Color Classification and Bin Detection using the techniques of Parametric Learning and Gaussian Naïve Bayes Model

Sanchit Gupta
Department of Electrical and
Computer Engineering
University of California San Diego
sag006@ucsd.edu

***Abstract*—This report discusses about the process followed to train a probabilistic color model to distinguish between red, green and blue pixels using the techniques of parametric learning through Maximum Likelihood Estimation (MLE) and generative classification via Gaussian Naïve Bayes model. The approach for further extended to detect objects like recycling bins in an image. The overall model was tested on a set of validation images and completely unknown images. The results for the same are also presented in this report.**

Keywords—Model training, Maximum Likelihood Estimation (MLE), Gaussian Naïve Bayes, Object detection

I. INTRODUCTION

Object detection using Computer Vision in real-time is finding huge applications across numerous domains in the world today. Few of the most popular applications include detection of vehicles, pedestrians and other objects in autonomous driving, detection of obstacles by warehouse robots, delivery robots and underwater robots as well as real-time face recognition algorithms. As the world is shifting its reliance to emerging technologies for autonomy in surveillance, vehicle driving and robotic motions among others, it becomes extremely important to ensure that the algorithms implemented to attain autonomy are robust, reliable and intelligent to make accurate decisions according to the circumstances.

With the hardware of the vision systems advancing with each passing day, it is has become very easy to capture high resolution colour images or record high resolution videos over a wide field of view. The dependence lies with the software at the backend to filter the information of the interest from the data set and execute the algorithms accordingly. The model created to extract the required information from an image is discussed further in this report.

Any colour image is composed of numerous pixels. Pixel is the smallest item of information in an image. For instance, a 720p HD image will generally have 1280 x 720 pixels. Each pixel in a colour image has three components namely Red, Green and Blue in RGB format. CMYK (Cyan, Magenta, Yellow and Key/Black) is another colour format similar to RGB. HSV is a cylindrical colour model representing Hue, Saturation and Value of a pixel.

In object detection algorithms, colour classification becomes an integral part of the problem. It becomes important to first train the model to look for the pixels of desired colours and then, extract those particular objects from the image which dominantly contain the desired pixels. The same approach has been implemented in this project. In the first part of this project, the parameters have been calculated from a large training set consisting of more than 1000-pixel samples each for red, green and blue colours using the estimation strategy of MLE. The estimation parameters are calculated for each of red, green and blue sample sets. Further, a Gaussian Naïve Bayes model has been defined to classify an unknown pixel as one of the three colours.

This approach is further extended to determine objects like blue recycling bins from an image. As an image would contain many objects with the same blue colour as the bin, three different training sets have been defined to differentiate between the blue colour of the bin, blue colour of other objects in the image and other completely different colours present in the image. Each of these training sets consist of more than 20 samples. The estimation parameters are again calculated for these three training sets using MLE. These parameters are fed to the Gaussian Naïve Bayes model to extract only the pixels of the blue recycling bin. The binary mask is then created using OpenCV functions, where the extracted pixels of the bin are shown in white and the unwanted pixels of the image are blacked out. The contours present in the mask are finally analyzed to determine the resemblance of the object detected with the recycling bin.

II. PROBLEM FORMULATION

This section will discuss about the formulation of the colour classification and bin detection problems in terms of mathematical formulas. As MLE strategy is used to Gaussian Naïve Bayes generative model, it requires maximizing equation 1 over ω.

$$\max \omega \; P(y, X; \omega) \tag{1}$$

In equation 1, ω represents the distribution of red, green and blue pixels and X represents the RGB values of a particular pixel. P() denotes the Probability Density Function (PDF).

Let θ be another variable added in the formulation of MLE. In the case of color classification, θ represents the probability of a pixel being one of red, green or blue. Further, in the case of object detection, θ will assume the probability of a pixel falling in one of the classes of blue colour of the bin, blue colour of other objects in the image and other completely different colours present in the image. Thus, the MLE formulation now modifies to as given in equation 2.

$$\max \omega, \theta \; P(y, X; \omega, \theta) \tag{2}$$

Applying Gaussian Naïve Bayes model, the MLE formulation transforms into equation 3.

$$\max \omega, \theta \prod_{i=1}^{n} P(y_i|\theta) \prod_{l=1}^{d} P(X_{il}|y_i, \omega \tag{3}$$

As each pixel will be a vector consisting of R, G and B values, l in equation 3 denotes different dimension of vector X. Taking log in equation 3 and solving it, we obtain the MLE estimates for θ and ω. Equations 4, 5 and 6 gives the MLE estimates for probability (θ), mean (μ) and covariance (σ^2).

$$\theta_k = \frac{1}{n} \sum_{i=1}^{n} 1\{y_i = k\} \tag{4}$$

$$\mu_{kl} = \frac{\sum_{i=1}^{n} x_{il} * 1\{y_i = k\}}{\sum_{i=1}^{n} 1\{y_i = k\}} \tag{5}$$

$$\sigma_{kl}^2 = \frac{(\sum_{i=1}^{n} (x_{il} - \mu_{kl})\text{^}2 \; 1\{y_i = k\})}{\sum_{i=1}^{n} 1\{y_i = k\}} \tag{6}$$

Color Classification: In the color classification problem, equations 4, 5 and 6 are used to get the probability, mean and covariance of red, green and blue pixels given in the training set. For instance, the probability (θ) of blue pixels represents the count of images present in the training set of blue divided by the combined images of training sets of blue, green and red. Similarly, mean (μ) is obtained from each training set. The mean here will be a 3-dimension vector, consisting of average pixel values of R, G and B across all samples of one color. The covariance will be a 3 x 3 matrix consisting of values of covariance between R, G and B pixel values. These estimates as obtained from the training set are important in classification of a new, unknown sample during testing.

Bin Detection: Similar to color classification, the probability (θ) represents the probability of a pixel to be either in one of the sets of blue color of bin, blue colour of other objects in the image and other completely different colours present in the image. Accordingly, mean (μ) and covariance are calculated and stored from the training set images. These estimates help in extracting the pixels of the blue recycling bin from the unknown image.

Once the estimates are obtained from the training sets, they are fed to equation 7 to calculate the Gaussian PDF with the input being the pixel values of a new pixel. Equation 7 shows the final, generalized form of the Gaussian PDF formula.

$$Gaussian\ PDF = log\theta + \sum_{l=1}^{d} -\frac{1}{2}\log(2\pi\sigma_l^2) - [(X_{*l} - \mu_l)^T(\sigma_l^2)^{-1}(X_{*l} - \mu_l)]/2 \tag{7}$$

In equation 7, θ, μ and σ^2 are inserted as obtained from the training set. X_{*l} denotes the pixel vector of an unknown pixel. With these, the Gaussian PDF is calculated for every pixel, which helps in classifying the pixel into the right set.

Color Classification: Using equation 7, three different Gaussian PDFs are calculated for every new pixel using the parameters obtained from the training sets of blue, green and red. After analyzing, if the Gaussian PDF calculated using the parameters of blue is maximum amongst the three Gaussian PDFs, then the pixel is classified to be a blue color pixel.

Bin Detection: Very similar to color classification, the Gaussian PDF is calculated for every pixel in the image using the parameters obtained from the training sets of blue color of bin, blue colour of other objects in the image and other completely different colours present in the image. If the Gaussian PDF of a pixel calculated using parameters of blue colour of bin is maximum amongst the three Gaussian PDFs, then that pixel is selected as a pixel belonging to the bin and is used for creating the binary mask of the bin.

III. TECHNICAL APPROACH

The technical approach to solve the problems of color classification and bin detection are discussed in this section and are elaborated in sub-sections A and B below.

A. Color Classification

A training set with more than 1000 samples of pixel images for each of red, green and blue colors are given. The sample images consist of different shades of each color. Using the read_pixels() function given in the starter code of generate_rgb_data, the pixel values of each of the sample for each color is determined and is stored in a n x 3 matrix. 3 such matrices are obtained, one each for each color.

Once the pixel values of the training set are obtained, equations 4, 5 and 6 are used to calculate probability (θ) for each color, mean (μ) vector for each color and covariance (σ^2) matrix for each color. In total, nine sets of values are calculated. These parameters as obtained from the training set

are stored in text files, in order to be loaded back for the PDF calculations.

In the class PixelClassifier(), θ, μ and σ^2 variables are initialized and their values are loaded from the text files simultaneously.

The task for classification of new, unknown pixels as red, green or blue is executed by the classify() function. This function takes the input of a vector of R, G and B pixel values for a particular image. This vector along with the values of θ, μ and σ^2 are fed in equation 7 to calculate the Gaussian PDF. Three Gaussian PDFs are calculated here, one for each color, using the appropriate estimation parameters. These three PDFs are compared and the highest PDF is determined. The given image is classified to be of the color for which the Gaussian PDF is calculated to be the highest.

B. Bin Detection

The bin detection problem is an extended version of the color classification problem. As a training set, nearly 60 images with recycling bins of different colours and in different surroundings are provided. These images are first classified into three training sets:

- Desired Blue – Training set for blue color of the bin
- Undesired Blue – Training set for blue colors other than the blue color of the bin
- Other Colors – Training set for all the other colors present in the image

The images are classified into these three sets in order to cover the exhaustive range of variation in blue color of the bin, light invariance and effects of the surrounding. After segregation, the RGB data for these training samples is generated using the RoiPoly() function (Region of Interest Polygon) provided along with the project. RoiPoly() function helps in collecting the RGB values of desired pixels from the image. An example of functioning of this function is shown in Figure 1.

Figure 1: An example of usage of RoiPoly function

In bin_detector, a new function named generate_rgb_data() is defined to collect pixel values for the region of interest using RoiPoly(). For each of the training sets defined above, the RGB pixel values of an appropriate region of interest from all the images are collected and stored in a matrix. The matrix for each training set consisted of more than 40000 pixel values in total. These values are stored in a text file, one text file for each training set.

Once the pixel values are collected, the estimation parameters are calculated similar to that in color classification problem. Equations 4, 5 and 6 are used to calculate probability (θ) for each set, mean (μ) vector for each set and covariance (σ^2) matrix for each set. These parameters as obtained from the training set are stored in text files, in order to be loaded back for the PDF calculations.

In the class BinDetector(), θ, μ and σ^2 variables are initialized and their values are loaded from the text files simultaneously.

The segment_image() function performs the task of classification of the new pixels into the 'Desired Blue' set and creation of a binary mask for the bin. This function takes an input of an image in RGB format. The image as obtained in this function is a tuple of width, height and RGB values of each pixel. Obtaining pixels from the image, the Gaussian PDF is calculated for each pixel using equation 7 and the parameters of θ, μ and σ^2 of each training set. Three Gaussian PDFs are calculated here, one for each training set, using the appropriate estimation parameters. These three PDFs are compared and the highest PDF is determined. The given pixel is classified to fall in the set for which the Gaussian PDF is calculated to be the highest.

If a particular pixel is analyzed to be a pixel of 'Desired Blue' set, a value of 255 is stored in another mask matrix at the co-ordinates of the pixel. For pixels of other two sets, a value of 0 is stored in the mask matrix. Once all the pixels of the image are analyzed, a mask matrix of exactly the same shape as original image is obtained, containing only the binary values of 0 and 255. An example of the mask of the bin for one of the validation images is shown in Figure 2.

Figure 2: An example to show the original bin image (*0063.jpg from the validation set*) on the left and the mask as created from the algorithm deployed on the right

Since the mask matrix obtained is in float data type, it is converted to unsigned 8-bit integer type for compliance with OpenCV functions.

Further to creation of mask, the problem statement requires to draw a bounding box around the identified bin and compare with the bounding box drawn by the test program. In order to find the contours present in the mask image, OpenCV's findContours() function is used. findContours() function takes the input of mask image in unsigned 8-bit integer format and gives the output of a list of contours present in the image.

Each of the obtained contour is analyzed to draw the bounding box around the bin. Each contour is passed to the OpenCV's boundingRect() function to get the co-ordinates of left corner of the contour, its width and height. These contours are then filtered on the basis of two conditions:

- Height of the contour should be greater than the width of the contour
- Aspect ratio of width to height should be greater than or equal to 0.5

The condition on the aspect ratio is assumed according to the dimensions of the bins given in the problem statement. The contours which satisfy these two conditions are then stored in a list and are also used to draw a bounding box around the bin. The bounding box around the bin for the same image as shown in Figure 2 is shown in Figure 3.

Figure 3: Bounding box around the recycling bin for the same image as shown in Figure 2

In Figure 3, the blue bounding box is the one generated by the test program and the red color one is the one generated using the algorithm presented in this report. It shows that the red bounding box covers the bin nearly appropriately. The small contours which are seen at few places in the image are some of the noise contours, which can be filtered out through more filtering conditions.

IV. Results

This section discusses about the results as obtained for both, color classification and bin detection.

A. Color Classification

Following the technical approach as elaborated in Section 3 of this report, the images with shades of blue are classified and returned to the test code. As per the validation samples provided, an accuracy of 100% is obtained, whereas, a score of 9.94 is obtained on Autograder in Gradescope. The final parameters used by the Gaussian Naïve Bayes model are given below:

- $\theta\_red = 0.366$, $\theta\_green = 0.309$, $\theta\_blue = 0.325$
- mean (μ) vector for red: [0.752, 0.348, 0.349]
- mean (μ) vector for green: [0.350, 0.735, 0.329]
- mean (μ) vector for blue: [0.347, 0.331, 0.735]
- Covariance for red: [[0.037, 0.018, 0.018],
  [0.018, 0.062, 0.008],
  [0.018, 0.008, 0.062]]
- Covariance for green: [[0.055, 0.017, 0.008],
  [0.017, 0.034, 0.017],
  [0.008, 0.017, 0.056]]
- Covariance for blue: [[0.054, 0.008, 0.017],
  [0.008, 0.056, 0.018],
  [0.017, 0.018, 0.035]]

B. Bin Detection

The technical approach as elaborated in Section 3 of this report is followed to solve the bin detection part of the problem statement. The final parameters used by the Gaussian Naïve Bayes model from the three training sets defined are given below:

- $\theta\_Desired\_Blue = 0.316$, $\theta\_Undesired\_Blue = 0.233$, $\theta\_Other\_Colors = 0.451$
- mean (μ) vector for Desired Blue Set: [0.208, 0.328, 0.658]
- mean (μ) vector for Undesired Blue Set: [0.395, 0.521, 0.673]
- mean (μ) vector for Other colors Set: [0.404, 0.419, 0.370]
- Covariance for Desired Blue Set: [[0.035, 0.035, 0.014],
  [0.035, 0.041, 0.023],
  [0.014, 0.023, 0.036]]

- Covariance for Undesired Blue Set: [[0.038, 0.035, 0.026],
  [0.035, 0.041, 0.040],
  [0.026, 0.040, 0.052]]
- Covariance for Other Colors Set: [[0.064, 0.054, 0.041],
  [0.054, 0.054, 0.042],
  [0.041, 0.042, 0.044]]

The accuracy in the case of bin detection is determined by comparing the area of the bounding box around the bin created following the technical approach elaborated in Section 3 of this report and the bounding box created by the test code. On executing the test code, 100% accuracy was obtained for all images given in the validation set except for 0066.jpg, 0068.jpg and 0070.jpg. Even though the mask image and the bounding boxes are being created appropriately, it would require deeper attention to identify the loss in accuracy in these cases. All these three images do not contain the bin; however, they have shades of blue similar to that of the bin. Since a large training set was used, the algorithm has rightly classified very few pixels in these images under the 'Desired Blue' set. The mask images of these include some noise pixels which can potentially be the reason for loss in accuracy. Further, the contour filter algorithm has correctly filtered out many of the noise pixels and hence, it would require deeper attention to identify the loss in accuracy for these images.

On uploading the code in autograder in Gradescope, the score of 6 is obtained, making the final score to be 15.9346 out of 20. Bin 1, 3, 5 and 9 are not detected by the algorithm implemented in this report in the autograder. The segmented mask images and the bounding box images for each of the image in validation set are shown in Figures 4 to 13. In all the images, the blue bounding box is as created originally by the test code and the red bounding box is the one created following the algorithm elaborated in this report.

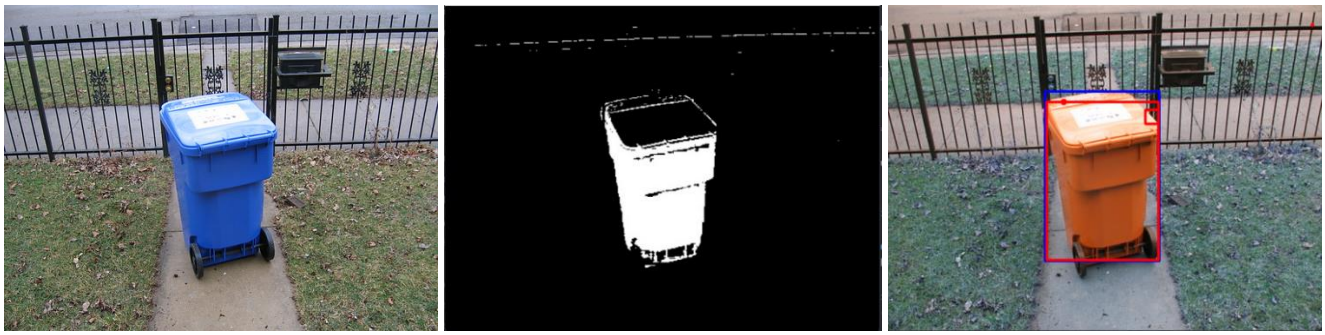

Figure 4: Original Image, segmented mask image and image with bounding box around the bin for validation image 0061.jpg

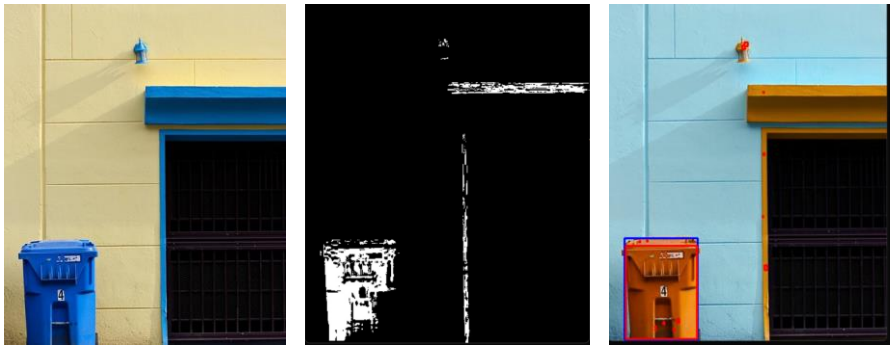

Figure 5: Original Image, segmented mask image and image with bounding box around the bin for validation image 0062.jpg

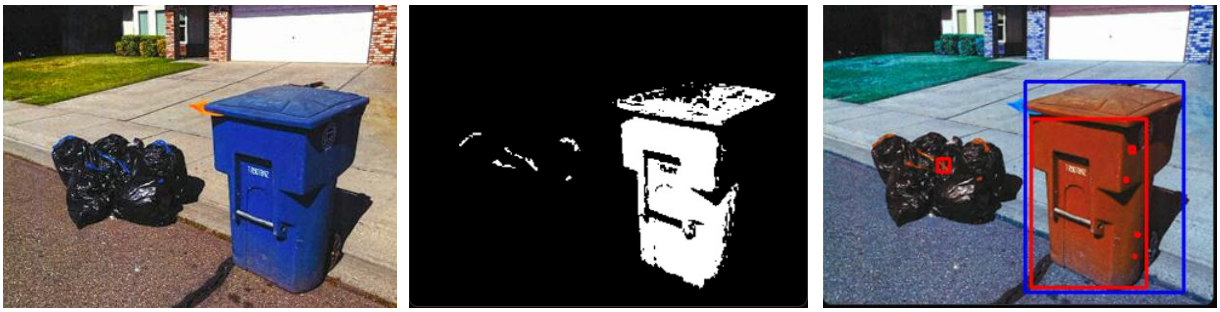

Figure 6: Original Image, segmented mask image and image with bounding box around the bin for validation image 0063.jpg

Figure 7: Original Image, segmented mask image and image with bounding box around the bin for validation image 0064.jpg

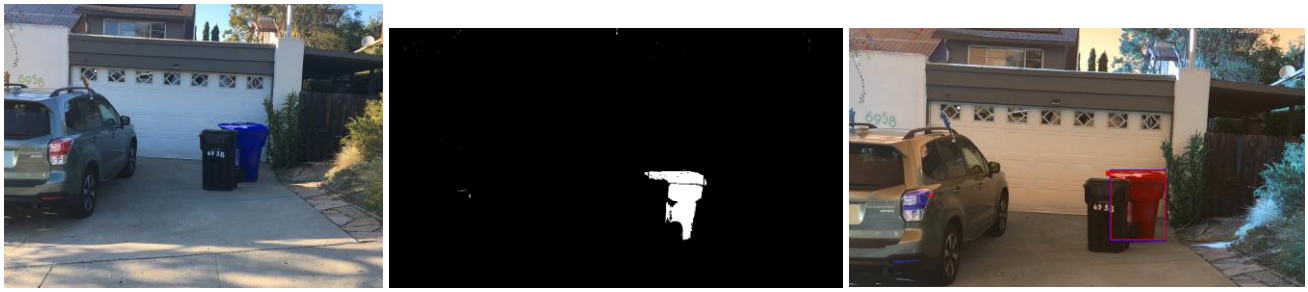

Figure 8: Original Image, segmented mask image and image with bounding box around the bin for validation image 0065.jpg

Figure 9: Original Image, segmented mask image and image with bounding box around the bin for validation image 0066.jpg

Figure 10: Original Image, segmented mask image and image with bounding box around the bin for validation image 0067.jpg

Figure 11: Original Image, segmented mask image and image with bounding box around the bin for validation image 0068.jpg

Figure 12: Original Image, segmented mask image and image with bounding box around the bin for validation image 0069.jpg

Figure 13: Original Image, segmented mask image and image with bounding box around the bin for validation image 0070.jpg