Blue Recycling Bin Detection

Steve Kim
Department of Electrical Computer Engineering
University of California, San Diego
stk002@eng.ucsd.edu

***Abstract*—This paper reviews the process of computer object recognition via Gaussian discriminant analysis and shape detection.**

***Index Terms*—Gaussian Discriminant Analysis, object detection, GDA, recycling bins, color segmentation**

I. Introduction

Computer vision object detection has matured quite a bit over the past decade. Its usage in autonomous vehicles has been a point of focus especially as the race to attain full autonomous driving nears its finish. In this paper, we review a model that is able to detect blue recycling bins in various real world environments in a two-step process. As a high level overview, the process involves first finding pixels of the image that have the same color as a recycling bin, and then taking those blue colored pixels and finding upright rectangular groupings. We rely on the assumption that blue upright shapes in images are recycling bins. This proves to be an insufficient assumption which we'll see later on. We detail various challenges that occur with visual object recognition, as well as methods to overcome them. Such a model could be further developed to one day automate self-driving dump trucks that can haul away refuse without human aid.

II. Problem Formulation

A. Image Color Segmentation

In this paper, the problem of detecting blue recycling bins is solved in two steps. The first step is color segmentation. We generate a Gaussian distribution of the RGB color values of blue recycling bins by sampling a number of diverse photos of bins in the real world. We then also create RGB distributions for environmental colors. In our approach, there are four total classes of distributions - Recycling bin blue, general background colors, red, and green.

$$y \in \{1,2,3,4\} \tag{1}$$

$$Class_y \sim N(\mu_y, \Sigma_y) \tag{2}$$

With these approximate distributions, for any new and unseen data, we can compute the likelihood that the data comes from one of these classes. By choosing the argument that maximizes these likelihoods, we can classify any given pixel as being a part of the blue recycling bin or otherwise.

$$argmax_y^* = N(\mu_y^*, \Sigma_y^*) \tag{3}$$

$$f(X|\mu_y,\Sigma_y) = \frac{1}{\sqrt{2\pi\Sigma_y}}\exp(\frac{-1}{2}(X-\mu_y)\Sigma^{-1}(X-\mu_y)^T) \tag{4}$$

Equation 4. Likelihood calculation for Gaussian class y

There are several challenges with creating a color mask for real world images with the recycling bin color 'blue'. One of them is that the color of the sky is often the same hue of blue as recycling bins. This results in color masks that contain extraneous information, which creates challenges when detecting recycling bins. The second step helps address this this issue.

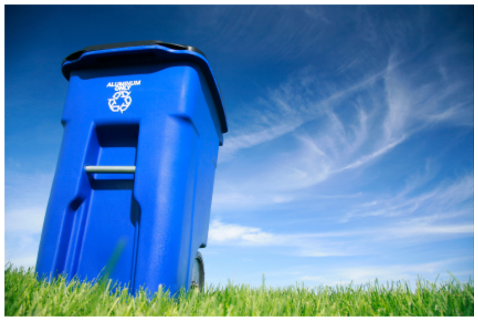

Fig. 1. Example of recycling bin. Note that the sky is a similar hue of blue, it's difficult for the Gaussian distribution to make a distinction. Image from Google https://images.app.goo.gl/DF1NWEgWVnDihVxV6

B. Shape Detection

The second step of this approach is to conduct shape detection on the image mask produced by the first step. This step entails creating a region box around elements in the color mask and deciding whether the proposed box is a recycling bin. However, given the issue mentioned above concerning noise in the color mask, we need to implement a filter for the bounding boxes. The criteria used to make this determination is fairly simple - it is a ratio of the width of the box in relation to its height.

$$\frac{height}{width} > ratio, ratio = hand - tuned \tag{5}$$

Equation 5. Recycling bins are upright rectangular boxes, so we would expect the ratio to be larger than one, aka the height should be larger than the width.

With the combination of using color segmentation to find all objects in a scene that is "recycling bin blue", and only choosing bounding the parts which have the shape of an

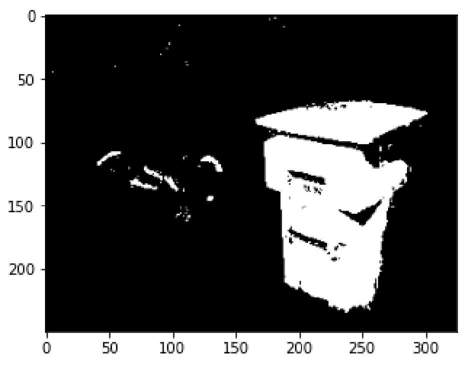

Fig. 2. Example of image mask on recycling bin from step one. We can see that there is visual noise to the left of the bin, which indicate that the original image has another object with blue hues in it. Original image from validation set 0063.jpg

upright rectangle, we can reliably, though perhaps naively, detect recycling bins in an unseen image.

III. Technical Approach

Here We will re-visit the two processes described above in step by step detail.

A. Data Generation and Gaussian Discriminant Analysis

Once again, the first step was to discriminate the blue color of recycling bins by conducting Gaussian Discriminant Analysis on each of the color distributions.

I. Generate data: We created four data-sets that would be representative of the hue of blue recycling bins in all real world conditions. This started by computing a conservative, but hard, 'blue' filter on all images. By converting the original image to HSV format, and taking only values from HSV [110, 80, 2] to HSV [126, 255, 255], we produce a black and blue image from the original. We then manually cropped these images to only contain the blue parts. The reverse was done to create the "non recycling bin" data class, which was the backgrounds to all of the original data minus blue hues. We also included data for green and red colors, to increase the distinguishing power of the classifier.

II. Calculate Likelihood: We sampled pixels from each class a different number of times; 1200 random pixel samples from each recycling bin image, 600 from background images, and 50 each from red and green images. These numbers were chosen so that despite the varying amount of training data, my distributions would be distinctive enough for each class. Given our data, we can see that the likelihood function becomes the following via Naïve Bayes:

$$data \in \{x_1, x_2, ...x_n\} \tag{6}$$

$$k - classes \in \{bin, background, red, green\} \tag{7}$$

$$P(y, X|\omega, \theta) = \prod_i^n \prod_l^k p(x_{il}|y_l, \omega) \prod_l^k p(y_l|\omega) \tag{8}$$

For ease of use and calculation, We used the log likelihood of the Gaussian, which is below:

$$\begin{split}\sum_i^n \log(\frac{1}{\sqrt{2\pi\Sigma_k}} \exp(-\frac{1}{2}(x - mu_k) \left(\Sigma_k^- 1(x - mu_k)^T\right) \\ + \left(\log \frac{1}{k} \sum \mathbb{1}\{y_i = k\}\right)\end{split} \tag{9}$$

Where the parameters are calculated via Maximum Likelihood Estimation

$$\mu_l^{MLE} = \frac{1}{N}\Sigma x_{il} \tag{10}$$

$$\Sigma_l^{MLE} = \frac{1}{N}(x_{il} - \mu_l^{MLE})(x_{il} - \mu_l^{MLE})^T \tag{11}$$

By selecting the argument that maximizes equation 9, We now have a classification for an unseen data point X.

B. Bounding Boxes

For the second step of creating bounded boxes around elements in the color map. We used opencv's findContours() function in order to find the regions in the color masks from the first step, but not without pre-processing the images beforehand.

I. Pre-processing images: The masks from the first step had quite a bit of noise, as well as stray pixels. As a result, we found the best results came from hand tuning the following morphological changes to each image mask.

i. Strong erosion: We first started by eroding our image with a relatively large kernel in order to drastically reduce the noise in each color mask.

ii. Gaussian Blur: Then by applying a Gaussian blur, we smoothed the edges of each element in the mask so that edges were more consistent across the image.

iii. Bounding Boxes: We applied findContours from opencv to find contour lines in the segmented image, and took the largest 5 from the result. Next, these contours were passed into approxPolyDP, which based on the contour shapes, produced a rectangle. Then, using boundingRect, we have the upper left and bottom right coordinates of the rectangle. Finally, these coordinates were passed into a filter that only chooses rectangular boxes that have a greater height than width (but also less than six times the width). After another filter of only choosing boxes with greater than 400 pixels squared and larger than 2 percent of the image to take care of small artifacts, we output our final bounding boxes. And finally, in figures 10 and 11, we see that despite the fact that there is a large blue pool, there are no bins detected, showing that the classifier is able to make that distinction.

IV. Results

From the first step, we found the following parameters for the GDA:

Below, we see the mean for each class k. Each column represents RGB values, and each row represents one of the four classes { Recycling bin, Background color, Red, Green}

$$\begin{pmatrix} 150.26 & 58.89 & 31.10 \\ 90.74 & 97.582 & 100.03 \\ 88.97 & 88.76 & 191.88 \\ 84.02 & 187.55 & 89.40 \end{pmatrix}$$

And the covariance for each class

$$\begin{pmatrix} 2420.31 & 1154.62 & 497.75 \\ 1154.62 & 1365.16 & 854.03 \\ 497.75 & 854.03 & 761.865 \\ \\ 5712.84 & 5389.18 & 5074.23 \\ 5389.18 & 5723.48 & 5569.17 \\ 5074.23 & 5569.19 & 6132.7 \\ \\ 4033.07 & 557.68 & 1210.77 \\ 557.61 & 4029.51 & 1198.27 \\ 1210.72 & 1198.20 & 2409.8 \\ \\ 3642.88 & 1106.32 & 567.88 \\ 1106.33 & 2262.11 & 1146.86 \\ 567.82 & 1146.86 & 3624.20 \end{pmatrix}$$

Using the above parameters and calculating the likelihood of each class, we generate the segment masks in step one. From there, we draw the bounding boxes, which see from the following examples as we display the output of each step that this approach takes.

In figures 3-5, we see that despite the fact that there is a blue door frame, our bin detector is able to successfully avoid creating a bounding box around it. In figures 6-8, we see that there are two recycling bins in a shadow, but thanks to our diverse training data and image transformation, we're able to detect and make distinct both bins. However, in images 9-11, we see that we are unable to avoid creating a false bounding box around the pool, showing that our classifier needs further development.

Below are the Accuracy, Precision, and Recall metrics of this classifier.

Set	Accuracy	Precision	Recall
Validation	80%	40%	67%
Test	67.5%	na	na

There is a lot of room for improvement, as we know that the GDA can perform upwards to 95% and higher. The author of this paper is continuing to try to increase each of these performance metrics.

V. Collaboration

Collaborated with Hala Abualsaud and Luis Martin Herrera Lezama

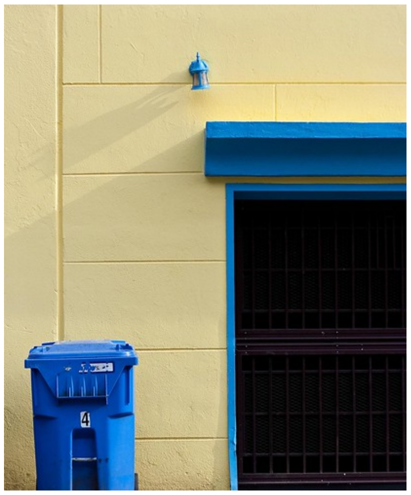

Fig. 3. Original image from validation set 0062.jpg

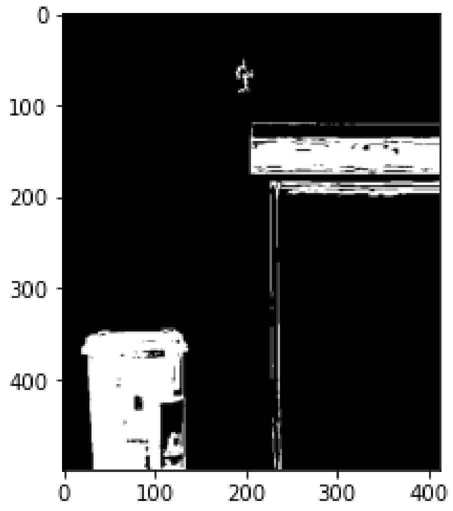

Fig. 4. Color segmentation of original image

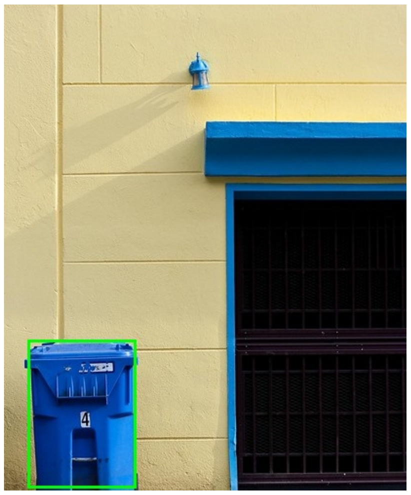

Fig. 5. Original image with bounding box

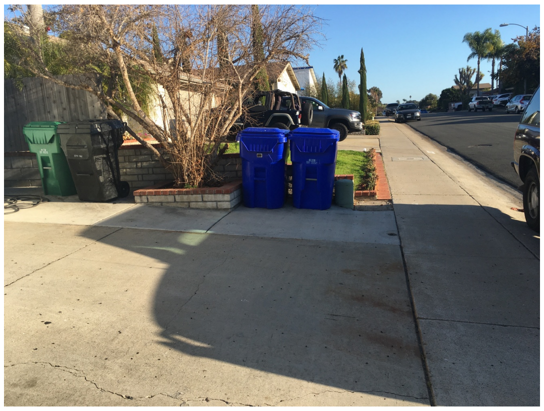

Fig. 6. Original image from validation set 0067.jpg

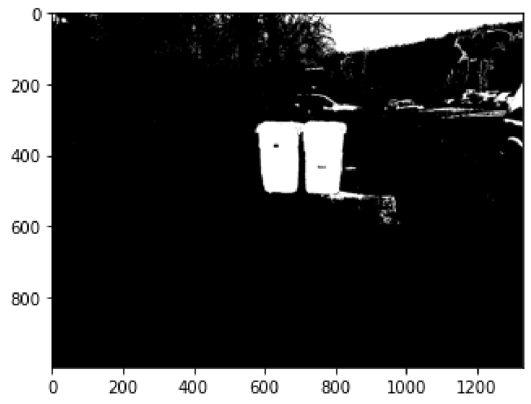

Fig. 7. Color segmentation of original image

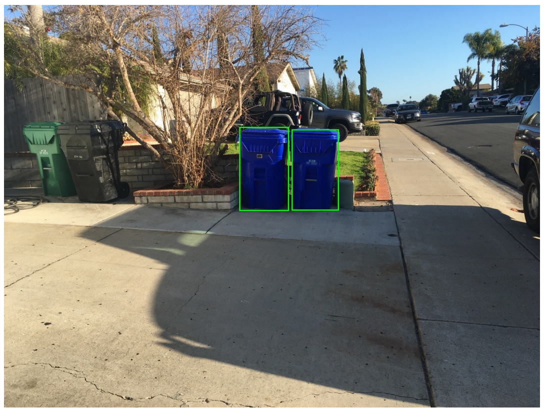

Fig. 8. Original image with bounding box

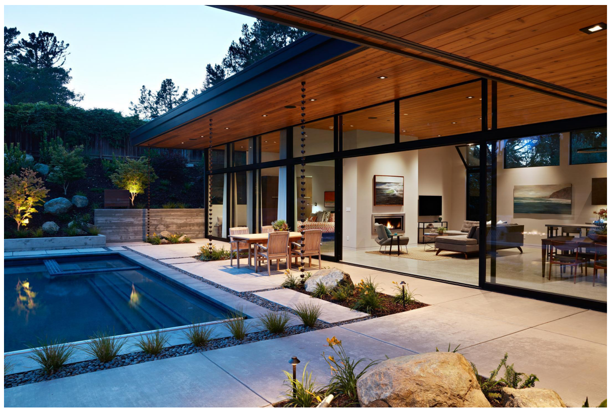

Fig. 9. Original image from validation set 0070.jpg

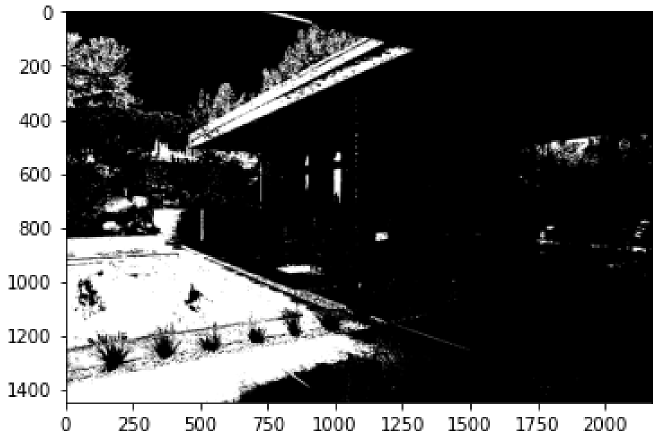

Fig. 10. Color segmentation of original image

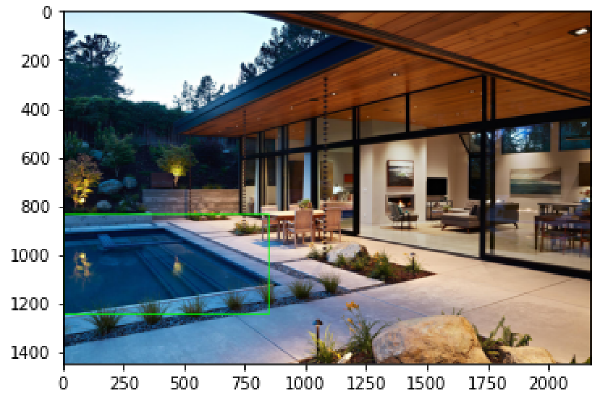

Fig. 11. Original image with no bins detected