Traffic road signs detection

Image Processing Project Documentation

Student:

* Moisa Oana Miruna
* Pop Ileana

**Group: 30432**

**Contents**

[1 Introduction 3](#_Toc103764090)

[1.1 Topic Context 3](#_Toc103764091)

[1.2 The issues that should be solved 3](#_Toc103764092)

[1.3 Proposed objectives 3](#_Toc103764093)

[1.4 Documentation structure 3](#_Toc103764094)

[2 Theoretical background 4](#_Toc103764095)

[2.1 The bibliographical study from the literature 4](#_Toc103764096)

[3 Design and implementation 6](#_Toc103764097)

[4 Experimental results 6](#_Toc103764098)

# Introduction

## Topic Context

We will provide the application with a set of color images containing traffic road signs with circular, triangular, and rectangular shape.

## The issues that should be solved

We aim to implement a solution that would detect all the traffic road signs appearing in the given images, using image processing algorithms and techniques.

## Proposed objectives

By processing the input images, the result will contain the original scene images in which all the traffic road signs, or areas are marked.

## Documentation structure

In the first part of the documentation, we will present our findings on traffic road signs detection from a theoretical point of view. We will also provide a brief description of the methods that can be applied to achieve this task and decide on the final solution.

Then, we will move on to design and implementation, section in which we will go into further detail about the chosen solution: implementation flow, functionalities, algorithms, and application usage.

In the experimental results section, we will present our practical results and validate them.

Finally, in the conclusions section we will assess the degree of accomplishing the objectives by making observations and thinking about further development directions.

# Theoretical background

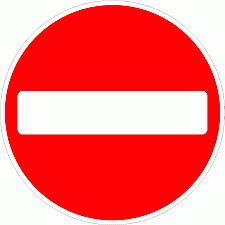
## The bibliographical study from the literature

The principal features of road traffic signs are their color and shape. For now, in our attempt to solve our problem, we will focus on 3 particular shapes:

* Triangular
* Circular





**

The detection of traffic road signs will consist of applying filters on the images.

1. Firstly, a color-based segmentation is used for a detection of red, blue, and yellow objects in the input image.
2. The objects obtained after applying the first filter are then filtered again by means of given geometrical and color conditions valid for traffic signs (e.g. only circles and rectangles are interesting in blue color etc.)

**Color based segmentation approaches:**

There are two possibilities when it comes to color-based segmentation: segmentation in RGB or HSV color space.

* **RGB color space**: Color-based segmentation in such space is simply given by limits of min and max values along all three axis related to R, G and B channel respectively. Each pixel in the input image of color given by a (R, G, B) triple lying inside the mentioned shape is classified as an element of an objects and vice versa. A binary map of pixels memberships’ to particular color is a result of such color-based segmentation.
* **HSV color space**: Segmentation in the HSV color space is very similar to the previous one with only one difference that shape determining an area of object colors is distorted in accordance with a shape of the HSV space.

**Shape segmentation approaches:**

One or more regions of interest are an output of the previous color-based segmentation step. Now classification of its shape has to be carried out. The task of traffic sign recognition is very limited by itself: only circles, triangles, octagonal and diamond shapes are considered as permissible classes of shapes.

The fast radial symmetry detection method (FRS) has been employed for **circles** recognition. This algorithm outputs a grayscale map with a peak at that position, where any circle is present in the input image. Recognition of valid circle-shape object in the input image is then done based on the peak level and its compactness.

The rest of the shapes are distinguished by means of the Harris corner detector. We use the knowledge from the previous steps that object of traffic sign is normalized in the ROI and is straight in it. Under these conditions, the Harris corner detector gives a strong response at the four principal corners of the ROI in case of rectangular and diamonds object and the three corners in case of triangular object. Based on mutual position of these corners the proper class of input object can be specified.

# Design and implementation

The principal function of our application is called *identify\_traffic\_signs.* This is where, for each object in the image, we verify if it represents a valid traffic sign or not by checking it against several functions. We have a function for identifying each traffic sign. (5 functions):

* identify\_access\_restricted
* identify\_no\_entry
* identify\_danger\_warning
* identify\_semaphore
* identify\_cedeaza

Steps for preparing the image:

First thing we do is to generate an array of colors which could appear in a traffic sign: red, blue, black, yellow, green, orange.

Then, we use this array in a ***standardize\_colors*** function meant to bring all pixels being a variation of one of the predefined colors to a pure color. For example, the range we accept for red is r in [100,255], g in [0,50] and b in [0,50]. The rest of the colors remain the same.

The next step is to call ***transform\_image*** function which operates on the standardized image. Here, we traverse the image, pixel by pixel and keep only the pixels that could potentially be part of a traffic sign (have one of the predefined colors). The rest of the pixels are discarded and their value is set to (255,0,255) as this color certainly is not part of the traffic signs identified by our application. For example we keep all the red pixels but the rest (black, white, yellow, green, orange) are kept only if they are inside the red contour.

Then, we apply the ***label\_object***function which is essentially a bfs algorithm exploring 4 neighbors in which we label each potential traffic sign with a label. It is different from colour labelling.

The next step in ***identify\_traffic\_signs*** function is to take each object and verify if it is a traffic sign. We will explain each such function.

In the ***identify\_access\_restricted*** function, we check if we encounter a pixel which is neither red nor white. If this is the case, then we certainly do not have this sign and we the exit the function returning false. Otherwise, we add each pixel in the labeled object to the overall area and each white pixel to the white area. We also need to compute the perimeter for both the red and white parts. We do this by counting the red pixels which have at least one neighbor of color purple (for red contour) and counting the white pixels which have at least one neighbor red. After all the pixels in the object are checked, we compute the thinness ratio (circularity) with the formula . Both red and white areas must be circles so the value should be around one. If this is the case, we return true.

***Identify\_no\_entry function***, we also check the color of each pixel as is can be only red and white. We apply the same method for computing the area and perimeter of the red and white parts as in the function explained above. In the case of the red area the test is the same, the thinness ratio must be around one. The white area should represent a rectangle, i.e, it has the thinness ratio around . If all these conditions are satisfied, we return true.

***Identify\_danger\_warning function.*** Here, we apply the same strategy by checking the color of each pixel (it has to be red, white or black). Then, we need to see if we have to triangles. (one inside the other). In order to see if the shape has three corners, we apply the canny edge detection algorithm. For the black circle inside the triangle, we apply the same formula for detecting circularity.

***Identify\_semaphore*** function detects the triangles as we mentioned above but now, we also need to find three different circles of colors: orange, yellow and green. The circles are found by applying the thinness ratio and the same strategy is applied in case of the red and white triangles.

***Identfiy\_cedeaza function.*** In this function, we also need to identify the triangles with the same method, but we need to make sure that the triangle is upside down. We do that by comparing the x coordinates of the corners array. If we have two coordinates with a minimum value and one grater, than the sign is detected.

Observation: In order to enhance the user experience we decided to mark each identified traffic sign with a blue rectangular border.

# Experimental results

In this section, we will present the practical results obtained for all of our test images in order to prove the accuracy of our application.

1. **Detection of the “NO ENTRY” (“Interzis”) traffic sign**

*O imagine care conține text, iarbă, arbore, exterior

Descriere generată automatOriginal image:*

The traffic sign is correctly detected:

O imagine care conține text, arbore, iarbă, exterior

Descriere generată automat

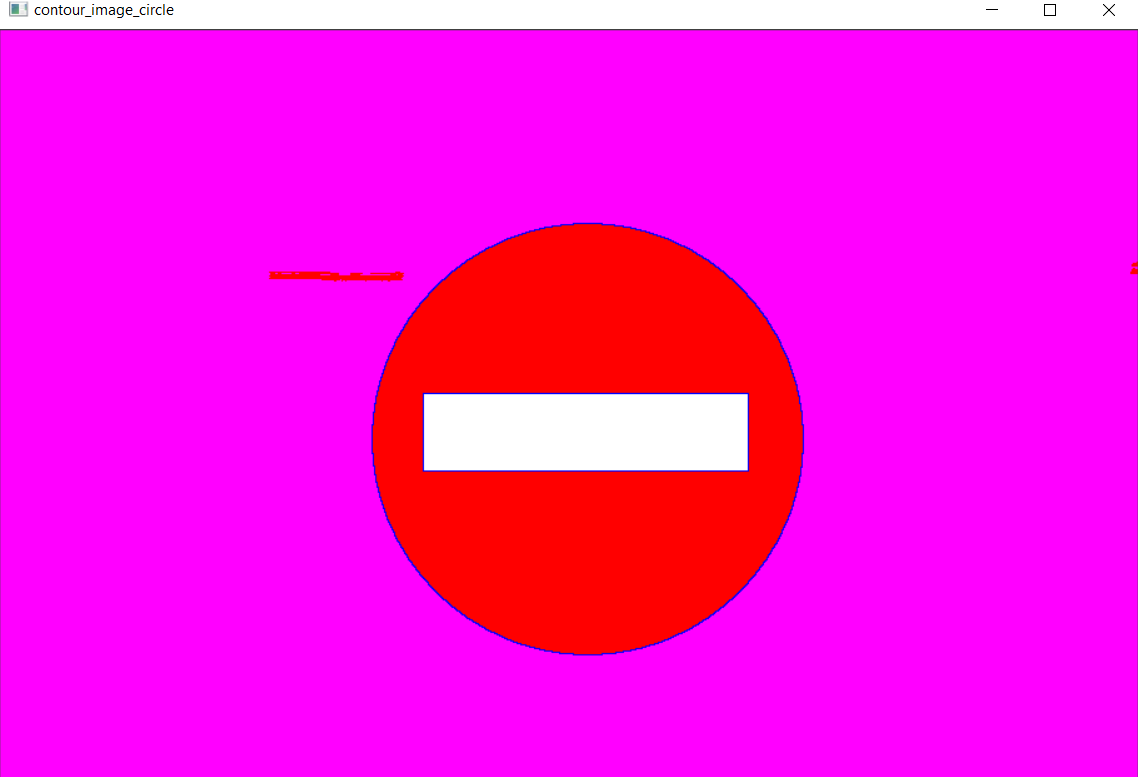
Steps to detection:

1. O imagine care conține text, arbore, iarbă, exterior

   Descriere generată automatThe colors are standardized. Here, the shades of red are all brought to (255,0,0) pure red.
2. Further, only the colors that could be part of a traffic sign are kept and the objects are labeled. In this particular case, there are other red pixels in the image that are detected and labeled and will undergo the detection process but will end in failure.



1. Find the contours of the objects for computing the perimeter.



O imagine care conține text

Descriere generată automatAll functions for detecting traffic signs are applied to the image, until a match is made.



The thinness ratio for the red part is exactly 1, meaning that we have a circular traffic sign. The circularity of the white part is close to the typical value for rectangular shapes, meaning that we have a “no entry” sign.

1. *O imagine care conține cer, exterior, sol, drum

   Descriere generată automat***Detection of the “All vehicles prohibited” (“Circulatia interzisa in ambele sensuri pentru toate vehiculele”) traffic sign and “Danger warning” (“Zona cu risc ridicat de accidente”)**

*Original image:*

O imagine care conține text, cer, exterior, sol

Descriere generată automatThe traffic signs are correctly detected:

Steps to detection:

1. Color standardization

O imagine care conține text, cer, exterior, sol

Descriere generată automatb

1. Object labeling



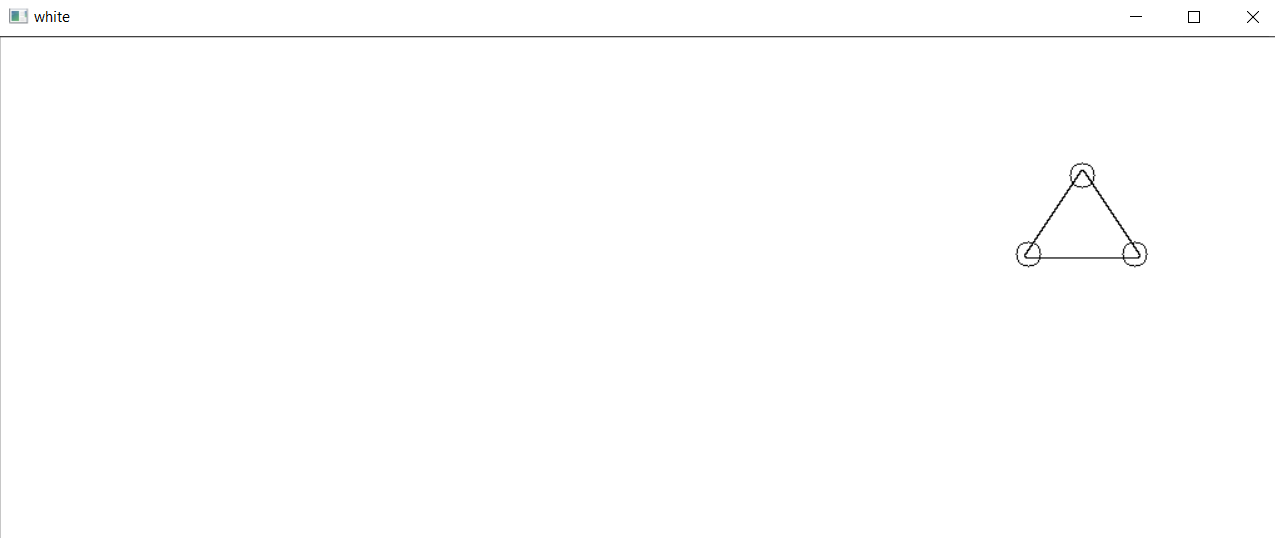
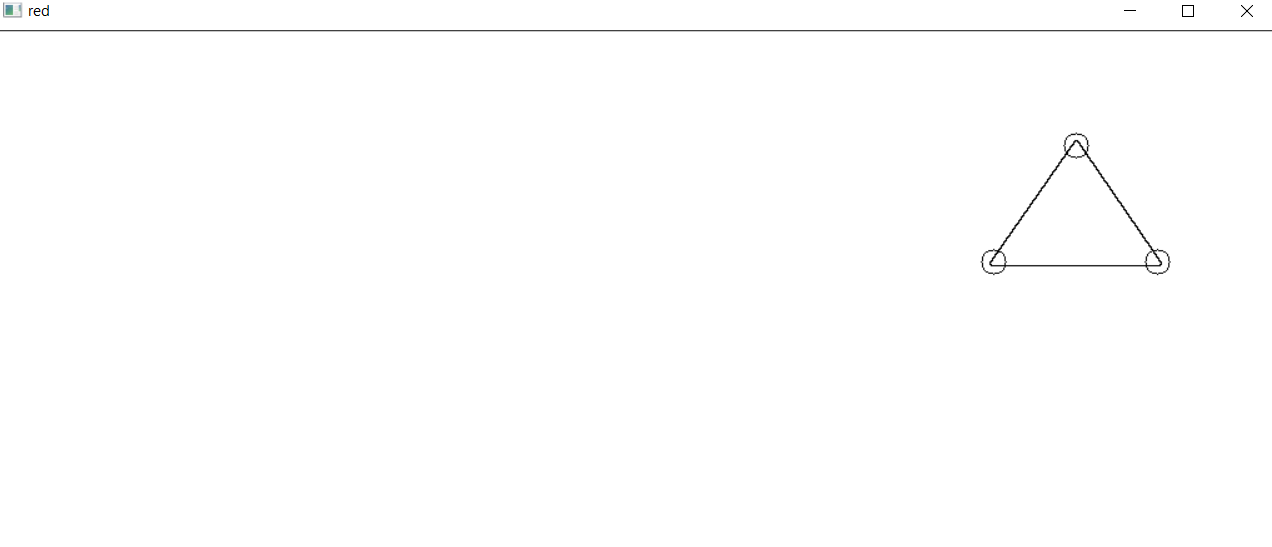
1. Contour detection





1. Triangle detection

white triangle: red triangle:



O imagine care conține text

Descriere generată automatO imagine care conține text

Descriere generată automatAll functions for detecting traffic signs are applied to the image for each labeled object, until a match is made.

1. **Detection of the “Semaphore” (“Atentie, semafor”) traffic sign**

*O imagine care conține text, exterior, cer, drum

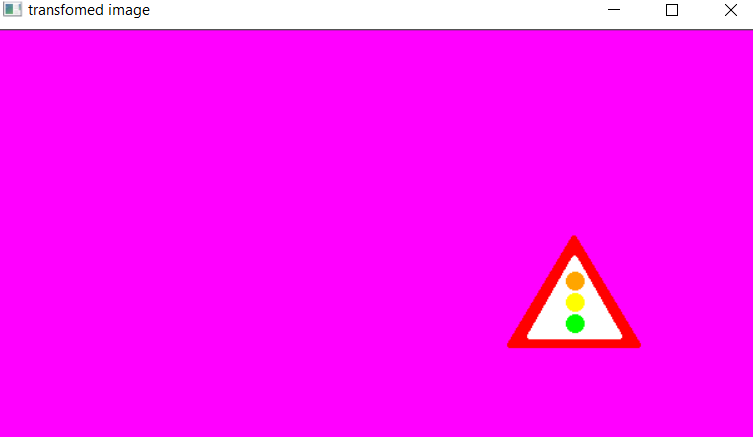
Descriere generată automatOriginal image:*

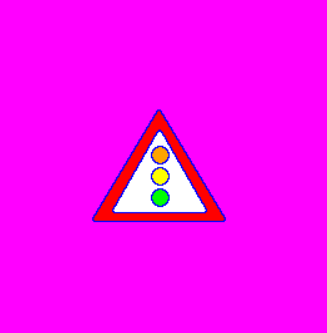
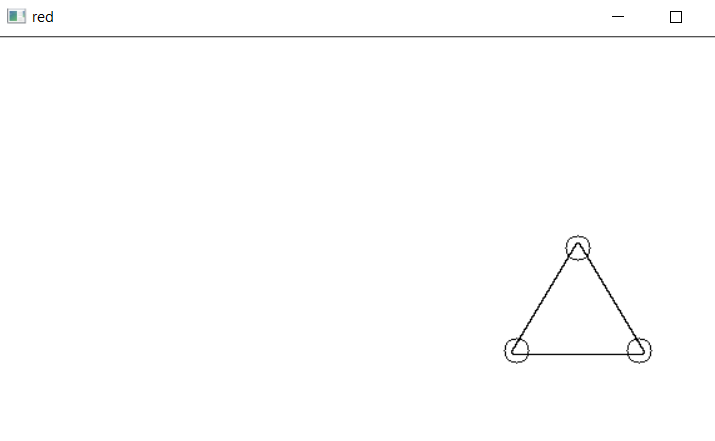
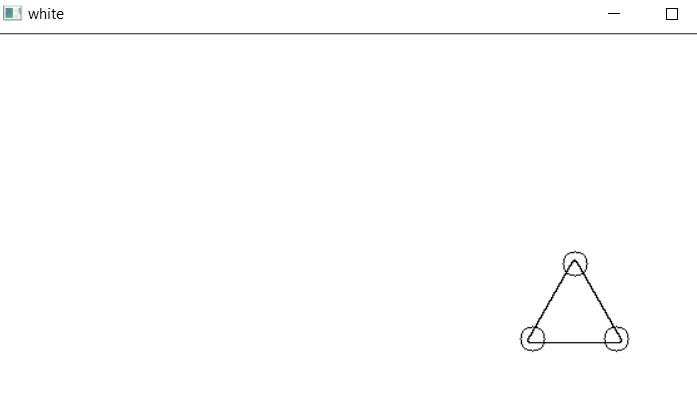
O imagine care conține text, cer, exterior, scenă

Descriere generată automatTraffic sign detected:

Steps to detection:

1. O imagine care conține text, exterior, cer, drum

   Descriere generată automatColor standardization
2. Object labeling
3. Contour detection d) Triangle detection



O imagine care conține text

Descriere generată automatAll functions for detecting traffic signs are applied to the image, until a match is made.

O imagine care conține text

Descriere generată automat

1. **Detection of the “Give way” (“Cedeaza”) traffic sign**

*Original image:*

*O imagine care conține text, cer, exterior, stradă

Descriere generată automat*

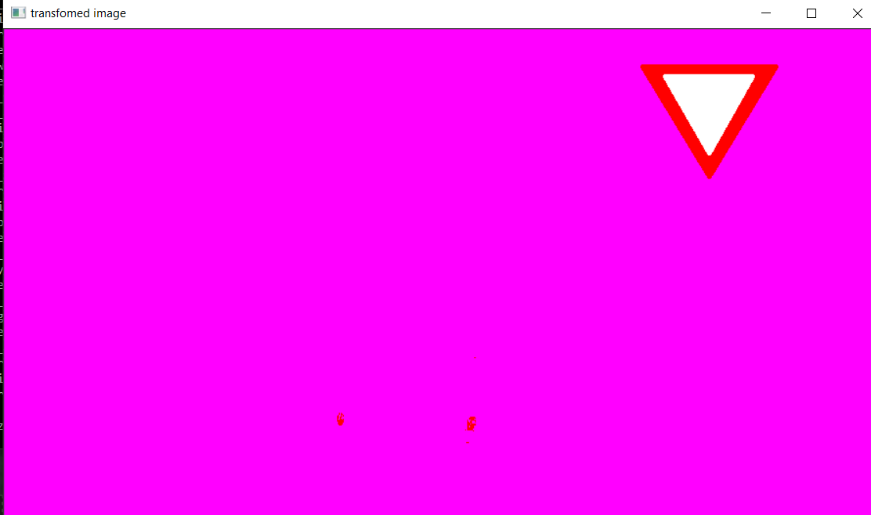
O imagine care conține text, cer, exterior, stradă

Descriere generată automatTraffic sign is correctly detected:

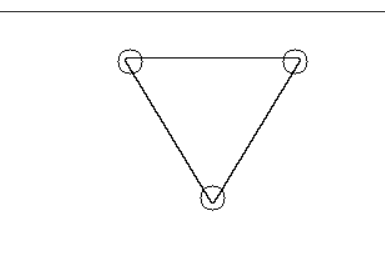
Steps to detection:

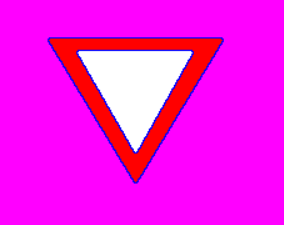
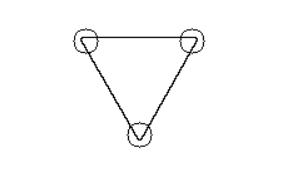
1. O imagine care conține text, cer, exterior, stradă

   Descriere generată automatColor standardization
2. Object labeling



1. Contour detection d) Triangle detection





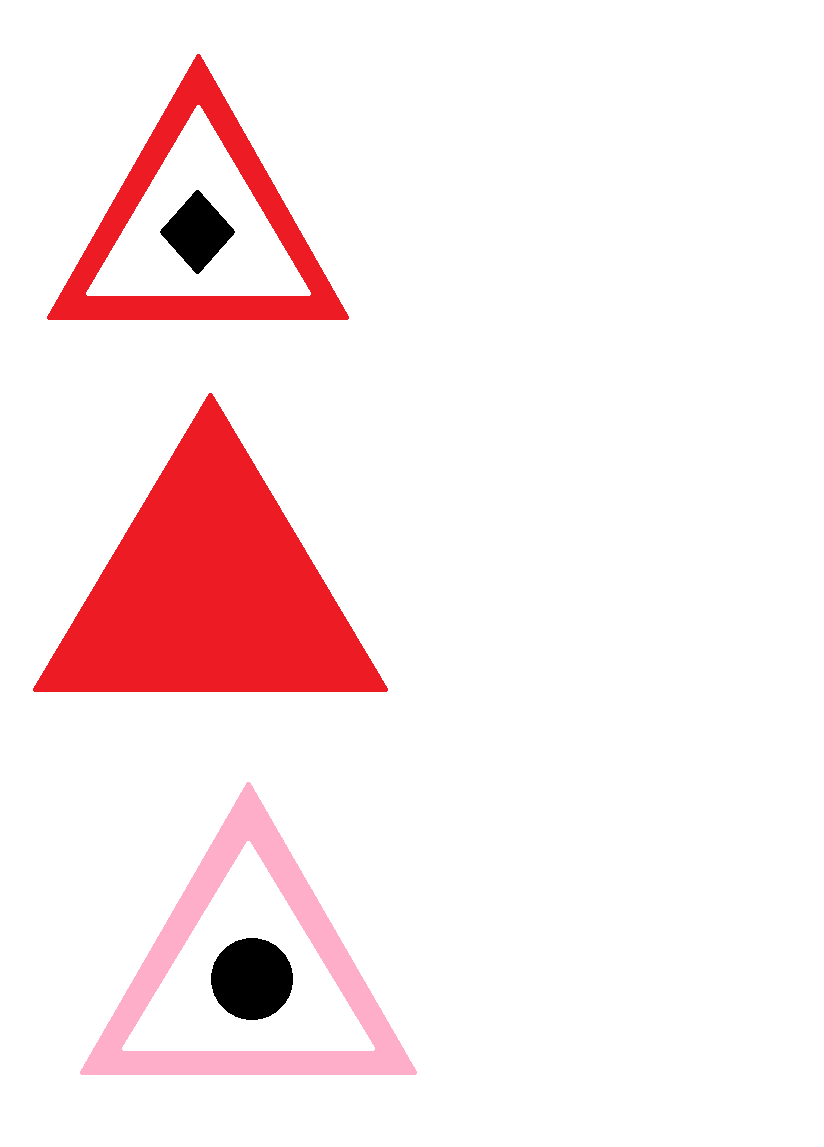
All functions for detecting traffic signs are applied to the image, until a match is made.

O imagine care conține text, portocaliu, set, închidere

Descriere generată automatO imagine care conține text

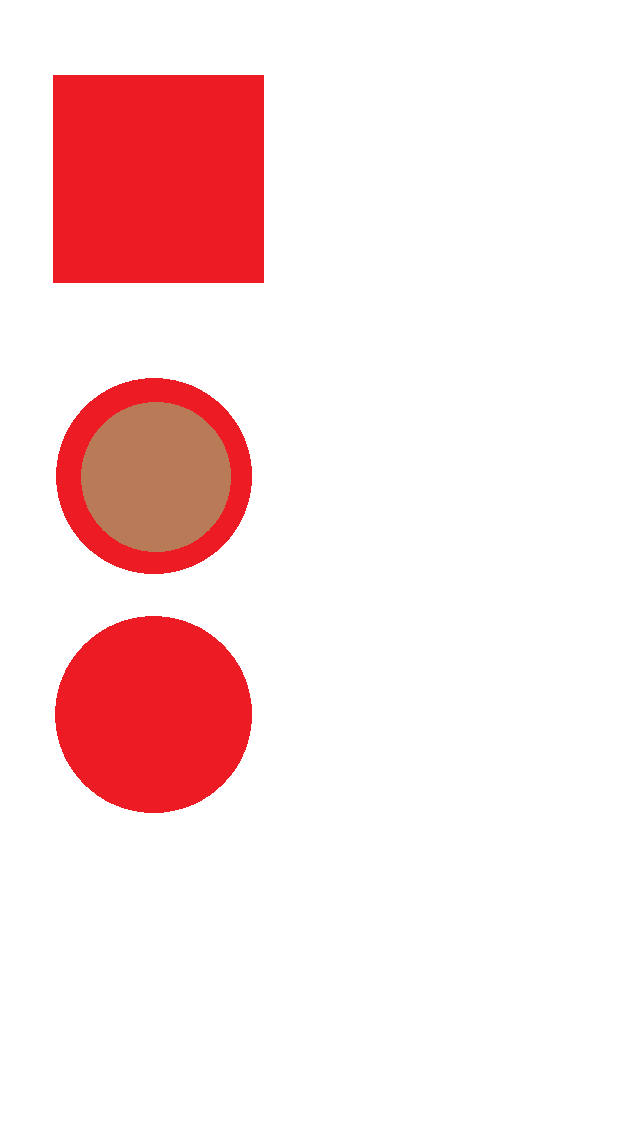
Descriere generată automat

1. **Wrong input images**



O imagine care conține text

Descriere generată automat

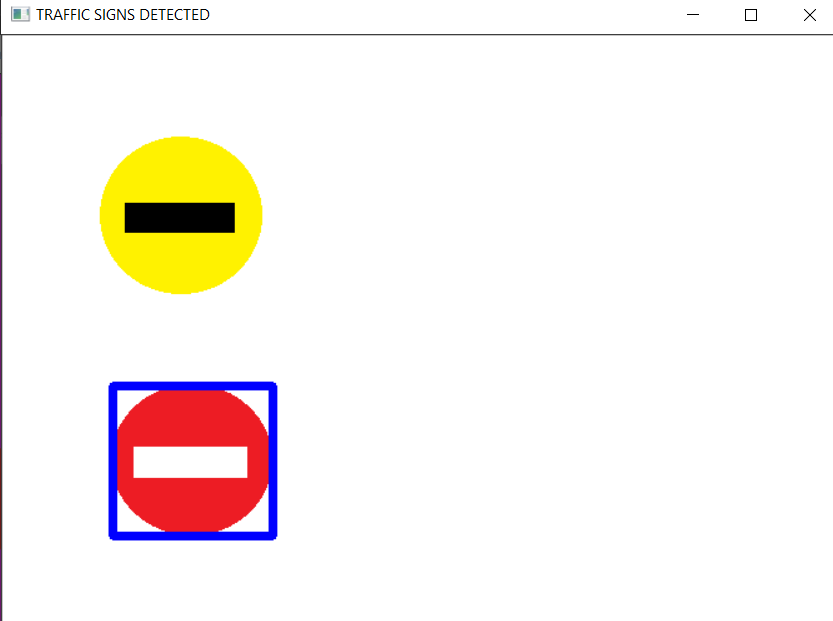




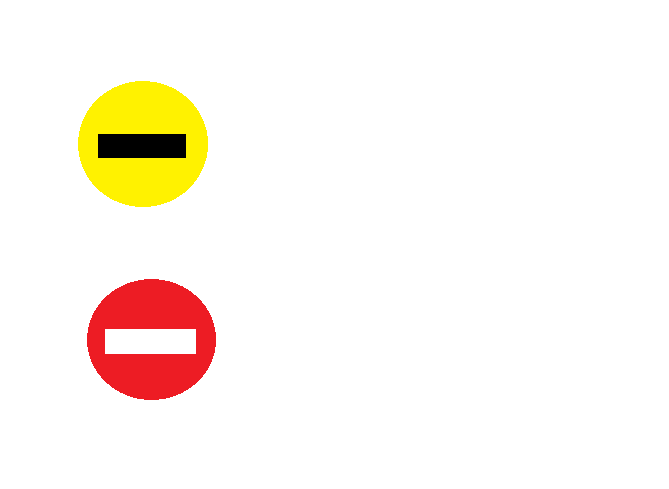
O imagine care conține săgeată

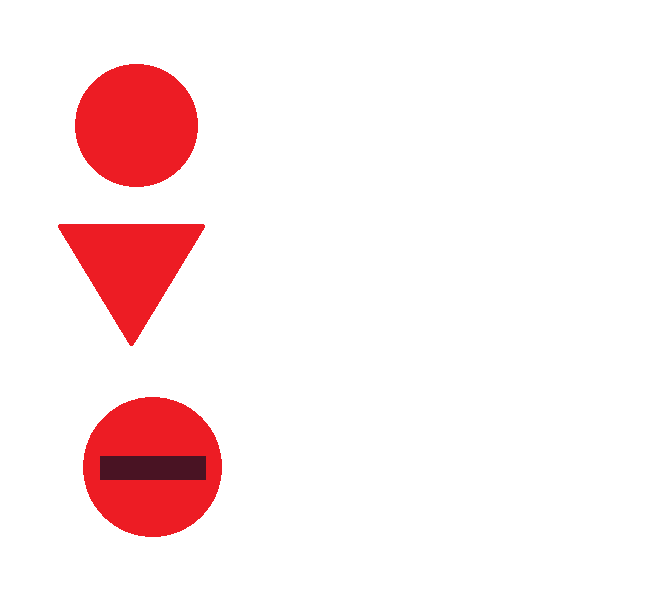
Descriere generată automat





O imagine care conține text

Descriere generată automat





# Conclusions

As far as achieving our objectives, we are very satisfied with the final result. We have realized along the way that the detection was far more complex than we have anticipated and there were a lot of steps and strategies that needed to be applied. Despite this, we tried to apply our image processing knowledge gained from the laboratories and we have succeeded in implementing all functionalities writing from scratch all the functions. We took concepts and inspiration from the literature, but the actual implementation is entirely our personal contribution.

Regarding further development, the first thing that comes to our minds is to expand the set of traffic road signs detected by our application, even in poor weather conditions.