Stop Sign Detection

1st Simon Fong
Electrical and Computer Engineering
University of California San Diego
La Jolla, United States
simonfong6@gmail.com

***Abstract*—In this paper, we discuss performing stop sign detection using a multivariate Gaussian for segmentation and various region properties for shape similarity.**

***Index Terms*—segmentation, stop sign, Gaussian**

I. Introduction

With the rise of autonomous vehicles from companies such as Waymo, Tesla, and Cruise Automation, the problem of identifying traffic signs accurately is important so that these autonomous systems can react appropriately to the signs. For our problem, we are identifying just stop signs. I attempt this using a multivariate Gaussian for image segmentation, and a combination of region properties to identify the bounding boxes.

II. Problem Formulation:

A. Image Segmentation

We will be training two multivariate Gaussian classifiers to identify positive (stop sign pixels) and negative (non-stop sign pixels).

For a given class C, and a sample x.

$$P(C|x) \propto P(x|C)P(C) \tag{1}$$

We can model $P(x|C)$ as Gaussian distribution with μ and Σ representing mean and co-variance of the training data, respectively.

$$P(x|C) = N(x; \mu, \Sigma) \tag{2}$$

So our likelihood becomes.

$$P(C|x) \propto \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} P(C) \tag{3}$$

B. Shape Similarity

Now that we have the regions segmented out of the image, we use properties of each region to determine if it resembles the shape of a stop sign. The attributes that we will consider are the following: area, number of corners of an approximate polygon, and the eccentricity of the region.

C. Area

Note: Add image of segmented image with parts.

During segmentation non-stop sign pixels are sometimes segmented along with the stop sign pixels. Often times, those non-stop sign regions are much smaller than the stop sign regions, so we can filter those out.

D. Polygon Approximation

Note: Add an image of a Polygon approximated.

Here we can use the convex hull of the region and use it approximate a polygon. Since most of the stop signs are octagonal in shape, we can count the number of corners of each region's polygon. If the number of corners of a polygon is close to 8, we can keep it, otherwise we discard it.

E. Eccentricity

Eccentricity is how close a given shape resembles a circle, with 0 representing a circle, and 1 not representing a circle. We can measure the eccentricity of several stop signs and non-stop sign regions. Then use those values to see which are close to being a stop sign.

III. Technical Approach

A. Image Labeling

First, we need a labeled training data set, so we use roipoly to create binary masks of each image with a stop sign.

B. Pixel Extraction

Now that we know which pixels are stop sign pixels. We can extract the stop sign pixels and the non-stop sign pixels from each image in the training sets.

C. Classifier Training

With each set of pixels, we can calculate the mean μ and co-variance Σ for each and use them in our Gaussian classifiers.

D. Segmentation

Now given an image, we use each classifier to generate a probability of each class, positive or negative, and label it using the class with the highest probability. This creates a black and white segmented image where the white pixels are supposedly stop sign pixels.

E. Regions

With this mask, we can separate each connected region and identify properties of each region.

F. Filtering by Area

One property that will filter by is area. We can filter out the smaller regions using a $MIN\_PIXEL\_COUNT\_THRESHOLD$. For our purposes, a the following threshold worked decently.

$$MIN\_PIXEL\_COUNT\_THRESHOLD = 1000 < area$$

G. Filtering by Polygon

Using the convex hull of the mask, we can approximate a polygon and count the number of corners. We used the following thresholds.

$$5 \leq NUM\_CORNERS \leq 12$$

H. Filtering by Eccentricity

We will keep regions with eccentricities within the following range:

$$0.8 \leq ECCENTRICITY \leq 0.9$$

IV. Results

A. Training Results

Here is the classifier run on the training data, specifically Picture 1.

Fig. 1. Picture 1

You can see in Fig. 2 that the image is perfectly segmented out the red stop sign pixels.

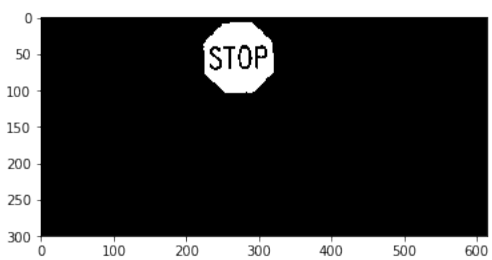

Fig. 2. Picture 1 Binary Mask After Segmentation

In Fig. 3, you the stop sign's bounding box is correctly identified. All these results were expected since this is the data that the classifier was trained on.

B. Test Results

Now we will see how the classifier worked on our validation set of data. In this example, we have chosen Picture 81.

You can see that segmentation was not as clean as in Fig. 2. There is much more noise and we have even segmented the green sign above the stop sign.

Now because of all that noise and non-stop sign pixels being segmented, that is why we used a the filtering techniques that we did. Filtering by area removes all the smaller regions which

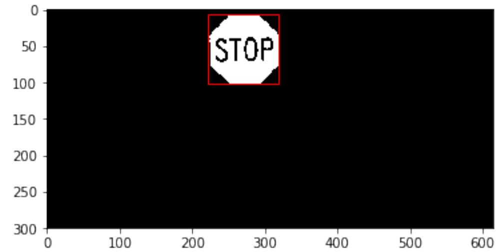

Fig. 3. Picture 1 Bounding Box
(min_x, max_y, max_x, min_y): (224, 291, 321, 195)
(min_row, min_col, max_row, max_col): (8, 224, 104, 321)

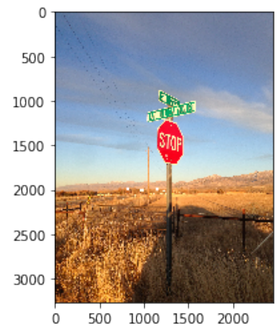

Fig. 4. Picture 81

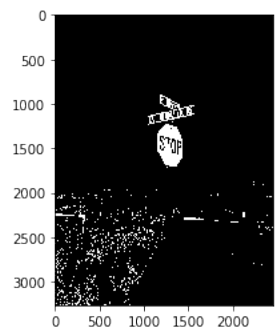

Fig. 5. Picture 81 Binary Mask After Segmentation

look like specs. Filtering by number of corners of polygon can help us remove regions which do not resemble a shape close to an octagon. Lastly, filtering by eccentricity gives us an extra tool to remove regions which aren't as circular as a stop sign should be.

References

[1] Hiroshi Shimodaira, Classifications with Gaussians, http://www.inf.ed.ac.uk/teaching/courses/inf2b/learnnotes/inf2b-learn-note09-2up.pdf.

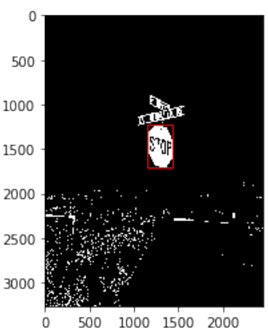

Fig. 6. Picture 81 Bounding Box
(min_x, max_y, max_x, min_y): (1153, 2043, 1441, 1552)
(min_row, min_col, max_row, max_col): (1220, 1153, 1711, 1441)