Blue color classification and detection of a blue recycling bin based on its rectangular shape: An academic project

Orish Jindal
Department of Electrical Computer Engineering
University of California, San Diego
ojindal@ucsd.edu

I. *Abstract*— This paper presented the Gaussian Discriminant Analysis approach to classify pixels of given images in the first part and then extending this approach to detect the recycling bins from real world images based on their blue color and typical rectangular shape in the second part of this paper.

Index Terms— Gaussian Discriminant Analysis.

II. Introduction

Object detection in an image is a computer vision technique that allows us to identify and locate objects. With this kind of identification and localization, object detection can be used to count objects in a scene and determine and track their precise locations, all while accurately labeling them. It is important to understand that Object detection is different from image recognition as image recognition assigns a label to an image. A picture of bins receives the label “bin”. A picture of two bins, still receives the label “bin” whereas object detection draws a box around each bin and labels the box “bin”. The model predicts where each object is and what label should be applied. In that way, object detection provides more information about an image than recognition.

Any progress in this particular area of object detection and classification is very important as this is a fundamental component in SLAM problem. A dream of autonomous vehicles is unachievable without perfecting the techniques of object detection. Also, Navigation, Pick and Place and additional robotics activities are vital in performing surgeries using the medical assistive robots.

Since we are clear about the importance of this this machine learning problem, we can be specific about what is done the project that is being presented in this paper. Now, this project is developed in two stages: Classification of pixels of the test images, Bin detection with the classifier of previous stage.

<u>First: Classification of pixels</u>

Data was provided in the form of training images labeled Red, Green and Blue in separate folders. The goal was to train a probabilistic color model from pixel data to distinguish among red, green, and blue pixels. The data consisted of a training set and validation set. Each example in the training or validation sets is a 28×28 image with a single RGB value at all of its pixels. There are various classification models that could have been used in this process such as Logistic Regression or Naive Bayes or Gaussian Discriminant Analysis. Every model has its advantages and disadvantages where the logistic regression model is quite calculation expensive while training and Naïve Bayes assumes that there is no co-relation between each feature of the feature vectors which leads to a diagonal co-variance matrix. This assumption leads to the loss of vital information which limits the accuracy of the model. Hence, a slightly superior version of Naïve Bayes model is used in this project which eliminates the naïve assumption of the independence of features in the feature vectors, called Gaussian Discriminant Analysis. This approach tens to be more calculation intensive than the Naïve Bayes approach but gives more accurate results.

<u>Second: Bin detection</u>

In this part of the project, the machine learning model of the previous part is used but this time, in a much more complex problem. In this problem, the training data was given as images where almost all of them consisted of a blue colored recycling bin in the picture. Data was hand-labeled using the RoiPoly technique with python. Data was labelled for four classes one of which, was our target class that must be detected. Then a machine learning model based on again, Gaussian Discriminant Analysis was built and the parameters were learned according to the labeled data which were further used to create mask of the testing images defining all the pixels that were not the part of bin (that has to be detected) as zero and rest as one. This resulted in a binary image formation according to which, a bounding box based on the shape of bins was placed on the original image.

III. Problem Formulation and Technical Approach

Since we already theorized the problem, it’s time to see that problem in some mathematical terms.

<u>First: Classification of pixels</u>

The images that were given, converted into a matrix format consisting of a vector V of dimension 3 as [R,G,B]. Since each image consisted of 28x28 pixels and each pixel was holding the same value throughout the image, only one pixel (top left corner) of each image is selected as this will not affect the parameters which are formed by averaging different aspects of the matrices.

A single matrix 'X' of dim: nx3 is then formed by concatenating the 1x3 vectors generated from each image of all the three sets {Red, Green, Blue} and a parallel list 'y' was formed that contained the label of images as {1,2,3} corresponding to the {Red, Green, Blue} pixels.

Now in the learning process, we initiated the tree parameters of the Gaussian Discriminator that are calculated by maximizing the likelihood of the data 'D' as follows:

$$\Theta = \text{arg.}\, max_{\Theta}\, P_{\frac{X}{Y}}\, (D^{\Theta} \mid i\, ;\, \Theta)$$

Where Θ represents our parameters: μ, cov and θ.

$$\theta_{\mathrm{k}}^{\mathrm{MLE}} = \frac{1}{n} \Sigma_{i=1}^{n} 1\{y_i = \mathrm{k}\}$$

- $\mu_{\mathrm{kl}}^{\mathrm{MLE}} = \frac{\Sigma_{i=1}^{n} x_{il} x_{il} 1\{y_i = \mathrm{k}\}}{\Sigma_{i=1}^{n} 1\{y_i \mathrm{i} = \mathrm{k}\}}$
- $\mathrm{cov}_{\mathrm{kl}}^{\mathrm{MLE}} = \frac{\Sigma_{i=1}^{n} (x_{il} - \mu_{\mathrm{kl}}^{\mathrm{MLE}})^2 1\{y_i = \mathrm{k}\}}{\Sigma_{i=1}^{n} 1\{y_i \mathrm{i} = \mathrm{k}\}}$

These parameters were calculated for each class and then used to calculate the Mahalanobis distance $(d_i(\mathrm{x}, \mu_i) + \alpha_i$) of the region of each class from the vector formed by the pixel that is being classified. The minimum distance is then considered, and the vector is put in the corresponding class. The formula is a as follows:

$$i^{*} = \text{arg.}\, min_{\mathrm{i}}\, [d_i(\mathrm{x}, \mu_i) + \alpha_i\,]$$

Where, i represents the classed and,

- $d_i(\mathrm{x}, \mu_i) = (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)$
- $\alpha_i = \log[(2\pi)$^d] $|\Sigma_i|$ - 2logP_Y (i)

That way, a vector consisting of labels of each testing image is formed which denotes the best prediction of its class.

<u>Second: Bin detection</u>

Same approach is then extended to solve this much larger problem. In this, four classes are created as:

a) bin_blue
b) blue_not_bin
c) green_yellow
d) red_black

Now, the three parameters were calculated according to these four classes and this time, four mahalanobis distances will be created. But this time, instead of labeling them 1,2,3,4, we will use a different approach.

Each pixel when assigned to one of the four classes will pass through a filter where if the class '1' is assigned (bin_blue), the pixel is reconstructed and given a scalar value 1 and it is given the value 0 if the pixels belong to any other class. This will result in the generation of a binary mask of the original image which will then be used to draw a bounding box over the pixel cluster assigned with value 1. The dimensions for the bounding box are limited according to the typical dimensions of recycling bins that are observed in real life. The conditions are as follows:

height*width > 4000

Width < height < 2*width

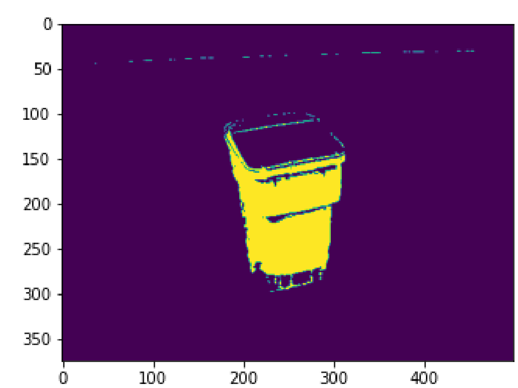
Where height and width denotes the height and width of the bounding box that has to be selected out of many boxes that will be generated.

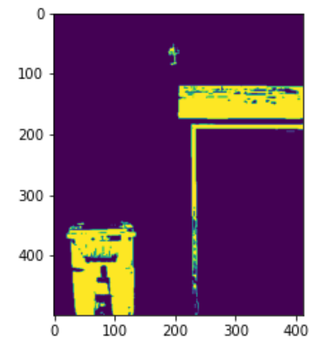
These classes are created to increase the accuracy of the discriminator as with a sufficiently large amount of training data, more number of class divisions will be considerate of the much smaller differences and hence will create a better segmentation of the image.

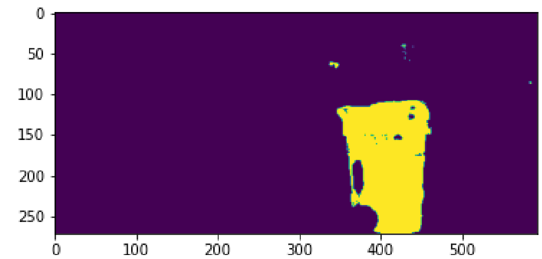
IV. Results

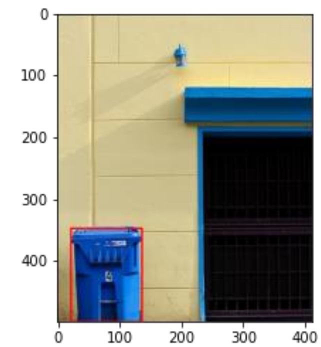
Part 1 resulted in the 100% accuracy on the validation data whereas the accuracy on the gradescope came out to be 99.3% .

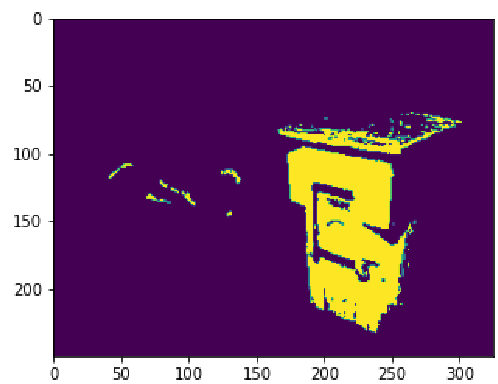
In part 2 of the project, 10 images were validated for the Gaussian Discriminant approach. The binary mask and the resulting bounding box are shown below for each validation image are shown below (a total of twenty images: two form of each image placed one below other respectively).

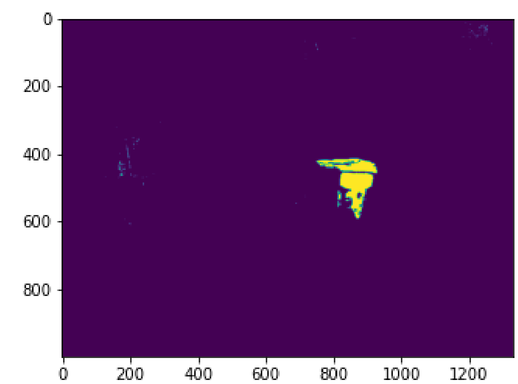


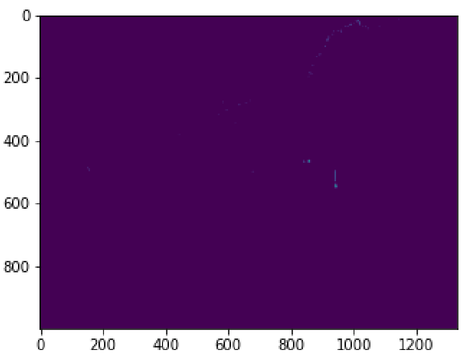
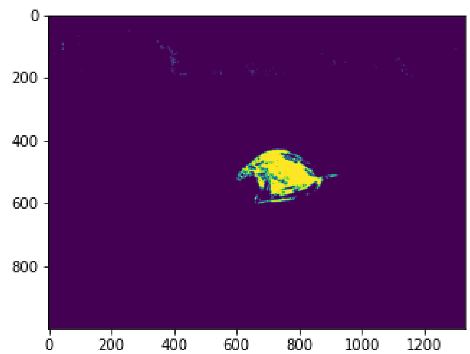


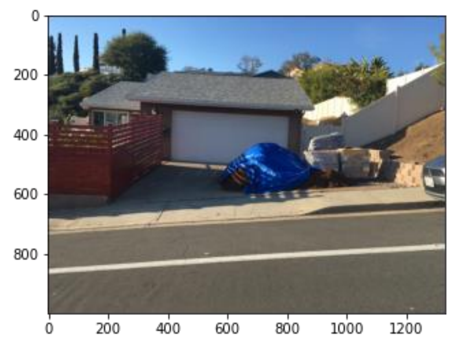
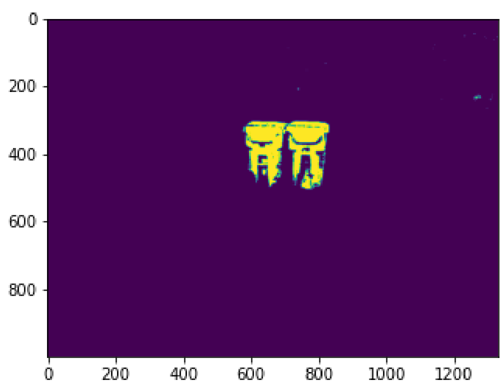
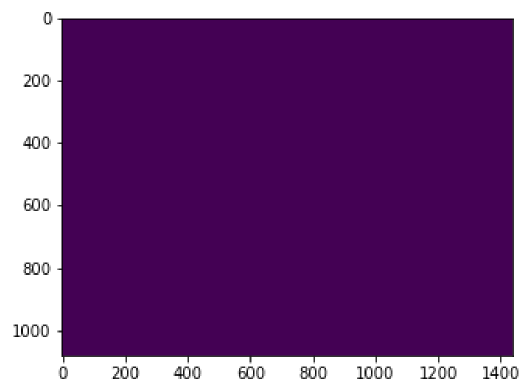


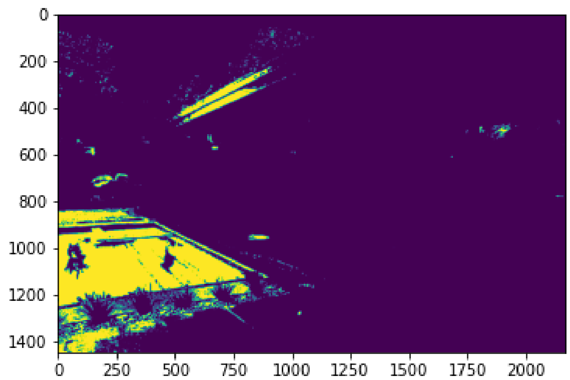








In set 1,2,3,4,5 (placed serial wise), the model is able to detect the blue bin accurately and in set 6,8,9,10, the bin was not there and the model avoided detecting wrong object which also corresponds to the full accuracy. It only failed to detect the bins in the set 7 where two bins are placed beside each other. The binary image is formed correctly but the program was not able to bound that portion with a rectangular box as its conditions did not allow the program to form a box that is more in width than height (width ere will be considered as the combined width of the two bins).

<u>The final parameters used in the model for part 1 of the project are as follows:</u>

- mean: 3x3 matrix with each row representing RGB approximate of a given class, μ =

```
0.7525060911938757 0.34808562478245914 0.34891228680821595
0.350609167770528 0.7355148898592014 0.3294935321918604
0.34735903110150496 0.3311135127716902 0.7352649546257728
```

- Covariance: 9x3 matrix with every 3 corresponding rows representing covariance of each class respectively, **Cov** =

```
0.037086702210567715 0.01844078389423107 0.018632848266471713
0.01844078389423107 0.06201456207728437 0.008581635746996158
0.018632848266471713 0.008581635746996158 0.06206845784048822

0.05578115269442662 0.017653267296060517 0.008739554602480012
0.017653267296060517 0.03481496393718223 0.01702340107506035
0.008739554602480012 0.01702340107506035 0.056068642903266026

0.05458537840583404 0.008552820244218096 0.01717350258926295
0.008552820244218096 0.056883076264007806 0.01830848688183290
0.01717350258926295 0.018308486881832904 0.03577190352880781
```

- Prior probability: 3x1 vector with each row representing of a given class, $\boldsymbol{\theta}$ =

```
0.36599891716296695
0.3245804006497022
0.3094206821873308
```

<u>The final parameters used in the model for part 2 of the project are as follows:</u>

- Mean: 4x3 matrix with each row representing RGB approximate of a given class, μ =

```
0.07561849928987706 0.21600890812404172 0.5930533485615389
0.5262430001974105 0.5787085534403065 0.46929080578295923
0.22415883884768822 0.14906764349236742 0.16538517946422188
0.31972349858207755 0.43398952722802814 0.600717106841458
```

- Covariance: 12x3 matrix with every 3 corresponding rows representing covariance of each class respectively, **Cov** =

```
0.00709540453936652 0.006899354033525645 0.005501986310413163
0.006899354033525645 0.008998846053464477 0.007327454586642257
0.005501986310413163 0.007327454586642257 0.011736169098115697

0.03677520333683274 0.03226928848044518 0.03508251275481498
0.03226928848044518 0.03220488005887255 0.03603000829604879
0.03508251275481498 0.03603000829604879 0.04924587381927316

0.03448645788186817 0.02143553520622551 0.01608139798726714
0.02143553520622551 0.01988958829917079 0.015658082638692245
0.01608139798726714 0.015658082638692245 0.013467696217403203

0.031699456043084794 0.03844482501254341 0.0388856421477982
0.03844482501254341 0.049274603010209014 0.05248483157612935
0.0388856421477982 0.05248483157612935 0.05918851804839117
```

- Prior probability: 4x1 vector with each row representing of a given class, $\boldsymbol{\theta}$ =

```
0.21217623689333986
0.21383408639545573
0.2663352324775649
0.3076544442336395
```

V. Refrences

[1] **Pattern Classification by Richard O. Duda, Peter E. Heart, David G. Stork.**