SAPIENZA
UNIVERSITÀ DI ROMA

AILab: computer vision and nlp

drAIve:
crosswalks recognition algorithm

Barbalata Radu Ionut 2002688
Dalla Ragione Ivan 2009768

Contents

1 Introduction

Observing the wide variety of driver assistance systems, we noticed that few support systems alert the user to the presence of crosswalks.
Inspired by the reading of this paper [1], which demonstrates a Yolov5-based network implemented on a Jetson device, we developed a different model that relies on image filtering and pattern recognition to identify crosswalks.

2 Methods

In analyzing crosswalks, we focused on the factors that characterize them, namely the white color and rectangular shapes in sequence. But after closer analysis, several variables that influence them arose:

1. Different brightness and colors: due to the difference in light between day, night and general surroundings. This results in seeing their color as white and light in some cases and much darker in others. The camera and the quality of the images also often alter the colors, giving rise to several problems that we will analyze later.

2. Different sizes: due to the structure of the road, paint that is not always sharp and missing in some places, or intersecting with other signs on the asphalt, it is not always easy to trace the stripes back to a succession of rectangles

2.1 First Approach

The first approach was to define the threshold [2] based on the colors that most closely resemble white through HSV color representation, which would allow us to easily find the colors.
However, this approach did not allow us to distinguish the crosswalk from the road in cases of high illumination or darkness.

In this first example, the lights are very good and the crosswalks are clear, so also this simple filter works well:

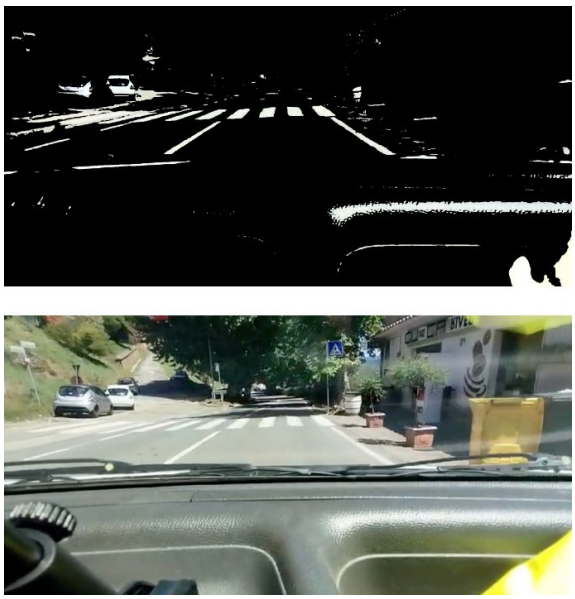

Figure 1: first approach example

As of easy intuition, as soon as the image becomes darker, so with very dark lights and dimly lit strips, the first approach fails the threshold by referring to sources that emit a lot of light:

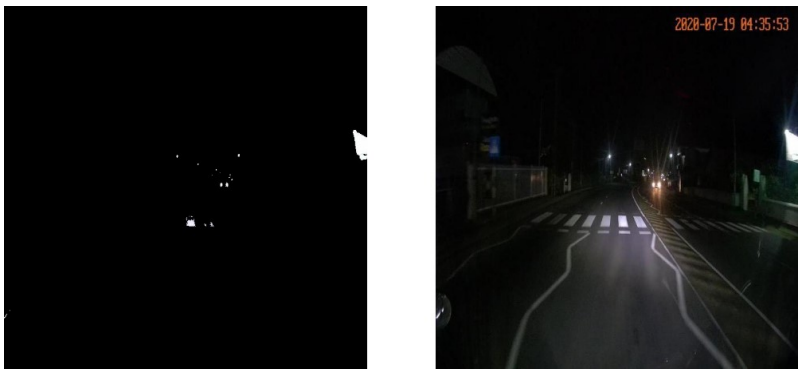

Figure 2: first approach night example

2.2 Second Approach

Looking at the results of the first approach, we realized that brightness is a decisive factor in crosswalks recognition, so we went on to analyze the Lab [3] representation of colors, more specifically the L channel, considering that it represents precisely the degree of brightness

We then discovered that crosswalks always consist of colors that are among the brightest, consequently, we tried two different ways:

1. Searching for the brightest absolute maximum: it looks for the brightest color and applies the threshold based on it, but it's not accurate enough, as in some frames with high brightness other elements might be brighter than the crosswalks, similar to the first approach.

2. Search for the brightest relative maximum: it looks for the brightest color with at least a given number of pixels and applies the threshold with a range of 10% error from the relative maximum, but the problem arises in the crosswalk size, which depending on factors such as proximity or type of shot, this makes the crosswalk size vary greatly and makes it very complicated to define a minimum correct pixel number. Also, in situations where the crosswalks are dimly lit, they might be excluded from the error range if other lights are present.

At the end we decided to test using the second way.

In this example we test this new approach on the same figure (figure2) where the crosswalks were not recognized previously, a small improvement from the first approach is observable, but still not precise enough

Figure 3: first and second approach night example

2.3 Third Approach

Given the previous cases, we wondered if there was a way to group objects based on a set of colors and decided to use a clustering approach: our model defines a number n of classes, based on the average brightness of the photo and applying the k-means [5] algorithm on the image.

The higher the average brightness reaches, more classes are defined.

The reason we increment the number of classes based on mean brightness are multiple:

- When there is less light, the crosswalks will also have less light to absorb, so they will appear darker. However, the entire environment would be darker except where there is a light source. This implies that with less classes, the crosswalks might not belong to the most illuminated class and therefore would not be detected.
- When there is more light, it will be easier to see the crosswalks. However, there will also be other colors that, with few classes, would be grouped together with the crosswalks, making it impossible to distinguish anything

It's possible to see an important improvement using this approach

Figure 4: first, second and third approach night example

2.4 Final implementation

The final implementation based on the third implementation performs the following steps:

1. A region of interest [4] is defined by cropping 1/3 of the image to remove unnecessary parts like the background.

Figure 5: Roi application

2. A K-means algorithm is applied with a variable number of clusters based on the average brightness of the L channel in LAB color representation.

Figure 6: K-means application

3. The cluster with the brightest color is turned into white and the rest into black and is applied a morphology [6] that allows us to eliminate noise such as small pieces of black in the middle of white.

Figure 7: threshold and morphology application

4. Contours are detected, and then we try to understand which one can be a part of the crosswalk, so given width w, height h, and area A of a contour we consider the ones that satisfies the following condition:

   - $0.15 \leq$ w/h ≤ 7
   - $h \geq 10$
   - $60 \leq A \leq 8000$

5. **Groups creation**: for each contour, a group is created, and we determine which contours belong to the same group. A contour is part of a group if it respect the following criteria on the respective group:

   - the height of the contour under consideration should be equal to the average height of the group, allowing a 2% tolerance
   - the width of the contour under consideration should be equal to the average width of the group with a 3% tolerance.
   - the width of the contour under consideration should be equal to the average width of the group with a 3% tolerance.
   - the center of the contour height must be at the same height as the average of the center of the height of the group contours with a tolerable error of up to 2%
   - the area bounded by the contour must be equal to the average of the areas in the group with a 60% tolerance

6. **Box creation**: The box is created based on the groups' boxes, and, if two or more groups' boxes intersect, then they will create a single box, allowing us to group different groups together.
   The group's box will be drawn only if there are at least three distinct contours in the group.

Figure 8: Defined contours and created boxes

3 Results

3.1 Test with positive results

The results in their entirety can be seen in the following **drive**, where is uploaded also a video with the model applied. Images from the following dataset [7] were used for our tests.

Some successful tests are shown here:

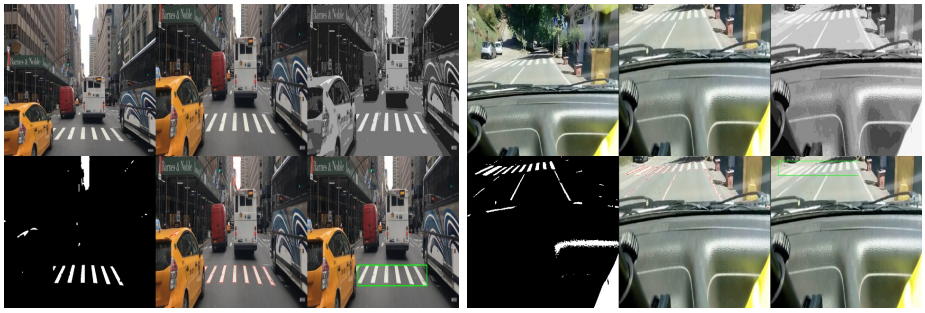

Figure 9: test 1 & 2

Figure 10: test 3 & 4

3.2 Test with negative results

There has been a significant improvement but still, there are images that have problems:
Images with high brightness, where even the low quality makes everything look too much like white

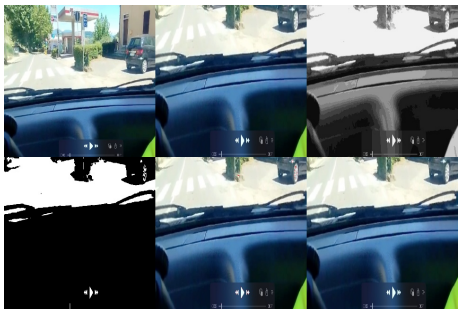

Figure 11: undetected 1

Photos with ruined crosswalks and undefined edges, where the group for the box building is not recognizable

Figure 12: undetected 2

Very dark photos with strong external light sources, which raise the bright average and in case of unlit crosswalks, lead the K-means algorithm to put them in very dark classes, which will then be transformed to black

Figure 13: undetected 3

References

[1] **CDNet:** a real-time and robust crosswalk detection network on Jetson nano based on YOLOv5

[2] **Threshold**

[3] **Lab color space**

[4] **Region of interest**

[5] **K-means**

[6] **Morphological Transformations**

[7] **Crosswalks Dataset**