King's Highway Sign Reading - COMPENG 4TN4 Report

Johnathan Spinelli
McMaster University, Hamilton, ON
spinellj@mcmaster.ca

Abstract

Global Positioning Systems use data signals to determine your a location on a map. In the absence of signals, landmarks can be used for determining location. Using images of the King's Highway road signs in Ontario, text will be read from the sign. These signs appear on the side of the road, and the images cannot have their text read immediately from the image. By applying a multi-stage preprocessing algorithm to these images, the sign can be isolated from its environment and the text can be extracted from the image. The gradient of the image is used to locate high contrast areas indicative of a sign. The background can be removed through HSV thresholding and by transforming the image to be squared with the frame. By applying binary operations on the result, the text can be read using Tesseract, an Optical Character Recognition tool. The algorithm only returns a text output if a number is read from the image, and returns nothing if the image cannot be read. Using a database of 54 images, 32 images had an output produced, with only 4 of those outputs being incorrect. When implemented with a recording of multiple frames, the algorithm can make multiple attempts to read the text from the sign.

1. Introduction

With the development of Global Positioning Systems (GPS), vehicles have the ability to know their location on the road and know the directions to reach their destination. The information stored in the GPS database was likely collected through satellite imaging, traffic cameras and mobile devices [1]. This information can be accurate to within 20 metres [1]. However, with poor GPS connection, information may not be properly communicated to the device, leaving the driver without directions and lost on the road. Being able to determine the driver's position on the road using distinguishable road signs can provide useful in scenarios where the GPS signal is weak, and can provide an offline solution for receiving directions.

Since 1925, Ontario's highways have had distinguished signs on the side on the road identifying the number of the highway [2]. Although these signs have changed since their implementation, they have maintained their numbering with a crown on the sign [2]. By identifying these signs on the side of the road, a camera system can identify which highway the driver is on, and can keep track of how many signs the driver has passed on their journey to estimate their position on the highway.

Optical Character Recognition (OCR) is a text recognition tool commonly used for reading text and numbers [3]. Tasks like reading house numbers and license plates make use of OCR libraries, and are not limited to single font styles or sizes [4]. While some OCR tools run fine without preprocessing, performance almost always improves with some implementation [4].

This project will attempt to preprocess an image containing a King's Highway sign to the point where it can have its text read by Tesseract. This will include many preprocessing steps, including transformations, thresholding, and binary operations to prepare the image for the Tesseract operation.

2. Related Work

There exists another project for Traffic Sign Recognition, where a cropped image is given to a CNN [5]. Keras is used to classify the signs based on their shape, colour and text [5]. This requires a large database of different signs, but the Ontario King's Highway sign is not included in this database.

Another related project is license plate detection. In this project, the developer designed an algorithm for Automatic Number Plate Recognition (ANPR) that detects the license plate and uses an OCR called Tesseract to read the text [6]. This project requires high quality images, and significant preprocessing before being able to read the text on the license plate [6].

3. Proposed Method

Collected images of the Ontario King's Highway signs will be used in this project. The objective is to properly classify an image of a highway sign. The algorithm will

include multiple steps of preprocessing, as the image will need to be cleaned of any noise and adjusted to be aligned with the camera, as is often required for OCR [3].

The first step of preprocessing is using Histogram of Gradients (HoG) in an unconventional way. The gradient of the image will be calculated to determine where the areas of high contrast and low contrast are in the image. Given that the white sign has black text, there should be high contrast on the sign, as well as contrast between the sign and the environment. Using these gradient values, the HoG result will be thresholded to remove areas of low contrast in an attempt to only display areas of high contrast. A bounding box will be placed around the remaining pixels to crop the original image to the areas of high contrast. The bounding box should have some padding around it in case the entire sign is not recognized as being a high contrast area.

The King's Highway signs appear on the side of the highway or rural roads. These environments often have trees and foliage behind the signs, as well as a blue sky in the background of the image. Any noise in the background may contribute to noise in the image that could affect the object detection stage. To remove the background, the algorithm will include an HSV thresholding stage, where blue and green colours above a certain saturation and value will be removed from the image.

The next step is object detection, where SIFT will be used between the image and a blank King's Highway sign with no text on it (see Figure 1). This reduces the number of bad matches between the image, as SIFT will not try to make matches between text or the numbers on the sign. The match ratio will be determined empirically to find a value that leads to the most correct readings of the signs. The resulting homography matrix is used to transform the image such that the sign is squared with the frame and takes up most of the image. This improves Tesseract's ability to read the numbers on the sign. To increase performance and to ensure the sign is as squared as possible, the is transformed up to three times depending on number of matches after each iteration.

With the major transformations complete, the final steps of the algorithm are focused on isolating the numbers on the sign. The image is binarized to increase the contrast between the black text and the white sign. The image then goes through morphology to remove any noise in the image that could come from wear or dents on the sign. To finally isolate the text and remove any background noise, the image is cropped to a rectangle where the expected text should be.

After properly aligning the image, Tesseract is used to extract the numbers on the sign as text. The text is then compared with the known characters on the sign to test the success of the algorithm. If Tesseract has difficulty turning the image into text, a different approach is taken in the previous step. The Canny operator is used on the image

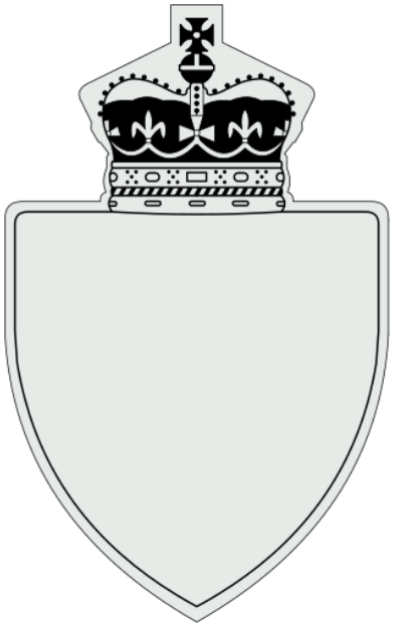


Figure 1. Blank sign used for SIFT object detection. It does not have any numbers or text that could lead to poor matches with the environment.

to see if that improves Tesseract's ability to read text from the image. If this is also unsuccessful, specific contours in the image are filled as a final attempt to prepare the text for recognition.

4. Experimental Results

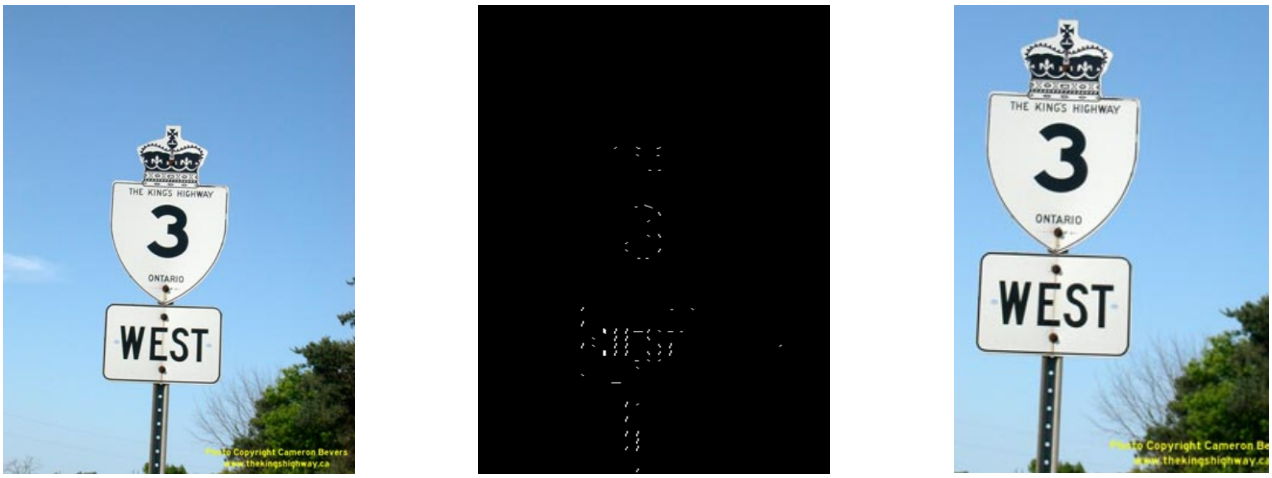
4.1. HoG

HoG is applied as a streamlined method for getting the gradient of the sample image. The gradient is used as a means to locate the high contrast areas which should correspond to the sign's location in the image. The gradient values are thresholded to remove areas of low contrast from the background. Figure 2 shows the gradient of a sample image, and the result after cropping the image to fit around the high contrast areas.

This method was successful in removing the background from most of the dataset. In instances where there is substantial background noise, this step may not yield beneficial results, but cannot harm the image as there will little or no affect of cropping.

4.2. HSV Threshold

Threshold values for saturation and value were determined empirically by sampling multiple images with green and blue scenery behind the signs. The boundaries for the thresholds must not remove the white sign associated with low saturation, and should retain the dark text on the sign



(a) Input image. (b) Thresholded gradient. (c) Resulting cropped image.

Figure 2. Progression of gradient analysis, resulting in a cropped image that removes background.

Figure 3. Images after HSV Thresholding, removing coloured regions from the image.

Figure 4. Image after HSV Thresholding is transformed to square the sign with the frame.

associated with low value. The hue values cover the entire range, as other background factors that could lead to noise could have any colour, not just blue or green. The applied threshold removes pixels with saturation between 84 to 255 and value between 30 and 255. This method proved to be mostly effective at assisting the SIFT procedure, as it removed more background information after the gradient cropping procedure. In the case where there are other signs of different colours in the image, those signs would also be thresholded too. Figure 3 shows two images after HSV thresholding, leaving the white sign and few other coloured pixels.

4.3. SIFT Recognition and Transform

SIFT recognition is used to match the sign in the thresholded image with the blank sign seen in Figure 1. Following the previous preprocessing stages, the SIFT algorithm produced the greatest results when using the ratio of 0.8. Using RANSAC, the homography matrix for the required transformation was computed, and the sign was transformed to be squared with the frame. Due to several factors such as noise remaining in the image, other signs in the image, and upscaling due to low resolution, the transformations would sometimes fail and produce unreadable results. This happened for 8 out of the 27 sample images used for refining the algorithm. For the images that transformed successfully, the sign became centreed in the frame. In some cases, the sign would still be slightly tilted, so to square the sign as much as possible for the text reading procedure, the sign was transformed two more times. Tesseract is very sensitive and can have issues reading stretched numbers, so this step is necessary. For samples that was successfully transformed the first time, the signs improved their human readability with every following transformation. Figure 4 shows an example of a transformed sign, squared with the frame for readability.

Figure 5. Binarized images ready for Tesseract reading. Left: original approach; Right: alternate approach with contours.

Figure 6. Successful image from the Complex dataset modified to be Lighter, Darker, and Rotated.

4.4. Text Isolation

With the image transformed to a readable position, the text can now be isolated in the image. The image is binarized with a threshold value of 120 to produce a black and white image. After performing morphology operations on the image, the numbers are isolated and clearly readable, assuming the transformations were successful. The image was then cropped to remove the outer edges of the sign, isolating just the text on the sign as much as possible. With this cropped result, the text is ready for reading. In some instances, the numbers in the frame were clearly distinguishable to the human eye, however, Tesseract failed to read some signs. To accommodate for this, an alternate method of reading was prepared. The image had the Canny operator applied to it, and the contours were filled to smoothly draw the numbers in the frame. This alternate method improved the results for images that were not read properly the first time. Figure 5 displays the results of these operations for an image that was read successfully with the first approach, and an image that failed with the first approach. From the original data set, six successfully transformed images failed to be read using the first approach and used the second approach instead. Of the six images that otherwise did not produce a result, three of them were recovered and produced the correct text output with the contour method.

4.5. Text Reading & Classification

Tesseract had varying success with reading the binary text in the frame. Images that look like they would be easy to read sometimes proved problematic for Tesseract. Noise in the image sometimes caused extra characters to be read, such as symbols or punctuation. A string of characters was created as a set of values that should be removed from the read text. In some cases, the number 1 was misread as being a close bracket, so this was accounted for in the text reading procedure.

Signs were split into two groups based on their complexity. Twelve images that were already squared with the frame with no background were put into the Simple group, while the remaining fifteen signs on the side of the road were put into the Complex group. Only six of the signs in the Simple group successfully had their numbers read from the sign. This is due to some signs having a weathered, yellow tint, and an older font style that is incompatible with Tesseract. From the Complex group, seven of the fifteen images were correctly read.

Excluding the signs that did not produce a text output, the Simple group had greater success, reading three out of the four signs correctly. In the case of the Complex group, seven out of ten signs were read correctly. This means that of the text outputs, the preprocessing algorithm had a 71% success rate across both categories. Tables 1 and 2 display the results from the text reading.

4.6. Extended Dataset

The internet has limited photos of the King's Highway sign available. In an attempt to increase the dataset, images that were successfully transformed and read were modified. These images were made 40 units brighter, 80 units darker, and rotated 25 degrees (see Figure 6). The goal is to see how well the algorithm functions for different conditions. The selected images came from both the Simple and Complex groups to have a diverse set of images.

Out of the 27 new images in the dataset, the algorithm was able to produce text for 17 images, with only 2 of the outputs being incorrect. This can be broken down further to compare each type of image. From the Lighter group, 7 out of the 9 images were read properly. From the Darker group, only 2 out of the 9 images were read properly. From the Rotated group, 6 of the 9 images were read properly. Images in the darker dataset struggled more than the others due to the significant difference in the image. By decreasing the brightness of the images by 80 units, some features were lost upon hitting the lower pixel value threshold. Additionally, the HSV thresholding stages are affected by the shift, causing the transformation stage to fail. Table 3 displays the results from the extended dataset reading.

5. Conclusion

Across all 54 images used in the testing of the preprocessing algorithm, 28 images had their text read correctly. Although this success rate is less than ideal, it is important to remember that the available images had low resolution, and often had old fonts that would be difficult for Tesseract

Test No.	1	2	3	4	5	6	7	8	9	10	11	12
Actual	69	7	83	9	2	70	17	17	11	69	7	2
Read	69		83	9			1	17		69	7	

Table 1. Complex Group Results

Test No.	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
Actual	401	105	12	401	400	401	64	401	400	96	7	400	35	3	89
Read	461	105		461	4001			401	400			400	35	3	89

Table 2. Simple Group Results

Modification	L	D	R	L	D	R	L	D	R	L	D	R	L	D	R
Actual	69	69	69	83	83	83	17	17	17	105	105	105	401	401	401
Read	69	69	69	83	83	83	17						1		401

Modification	L	D	R	L	D	R	L	D	R	L	D	R
Actual	400	400	400	401	401	401	400	400	400	3	3	3
Read	400		0	401		401	400		400	3		3

Table 3. Extended Dataset Results for Lighter, Darker, and Rotated Images

to read in an ideal scenario. Additionally, across the 54 images, only 4 signs had an output with an incorrect number, giving a 7% error rate. In the other test cases that did not produce an output, it is safer to not output any text than to output an incorrect value.

In a real-world application of this algorithm, a camera would be recording a live feed of the side of the road. When approaching a sign, the camera would capture multiple images of the sign, providing variety of samples for the algorithm to read from. A voting system could be implemented to compare the results from each frame when determining what number is on the sing. In the case where one frame does not produce a result, the other frames will make up for the lost contribution.

The project could be made more successful with implementing a different OCR method, or further improving the Tesseract procedure.

References

[1] C. Lawson, "Google maps and the role of Data Analytics," Selerity, 14-May-2021. [Online]. Available: https://seleritysas.com/blog/2021/05/14/google-maps-and-the-role-of-data-analytics/#:~:text=The%20search%20giant%20collects%20data,accurate%20information%20for%20their%20users. [Accessed: 04-Mar-2022]. 1

[2] C. Bevers, "History of Ontario's King's Highway Signs," Ontario Highway Sign History. [Online]. Available: https://www.thekingshighway.ca/signs.htm. [Accessed: 04-Mar-2022]. 1

[3] F. Zelic and A. Sable, "How to Ocr with Tesseract, OpenCV and Python," Nanonets, 10-Feb-2022. [Online]. Available: https://nanonets.com/blog/ocr-with-tesseract/. [Accessed: 04-Mar-2022]. 1, 2

[4] G. Shperber, "A gentle introduction to OCR," Towards Data Science, 22-Oct-2018. [Online]. Available: https://towardsdatascience.com/a-gentle-introduction-to-ocr-ee1469a201aa. [Accessed: 04-Mar-2022]. 1

[5] "Python project on traffic signs recognition with 95% accuracy using CNN & Keras," *DataFlair*, 06-Aug-2020. [Online]. Available: https://data-flair.training/blogs/python-project-traffic-signs-recognition/. [Accessed: 04-Apr-2022]. 1

[6] "OpenCV: Automatic License/Number Plate Recognition (ANPR) with Python," *PyImageSearch*, 01-Sep-2020. [Online]. Available: https://pyimagesearch.com/2020/09/21/opencv-automatic-license-number-plate-recognition-anpr-with-python/. [Accessed: 04-Apr-2022]. 1