Pixel Classification and Object Detection

Sambaran Ghosal
Department of Electrical and Computer Engineering
UC San Diego
La Jolla, USA
sghosal@ucsd.edu

Nikolay Atanasov
Department of Electrical and Computer Engineering
UC San Diego
La Jolla, USA
natanasov@ucsd.edu

***Abstract*—The aim of this project was to implement a Gaussian Discriminant Model and Mixture of Gaussians algorithm on the tasks of pixel classification and using these classifiers to detect blue recycling bins in a given image. We try to tackle this problem in two color spaces : RGB space and the YUV space.**

***Index Terms*—Classification, Detection, Gaussian Discriminant Analysis, Mixture of Gaussians**

I. Introduction

Robots often have to deal with unknown environments. It has to recognize objects of interest to reach to, or objects that may provide an obstacle for their movement and plan out a path to avoid such objects. Unlike us humans who can process the visual input of the world and differentiate between objects at go, robots do not have that kind of capability yet(perhaps in the future). Instead, the way robots differentiate between different objects is via training. We have to train the robot on several images using Machine Learning methods. We have a training dataset where we provide real life images to the robot. We have a label for an object of interest(for ex : a bounding box within which our object of interest lies) and we train our model using this. After the model has been trained, it can be deployed on the real robot and it can obtain objects of interest in the environment it is present in.

The aim of this project is to train a color classification model and then use the trained model to detect a blue recycling bin in the given image. Our approach is to first obtain a segmented image from the original image of objects that are of the interested color i.e. blue. Using the segmented image, we deploy open-cv techniques such as findContours and use shape statistics of recycling bins to obtain the region that is closest in resemblance to a recycling bin such as ratio of height to width of the contours bounding box, area of the bounding box compared to the total area of the image. Our approach is summarised in Figure 1.

II. Problem Formulation

A. Pixel classification

Our main problem is given a pixel, we have to identify whether it is red, green or blue in color. A pixel can be represented as a vector $x \in R^3$, and the corresponding color can be represented as a label $y \in \{1, 2, 3\}$ where 1 corresponds to red, 2 corresponds to green and 3 corresponds to blue. We train our model on the training dataset, and we evaluate the performance of our trained model in a validation set.

Fig. 1: The aim of this project is to find out blue recycling bins in a given image

B. Bin Detection

After we have a model for the pixel classification task, we have to use it to detect a blue recycling bin in a given image and draw a bounding box around it.

- Our pixel classification model will produce a binary mask of the original image where white represents regions that have a blue object.
- The binary masked image is taken as input and open-cv contour methods and shape statistics are applied to obtain regions that closely resemble a recycling bin.
- The output will be four numbers x_1, y_1, x_2, y_2 where (x_1, y_1) represents the top left coordinate of the bounding box, and (x_2, y_2) represents the bottom right coordinate of the bounding box.

III. Technical Approach

A. Pixel Classification

In this project, we try to implement a Gaussian Discriminant Analysis model for pixel classification. A gaussian discriminant model is given by the following formula

$$p(x) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} e^{\frac{-1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)} \quad (1)$$

where d is the dimension of x, Σ is the covariance matrix and μ is the expectation of the distribution. Our model for pixel classification is a generative model given as

$$p(x, y) = p(x|y) * p(y) \quad (2)$$

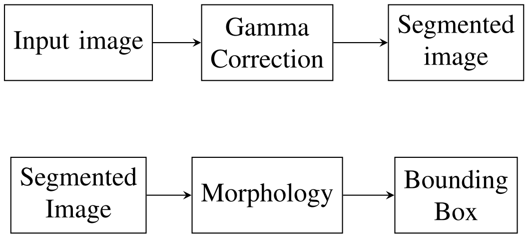

Fig. 2: Proposed method for bin detection

where $p(x|y)$ is the class conditional probability of a given pixel modeled as a Gaussian PDF as given by Eq(1) and $p(y)$ is the prior class probability which is simply the probability of a given class. Let us denote $p(y = k) = \theta_k$ where $\sum_{k=1}^{3} \theta_k = 1$.

Given a training dataset of pixels and corresponding labels as $D = (\{x_i, y_i\})|_{i=1}^{N}$ of iid samples, we apply **Maximum Likelihood Estimation(MLE)** to estimate the parameters of the gaussian model corresponding to each class $p(x, y_k)$ where $y_k = \{1, 2, 3\}$ where 1 indicates red, 2 indicates green and 3 indicates blue pixels. The MLE approach is to find optimal parameters θ_k, μ_k and Σ_k for $k \in \{1, 2, 3\}$ that maximises the data likelihood function

$$L(\mu_k, \Sigma_k; D) = \sum log\, p(x|y_k) * p(y_k) \tag{3}$$

From calculus, we can find the parameters that maximises the above function and we get the following result

$$\theta_k = \frac{\sum_{j=1}^{N} 1\{y_j = k\}}{N} \tag{4}$$

$$\mu_k^{MLE} = \frac{\sum_{j=1}^{N} x * 1\{y_j = k\}}{\sum_{j=1}^{N} 1\{y_j = k\}} \tag{5}$$

$$\Sigma_k^{MLE} = \frac{\sum_{j=1}^{N} (x - \mu_k^{MLE})(x - \mu_k^{MLE})^T * 1\{y_j = k\}}{\sum_{j=1}^{N} 1\{y_j = k\}} \tag{6}$$

Now we have the optimal parameters μ_k^{MLE} and Σ_k^{MLE} for $k \in \{1, 2, 3\}$, we can use our models for predicting colors of new pixels. Given an unseen pixel x_*, the predicted label will given as

$$y_* = \arg\max_{k \in \{1,2,3\}} p(x_*|y = k) * \theta_k \tag{7}$$

We can get the accuracy of our model using the validation set as follows

$$accuracy = \frac{\sum_{v=1}^{V} 1\{y_*^{(v)} = y^{(v)}\}}{V} \tag{8}$$

where V is the number of samples in the validation set, $y_*^{(v)}$ is the predicted class using our model and $y^{(v)}$ is the true label of the pixel provided in the validation set.

B. Bin Detection

For the bin detection, we first create a binary mask of the image provided using our pixel classifier model. Unlike in pixel classification, we are not given a dataset that we can directly use to train our classifier model further. Instead, we are given some images that may contain blue recycle bins, along with other objects. To gather a dataset for further training our classification model, we make use of the **roipoly** library to select regions of interest. For this project, we selected 2 classes : positive class containing pixels belonging to blue recycling bin and negative class containing pixels that do not belong to the blue recycling bin. The negative class contains pixels that are blue but not bins, and other objects that are of different colors. The pixels are collected in two color spaces : **RGB** and **YUV**. The **YUV** space captures the brightness of pixels, the blueness and redness of the pixels. For both RGB and YUV space, we try to fit two models as follows :-

- A single Gaussian Discriminant Model. This is the same that we did for the pixel classification task except that now we have only two classes : the positive class and the negative class. We obtain the parameters of these two classes using MLE as discussed before.
- A Mixture of Gaussian Model. Instead of trying to fit only one gaussian model for the entire positive and negative class, we propose that a mixture model that can learn different clusters for the same class depending on the variants of the colors. For ex: a single gaussian model may classify a light blue color into the positive example but may fail to classify a darker variant of the blue color into the same class. Whereas a mixture model can learn to classify both the lighter and darker variant of the blue color into the same class cluster because of it having learnt multiple clusters for the blue class.

1) Mixture of Gaussians: As discussed above, a mixture of gaussians model has the capability to learn multiple clusters for the same class of data. This helps in classifying different variants of a given class of data into the same category. For training mixture models, we normalise the data, and hence we divide our pixel values by 255 before training. We considered a model with 3 mixtures for each class. The class conditional probability of a given pixel x can then be given as

$$p(x|y_k) = \sum_{c=1}^{3} p(x|y_k, z = c) * p(z = c) \tag{9}$$

where $y_k \in \{1, 0\}$ where 1 represents positive class and 0 represents negative class, z is a latent variable which denotes to which cluster a given data belongs to. For our work, $z \in \{1, 2, 3\}$. $p(z = c)$ is the probability that cluster c is selected. Let us denote $p(z = c) = \pi_c$ for $c \in \{1, 2, 3\}$. Finally, we again have the assumption that the class conditional probability for a pixel is a gaussian pdf and hence, Eq(9) can be rewritten as

$$p(x|y_k) = \sum_{c=1}^{3} \pi_c N(x; \mu_{kc}, \Sigma_{kc}) \tag{10}$$

where $N(x; \mu, \Sigma)$ represents the gaussian pdf with expectation μ and covariance Σ, μ_{kc} represents the mean of cluster c of class k and Σ_{kc} denotes the covariance matrix of cluster c of class k. Thus we will have a total of $k \times c$ parameters to learn.

We implement an **Expectation Maximisation(EM)** algorithm to train the parameters for our gaussian mixture model. The steps for EM are as follows :-

- Step 1 (Initialise the parameters) : Initialise the parameters π_{kc}, μ_{kc} and Σ_{kc} for $k \in \{1, 0\}$ and $c \in \{1, 2, 3\}$. Denote the set of parameters as $\Psi^o = \{\pi^o_{kc},\ \mu^o_{kc},\ \Sigma^o_{kc}\}$
- Step 2 (Expectation step) : For each class, compute the expectation of the latent variables given the observed data $E_{Z_{kc}|X_i;\Psi^o}[Z_{kc}]$

$$\gamma^o_k = \frac{\pi^o_{kc}\, N(x_i; \mu^o_{kc}, \Sigma^o_{kc})}{\sum_{c=1}^{3} \pi^o_{kc}\, N(x_i; \mu^o_{kc}, \Sigma^o_{kc})} \tag{11}$$

  where superscript o stands for current iteration values of parameters.
- Step 3 (Maximisation Step) : Obtain the new parameter values $\Psi^{o+1} = \{\pi^{o+1}_{kc},\ \mu^{o+1}_{kc},\ \Sigma^{o+1}_{kc}\}$ as

$$\pi^{o+1}_{kc} = \frac{\sum_{i=1}^{N} 1\{y_i = k\}\gamma^o_{kc}}{\sum_{i=1}^{N} 1\{y_i = k\}} \tag{12}$$

$$\mu^{o+1}_{kc} = \frac{\sum_{i=1}^{N} 1\{y_i = k\}\gamma^o_{kc} x_i}{\sum_{i=1}^{N} 1\{y_i = k\}\gamma^o_{kc}} \tag{13}$$

$$\Sigma^{o+1}_{kc} = \frac{\sum_{i=1}^{N} 1\{y_i = k\}\gamma^o_{kc}(x_i - \mu^{o+1}_{kc})(x_i - \mu^{o+1}_{kc})^T}{\sum_{i=1}^{N} 1\{y_i = k\}\gamma^o_{kc}} \tag{14}$$

- Step 4 (Check for stopping condition) : If

$$||\mu^{o+1} - \mu^o|| < 0.001 \tag{15}$$

  stop else repeat steps 1-3 using new parameters Ψ^{o+1}

It becomes difficult for EM to converge if the data size is not sufficient and the covariance matrix is considered in the most general form. To help EM achieve convergence, the covariance matrix can be assumed as a diagonal matrix i.e. $\Sigma_{kc} = \mathbf{diag}([\sigma^2_{kci}]^T)$ and then Eq(14) can be modified as

$$\Sigma^{o+1}_{kc} = \frac{\sum_{i=1}^{N} 1\{y_i = k\}\gamma^o_{kc}\, \mathbf{diag}((x_i - \mu^{o+1}_{kc})^2)}{\sum_{i=1}^{N} 1\{y_i = k\}\gamma^o_{kc}} \tag{16}$$

Now we have the optimal parameters of color classification for both Single Gaussian Discriminant Analysis as well as Mixture of Gaussians. We produce a segmented binary mask for the original image and proceed to the next step.

2) Getting the bounding box: After obtaining the segmented image, some pre-processing is performed on the masked image. This includes morphology operations such as erode, dilation to reduce noise in mask and to fill gaps within the mask. Smoothening operations such as Gaussian Blur and Thresholding are also performed. These functions are all available in open-cv. Next, open-cv findContours method is used to draw contours along the boundaries of the mask. The findContours methods will draw boundaries to separate regions of white pixels from black pixels. For each contour that open-cv draws, a bounding box of the contour is obtained using open-cv boundingRect method. From data, the ideal shape of a recycling bin is such that

- The ratio of height to the width of the bin $\frac{h}{w}$ is between 0.9 - 1.85
- Area occupied by the bin is greater than 0.8% of the total area of the image.

The above conditions are obtained using various experiments and selecting values that work best with all images provided in the validation set. The output is defined as a list of bounding box coordinates of possible blue recycling bin regions in the image.

For the validation set, we are provided the coordinates of the bounding box in the image as a reference. An accuracy measure in this case is defined as the **Intersection over Union(IOU)** of the reference bounding box and predicted bounding box. For a given image, the intersection over union is computed between predicted bounding boxes and the actual bounding box. If IOU is more that 50%, an accuracy of 1 is given. The overall accuracy for the given image is then defined as the total accuracy over the total number of bounding boxes.

Using these conditions, the contours are filtered out and the bounding box coordinates of the contours that satisfy the above conditions is returned.

IV. RESULTS

We trained two models : A single gaussian model and a Mixture of Gaussians Model on two color spaces, the RGB space and the YUV space. The parameters obtained are reported in section A

A. RGB Color Space

1) Single Gaussian Discriminant Model: The single gaussian discriminant model assumes a single cluster for each image class. The input image is first fed to the segment_image method and then the output of the segment_mask image is fed to the get_bounding_box method to get the bounding boxes around blue recycling bins. Results are shown in Fig(5).

2) Mixture of Gaussians: The mixture of gaussian model assumes several clusters for each class. This way it can classify multiple variants of the same color into the same class in contrary to single Gaussian Discriminant model which may assign different classes to different variants of the same color. Results obtained using this model are shown in Fig(6)

From the results show in Fig(5) and Fig(6), it can be observed that the mask obtained using Mixture of Models is more accurate than the one obtained from Single Gaussian Model, as was proposed in the formulation because of its ability to learn different clusters for same class. It is also observed that the Single GDA model fails to get a bounding box around the blue recycling bin in the 3rd row image because it was not able to produce a proper mask for the blue bin as it is obstructed by the black bin, and in the first row image, it is able to obtain only one bounding box for the left blue bin but fails to get it for the right blue bin. In contrary, using a Mixture model, we can see that in the first row image, although it is not able to obtain two separate bounding boxes for the two bins, it generates a bounding box that encloses both the bins. In the third row, the mixture model is able to produce a more accurate mask than obtained using single gaussian model and this helps our shape statistics method to obtain a bounding box for the blue bin even though it is obstructed by the black bin. It is observed that without any gamma correction to the original image, a mixture model is able to perform better than a single model. The comparison between using gamma corrections is done in later section.

B. YUV color space

1) Single Gaussian Discriminant Model: Using the YUV space and no gamma correction, it is observed from Fig(7) that even a single gaussian discriminant model is able to give very accurate results compared to the RGB space. Particulary, it can be seen that YUV space gives accurate results for both the 1st and 3rd row images on which RGB space model without gamma correction was not fully accurate. Hence, we can conclude that YUV space is more beneficial in segmentation than the RGB space because one of the channel values of YUV space is just the intensity of the pixel and hence different intensities of the same color are often in a similar range of Y values.

2) Mixture of Gaussians: The results obtained using a Mixture of Gaussian Model is shown in Fig(7). It is observed that the mask is a little bit noisy compared to that obtained using a single gaussian model. This is attributed to the fact that in this space too, the mixture model learns multiple clusters for different variants. Since in YUV space, pixels are converted to grayscale, redness and blueness, even if two pixels are slightly different in RGB space, it is possible that they are almost the same in YUV space because of similar intensity, redness and blueness and hence they get clustered to the same class. For example, in the third row image, it is observed that the garage door and car have some shades of blue. In RGB space it looks quite different than the recycling bin blue but in YUV space, these are quite indistinguishable. This is shown in Fig(3) This does not affect our bin detection since the shape statistics still work out perfectly with the given parameters.

Fig. 3: YUV space similarity between different shades of blue

C. Effect of Gamma Correction

Gamma correction is a method of image processing that controls the overall brightness of an image. Images that are not corrected properly may look too dark or bleached. Varying the amount of gamma correction also influences the ratio of red to green to blue in a given image. We compare the results of using gamma correction to not using gamma correction for the rgb color space since we had two failure cases. It is observed from Fig(9) that gamma correction produces a much less noisy mask and helps in accurate prediction of the bounding box around the blue recycling bin. It is also able to get two bounding boxes in the fourth row image whereas, as seen before, without application of gamma correction, we were able to get only one bounding box for the left bin using single gaussian model, and a single bounding box enclosing both the bins using a Mixture of Gaussians model. Hence, we conclude that for the RGB space, using gamma correction is beneficial to us. Although the amount of gamma correction is an important factor. A large value of the gamma correction factor will increase brightness too much and the classifier will fail to differentiate pixels of different colors. From several experiments conducted on the validation set, it was concluded that a gamma correction factor of 2 yielded the best results.

D. Effect of Morphology

Morphology is defined as a series of transformation operations on a binary masked image that depend on the shape of the image. The input of morphology operations is the masked image and a structuring element that defines the nature of the transformation. Open-cv has many morphology operations such as dilation, erosion, opening, closing, gradient. For this project we are using two morphological operations namely erosion and dilation. Erosion is used to remove the borders of the foreground object which is represented as white pixels in the image. Given a kernel, erosion operation will convert all pixels under a given kernel to 0 if any of the pixels inside the kernel is 0. Hence this has the effect of removing borders from the foreground object. Dilation is the opposite of erosion. Given a kernel, if any element inside ther kernel is 1, it will transform all pixels under the kernel to 1. This has the effect of removing small openings inside the foreground object. These operations are beneficial when we have other blue colored objects that are not actually a recycling bin so that we can remove these kind of noise in the masked image. It is also beneficial in cases where two recycling bins are located very close to each other such that their masks are concurrent as seen in Fig(5) and Fig(6) first row image. We observe that in both the cases, we are not able to correctly draw a bounding box over both the recycling bins. In this case, doing an erosion operation first will remove the concurrency between the masks and create a mask that has two separate entities for both the bins. Then we can perform a dilation operation to close the gaps inside the bins. Fig(4) shows the comparison of using morphology operations against not using them. We observe that in the case where there was no morphology operation, we got a single bounding box enclosing both the blue recycling bins, but using a combination of erosion + dilation, we were able to get two separate bounding boxes for both the blue bins. Although the kernel size used during erosion and dilation should be carefully chosen because if the kernel size is too large, erosion can remove majority of the mask and this will create a bounding box for the recycling bin that does not satisfy the conditions and hence is not selected or create a bounding box that is small and the IOU with the true bounding box will be low.

E. Comparison of accuracy with different methods

As we have discussed, a metric of accuracy for this problem is the **Intersection Over Union(IOU)** between the true bounding box and the predicted bounding box. We compare the IOU values obtained using different methods mentioned in sections IV-A1, IV-A2, IV-B1 and IV-B2. All methods were compared keeping the gamma correction factor as 1.5, and all using morphology operations discussed in section IV-D. We observe the following from the table :

- The performance is better in the YUV space where we can see a higher IOU for difficult images like 0065 and 0067.
- In the RGB space, the mixture of gaussians model is able to perform slightly better than a single gaussian model. Although the performance is almost the same in easier images, a mixture model comes up on top in difficult images like 0065 and 0067.
- Both single gaussian model and mixture model have relatively the same performance in the YUV space. Although we get a slight better performance in the mixture model case with the specified gamma correction factor of 1.5, the time taken to train a mixture model and to make predictions os significantly larger than that of a single gaussian model and hence, for a gamma value of 2, it was observed that a single gaussian model in YUV space is more beneficial than a mixture model because of significantly less time to train and make predictions and very little difference in performance.

Fig. 4: No morphology vs morphology

TABLE I: Intersection over union for different methods

Image number	RGB		YUV	
	GDA	MOG	GDA	MOG
0061	61.10%	56.80%	60.64%	64.98%
0062	90.16%	89.52%	89.22%	95.31%
0063	49.23%	52.09%	47.2%	50%
0064	87.94%	87.37%	86.7%	90.95%
0065	0%	64.70%	74.16%	64.08%
0066	NA	NA	NA	NA
0067	81.90%	93.61%	91.68%	95.25%
0068	NA	NA	NA	NA
0069	NA	NA	NA	NA
0070	NA	NA	NA	NA

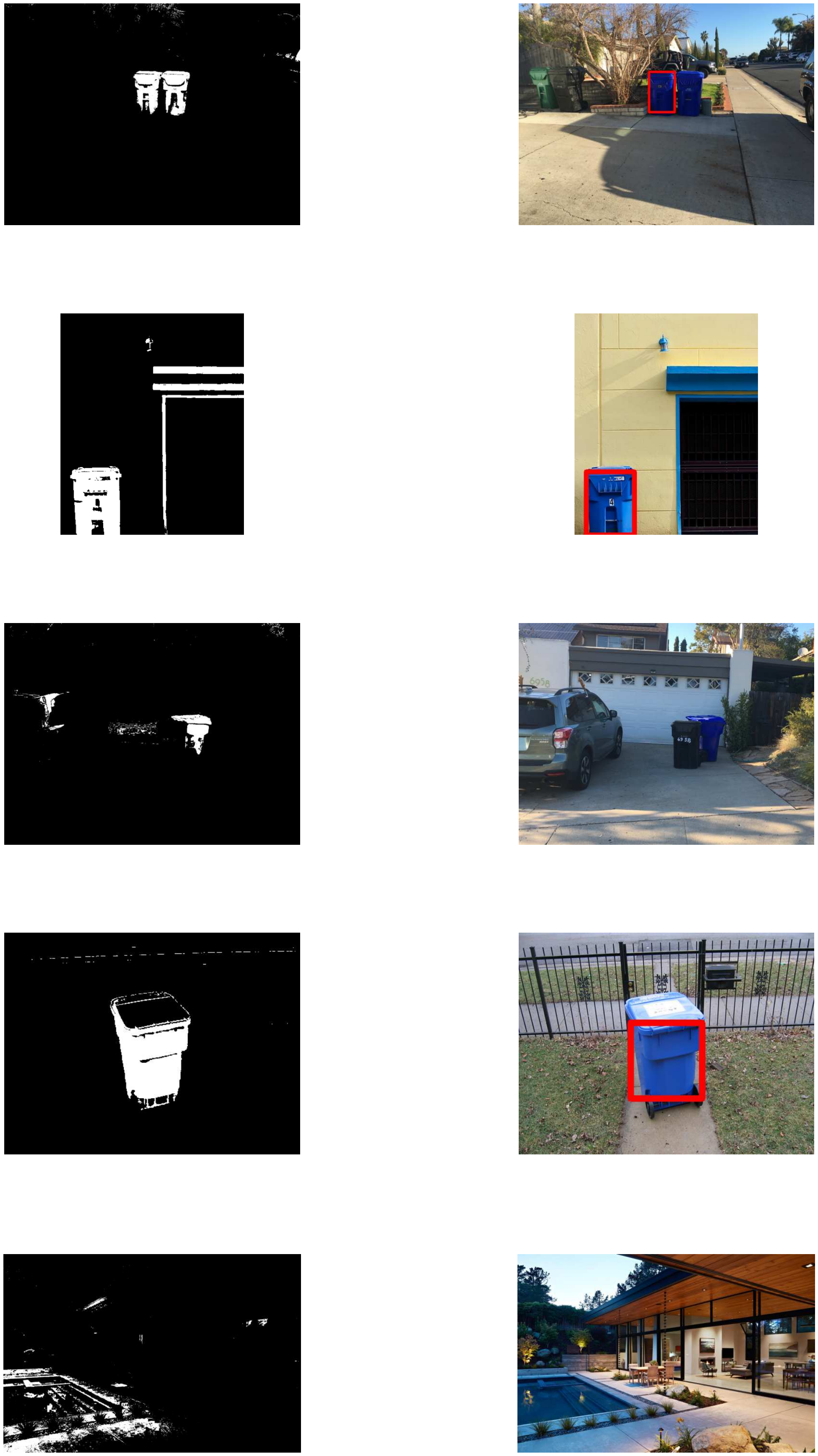

Fig. 5: RGB Single Gaussian Discriminant Model masks and bounding box

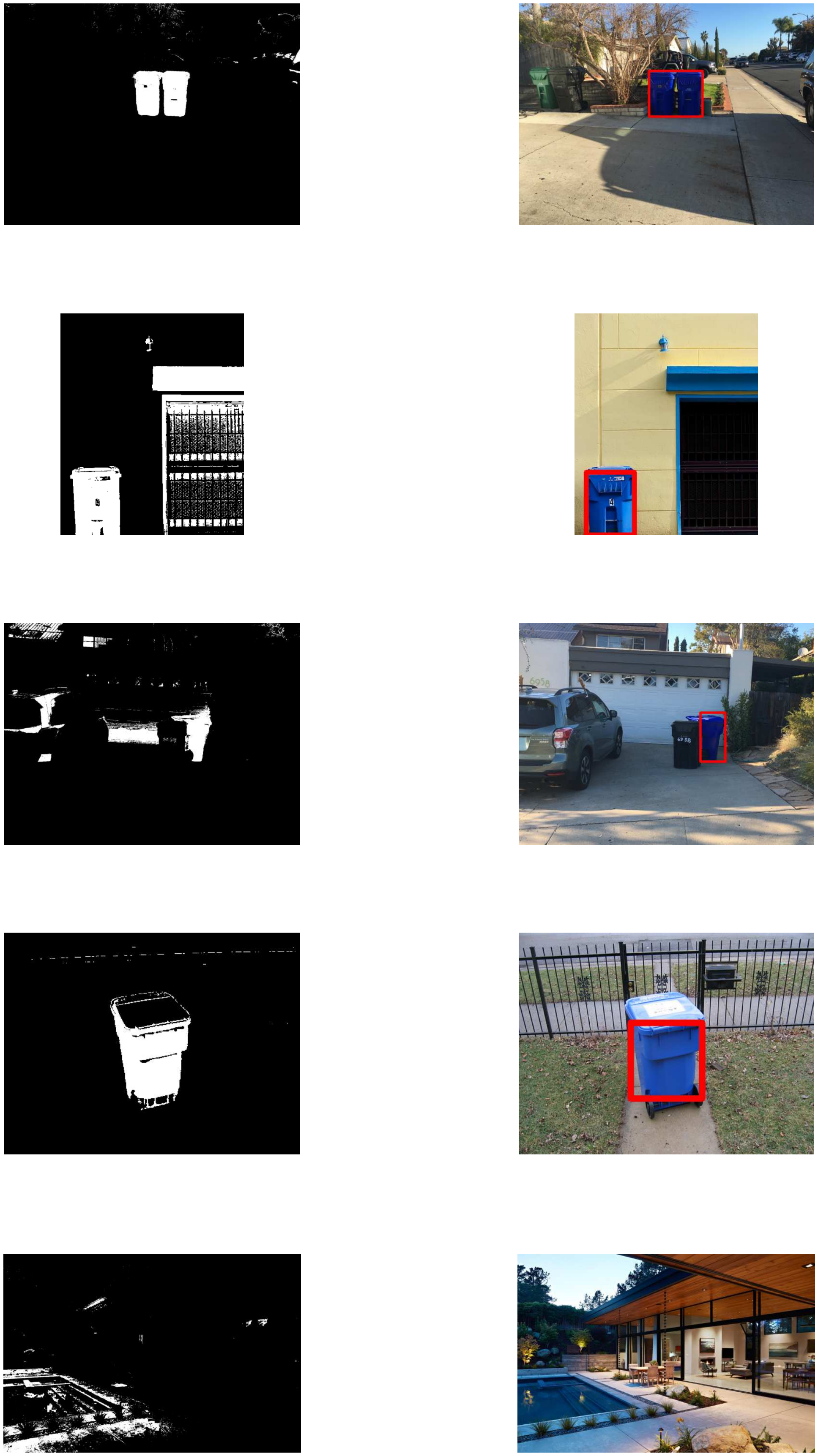

Fig. 6: RGB Mxiture of Gaussian Model masks and bounding box

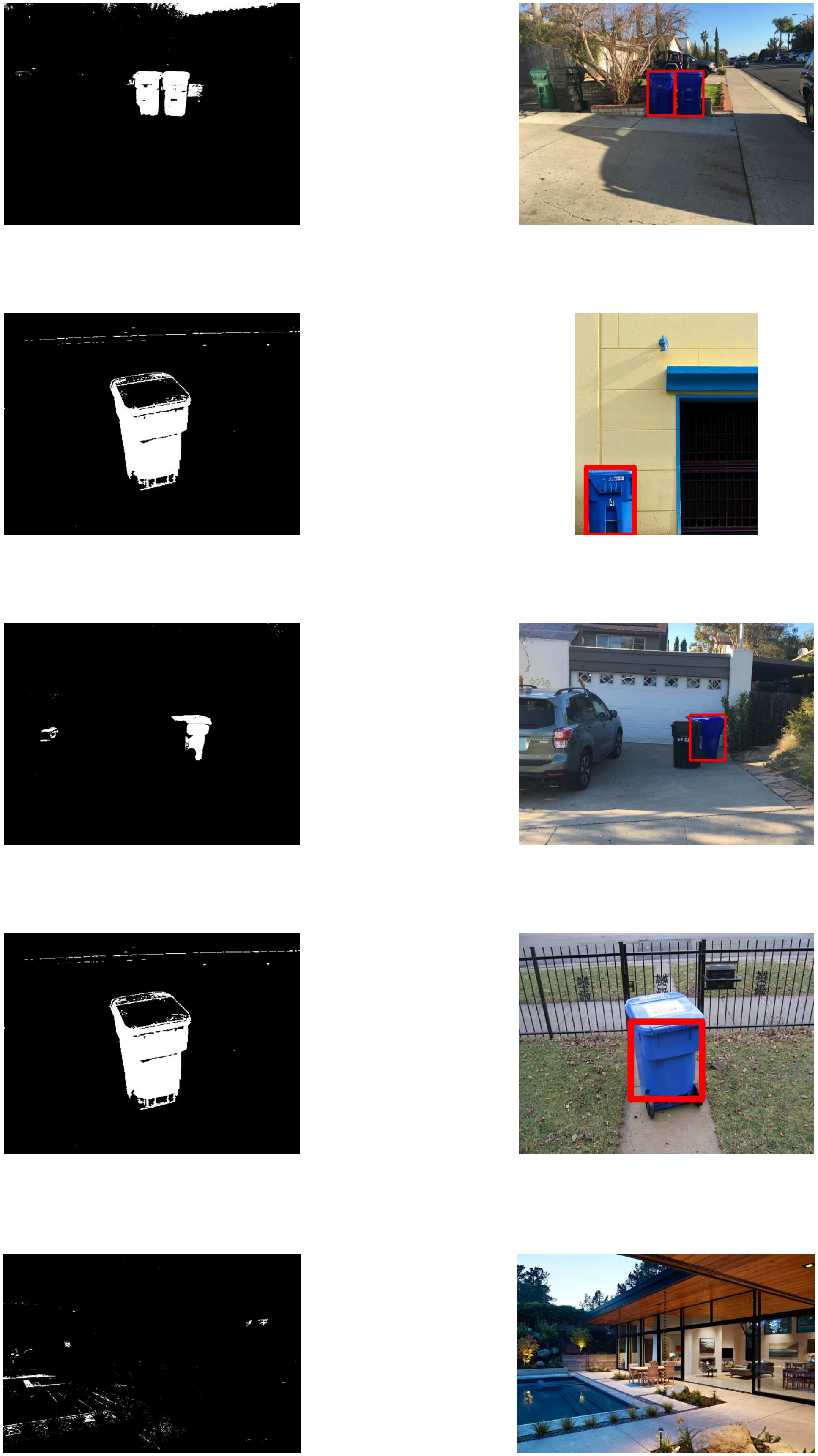

Fig. 7: YUV Single Gaussian Discriminant Model masks and bounding box

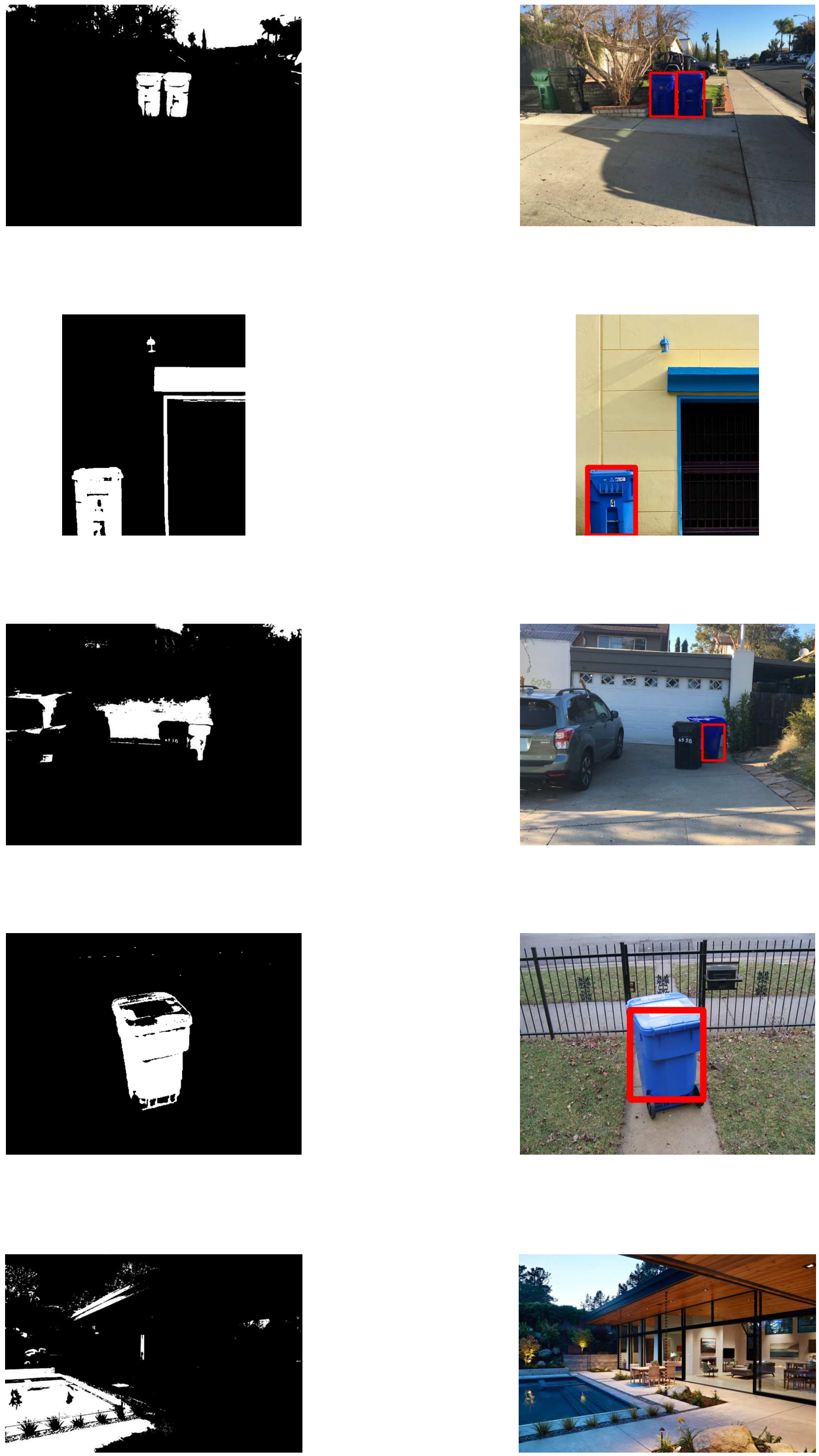

Fig. 8: YUV Mixture of Gaussian Model masks and bounding box

Fig. 9: Effect of Gamma correction

V. Conclusion and Future Work

In this project, we implemented a Gaussian Discriminant Model and a Mixture of Gaussians Model to segment a given image where our region of interest was blue recycling bins. After the segmentation, we get a masked image of all blue regions in the image. Using open-cv findContour, boundingRect methods and shape statistics, we filtered out the regions that actually are most probable to belong to the blue bin. We used image processing techniques such as gamma correction, morphology operations such as dilation and erosion in addition to smoothening operations like gaussian blur and thresholding. We solved the problem in the RGB and YUV color space and concluded that YUV space was easier to work with for the given problem because of its capability to handle intensities efficiently.

For future work, we can try to implement Bayesian Estimation methods instead of Maximum Likelihood Estimation. This would require a prior knowledge of parameters like the mean and covariance of the models. Use of other machine learning techniques such as R-CNN, Fast R-CNN, YOLO can be used to solve the given problem because of the capability of such methods to learn important features for objects of interest.

References

[1] "https://docs.opencv.org/3.4/d9/d61/tutorial_py_morphological_ops.html"
[2] "https://en.wikipedia.org/wiki/Gamma_correction"
[3] "https://github.com/jdoepfert/roipoly.py"
[4] Nikolay Atanasov, "https://natanaso.github.io/ece276a/"

Appendix

Training on RGB and YUV color space was done using two methods : Single Gaussian Discriminant Analysis amd Mixture of Gaussians Model. Parameters obtained for the single gaussian model are as follows :-

- **RGB** :

$$\mu_{pos} = \begin{bmatrix} 99.80 & 112.08 & 169.73 \end{bmatrix}$$

$$\mu_{neg} = \begin{bmatrix} 106.58 & 106.30 & 97.22 \end{bmatrix}$$

$$\Sigma_{pos} = \begin{bmatrix} 3808.03 & 3048.27 & 829.06 \\ 3048.27 & 2687.61 & 982.44 \\ 829.06 & 982.44 & 1252.71 \end{bmatrix}$$

$$\Sigma_{neg} = \begin{bmatrix} 4507.23 & 3843.56 & 3125.46 \\ 3843.56 & 3902.12 & 3530.84 \\ 3125.46 & 3530.84 & 4103.70 \end{bmatrix}$$

- **YUV** :

$$\mu_{pos} = \begin{bmatrix} 64.17 & 171.10 & 103.42 \end{bmatrix}$$

$$\mu_{neg} = \begin{bmatrix} 106.88 & 123.95 & 130.28 \end{bmatrix}$$

$$\Sigma_{pos} = \begin{bmatrix} 2008.48 & -286.83 & -68.70 \\ -286.83 & 492.68 & -254.70 \\ -68.69 & -254.70 & 275.96 \end{bmatrix}$$

$$\Sigma_{neg} = \begin{bmatrix} 4559.46 & -245.54 & 172.85 \\ -245.54 & 201.11 & -196.40 \\ 172.85 & -196.40 & 349.38 \end{bmatrix}$$

For the mixture of gaussian model, we trained the models by normalising the pixel values. Hence each pixel is in the range of 0-1. We assumed the covariance matrix of each cluster as a diagonal matrix. Hence we report the diagonal elements of each covariance matrix in the results below. Number of clusters was taken as 3. The parameters obtained in this case are as follows :-

- **RGB**

$$\mu_{pos} = \begin{bmatrix} 0.10 & 0.08 & 0.25 \\ 0.25 & 0.19 & 0.54 \\ 0.53 & 0.50 & 0.84 \end{bmatrix}$$

$$\mu_{neg} = \begin{bmatrix} 0.72 & 0.66 & 0.59 \\ 0.33 & 0.34 & 0.35 \\ 0.16 & 0.17 & 0.14 \end{bmatrix}$$

$$\Sigma_{pos} = \begin{bmatrix} 0.01 & 0.01 & 0.03 \\ 0.016 & 0.01 & 0.011 \\ 0.04 & 0.01 & 0.01 \end{bmatrix}$$

$$\Sigma_{neg} = \begin{bmatrix} 0.02 & 0.014 & 0.03 \\ 0.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.01 \end{bmatrix}$$

$$\pi_{pos} = \begin{bmatrix} 0.22 & 0.41 & 0.37 \end{bmatrix}$$

$$\pi_{neg} = \begin{bmatrix} 0.32 & 0.34 & 0.34 \end{bmatrix}$$

- **YUV**

$$\mu_{pos} = \begin{bmatrix} 0.06 & 0.04 & 0.03 \\ 2.16 & 0.79 & 0.40 \\ 2.15 & 0.79 & 0.39 \end{bmatrix}$$

$$\mu_{neg} = \begin{bmatrix} 0.29 & 0.02 & 0.03 \\ 11.06 & 0.48 & 0.52 \\ 11.06 & 0.48 & 0.52 \end{bmatrix}$$

$$\Sigma_{pos} = \begin{bmatrix} 0.01 & 0.34 & 0.18 \\ 4.02 & 0.01 & 0.011 \\ 4.01 & 0.01 & 0.01 \end{bmatrix}$$

$$\Sigma_{neg} = \begin{bmatrix} 0.01 & 0.2 & 0.2 \\ 110.22 & 0.01 & 0.01 \\ 110.22 & 0.01 & 0.01 \end{bmatrix}$$

$$\pi_{pos} = \begin{bmatrix} 0.03 & 0.48 & 0.49 \end{bmatrix}$$

$$\pi_{neg} = \begin{bmatrix} 0.03 & 0.48 & 0.49 \end{bmatrix}$$

where each row of Σ_{pos} and Σ_{neg} are the diagonal elements of the covariance matrix of the corresponding cluster.