ENGDATA201 – Image Processing and Computer Vision

Project 3 - Practical Research Project:

Recognition of Road Signs

Authors:

Joanikij Chulev

Maxime Chopin

May, 2024

Professor:

Dr. Ir. Frank
van der Stappen

Abstract

In this project we create three pipelines for road sign detection. The first trading efficiency for speed, which is an important aspect of a road sign detector as it would need to be applied in a changing and high-speed environment. The second pipeline focuses on a more balanced approach. And the last of the three adds a grid search with a scoring function to the second pipeline. This essentially puts all of its effort into performance at the cost of computational power and storage being used for the comparisons, grading and grid search. All three were tested on different images, with different environments, lighting, numbers of signs and angles to address system performance.

Contents

Experimental platform

Hardware technology platform:

OS Name: Microsoft Windows 11 Pro - Windows version: 22H2

System Type: x64-based PC

Processor: Intel(R) Core (TM) i7-1065G7 CPU @ 1.30GHz, 1498 Mhz, 4 Core(s), 8 Logical Proc.

Installed Physical Memory (RAM): 4.00 GB

GPU: Integrated Intel GPU (Intel Graphics)

Software technology platform:

Spyder version: 5.4.3 (conda)

Python version: 3.10.17 64-bit

Qt version: 5.15.2

PyQt5 version: 5.15.7

Operating System: Windows 11

Library - OpenCV: cv2.

1 Introduction

Road sign recognition is a crucial component of modern autonomous driving and intelligent transportation networks. Accurately identifying and interpreting road signs enhances road safety, facilitates navigation, and is inevitable for compliance with traffic regulations. Generally, road sign recognition technologies have been directed by the need to improve automated systems in the dynamic environment that are roadways. These changes and techinques are possible due to the advancements in image processing and computer vision. Work in this field previously has largely centered around using computer vision and image processing to find and classify different road signs. Machine learning models, especially those that use convolutional neural networks (CNNs), have been used due to them handling the tasks, and complexities associated with varied lighting conditions, occlusions, multiple sign designs and other factors that may get in the way of a straightforward sign recognition.

European road signs have multiple classes such as: warning signs, prohibitory or restriction signs, mandatory signs, or informatory signs. Among these, prohibitory signs, which typically feature distinct geometric shapes such as circles, and are characterized by their striking red borders, are critical for roadway safety and regulation.

This project focuses specifically on the recognition of prohibition signs. These signs are easily identifiable by their red annulus (a red ring) with a white background and an additional symbol indicating the specific prohibition. In the Netherlands, as referenced from an overview of Dutch road signs, relevant examples include signs denoted as A1, C1, and series C6 through C22a, among others. Our goal is to develop a pipeline capable of detecting these prohibition signs within images. This pipeline will be designed to recognize the presence of red annuli without the need to classify the specific type of prohibition depicted by each sign. This simplification allows for a lighter load on the system and no Machine Learning models to be involved. By focusing on this particular class of signs, the project aims to detect this class of signs consistently, correctly and fast.

Figure 1: Example of a Prohibition traffic Sign.

2 Prohibition Sign Detection Pipelines

2.1 Implementation – HBEHC

In the construction of a unique custom pipeline for the recognition of prohibition road signs, which are characterized by their distinctive red annuli, a multi-faceted approach leveraging both color segmentation and geometric transformations was developed. This pipeline incorporates a series of selected preprocessing and detection steps aimed at maximizing the efficiency and effectiveness of the sign recognition process, while taking inspiration from our reasoning of a logical process for solving this task. It incorporates: Hue thresholding + Blur + Edge Detection + Hough Circle Detection, hence the acronym HBEHC.

The pipeline initiates with the conversion of RGB images to grayscale. This step reduces the complexity of the image data, focusing on luminance rather than color, which is beneficial for enhancing the visibility of structural features like edges and may help with lighting invariance. This change reduces computational load in later steps but also decreases the amount of data that needs to be managed, making the data stored and computational load more manageable.

At the same time, the images are converted to the HSV (Hue, Saturation, Value) color space, which is particularly suited for color-based segmentation tasks and has adjustable luminosity. This conversion is needed for the effective isolation of red hues, which are necessary for finding the specific class of prohibition signs. Color segmentation in the HSV space means defining thresholds that can isolate shades of red regardless of some potential variations due to lighting conditions or partial occlusions. By applying masks for these thresholds, the pipeline focuses only on areas of interest that are likely to contain the signs, avoiding false positives.

After isolating the red components, a Gaussian Blur is applied. This step is for noise reduction, it smooths out the image, reducing the impact of minor color variations and other visual noise, which could otherwise lead to incorrect edge detection. The choice of Gaussian Blur balances the need for a smooth image against the loss of edge clarity, ensuring that the essential features of the prohibition signs remain detectable.

Edge detection is performed with the Canny algorithm. The Canny edge detector is known for its efficacy in detecting a wide range of edges in images. It identifies areas of high gradient where color or intensity changes sharply. The thresholds used for the Canny detector are important, as we want to avoid unnecessary visual noise from being detected and cannot miss. The edges detected serve as the basis for the Hough Transform, which is used to identify the circular shapes of the prohibition signs.

The Hough Circle Transform is a specialized version of the Hough Transform modified for detecting circles. Parameters such as the resolution of the accumulator space, the minimum distance between circles, and the radii range considered must be chosen based on the awaited size and spacing of the signs in the image. The computational cost of the Hough Transform is significant. Because it involves a comprehensive search of possible circle centers and sizes.

Hence, optimizing these parameters are paramount for maintaining performance without giving up accuracy. Furthermore, we dynamically adjusted the parameters based on the input image dimensions. Which allows for scalability and responsiveness to different image resolutions.

This pipeline's design emphasizes simplicity and efficiency, making it well-suited for integration into real-time systems where computational resources and response times are extremely important. Each step in the pipeline, from color segmentation to circle detection, is made with the goal of minimizing unnecessary computations and focusing resources on the most promising parts of the image. As this is done in real time with a fast-moving camera point speed of detection needs to be good. This approach not only improves the speed and efficiency of the detection process but also makes sure that the system remains stable across varying operational conditions and settings. Overall, the pipeline represents a balanced approach to the challenge of road sign recognition, addressing key issues such as computational load, storage requirements, and operational simplicity. You can see some results in the figure below.

Figure 2: Detection results using our first implementation - HBEHC approach.

2.2 Implementation - CSMDHC

In their recent study, Yakimov and Fursov developed a comprehensive Traffic Sign Recognition (TSR) system that enhances both detection accuracy and processing speed through several innovative methods [1]. Recognizing the detrimental effect of noise on detection accuracy, the authors introduced an advanced noise reduction algorithm. Which was not a feature we thought of adding before. They also utilized a lot of different color segmentation thresholds for the hue of red. Finally, they implemented a Modified Generalized Hough Transform approach for recognizing multiple sign types, although for our task this is not necessarily needed [1].

Inspired by their work, we made a pipeline that uses Color Segmentation + Morphological Denoising + Hough Circle Detection.

The process initiates with the transformation of the image into a format that isolates key features necessary for sign recognition. By converting the image into the HSV color space, the pipeline effectively separates color information from intensity like in the previous pipeline. This allows for precise segmentation of red hues. In turn those help with identifying the specific class of prohibition signs. This method is not only computationally efficient but also robust against variations in lighting and environmental conditions, ensuring consistent performance across different scenarios.

Following color segmentation, our second pipeline incorporates a noise reduction step once again. This involves applying morphological operations to the segmented image, smoothing out irregularities and erasing small artifacts that may get in the way of the detection process. The function defines a kernel, a 5x5 mixed matrix of ones, which functions as a structuring element influencing the area of the closing operation. Morphological closing itself consists of two actions: dilation followed by erosion. Dilation first expands the brighter regions of the image, which helps with closing small holes and connecting nearby areas. Thereafter, erosion is applied, which shrinks the now expanded areas back to their original sizes but keeps the newly closed gaps in the structure. This sequence effectively eliminates small holes and noise from the image without significantly altering the proportions of prominent features, making it adequate for preparing images for more precise feature detection like edge detection or circle detection in subsequent steps.

The detection of circular patterns, which are characteristic of the prohibition signs, is achieved through a modified Hough Transform. This technique identifies geometric shapes in a image and is particularly useful for circle detection. By carefully tuning the parameters of this transform, the pipeline achieves high specificity in detecting the rounded shapes of the signs, even in the presence of partial occlusions or varying sign sizes. This step, while computationally more demanding, is essential for the accurate localization of signs, enabling precise and reliable recognition.

The final step of the pipeline involves sticking the detected circles over the original image to provide visual confirmation of the results. This helps with verifying if the detection is accurate but also serves practical purposes, such as informing the user or a system that may want to view these results and check how accurate they are.

Together, these steps form our second pipeline to effectively balances the trade-offs between detection accuracy, computational demands, and storage efficiency. By focusing on the most critical aspects of the image processing needed for sign detection, this pipeline ensures optimal performance. This is achieved without less complexity than often associated with such systems. Again, simplicity is our aim and aid. It helps to facilitate ease of integration and operation in diverse environments. Thus, this pipeline stands out with better performance than our first pipeline, although a slight bit more computationally intensive. Below we can we clear superior performance to our first implementation.

Figure 3: Detection results using our second implementation - CSMDHC approach.

As we can see this method has far better performance, in fact so good that in some pictures the green outlines are so precisely put on the perimeter of the image, they are not as visible. Although we assure, they are there if you zoom in.

2.3 Grid Search Implementation

This advanced pipeline for detecting prohibition signs significantly expands upon the foundational principles of image processing explored so far by incorporating adaptive parameter tuning and a scoring system. This scoring system is to evaluate the detection accuracy.

For the Hough Circle Transform, this pipeline employs a grid search methodology using Parameter Grid Search to systematically explore a range of parameter setups and combinations. This method allows the algorithm to adapt to different environments by selecting the optimal parameter set that maximizes the detection score. This feature represents a significant shift towards a more dynamic and responsive detection system. However, this will cost more computational power. The parameters include the resolution of the accumulator in the Hough Transform (dp), the minimum distance between detected centers (minDist), the upper and lower thresholds for the internal Canny edge detector (param1 and param2), and the minimum and maximum radii of circles to find (minRadius and maxRadius). This search is designed to identify the most effective combination of parameters for circle detection in different conditions, adjusting the detection mechanism to specific characteristics of the input image.

Furthermore, adding the custom scoring function and calculate_score, makes a big improvement. This function grades the quality of detected circles on how well they correspond to what we'd expect from a prohibition signs. It evaluates each detected circle by checking the boundary conditions using check_boundary, which inspects the perimeter of the detected circle for a sufficient number of red pixels. Thereby ensuring that the detection was relevant. The scoring mechanism also penalizes scenarios in which multiple circles are found for a single sign, addressing the issue of over-detection noticed in some cases with our first pipeline. This step increases the precision of the pipeline even more.

This pipeline's complexity comes at a cost. More specifically in terms of computation, it increases due to the iterative testing of parameter sets and the evaluation of detection results against a scoring system. This not only requires more computational time but also increases the demand on processing resources. Nevertheless, this trade-off is worthwhile in some cases because of the potential for significantly improved accuracy and adaptability in detection.

In terms of storage, the algorithm's requirements are weighed down by the need to store multiple configurations and intermediate detection results for comparison, unlike simpler, single-pass detection systems.

3 Captcha tests

We also experimented with some captcha images to test if our detection methods will bypass the anti AI measures. We did find somewhat satisfactory results with our detectors. See examples.

Figure 4: Detection results of Captcha verifiers using our second implementation - CSMDHC approach.

4 Conclusions

In conclusion, as expected the HBEHC model was the most resource efficient. However, it came at the cost of performance. It was a bit less accurate, had some false negatives and a few false positives. Despite this making it unsafe, a pipeline that prioritizes speed and efficiency may be the best in some cases where the scenery changes too fast for other models, or in cases where the hardware can't manage more demanding pipelines in real time. The second pipeline CSMDHC which put more emphasis on accuracy was also as expected far more accurate. This was managed while still being considerably computationally cheap. It was impressively reliable and as it is a good middle ground between speed and maximum accuracy is likely the most applicable one of the three. It proved to have almost perfect accuracy. The last pipeline was a modification of the CSMDHC pipeline with a grid search and scoring system to choose the optimal parameters for every scenario. This was the maximum accuracy model and traded off much more resource intensity than we were willing to allow in the other two pipelines. This gave the best results although past a certain accuracy we have diminishing returns. The standard CSMDHC was very accurate and with the addition of the grid search and scoring system the improvements weren't as big as between the CSMDHC models. Despite that missing even a single road sign could be fatal if these models were applied in real time so this might be the model that is necessary, despite being costly in computation, storage and time.

View the source code here - https://github.com/JoanikijChulev/Prohibition-Sign-Detection-IPCV

References

[1] P. Yakimov and V. Fursov, “Traffic signs detection and tracking using modified hough transform,” in *SIGMAP 2015 - 12th International Conference on Signal Processing and Multimedia Applications, Proceedings; Part of 12th International Joint Conference on e-Business and Telecommunications, ICETE 2015*, SciTePress, 2015, pp. 22–28. doi: 10.5220/0005543200220028.