Street Sign Image Segmentation with Computer Vision Methods

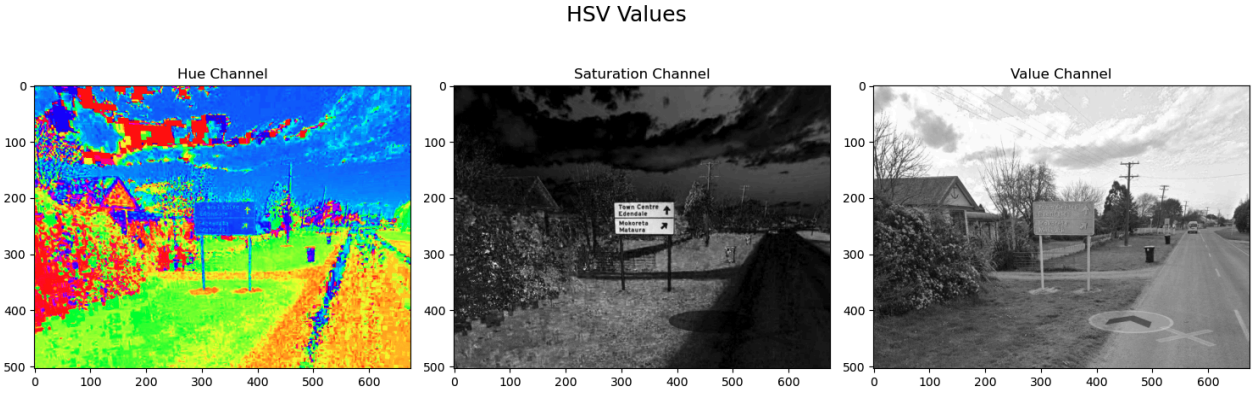
1 Introduction

Motivation: My original idea for this project was to create a machine learning model that takes a street view image containing a street sign as input and outputs the approximate location of where the image was taken. I decided I would attempt to use the computer vision techniques I have learned in the course for a preprocessing step to segment the street sign of interest in the image to make it easier for a machine learning model to learn from street signs without distractions. This is because in a typical street view image, there are many features (road, vehicles, houses, vegetation) that will make it more difficult to train a model that is meant to focus on street signs specifically. This report focuses on the image segmentation as a preprocessing step for my original idea that I plan to try and implement in the future.

Problem definition: Given a street view image containing a street sign like the one shown below, we want to perform image segmentation to isolate the street sign. This will remove all other parts of the image and leave only the sign of interest.

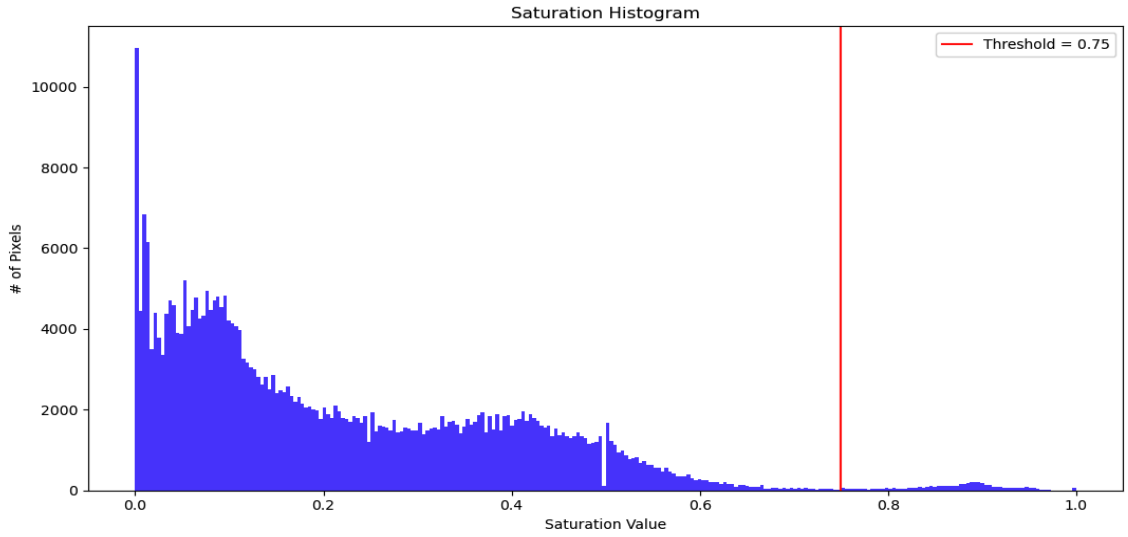
2 Methods

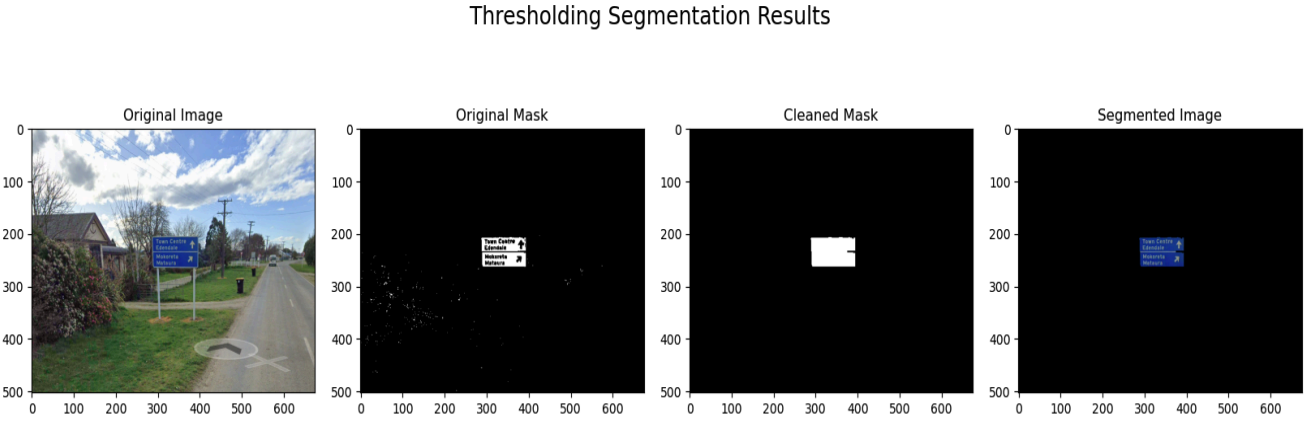
Preprocessing: When thinking about the best approach to take for this project, I decided that taking advantage of the bright colors of the signs would be key as street signs are typically brightly colored for visibility. I converted my input images from RGB to the HSV color scale since HSV is typically recommended opposed to RGB when trying to segment based on color. After conversion, I visualized the images on this new color scale to try and understand how I can use this to accomplish my goal.



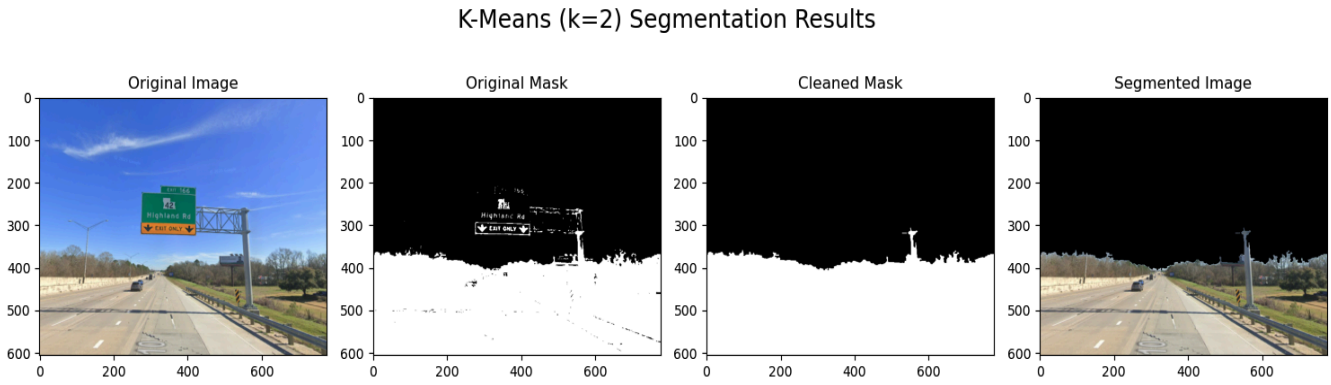
Thresholding: When my HSV images were visualized, I noticed that the signs always had very high saturation values compared to the rest of the image as can be seen in the middle image above. With this in mind, I decided to use thresholding with the saturation values to try and segment my images. I created a binary mask to represent the signs in the images by choosing all pixel values with a saturation value above .75. I chose the threshold values by plotting the saturation histograms and testing different threshold values until I reached one that worked the most consistently with different images. Shown below is an example of a result I got using this

technique.

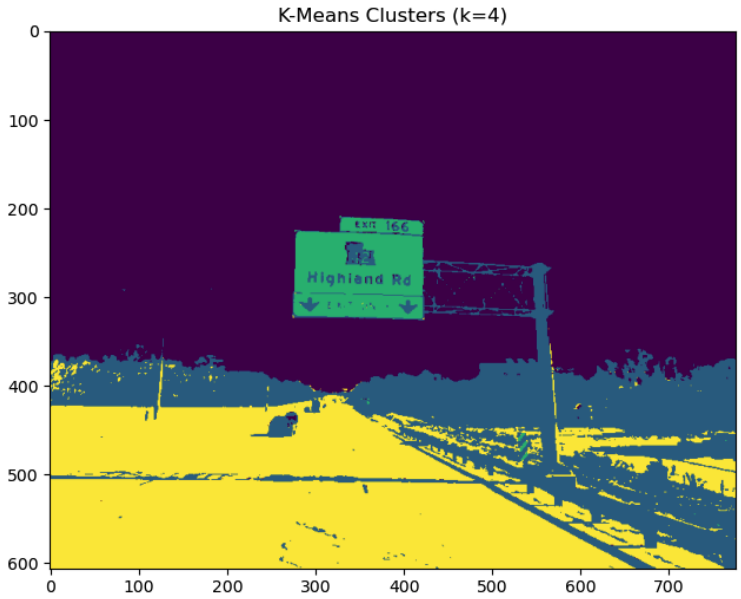


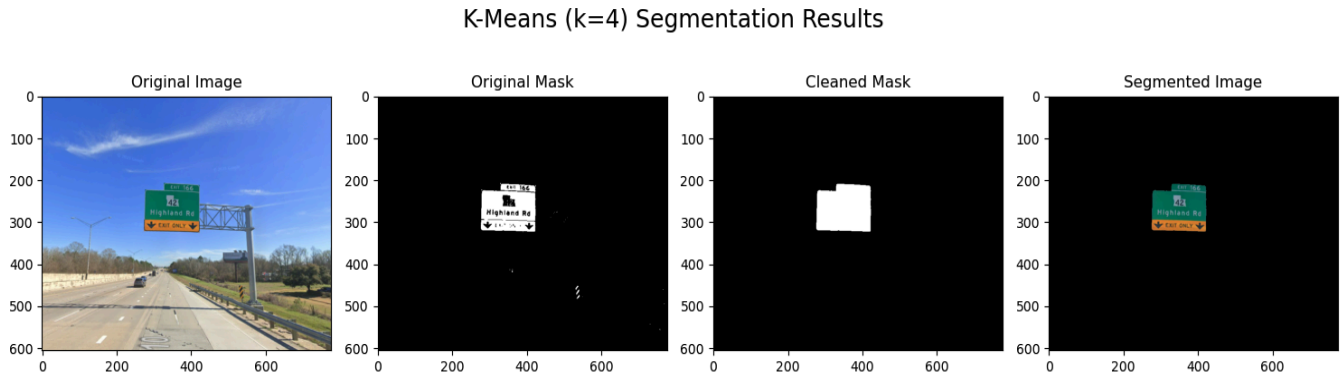


K-Means Clustering: I wanted to try using another segmentation technique since thresholding did not always work as I expected, especially when there are multiple signs in the image or there is poor lighting. So I decided to try using k-means clustering. I first started by simply feeding the image into a scikit-learn KMeans model with 2 clusters to see what kind of result that would get. Unsurprisingly, this would simply segment the foreground from the background in most cases and not focus on the signs. This result is shown below.



My next approach was to divide the images into more than 2 clusters and choose from these clusters with some criteria to hopefully choose the cluster representing the sign. When testing different numbers of clusters, I decided that 4 clusters worked the best for my case. This is because the signs would consistently be given their own cluster without there being too many small, insignificant clusters. This was perfect because to decide which cluster was representing the sign, I simply chose the smallest cluster of the 4 and used that cluster as the binary mask to represent the sign. This method gave me pretty good results with simple images that didn't have too many distracting features like other signs and bright objects. A result from this technique is shown below.





3 Conclusion

Results: The thresholding technique as well as k-means clustering both produced fair results given the simplicity and minimal processing that the techniques required. K-means produced slightly better results more consistently than thresholding, but generally they both produced similar results in most cases. The techniques worked well with the images I chose because of the small number of distracting elements and the high contrast between the signs and the background in most of the images. But, when images contained lots of signs or other objects, like there are in many urban environments, these techniques failed. If I was to implement this preprocessing step in a machine learning model, I would probably try to use a more sophisticated machine learning technique to segment the street signs to get more consistent results with more difficult images.