Color Classification and Recycling Bin Detection

Vaibhav Bishi
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, USA
vbishi@ucsd.edu

***Abstract*—This project presents an approach for color classification and blue recycling bin detection using Gaussian Discriminative Analysis. In this project, we have built probabilistic color classification models to distinguish among different colors and recognize recycling-bin blue color. Our objective is to detect blue recycling bins using these models. The results show very good bin detection rates.**

Keywords—Gaussian Discriminant Analysis, Maximum Likelihood Estimation, Bayes Decision Rule, OpenCV, roipoly, scikit-image

I. Introduction

Vision is the primary artifact that allows us to recognise, analyse and interact with objects in the real world. Human eyes can register light in a wide spectrum of colors which essentially helps us differentiate between objects. Thus, color classification is of great significance and is crucial to many real world applications such as colorblind assistance, image segmentation, image retrieval and object detection. We can then implement our knowledge of color classification and object detection in the industry, traffic monitoring and autonomous vehicles.

In recent times, with the advent of smart cities and self-driving cars, object detection is extremely important for safety of people and smooth traffic on the roads. Recycling bins make a good case to demonstrate object detection. This is because recycling bins are really important for proper waste disposal and thus, are present in abundance on every corner on every street in every city. Detection of recycling bins can prove essential to autonomous vehicles as they can set the bins as references on the street map and thus, they can know where the end of a road or a corner is making it impossible for them to go outside the bounds of the lane. This will prevent cars from harming any pedestrian. Another application of recycling bin detection can be automatic loading and unloading of the bins by robots for which clearly tracking the bin first is important.

In the first part of this project, we build a probabilistic color classification model from the pixel data of various shades of red, green and blue pixels that classifies these pixel values into red, green and blue colors. For this end, we take a dataset consisting of training images and validation images, where each example in the set is a 28 x 28 image with a single RGB value at all pixels. Based on their labels of red, green or blue, we train the data using a single Gaussian generative model. Specifically, we implement a trenary model using Gaussian Discriminant Analysis for pixel color classification into red, green or blue colors.

In the second part of the project, we build a probabilistic color classification model from various images of blue recycling bins and other objects on the street that recognizes recycling-bin blue color. For this end, we implement the Gaussian Discriminant Analysis based model from the first part. We use the trained model to segment unseen images into the desired blue regions and then, given those blue regions, we detect the blue recycling bins by drawing bounding boxes around them.

II. Problem Formulations

A. *Color Classification*

We are given a set of training images, where each example in the set is a 28 x 28 image with a single RGB value at all pixels. Consider that we have a labelled training dataset in the form of $X \in \mathbb{R}^3$ representing the RGB pixel values and $y = \{1,2,3\}^n$ representing the labels: Red=1, Green=2 and Blue=3, where n is the number of examples.

Our objective is to train this model for parameters that best represent our (X, y) data for color classification. In other words, we need to find some parameter θ and ω that maximizes the probability distribution of our data given by $p(y, X|w, \theta)$.

B. *Recycling Bin Detection*

We are given a set of training images consisting of blue recycling bins and other objects. After implementing roipoly, we have a set of labelled RGB values as examples. Consider that we have a labelled training dataset in the form of $X \in \mathbb{R}^3$ representing the RGB pixel values and $y = \{1,2,3,4\}^n$ representing the labels: binblue=1, otherblue=2, green=3 and brown=4, where n is the number of examples.

. Our objective is to train this model for parameters that best represent our (X, y) data for color classification. In other words, we need to find some parameter θ and ω that maximizes the probability distribution of our data given by $p(y, X|w, \theta)$. Given these parameters, we need to segment unseen images into the desired blue regions and then, given those blue regions, we need to detect the blue recycling bins by drawing bounding boxes around them. Finally, we need to get the bounding box coordinates (x_1, y_1, x_2, y_2) that represent the top-left and bottom-right corners of the boxes.

III. Technical Approach

Our main objective here is to build a probabilistic color classification model. For this end, we make use of a particular kind of Machine Learning model which is a single Gaussian Generative model. This type of model can approximate the required unknown data-generating probability density functions. Thus, it can generate new examples (x, y) by sampling from the optimal distribution $p(x, y)$ which makes it very effective for classifying new examples. For our project, we have implemented the Gaussian Discriminant Analysis model as opposed to the Naïve Gaussian Bayes model, since it generates a model without any conditional independence assumptions on $p(x_i|y_i,\omega)$, thus making our predictions more accurate.

Gaussian Discriminant Analysis uses a generative model $p(y,x|\omega,\theta)$ for discrete labels $y \in \{1,\dots,K\}$ where θ is set of parameter that models the marginal $p(y)$ and ω is a set of parameters that models the conditional $p(x|y,\omega)$. For optimal distribution, we need to maximise the following expression:

$$p(y, X \mid \omega, \theta)$$
$$= p(y \mid \theta) p(X \mid y, \omega)$$
$$= \prod_{i=1}^{n} p(y_i \mid \theta) p(x_i \mid y_i, \omega)$$

where, $p(yi \mid \theta) := \prod_{k=1}^{K} \theta_k^{1\{y_i=k\}}$

$$p(x_i \mid y_i = k, \omega) := \phi(x_i ; \mu_k, \Sigma_k)$$

We obtain the MLE parameters for this model by solving the following constrained optimization:

$$max_{\theta,\omega} \ log\, p(y, X \mid \omega, \theta)$$

where, $\sum_{k=1}^{K} \theta_k = 1$

The MLE estimates of θ and ω are:

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

$$\mu_k^{MLE} = \frac{\sum_{i=1}^{n} x_i 1\{y_i = k\}}{\sum_{i=1}^{n} 1\{y_i = k\}}$$

$$\Sigma_k^{MLE} = \frac{\sum_{i=1}^{n}(x_i - \mu_k^{MLE})(x_i - \mu_k^{MLE})^T 1\{y_i = k\}}{\sum_{i=1}^{n} 1\{y_i = k\}}$$

Now, given any new example we can get the class it belongs to using our trained model as:

$$y_* = argmax_y \ \log p(x_*, y|\theta^{MLE}, \omega^{MLE})$$

A. *Color Classification*

Given the labelled training images, we implement the Gaussian Discriminant Analysis model discussed above to train our model and get the required parameters θ, μ and Σ. Using these parameters, we classify the images based on the Bayes decision rule as shown in the last equation. We implemented our code for classifying blue pixels on the validation set and tested it on unknown images.

B. *Recycling Bin Detection*

Given the training set of images containing blue recycling bins and other objects, we use the roipoly function in python to label multiple sets of RGB pixels extracted from these images into 4 broad classes: bluebin=1 (for recycling-bin blue), otherblue=2 (for other shades of blue), green=3 (for green, yellow, white) and brown=4 (for brown, black, red). We implement the same Gaussian Discriminant Analysis model as in color classification case, but now with four classes instead of three. Using the trained parameters, we then segment the recycling-bin blue-like regions from the validation images to get binary masks of bin-blue and not bin-blue regions. Afterwards, we detect the recycling bins using regionprops and label functions from scikit-image by drawing bounding boxes on the segmented regions while considering a similarity index for the typical recycling bin shape and size by checking for area of the box to be greater than 5000 pixels so that it doesn't take other smaller patches of blue regions and also for the height-width ratio to be within a range of 0.8 and 2.5. We implemented our code for bin detection on the validation set and unknown test images.

IV. Results

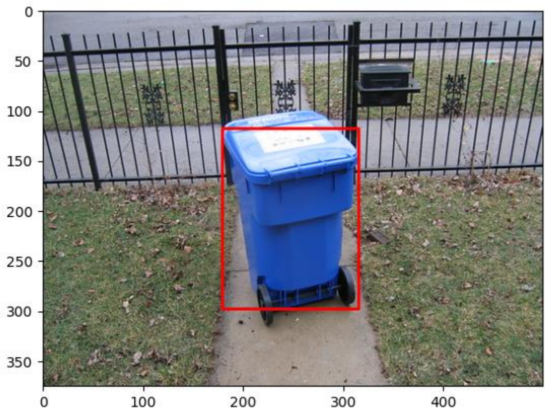
A. *Color Classification*

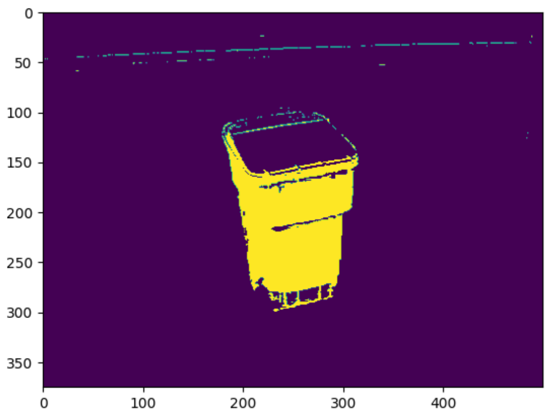
We successfully implemented our code for blue pixels on the validation set with precision=1 (all correct) and tested it on unknown images with a score of 9.93/10.

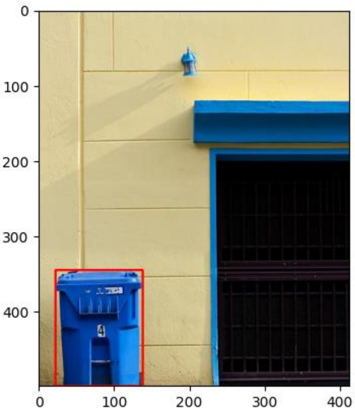
B. *Recycling Bin Detection*

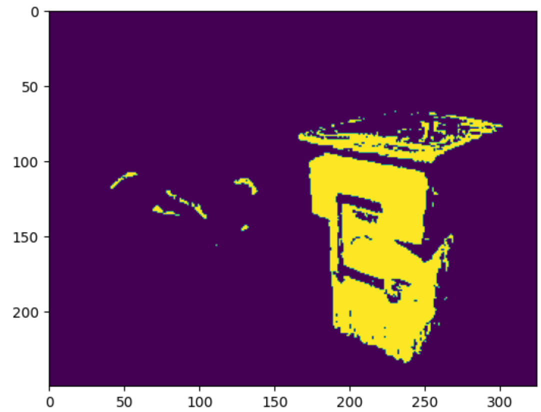
We successfully implemented our code for bin detection on the ten validation images with all ten being 100% accurate and tested it on unknown test images with a score of 7.18/10. This inaccuracy on unknown images can be attributed to the fact that some images may contain bluebin-like-colored but not exactly blue-bin regions which maybe too big since we did not set a bound on the maximum area of the bounding box.

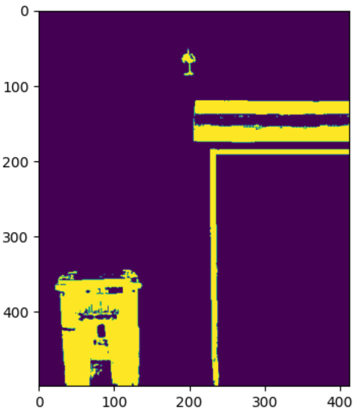
For the ten validation images, their segmented mask images and these images with corresponding bounding boxes for the bins have been shown below.





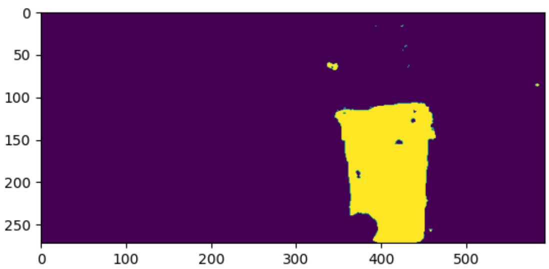

0
100
200
300
400
0 100 200 300 400


0
50
100
150
200
0 50 100 150 200 250 300

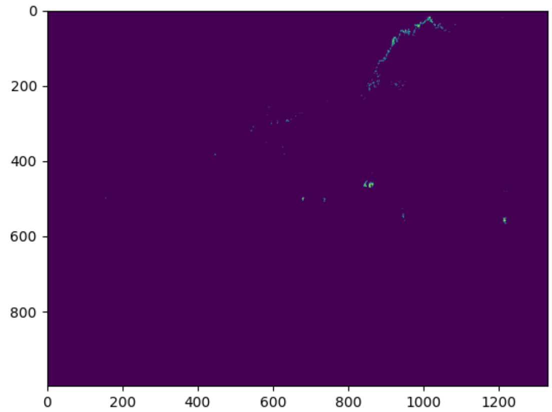

0
100
200
300
400
0 100 200 300 400


0
50
100
150
200
250
0 100 200 300 400 500

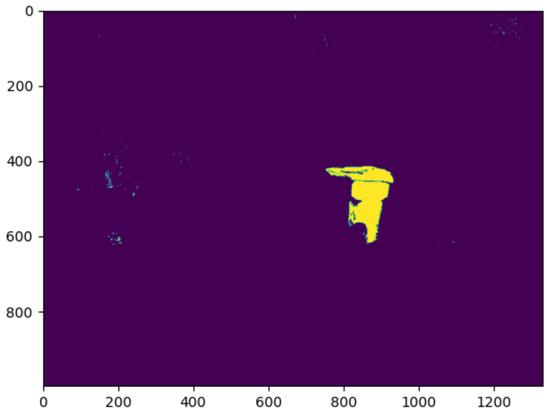

0
50
100
150
200
0 50 100 150 200 250 300


0
50
100
150
200
250
0 100 200 300 400 500

0
200
400
600
800
0
200
400
600
800
1000
1200



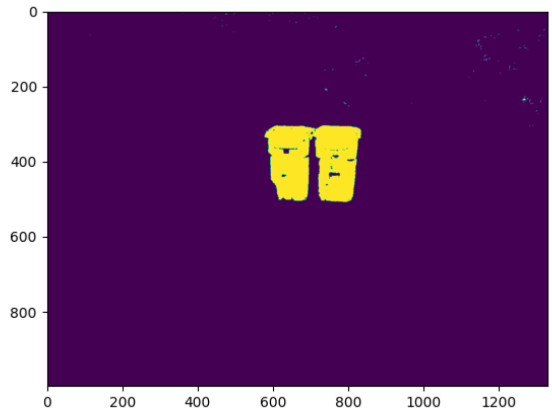
0
200
400
600
800
0
200
400
600
800
1000
1200



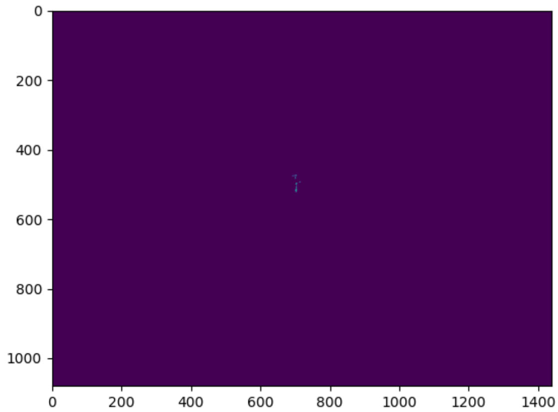
0
200
400
600
800
0
200
400
600
800
1000
1200

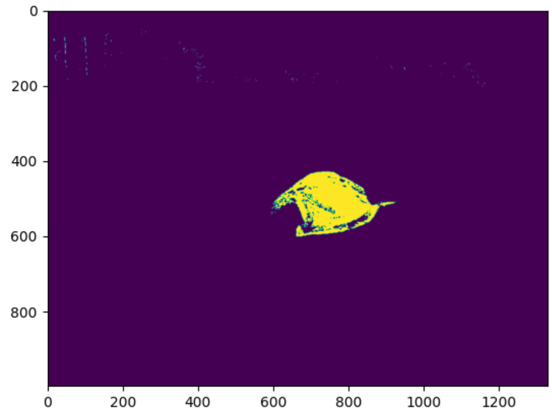
0
200
400
600
800
0
200
400
600
800
1000
1200

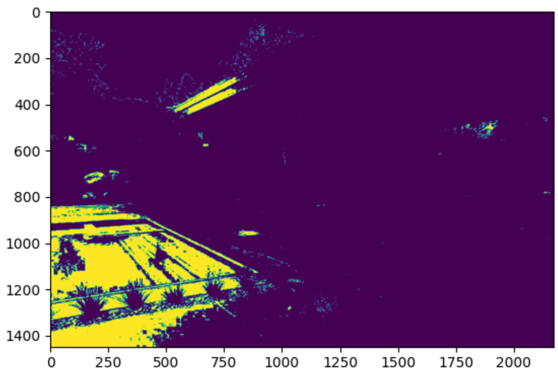
0
200
400
600
800
0
200
400
600
800
1000
1200



0
200
400
600
800
0
200
400
600
800
1000
1200







In conclusion, our model of Gaussian Discriminant Analysis and our code did pretty well on both the validation and test images. Although it can be improved still using better parameters and training sets and also implementation of better Machine Learning algorithms.