CIS * 4720
Denis Nikitenko

Assignment 3
April 10, 2019

Abhishek Jhoree - 0986820
Miriam Snow - 0954174
Areeb Mustafa - 0932649

CIS*4720 Assignment 3
April 10, 2019

Abstract

For the purpose of this assignment, we implemented a road sign detection algorithm to identify, locate and interpret road signs. Although this genre of algorithm has many real-world applications, we found the most interesting one to be driverless cars. Keeping this in mind, we decided to extend the assignment past still images to video analysis. A driverless car would likely be moving while detecting road signs, so we decided to include videos in our testing regiment too. After identifying the road sign, interpretation of road signs was another key aspect of this algorithm. This included translation of non-english signs, and outputting instructions to the user based on the text recognized on the signs. In conclusion, we accurately identified signs 75% of the time and outputted the correct instruction 68% of the time.

Introduction

While deciding on how to implement our algorithm, we did a lot of research and read many reference papers to understand what was (and was not) successful for others in the past. In particular, we found the “Image Segmentation and Shape Analysis for Road-Sign Detection” by Khan et al. very useful, as it walked us through an in-depth solution for shape detection. The flow charts and equations in this paper were helpful for us to understand how shape detection is implemented. We also found “Road Sign Segmentation Based On Color Spaces: A Comparative Study” by Zakir et al. particularly interesting, as it explained color segmentation and gave us charts to indicate successful RGB and HSV values for accurate segmentation results.

As we researched, we found that many of the authors explained that their proposed concepts should be combined with other algorithms to improve success rates. Therefore, we took various ideas from many papers to implement our final algorithm. So, after much research, we decided the three main goals of our algorithm were color segmentation, shape detection, and text recognition. After doing an excessive amount of testing various images and videos, we found that all of these goals were met to a satisfactory degree.

Even though our algorithm was successful, we faced a few challenges, including finding RGB and HSV ranges for specific colors and working with non-normal images. The edge cases of our algorithm, as described below, included images of road signs that had low lighting, were far away, contained poor weather conditions, or were destroyed, causing the sign to lose its shape or color. We also faced some issues with the text recognition, which did not always output the correct information every time.

Algorithm Description

The task at hand was broken down into three main steps. The first step of the proposed algorithm is to segment the images based on color. The color segmentation algorithm is carried out by taking the RGB image to be tested and converting it to the HSV color space. HSV color space was chosen for this algorithm as it was found to have the highest success rate in detecting the color cues of traffic signs than any other color spaces (about 95% accuracy as indicated in the chart below).

Colour Space	CIE Lab	RGB	YCbCr	CYMK	YIQ	HSV
Accuracy %	87.7	62.2	77.2	76.5	74.6	94.7

Table-5.5: Overall average accuracy figures obtained for individual colour spaces.

This was demonstrated by Zakir et al. in their paper on segmenting road signs based on color spaces. The HSV color space works so well because it is very close to a human's perception of colors. Hue is invariant to the variations in light conditions as it is multiplicative/scale invariant, additive/shift invariant, and it is invariant under saturation changes. In other words, this means that it is still possible to recover the tint of the object when it is lit with intensity varying illumination space. The hue is unaffected by shadows and highlights on the object when the illumination is white. For our algorithm, we only implemented red, yellow, blue and green for color detection. We found these to be the most common sign colors. At a later time, this could be extended to identify other colors as well.

Below is an example of a stop sign and the output after it has been segmented by our algorithm in the HSV color space.

The next step is the shape classification process. The segmented images (still in the HSV color space) have a contour detection algorithm applied to them. This helps classify the shape of the road sign as well as outline the actual contour of the road sign which will later be superimposed back on the original image to show the sign region. Road signs fall into 4 main shape categories; namely circular, triangular, rectangular and octagonal. The number of sides detected by the contour detection algorithm will classify the shapes and confirm that the segmented blob is indeed a road sign. For instance, a circle will have 0 sides, a triangle has 3 sides and so on.

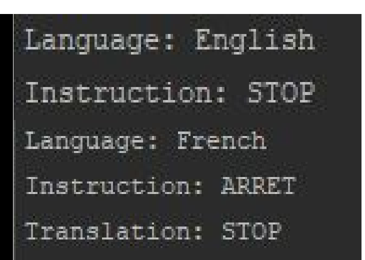
After the contour is drawn around the segmented image and the shape is confirmed to fall within the permitted road sign shapes, the processed image is used to draw the contours in the original image to outline the road sign. This is achieved by using the same coordinated that were obtained in the segmented image and applying them onto the original image.

Below is the same stop sign which has be processed through the shape classification step. It is identified as an octagon and the shape outline has been superimposed back onto the original image.

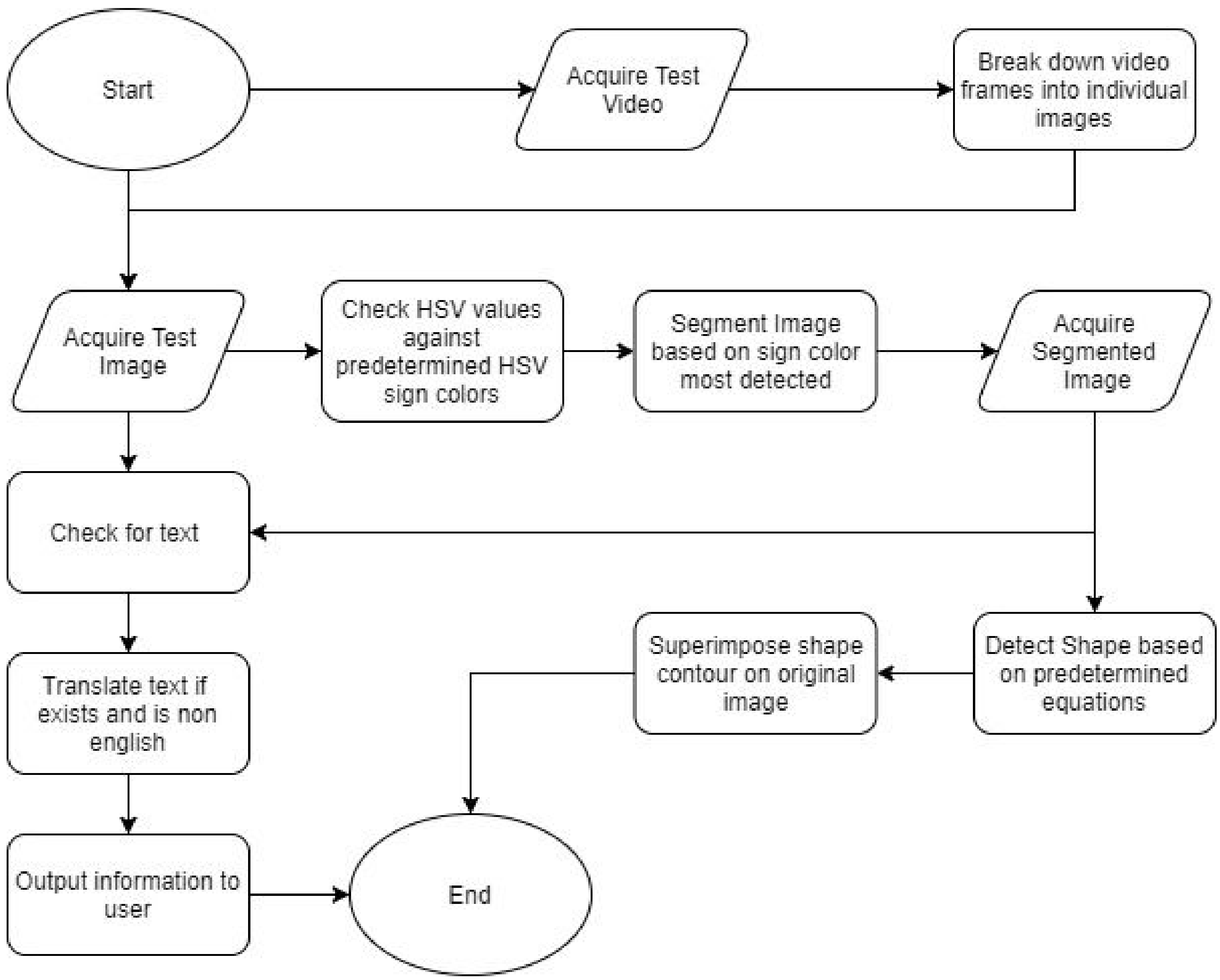
The final step is the text recognition process. Once the image is confirmed to contain a road sign, it undergoes text detection. The same segmented image is used in this step as the better contrast the HSV image provided, often makes the text detection process more accurate. However, in certain cases, the text was recognized with more success on the original image. Therefore, we apply text recognition to both the original and HSV segmented image. This step will help convey textual information to the driver such as stop instructions and speed limits. The algorithm is also capable of translating stop instructions specifically from French or Spanish into English thereby assisting an English speaking driver who might be visiting those countries.

This step of the algorithm uses an EAST text detector (Efficient and Accurate Scene Text Detector) which is a pre trained machine learning algorithm to correctly identify letters in images and interpret them into words.

Below are some screenshots of the text recognition algorithm in process, and the output given to the user based on what the algorithm detected.



Algorithm Overview



Problems Encountered

Though the algorithm works well for many of the test images, there were a few shortfalls that we found when implementing the algorithm. The first problem that was encountered was defining the HSV color space threshold values for specific colors. Our algorithm identifies signs that are blue, red, yellow or green. We started by using HSV color ranges for these colors that were stated in the reference papers we read. However, these values often had to be tweaked and we found while testing that some signs were segmented more accurately than others. Ranges were found for blue, red and green very easily, however, yellow was more challenging. We also tried to implement color segmentation for white, but we weren't able to complete this given the time allotted.

In our algorithm, the road sign is segmented by a specific color depending on the percentage of that color in the image. For example, a high percentage of green (indicating a green road sign is in the image) would cause the green segmentation function to be called. However, another problem that was encountered was that stop signs were segmented with the green segmentation mask instead of red. We think this was because the image had a green background and so there was a higher percentage of green in the image, than red. This caused the algorithm to not segment the stop signs as well as it would have with the red mask, however, they were still identified correctly.

A third issue we faced was with text detection of road signs that contained letters which are not used in the English language. The algorithm worked well in detecting signs that are understood by the English language, words like 'stop', 'speed', 'exit', or numeric values. However, signs that are in a foreign language that do not use the alphabet were not detected with the text detection algorithm. We also had encountered a problem where the text detection algorithm did not always detect signs that contained letters with accents (e.g arrêt).

The last issue that arose was processing non-normal images which contained road signs which were destroyed or hard to see due to bad weather conditions. This was a key issue because road signs should still be detected and followed regardless of the fact it is destroyed, bent out of shape or discolored. They also need to be followed under bad weather conditions, especially in canada where there is so much snow and ice, however our algorithm did not detect a stop sign covered in ice.

Experimentation Description

Our experiment consists of processing various test images and some videos through our algorithm. This includes images that may or may not contain road signs, contain different weather and lighting conditions, or contain road signs that are discolored or broken. We chose approximately thirty test images to test. We thought that this would be a good collection to determine the effectiveness of the algorithm. In addition to still images, we also processed some videos with our algorithm. The idea here is to break down the video frame by frame and process a video frame through the algorithm by the same method as if is was a still images. All of our test images are located in a folder called "TestImages".

We split our test results into three different categories. The first was the accuracy, correctness and false detection rate of the identification of the road signs. This focused mainly on the first two steps of our algorithm; color segmentation and shape detection. The results were determined by visual inspection of whether the road sign was correctly identified in the image.

The second category was the accuracy, correctness and false detection rate of the text output. The results of this were determined based on whether or not the output of the program to the user corresponded with what the road sign displayed.

The third category was the speed of our algorithm. Here, we used a timer to display how many seconds our algorithm took to process the image and give output to the user. Each image we tested was not larger than 500px wide, so the average time computed should be an accurate representation of the efficiency of our algorithm.

After processing our test images, the results from the algorithm can be used to analyze how well our algorithm correctly segments and detects road signs, and gives meaningful information to the user.

Experimentation Results

Identification of Road Signs

True Positives - road sign exists - road sign detected	False Positives - road sign does not exist - road sign detected	True Negatives - road sign does not exist - road sign not detected	False Negatives - road sign exists - road sign not detected
16	2	5	5

Accuracy
$$\begin{aligned} &= (TP + TN) / (TP + TN + FN + FP) \\ &= (16 + 5) / (16 + 5 + 5 + 2) \\ &= 21 / 28 \\ &= 75\ \% \end{aligned}$$

Correctness = TP / (TP + FP)
= 16 / (16 + 2)
= 16 / 18
= 89 %

False Detection Rate = FP / (TP + FP)
= 2 / (16 + 2)
= 2 / 18
= 11 %

Text Recognition

True Positives - text exists - text is outputted	False Positives - text does not exist - text is outputted	True Negatives - text does not exist - text is not outputted	False Negatives - text exists - text is not outputted
10	1	9	8

Accuracy = (TP + TN) / (TP + TN + FN + FP)
= (10 + 9) / (10 + 9 + 8 + 1)
= 19 / 28
= 68 %

Correctness = TP / (TP + FP)
= 10 / (10 + 1)
= 10 / 11
= 91 %

False Detection Rate = FP / (TP + FP)
= 1 / (10 + 1)
= 1 / 11
= 9 %

Speed
Average speed of algorithm processing = 512 seconds / 28 images
= 18 seconds per image

Analysis of Results

In conclusion, we were very pleased with the results of the experiments we conducted. The first category analyzed determined that our algorithm was 75% accurate at identifying if there was a road sign in the image or not. When there was a road sign, our algorithm detected it 89% of the time, and it only falsely detected road signs when they weren't there 11% of the time. Images where our algorithm failed to correctly identify the road sign were special cases where road signs did not quite look like road signs. Examples are shown below.

Here we can see that the stop signs are not their typical vibrant red color, and they aren't their regular octogonal shape either.

In some cases, our algorithm indicated that there was a road sign in the image, when there was not. This is called a false detection rate. Although this did not happen very often, an example where this occurred can be seen below.

Here, the green colored stripe on the road is being identified as a road sign because it is the same color as typical green road signs. Although it is not being identified as clearly as a regular road sign would be, we can still see that this is an area where the algorithm would need to be improved.

The second category of metrics indicated that 68% of the time, the road sign was correctly identified as instructional (sign had no text) versus informational (sign had text). The remaining 32% where the text recognition algorithm was inaccurate, was primarily caused because the text on the road sign was not outputted to the user. There were not very many times when text did not exist, but it was outputted to the user. This is indicated by the low false detection rate of 9%. Although we would have liked to improve our accuracy rating, we determined a correctness rate of 91%. This indicates that almost every time there was text outputted to the user, it was correct.

Lastly, we determined that our algorithm processed our test images at about 18 seconds per image. The speed of our algorithm was definitely something we were concerned about because we have so many steps to our algorithm. The maximum number of seconds to process an image was about 45, and the minimum was about 13. We also tried to keep our test images small so that our algorithm would be able to process them faster. This metric does not take into account the amount of time to process videos. However, we determined that a 2 second video only took about 30 seconds to process.

Factor of Originality

The referenced algorithms provided in the assignment description were read, but often the authors indicated that their algorithms were best combined with other concepts. Most focused on color segmentation only. Therefore, we decided to do more research and combine various techniques from the reference papers we found to develop our final algorithm. Thus, resulting in our three step analysis. We also searched and acquired all of our own testing images to verify a thorough experiment. This included various edge cases as described above. For the extended video portion of our algorithm, we acquired our own video to mimic as closely as possible how our algorithm would perform in a real-world, driverless car situation.

References

Fleyeh, H., “*Road and traffic sign color detection and segmentation - a fuzzy approach*”, IAPR Conference on Machine Vision Applications, pp.124-127 (2005)

Fleyeh, H., “*Traffic sign color detection and segmentation in poor light conditions*”, IAPR Conference on Machine Vision Applications, pp.124-127 (2005)

Khan, J., Bhuiyan, S., Adhami, R., “*Image Segmentation and Shape Analysis for Road-Sign Detection*”, IEEE Transactions on Intelligent Transportation Systems, pp. 83-96 (2011)

Oh Song, H., Bolouki, S., Yen, A., “*Detection of Stop Signs*”, Stanford University CS 223B Project Report, pp. (2008)

Piccioli, G., De Micheli, E., Parodi, P., Campani, M., “*Robust road sign detection and recognition from image sequences*”, in: Proc. Symp. Intelligent Vehicles, (1994)

Rosebrock, Adrian. *“OpenCV OCR and Text Recognition with Tesseract.”* PyImageSearch, 4 Feb. 2019, www.pyimagesearch.com/2018/09/17/opencv-ocr-and-text-recognition-with-tesseract/.

Rosebrock, Adrian. “*OpenCV Shape Detection.*” PyImageSearch, 5 Feb. 2019, www.pyimagesearch.com/2016/02/08/opencv-shape-detection/.

Zakir, U., et al. “*Road Sign Segmentation Based on Colour Spaces: A Comparative Study.*” Computer Graphics and Imaging, 17 Feb. 2010.