside a given threshold. ROIs with bad aspect ratios or small bounding boxes were discarded, reducing much of the noise from the edge images. However, this proved to be insufficient in the removal of undesired ROIs. To improve this, the point polygon test was employed. Very simply, it checks if a given point is a part of a closed polygon. This was used to obtain the ratios of red, blue, white and black pixels inside the detected contours. With this, we can establish a minimum ratio of colours for the ROI and thus filter out irrelevant areas. Furthermore, ROIs that are completely enclosed by another ROI are discarded as they would be either redundant or irrelevant. This helped further reduce the amount of unwanted ROIs.

5 Shape detection

The detection of shapes within ROI is done through a combination of different methods, with the intention of having a more resilient sign detector. The main methods explored in our approach were corner detection and geometrical ratio analysis. Each method outputs the probability of a given shape present in the image.

Another method we intended to implement was the Fast Radial Symmetry Detection, following Valdenegro's research[14]. This approach is not implemented in any known library, and we only found some GitHub repositories implementing it, however, they were either incorrect, incomplete, or did not follow the research in its entirety. We attempted to create our own implementation, but due to time constraints, the complexity of this detector and its relatively low success rate (57% average detection rate and 86% average false positive rate), we decided against using it.

5.1 Corner detection

The chosen approach was based on the research of Carlos Paulo and Paulo Correia[11] on triangle and square shape identification. These shapes are identified by using the Harris corner detection algorithm to find the corners in the ROI, which are later checked for their presence in six control areas. Our implementation proved to have poor results regarding the presence of circles in the image, as their corners would end up within the control areas for the square.

In an attempt to amend this situation, two new control areas were added in the middle left and right borders, and to compensate for the addition of these two control areas, the weights were adjusted to be all 0.25. Furthermore, the equations from the aforementioned paper [11] were adapted, which allowed us to find circles, skewed triangles and skewed squares, and can be found in the appendix (see Eq. (9)).

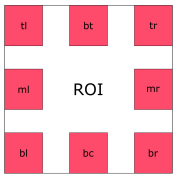


Figure 3: Corner control areas

In our implementation, each control area within the ROI, described in Fig. 3, is set to 20% of the total ROI width and height, and the presence (or absence) of corners count towards the probability of a shape being present. To account for perspective images, we allow for skewed shapes, for instance, the base of a triangle must occupy both corners of the ROI, but the third point may be in the centre or lateral corners.

The corners obtained via the Harris detector are normalized into the $[0,255]$ interval, and then the corner is considered present if the maximum intensity on the control area is above a given threshold. The value of this threshold was 75, which was obtained empirically.

5.2 Geometrical ratio analysis

To further refine our shape detection method, multiple ratios were used, both obtained mathematically and empirically. In the following sections, we will present a brief description of the ratio and how it was obtained. To obtain the probability of a shape based on the ratios, we calculate the errors of the ratio calculated for the contour to be identifier, R_c, to the base ratios of the shapes, R_{s_i}. This error can be defined as the absolute error, ε, the relative error, η, or other. The error observed for the shape will be referred as E_{s_i}, and the probability of the shape, $P(s_i)$, can be defined as

$$P(s_i) = 1 - \frac{E_{s_i}}{\sum_{k=1}^{N} E_{s_k}} \tag{5}$$

5.2.1 Circularity

One of the most standard circularity measures [15], often called compactness, derives from the relation between the contour area and its perimeter. The contour circularity, C, can be defined as

$$C = \frac{4 \cdot \pi \cdot Area_{contour}}{Perimeter^2_{contour}}. \tag{6}$$

We can now obtain the base ratios for this measure for each of the target shapes we want to identify

$$C(square) = 0.7854 \qquad C(triangle) = 0.600$$
$$C(circle) = 1.000 \qquad C(octagon) = 0.948$$

5.2.2 Extent

The extent of a shape can be defined as the ratio between the area of the shape and the area of its enclosing rectangle

$$Extent = \frac{Area_{shape}}{Area_{enclosingRectangle}} \tag{7}$$

Another ratio was derived from this definition, minimum extent, which is the same as extent but the enclosing rectangle takes into account the orientation of the shape and finds the minimum enclosing rectangle to the shape. This ratio was created as an attempt to boost identification of slanted traffic signs.

We can now obtain the base ratios for this measure for each of the target shapes we want to identify

$$Extent(square) = 1.000 \qquad Extent(triangle) = 0.500$$
$$Extent(circle) = 0.785 \qquad Extent(octagon) = 0.829$$

5.2.3 Circle Extent

Similarly to the extent, the circle extent can be defined as the ratio between the area of the shape and the area of its minimum enclosing circle

$$CircleExtent = \frac{Area_{shape}}{Area_{enclosingCircle}} \tag{8}$$

We can now obtain the base ratios for this measure for each of the target shapes we want to identify

$$CircleExtent = 0.637 \qquad CircleExtent(triangle) = 0.413$$
$$CircleExtent(circle) = 1.000 \qquad CircleExtent(octagon) = 0.900$$

5.2.4 Colour Ratios

In addition to the aforementioned mathematical ratios, the colour ratios of the blue and red colour on the traffic signs were also considered. These ratios were obtained empirically from analysing the traffic sign images in Traffic Rules [1].

6 Sign classification

By making use of the probabilities calculated in the previous step, a decision tree was defined to handle the classification of the shape, which can be found in annex on Fig. 9. Operations on the ratios and thresholds were defined empirically with the objective of generalizing the decision tree classification capabilities.

7 Results

Results	
Class	Accuracy
Information	57.00%
Prohibitory	57.47%
Stop	75.82%
Global	58.94%

Table 1: Experimental Results.

From table 1, it can be concluded that our implementation performed much better on stop signs in comparison with the others. This can be attributed to the segmentation and ROI extraction difficulties that were experienced. Many of the images from the used dataset [10] were very affected by the blue lighting of the sky (refer to Section. 2.2), which greatly impacted the segmentation of blue colours. Furthermore, the noise that resulted from the bad segmentation, and consequently edge detection, was generating a lot of ROIs that would later be misclassified as signals, increasing the rate of false positives.

To conclude, there was also a pattern of traffic signs not being identified whenever multiple signs of the same colour were too close to each other. For example, in a case with a priority and prohibitory sign side by side 6, if the gap between the traffic signs wasn't noticeable, the contours would merge and lead to an undesired ROI. This greatly impacted our results on the prohibitory and priority signs that would often be in close proximity. Another situation in which this was observed was the merger of information signs with the sky.

The annotations in the dataset only contained crosswalk, speed limit and stop signs. Given the fact that our solution can detect other signs beyond those, such as mandatory, priority, and even other prohibitory or information signs, the global accuracy isn't indicative of the true potential of the approach developed.

An example of the output of our system can be observed in Fig. 1, where, (1) result from processing step, (2) result of red colour segmentation, (3) result from edge detector on red colour segmented image, (4) result from ROI extraction of the red edges image, (5) result of blue colour segmentation, (6) result from edge detector on blue colour segmented image, (7) result from ROI extraction of the blue edges image, (8) output image with signs detected annotated over the original image. For more results, consult appendix on Section A.4.

8 Conclusion

In this article, a simplified traffic sign recognition (TSR) system was developed. The system is capable of detecting multiple traffic signs in an image and classify them into one of the five classes: information, prohibitory, stop, priority, mandatory. It uses a colour and shape based approach to tackle the problem, using a multitude of techniques to achieve the result, ranging from processing, segmentation, feature extraction and shape classification. From the experimental results, it can be concluded that there is still much room for improvement.

Namely, in the recognition of prohibitory and information signs. These could be achieved by improving the processing step, that while already partially corrects various defects, such as the "blue filter", isn't completely reliable and may have hampered the potential of our system. As such, other techniques could be further explored, such as the variational contrast-saturation enhancement model proposed by Po-Wen Hsieh and Pei-Chiang Shao [7]. Other segmentation techniques that weren't completely explored could be used to improve our segmentation step, such as the method proposed by Haojie Li *et al.* [8], which takes advantage of Gaussian colour models to achieve a robust segmentation invariant to several lighting conditions. It could help segment red colours that, due to lighting effects, appear as tones of orange. Another critical step that could be improved upon is the ROI extraction, as most problems leading to a bad classification occured in this step. In order to do this, we could further explore the maximally stable extremal regions method [9] and explore techniques to reduce edge noise and better separate potential ROIs with the objective of reducing both false positives and bad classifications due to merging of contours, see Fig. 6. Finally, the shape classification could be improved by exploring feature matching techniques, such as feature matching based on the FLANN matcher, where we could extract feature points from the ROI and then match with a database of feature points. The latter could be obtained from generating the feature points for the traffic signs images in Traffic Rules[1]. Another option found when researching for this project was a method of feature matching based on Hidden Markov Models presented by Jung-Chieh *et al.* [3].

References

[1] Traffic signs portugal, 2022. URL `https://traffic-rules.com/en/portugal/traffic-signs/warning-signs`. Accessed: 2022-05-04.

[2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[3] Jung chieh Hsien, Yi sheng Liou, and Shu yuan Chen. Road sign detection and recognition using hidden markov model †.

[4] OBA] [Daisuke. Python-image-enhancement-with-bright-dark-prior, 2020.

[5] Zaki T. El Baz M. and Douzi H. Red color segmentation based traffic signs detection. *International Journal on Emerging Technologies*, 11, 2020.

[6] Hasan Fleyeh. Traffic signs color detection and segmentation in poor light conditions. In *MVA*, 2005.

[7] Po-Wen Hsieh and Pei-Chiang Shao. Variational contrast-saturation enhancement model for effective single image dehazing. *Signal Processing*, 192:108396, 2022. ISSN 0165-1684. doi: https://doi.org/10.1016/j.sigpro.2021.108396. URL `https://www.sciencedirect.com/science/article/pii/S0165168421004333`.

[8] Haojie Li, Fuming Sun, Lijuan Liu, and Ling Wang. A novel traffic sign detection method via color segmentation and robust shape matching. *Neurocomputing*, 169:77–88, 2015. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2014.12.111. URL `https://www.sciencedirect.com/science/article/pii/S0925231215006712`. Learning for Visual Semantic Understanding in Big Data ESANN 2014 Industrial Data Processing and Analysis.

[9] Hendra Maulana. Traffic sign detection using region and corner feature extraction method. 2021.

[10] Larxel (Owner). Road sign detection, 2020.

[11] Carlos Filipe Paulo and Paulo Lobato Correia. Automatic detection and classification of traffic signs. *Eighth International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS '07)*, 2007.

[12] Zhenghao Shi, Meimei Zhu, Bin Guo, Minghua Zhao, and Changqing Zhang. Nighttime low illumination image enhancement with single image using bright/dark channel prior. *EURASIP Journal on Image and Video Processing*, 2018, 2018.

[13] Richard Szeliski. *Computer Vision, Algorithms and Applications*. Springer Cham, January 2022. ISBN 978-3-030-34372-9. doi: https://doi.org/10.1007/978-3-030-34372-9.

[14] Valdenegro Toro and Matias Alejandro. Fast radial symmetry detection for traffic sign recognition. 2015.

[15] Joviša D. Žunic and Kaoru Hirota. Measuring shape circularity. In *CIARP*, 2008.

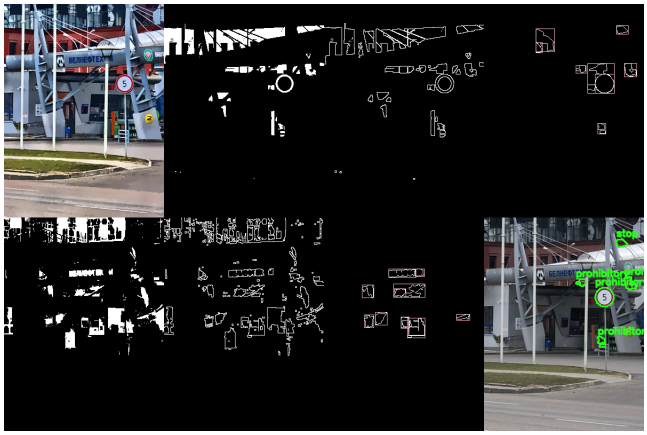
A Appendix

A.1 Processing Results

(a) Original Image (b) Processed Image

Figure 4: Result of processing step on an image with intense blue lighting effect

A.2 Problems with convex hull



(a) ROI Extraction without convex hulls

(b) ROI Extraction with convex hulls

Figure 5: Effects of applying convex hulls on ROI extraction step

A.3 Contour Merging Effect

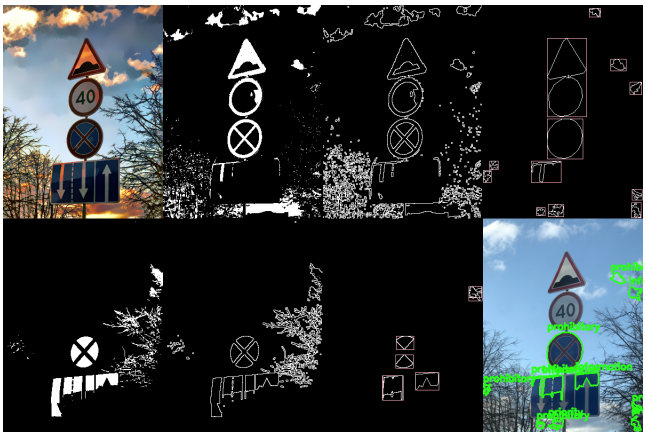


Figure 6: Vertical signs' contours merging

A.4 More results

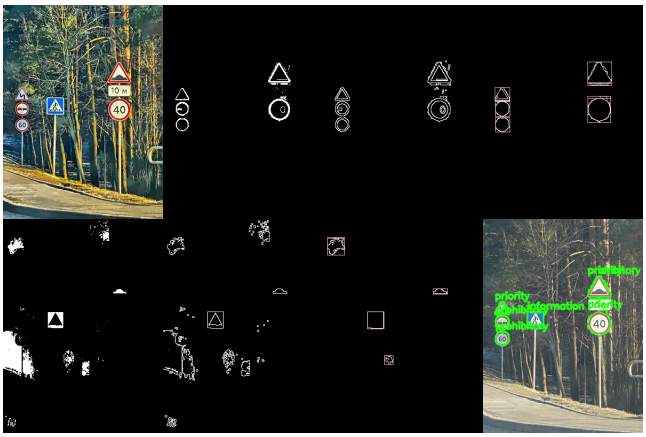


Figure 7: Multiple sign detection

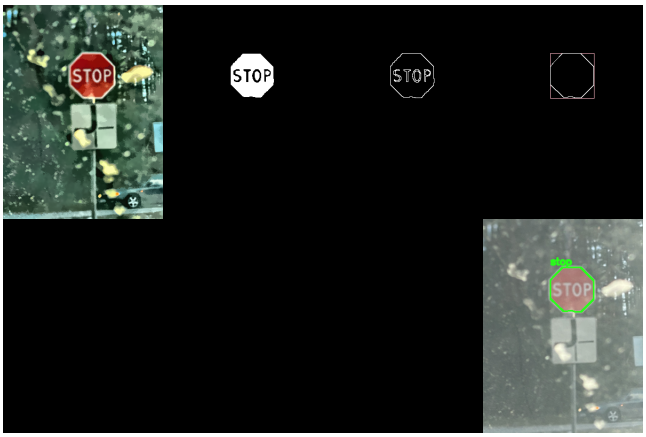


Figure 8: Detection on noisy image

A.5 Shape probability based on corner detection

$$
\begin{aligned}
sqp = max\{ & \\
& clamp(tl+tr+br+bl-0.5\cdot(ml+mr+tc+bc)), \\
& clamp(0.65\cdot(bl+br+tr+tc)-tl-0.5\cdot ml-0.8\cdot(bc+mr)), \\
& clamp(0.65\cdot(bl+br+tl+tc)-tr-0.5\cdot mr-0.8\cdot(bc+ml)), \\
& clamp(0.65\cdot(tl+tr+bl+bc)-br-0.5\cdot mr-0.8\cdot(tc+ml)), \\
& clamp(0.65\cdot(tl+tr+br+bc)-bl-0.5\cdot ml-0.8\cdot(tc+mr)) \\
\} & \\
triangle = max\{ & \\
& clamp(\frac{1}{0.75}\cdot(bl+br+tc)-2.0\cdot(tl+tr)-0.9\cdot(ml+mr)), \\
& clamp(\frac{1}{0.90}\cdot(bl+br+tl)-2.5\cdot(tr+mr)-0.9\cdot(ml+tc)), \\
& clamp(\frac{1}{0.90}\cdot(bl+br+tr)-2.5\cdot(tl+ml)-0.9\cdot(mr+tc)) \\
& clamp(\frac{1}{0.75}\cdot(tl+tr+bc)-1.5\cdot(bl+br)-0.9\cdot(ml+mr)), \\
& clamp(\frac{1}{0.90}\cdot(tl+tr+bl)-2.5\cdot(br+mr)-0.9\cdot(ml+bc)), \\
& clamp(\frac{1}{0.90}\cdot(tl+tr+br)-2.5\cdot(bl+ml)-0.9\cdot(mr+bc)) \\
\} & \\
circle = clamp(tc+bc+ml+mr+0.2\cdot(tl+tr+bl+br)) & \qquad (9)
\end{aligned}
$$

A.6 Shape detection decision tree

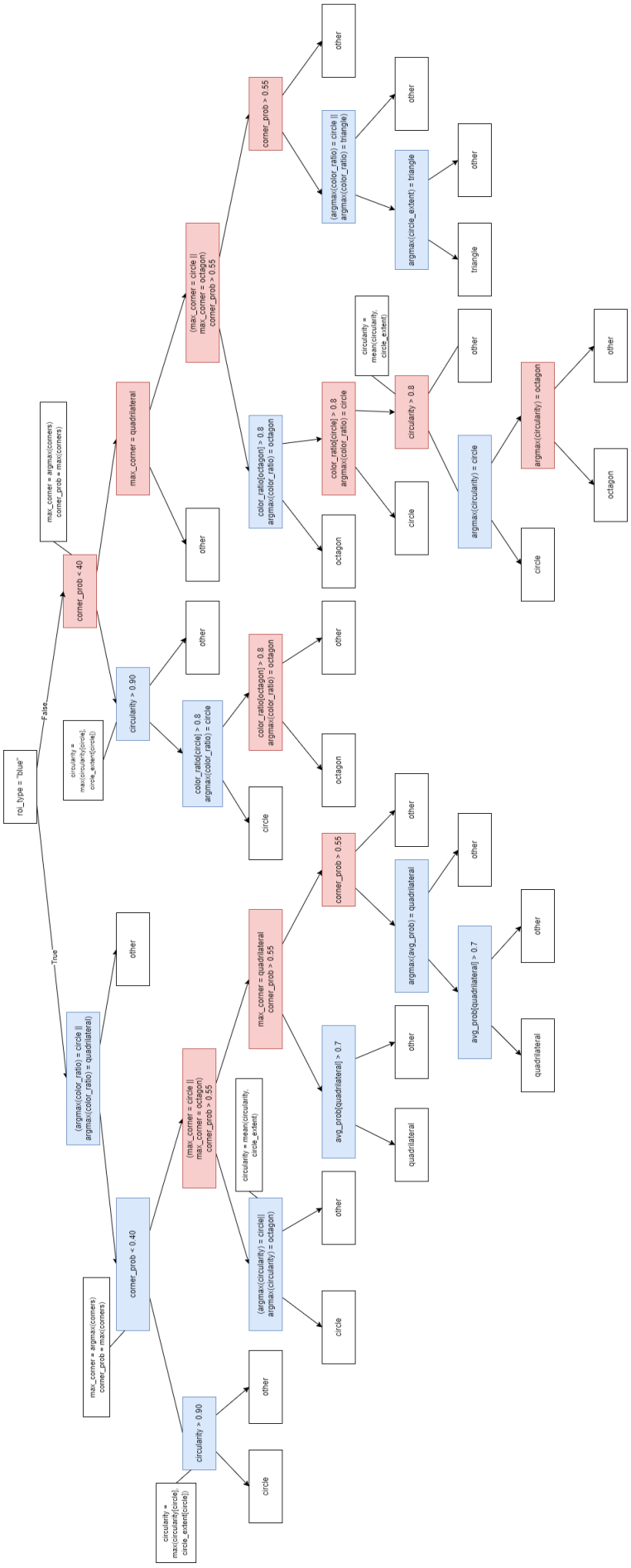


Figure 9: Decision Tree for shape detection