Color Classification and Recycling Bin Detection

Harsh Kishorsingh Thakur
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, U.S.A.
HThakur@ucsd.edu

Nikolay Atanasov
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, U.S.A.
natanasov@eng.ucsd.edu

Abstract— The main idea behind this project is sensing part of the Robot. In order to do such sensing, I am assuming that the robot has a camera embedded in it and it takes pictures of the surrounding. Now the goal of this paper is to classify the image find a blue color and check whether this blue color represents the recycling bin or not. In order to fulfill this goal, it is necessary to design an unconstraint optimization algorithm that will help us solve this problem.

I. Introduction

Nowadays, people are getting lazier than ever before. Also, one of the most important challenges in the current pandemic situation is we are short of labor. Thus, we need automation to do certain kinds of jobs to fulfill the labor demand. One of such works is picking up the garbage from door to door.

Now my goal is to make a robot that is embedded in a truck and check whether the object is a recycling bin or not. I have different criteria to check whether this is my object or not. First of all, I need to check the color of my object. In this case, the object needs to be blue. The main problem here arises due to different lighting situations, in which the color blue might look different than the actual blue. So, I need to train the robot to check the different shades of blue as well. Secondly, once we find a blue color or shade of blue color, we have to check whether that blue color represents a rectangle or not. Here I am assuming that the shape of the recycling bin will be a rectangle and this is the universal case.

Using these two conditions I have to train a model that accurately classifies the color and detects the recycling bin.

II. Problem Formulation

Color classification for image:-

Color Classification

1: Input: A color image. [23, 23, 3] or 1st pixel of this image.
2: We have a labeled data set. (Supervised learning)
3: D := $\{x_i, y_i\}$ where i=1, 2, .., n of iid examples
    $x_i \in R^3$ and $R \in [0, 1]$ and
    $y_i \in \{1, 2, 3\}$
4: Pass the input to the function
4: This function should do the classification of colors
5: Output: Give a label to given pixel i.e. $y \in \{r, g, b\}$ or $\{1, 2, 3\}$

Bin Detection from the given image.

Bin Detection

1: Input: An image of any size.
2: We have an unlabelled data set.
3: In this case, humans can label it so technically it is labeled data-set.
4: General measurement of the recycling bin, i.e. height and width.
5: Task 1 is to do the labeling of color blue associated with recycle bin. Note: Blue of the bin can be different from blue of the sky.
6: Pass it our the function
7: This function does the classification of colors and should give each pixel their label similar to color classification.
8: Output: Coordinated of the bin - [[x1, y1, x2, y2]] in a list, where (x1, y1) and (x2, y2) are the top left and bottom right coordinate respectively.

The main problem is to find the recycling bin from an image using color classification and general shape statistics of the recycling bin.

III. Technical Approach

A. Color classification using Naive Bayes approach

I have defined the table which is a simple pseudo code of what I have written in my code.

Color Classification

```
1: Input: A color image. [23, 23, 3]
2: Pick the first pixel of this image: [r, g, b]
3: We have a labelled data-set. (Superwised learning)
4: Classify the dataset in three label i.e. Red, Blue, Green
5: From the data set train the model.
6: probability= total number of y_i / total number of data
7:          y_i just refers to ith label
8:          find the parameters from the data-set
9:          mean_i <- summation( r_i, g_i, b_i) / total number of i
10:         varince_i <- summation(x_i - mean_i) / total number of i
11:
12: Once we have our parameters we find the label for new data using
    Normal Distribution.
13: We predict the data and find our output label
```

Here I assume that our data is IID and the distribution is Gaussian Random Variable. The given data is in D:= $\{X_i, Y_i\}$ where i=1, 2, …, n and each of the data is Independent Identically Distributed. Where $X_i \in R^3$ and $R \in [0, 1]$ which defines the range of each pixel it can be. Here the image are initially read between [0, 255] ∈ Integers and inside the code they are converted to Real Numbers. And here we are doing classification so we use 0-1 Loss Function.

$$\min_h Loss_{0\text{-}1}(h) := \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{h(\mathbf{x}_i)\neq y_i} = \#\text{ of times } h \text{ is wrong about the labels}$$

Here $h(x_i)$ is the predicted output we find from our trained model.

I am using the generative model h(x) = arg $\max_y$ p(y, x; w) with parameters w and w ∈ R^3. Along with generative model I am Maximum Likelihood estimation (MLE) that maximizes the likelihood of the data D given the parameters w. Where my w is mean and variance of the data set. Thus the distribution model is $p(y_i \mid \theta)$ and he Gaussian distribution for $p(x_i \mid y_i, w)$ where w := {μ, σ^2}. Thus,

$$p(y_i \mid \boldsymbol{\theta}) := \prod_{k=1}^{K} \theta_k^{\mathbb{1}\{y_i=k\}} \qquad p(x_{il} \mid y_i = k, \boldsymbol{\omega}) := \phi\left(x_{il}; \mu_{kl}, \sigma_{kl}^2\right)$$

[1]where 1 represents the indicator function and ∅ represent Gaussian distribution.

I obtain parameters of the Gaussian distribution using the following equation.

$$\theta_k^{MLE} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\{y_i = k\}$$

$$\mu_{kl}^{MLE} = \frac{\sum_{i=1}^{n} x_{il}\mathbb{1}\{y_i = k\}}{\sum_{i=1}^{n}\mathbb{1}\{y_i = k\}} \qquad \sigma_{kl}^{MLE} = \sqrt{\frac{\sum_{i=1}^{n}(x_{il} - \mu_{kl}^{MLE})^2\mathbb{1}\{y_i = k\}}{\sum_{i=1}^{n}\mathbb{1}\{y_i = k\}}}$$

[1] Here x represents the input data, y represents the output label and k represents the class I am finding the parameters for.

Now in order to predict the label for new pixel we use the previous obtained parameters of our model and find the predicted label using the given equation.

$$y_* = \arg\min_k \left\{\log\frac{1}{\theta_k^2} + \sum_{l=1}^{d}\log\sigma_{kl}^2 + \frac{(x_{*l} - \mu_{kl})^2}{\sigma_{kl}^2}\right\}$$

[1] For my project for classification k ∈ {1, 2, 3}.

B. Bin Detection using Naive Bayes approach

In order to classify the bin from a given image. I first need to label the data as the data is not labelled. I used the provided roipoly function to classify the image in blue and not blue bin. To find the blue region in roipoly I create a bounding polygon around the bin and all the blue inside the polygon will be be blue. And for non blue region I create a slightly big polygon outside the recyclin bin and all the region outside the polygon are considered non-blue. This is how I create the dataset.

Once the dataset is ready, I find the parameters as discussed in part A. Color Classification. Thus, in this case the label are just y ∈ {0, 1}. i.e. {non-blue, blue}.

Bin Detection

1: Input: An image of size [m, n, 3] (3 represents RGB space)
2: Label the image using Roipoly function
3: Now we have a labelled data-set. (Superwised learning)
4: Classify the dataset in two label i.e. Blue and Non-Blue. This is not just normal blue but blue resembling to recycling bin.
5: From the data set train the model. (Similar to Color Classification)
6: We create a mask image. This is done by iterating over each pixel and classify (Testing) whether the given pixel is our blue or not.
7: If pixel == blue:
8: mask = 1
9: Else:
10: mask = 0
11: Thus we obtain a masked image that is either 0 or 1.
12: Once we have the masked image, I do 4 morphological operations.
13: a. Gamma Corection
14: b. Dilation
15: c. Erosion
16: d. Dilation
17: This helps me getting the desired bounding box.
18: I choose the bounding box, only if
19: a. The box is greater than 1% of the image
20: b. Size of height is more than width of the box.
21. Once I have the bounding box I return the coordinates in the form of a list.

For the testing phase, our k is now 2 instead of 3 in first case. But the data is similar X ∈ {red, green, blue}. But, the now the representation is in matrix instead of a single column vector. In order to increase the efficiency of my algorithm I convert the data into a column vector using the np.reshape function. Thus after labeeling the data, I get the bounding box according to my project. Thus the output is the final list.

IV. Evaluation

A. Color classification using Naive Bayes approach

After running the Naive Bayes Classifier on my dataset for color classification, I obtain the following parameters (rounded up to 6 decimal points):

	red	green	blue
probability	0.365999	0.324580	0.309421
mean	[0.752506, 0.348086, 0.348912]	[0.350609, 0.735515, 0.329494]	[0.347359, 0.331114, 0.735265]
variance	[0.037059, 0.061969, 0.062023]	[0.055735, 0.034786, 0.056022]	[0.054538, 0.056833, 0.035741]

Table 1. Parameters for Color Classification

The accuracy that I get using this parameters is as follows:

	red	green	blue
validation	1.0	0.975904	0.975904
test	-	-	0.980392

Table 2. Accuracy of Color Classification

These test results are pretty good just considering the algorithm I am using i.e. Naive Bayes. I don't have accuracy score for red and green color for test set because I was not able to on red and green pixels.

The way that I think can improve the model performance might be adding more data to the data set which would tweak the parameters in the right direction. Or trying a Logistic Regression which would have less variance, which will help in getting better results.

B. Bin Detection using Naive Bayes approach

After training the data using the Roipoly function, the parameters for the Naive Bayes Classifier that I get are as follows(rounded up to 6 decimal points):-

	not blue	not blue	blue
probability	0.920678	0.920678	0.079322
mean	[124.5708292, 123.684155, 117.645483]	[124.5708292, 123.684155, 117.645483]	[29.353045, 63.731723, 153.042128]
variance	[4660.420820, 4172.190055, 4778.856957]	[4660.420820, 4172.190055, 4778.856957]	[691.239347, 1388.344192, 3004.799925]

Table 3. Parameters for Bin Classification

I duplicated the column for non-blue class just to do vectorization in python.

One important that I noticed while labelling the dataset. Class imbalance plays a very important role while it comes to training. I trained all 60 images and the blue classes had very very less data compared to non-blue class. This made the parameters susceptible to a large variance, and all of the bin in my classifier turned out to be non-bin. I think this is one reason, also in some images there was blue sky, which I labelled as non-blue class. This made lot of my blue recycling bin to classify as non-blue and it went terribly wrong.

I also tried the image classificationusing the parameters obtained from part A but this went wrong, because the blue images in part A are quite different from what is the color of the recycling bin.

The most important thing gamma correction, dilation and erosion [2]. This helps me to erode the non-important features from the recycling bin, also dilation helps me to retain the features inside the dustbin. Because sometimes dustbin have things written on them and this helps me just retain them back and considered them as recycling bin.

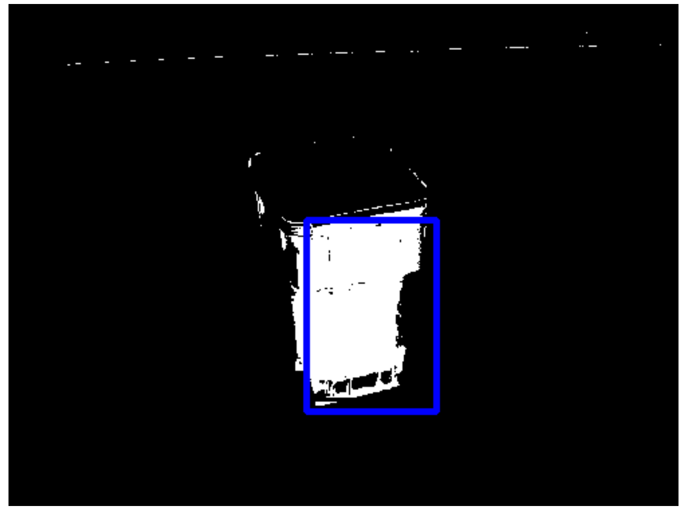

Fig 1. Image 61

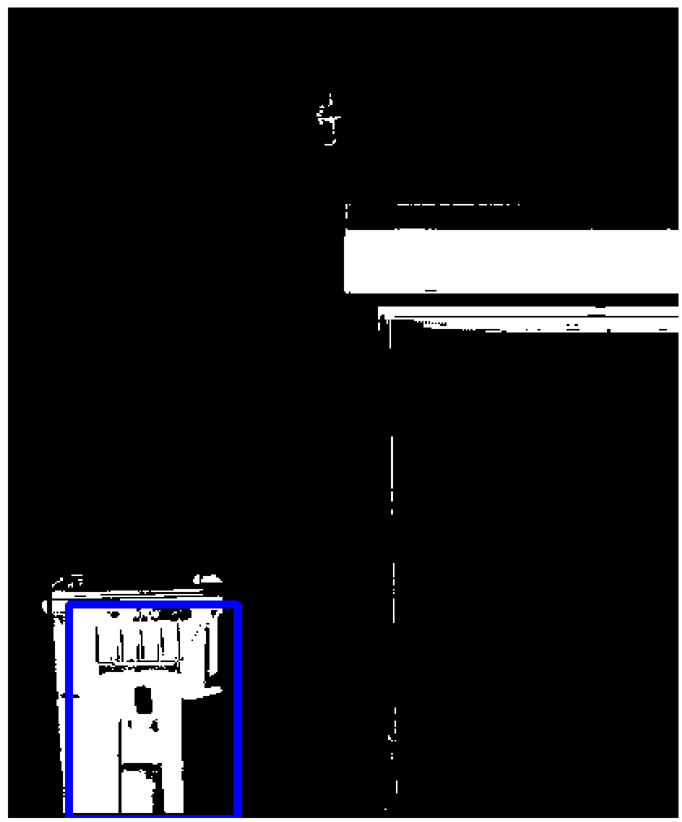

Fig 2. Image 62

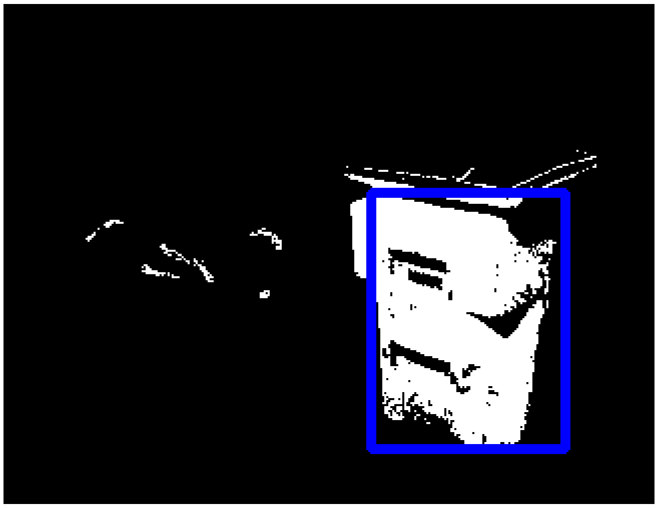

Fig 3. Image 63

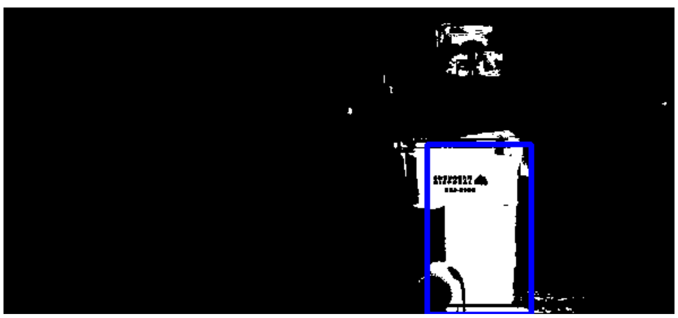

Fig 4. Image 64

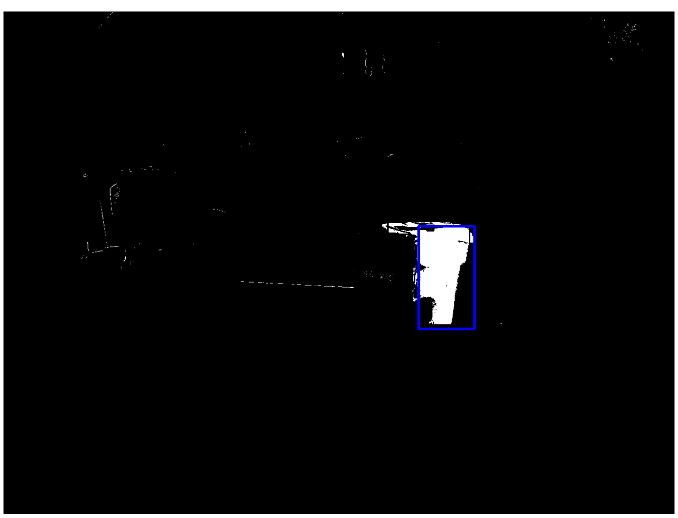

Fig 5. Image 65

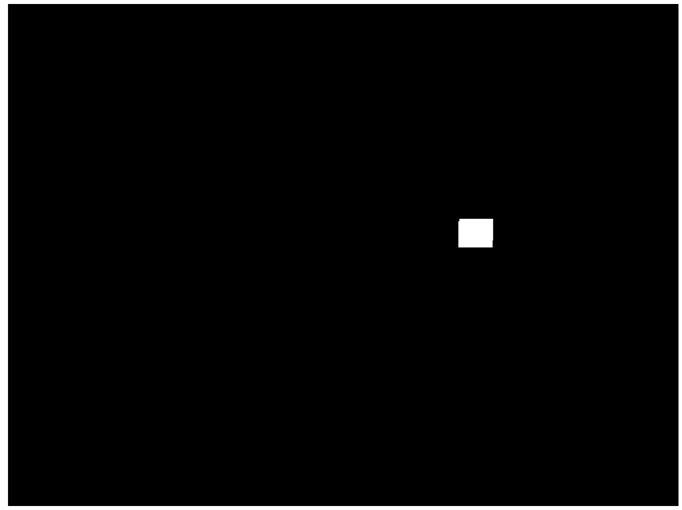

Fig 6. Image 66

In this case, small part of car is considered as dustbin but with the help of shape statistics, we can classify that this is not a dustbin.

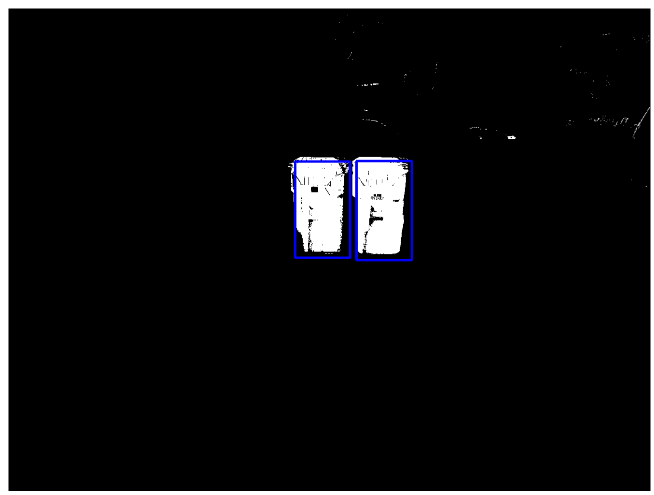

Fig 7. Image 67

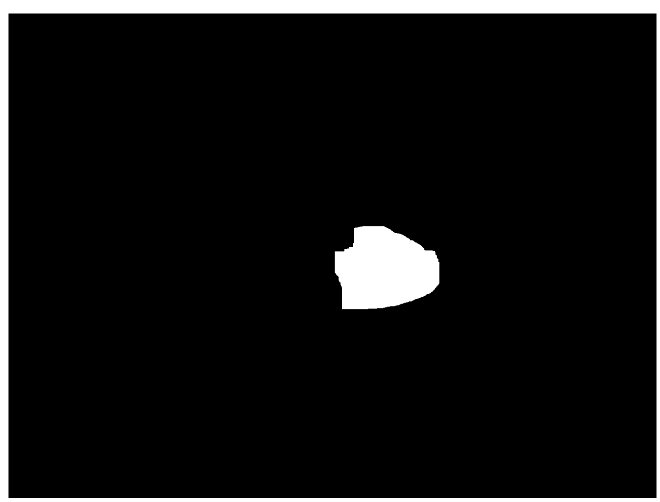

Fig 8. Image 68

Fig 9. Image 69

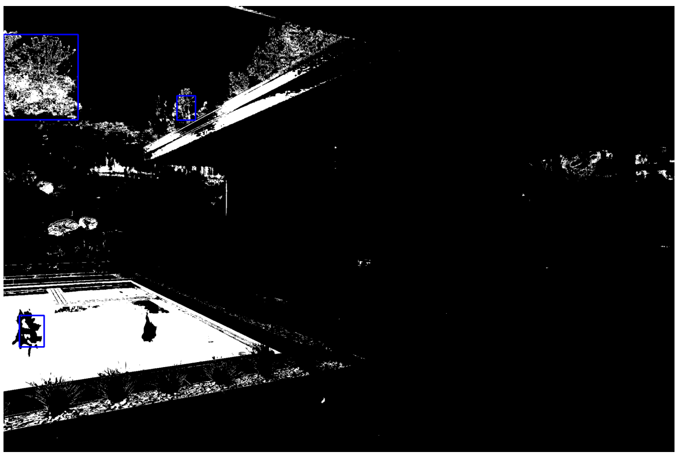

Fig 10. Image 70

In the case of this image, for some reason I was not able to classify the image properly, but this can be made better by training a different class of water blue and sky blue, which would help me a lot.

And so in this case, the case of mixture of Gaussian model would be perfect, because we have lot of data and inside of a recycling bin there are different shades of blue. The mixture of Gaussian model will be easily able to detect the different shades of a single color and thus making the bin detection more accurate and susceptible to lighting conditions.

The final result that I obtained on my validation dataset is 8/10. For some reason the first image is getting correction bin but it is not aligning with the required plot. And for the testing data I have 9.75/10.

References

[1] All the equations in Techinal Approac are taken from the lecture slide number 5.

[2] Discussed with Sambharan, Lando Li.