Color Segmentation and Recycling Bin Detection: A Project Review

Hao-Yuan Tang
Department of Electrical Engineering
University of California, San Diego
h6tang@ucsd.edu

I. Introduction

Color classification is a predicting problem that involves using labeled data to find and determine characteristics of colors. In order to identify colors, which are represented as 3D vector in conventional RGB images, it's necessary to understand how these data, i.e., pixel intensities, distributed within each dimension and develop mathematical model to describe it. The approach employed in this project is a probabilistic method called Gaussian Discriminative Analysis (GDA) to classify single pixel into red, blue, or green color.

In the first part of this project, GDA model for single pixel color classification was built in the first part. The main purpose of this part is to validate GDA performance on classification in such application. Parameters in GDA model was calculated from data distribution in original color space, and the whole problem was taken as supervise learning.

In the second part of this project, similar idea was used to detect interested object – recycling bin, from 60 street images. Image processing methods and concepts related to task, such as contrast stretching, YUV space, dilation and erosion, were discussed in this project. After this, the main problem is a binary classification using GDA model, which is trained by images in new color space.

II. Problem Formulation

A. Color classification

The first topic in this project is to train a probabilistic color model from pixel data to distinguish among red, green, and blue pixels. Each example in the training or validation sets is a 28×28 images with a single RGB value at all of its pixels. The images are split according to the three labels: red, green, blue, and the task is to classify single pixel into one of the three color classes, i.e.

$$\boldsymbol{x} \in \mathbb{R}^3 \mapsto y \in \{R, G, B\} \tag{1}$$

The problem was defined as supervised learning, which means the data was labeled as one particular class and the goal is to match it to predicted one. A dataset consist of RGB intensities and true label was denoted as $\mathcal{D} = \{(\boldsymbol{x_i}, y_i)\}_i$ of pixel values $\boldsymbol{x_i} \in \mathbb{R}^3$, labeled with colors y_i , $i = \{1,2,3\}$, represented intensities in RGB channel, respectively. And the task is to build a probabilistic model, such as generative model in this project, and optimize the parameters of model $p(y, \boldsymbol{x})$ using training data.

To use generative probabilistic model $p(y, \boldsymbol{x})$, one necessary task is to understand the data distribution of each class, e.g., mean value, covariance, and class prior, which can all be learned within training set. Then, we can use generative model to assign data to the class with highest class-joint probability, as following equation:

$$y^* \in \arg\max_y p(y, \boldsymbol{x}|w) \tag{2}$$

where * notation for optimal and w represents parameters in our probabilistic model. More details would be discussed in method section.

B. Bin detection

The second part in this project is to find interested objects in images. Different from the first part, the training set contained not just single pixel intensities, but the whole images in various scenes. Most of images contain one or more recycling bin and our task is to find all of them with probabilistic models.

The basic idea is to "decompose" images into lots of pixels, then classify each pixel into positive or negative cases. This part is the same as in color classification section, but only two classes in this model.

$$\boldsymbol{x} \in \mathbb{R}^3 \mapsto y \in \{Postive, Negative\} \tag{2}$$

III. Method

A. Gaussian Discriminant Analysis

GDA is a specific generative classification method in which the class conditional probability distributions are assumed to be gaussian distribution: $(\boldsymbol{X}|Y = k) \sim N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ is mean value and covariance of all data belong to class k. The conditional probability of a label y_i given parameter θ is modeled by categorical distribution as the following equation:

$$p(\boldsymbol{y_i}|\theta) = \prod_{k=1}^{K} \theta_k^{\mathbb{1}\{\boldsymbol{y_i=k}\}} \tag{3}$$

and the classification problem in GDA can be described as:

$$p(\boldsymbol{y}, X|w, \boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{\theta})p(X|\boldsymbol{y}, \boldsymbol{w}) = \prod_{i=1}^{n} p(y_i|\boldsymbol{\theta})\, p(\boldsymbol{x_i}|y_i, \boldsymbol{w}) \tag{4}$$

$$p(\boldsymbol{x_i}|y_i = k, \boldsymbol{w}) := \phi(\boldsymbol{x_i}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{5}$$

where equation (5) shows the assumption on gaussian class-conditional distribution in GDA and class-joint probability $p(\boldsymbol{y}, X|w, \boldsymbol{\theta})$ can be used to generate new examples $\boldsymbol{x}$ with label y be sampling this distribution in generative model.

We can estimate the parameters of the conditional distributions with maximum likelihood estimation(MLE):

$$\theta_k^{MLE} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{y_i = k\} \tag{6}$$

$$\mu_k^{MLE} = \frac{\sum_{i=1}^{n} \boldsymbol{x_i}\mathbb{1}\{y_i=k\}}{\sum_{i=1}^{n} \mathbb{1}\{y_i=k\}} \tag{7}$$

$$\Sigma_k^{MLE} = \frac{\sum_{i=1}^{n}(x_i-\mu_k^{MLE})(x_i-\mu_k^{MLE})^T\mathbb{1}\{y_i=k\}}{\sum_{i=1}^{n} \mathbb{1}\{y_i=k\}} \tag{8}$$

Once we form the conditional distributions with parameters by MLE, we use Bayes' Rule to build the GDA model:

$$y^* = \arg\max_y p(y, \boldsymbol{X}|w, \boldsymbol{\theta}) \tag{9}$$

$$= \arg\max_y \log P_{X|Y}(\boldsymbol{x}|y) + \log P(k) \tag{10}$$

$$= \arg\min_k \log((2\pi)^d|\Sigma_k|) + (\boldsymbol{x} - \mu_k)^T \Sigma_k^{-1}(\boldsymbol{x} - \mu_k) - 2\log P(k) \tag{11}$$

B. Color space and Image processing

The robustness of probabilistic models depends strongly on data distribution. For a simple case, a 'red-like' pixel should have stronger intensity in R-channel compared to the other two. As expected, if we look at mean values of each RGB color, it's obvious that the corresponding channel has highest value. Covariance was a measure of the joint probability of pixel values in two channels.

Parameters in GDA, which were calculated in training process, would influence the performance on prediction. As a result, different color space, such as HSV or YUV, can produce different data distribution in feature vector $\boldsymbol{x}$. Unlike RGB space, HSV/YUV color spaces separate the intensity (brightness/darkness) from the color hue. This allows classifier to easily recognize the same color with different brightness, e.g., bright blue or dark blue. The blue colors, for example, are clustered closer together in the 3D YUV color space compared to 3D RGB space.

In this project, images were transformed into YUV space since our target "blue" color can be clearly differentiated in eyes in this space.

Also, in training set, there are many images consist of "shaded" recycling bin, which made detection in such area more difficult from background, so contrast stretching was used to enlarge difference between light variance and the final results were better after processing. The contrast stretching can be defined as following:

$$P_{out} = \left(\frac{P_{in} - P_{min}}{P_{max} - P_{min}}\right) \times 255 \tag{12}$$

Fig. 1. Image processing and color space comparison: (a) Original RGB space, (b) YUV space transformed by (a), (c) contrast stretching on (a), and (d) YUV space transformed by (c).

Since there was no training data as pixels intensities $\boldsymbol{x} \in \mathbb{R}^3$, manually labeled was necessary. Training images were first processed as the same flow, and ranges of interest were collected.

C. Python Implementation Flow

For pixel classification, since data with true class was given, the primary task is to build one classifier for supervise learning. In gaussianClassifier.py, mean, covariance and class prior for each RGB channel were calculated and packaged into parameters.pkl file. Then it was loaded to pixel_classifier.py to calculate log-likelihood $p(\boldsymbol{y}, X|w, \boldsymbol{\theta})$ of validation data and assigned its predicted color. The training/validation data $\boldsymbol{x} \in \mathbb{R}^3$, which was one color single pixel RGB intensity, was within RGB space from import to prediction.

For recycling bin detection, training/validation images were first enhanced by contrast stretching and transformed into YUV space. GDA was used again in this part, but the task became binary classification: blue or non-blue. For data labelling, label_test.py was used to process training images and collect hand-label data by applying roipoly.py modulus. And then, gaussianClassifier.py calculate parameters and packaged them into parameters_bin_detection.pkl file. And the main model was in bin_detector.py with the same procedure as in pixel classification.

The segment_image method returns a binary image mask generated by GDA with YUV image input. And the mask was used to find ROI in get_bounding_boxes method by applying dilation and erosion with cv2.dilate and cv2.erode function. After dilation, scattered positive pixels would connect to each other and fill holes within the recycling bin, and erosion can clean most of positive pixels outside the main area.

After labelling connected pixels, rectangular ROIs are extracted for each of the region. cv2.findcontours was used to find the contours in these areas, and cv2.RETR_EXTERNAL argument is used to approximate external contour in recycling bin. To choose the "best" bounding box, aspect ratio of bounding boxes was examined and qualified boxes were returned in get_bounding_boxes method.

since data in the same color was very clustered together and GDA can easily separate them from the other classes.

Unlike pixel classification, recycling bin detecti

The first step of implementation to find

IV. Results

This section showcases several test cases that the complete program was put through and the success and failures are discussed. For all validation images, the segmentation mask and images with bounding box are displayed. For images where recycling bin was not found, all the predicted bounding boxes are displayed to see whether the result was rejected because of its morphology or model performance.

A. Pixel classification

The validation set consists of small numbers of single pixel color. The performance of GDA model has 99.3% overall accuracy on the classification task. By properties of GDA, class-conditional distribution can represent clusters of each class very well, and only small portion of outliers were misclassified, such as colors shown in fig. 2.

Fig. 2. Failed prediction in pixel classification

The reason for such failure is not difficult to guess: when the data locates near decision boundary, i.e., color is "close" to mixtures two or three of RGB colors, such as dark purple which consists of red and blue components. One advantage of probabilistic models is to use probability distribution of training data, which represents data from different classes, and apply probability rules, such as Bayes rules, to handle classification problem. In this case, most of validation set can be correctly predicted because training data provided very clustered distribution within each class and be learned very well in GDA model.

But one disadvantage is also there: model performance depends heavily on data distribution, which means well-predicted data was often with "similar" feature vector $\boldsymbol{x} \in \mathbb{R}^3$ as class-conditional distribution. Since that, outliers, which are far from all each class distribution, would be predicted with higher error rate compared to those close to class clusters.

B. Bin detection

As mention in pixel classification, the performance of probabilistic model depends on data distribution. In other words, "bad quality" labelling would decrease robustness of model dramatically. As a result, the balance between positive and negative training data must be considered. In this project, the ratio of positive to negative training data is 0.3. Also, since there were only two classes in this GDA model, the ROIs of each class were carefully considered. For example, ROIs in negative class could easily consist of wide range of colors, i.e., all "not interested blue" colors, so ROIs in class were selected in monotonic color range, e.g. from coral orange to orange or from black to dark green. The disadvantage of only two classes in GDA model was easily expected: coral orange, rose red, dark green, blue, purple, and all the other "not interested blue" colors would be considered as the same class-conditional distribution, which might not be able to well-represented their characteristics of true color.

Intuitively, how probabilistic model is the similar to how we separate particular objects from others: the more distinctive color such objects have, the more easier to extract them from others. Therefore, during labelling procedure, better color space provided a clearer way to extract interested and uninterested regions. In this experiment, YUV images were optimal with better performance than HSV or RGB images since the "blue" color of our interested objects was presented by "bright green" color in YUV space and its quite different from other objects, such as bushes/trees/grass (green → rose red), sky (sky blue → yellow green), sunshine (white → coral orange), and shadow. YUV color spaces separate the intensity (brightness/darkness) from the color hue, so it's easier to observe difference between normal lighted, shaded and bright recycling bin than RGB space.

Moreover, HSV space in this experiment didn't perform very well than in YUV space. This result was probably because of how HSV represent color. One of the coordinates in HSV space is angle and the colors might be split into two clusters in the 3D space because of angle wrap-around, which requires additional computation to capture. Also, one observation is that the difference between light variance would give more influence on model performance than in YUV space.

Fig. 3. shows an image in different color spaces, we can simply observe the difference between YUV and HSV space. For example, colors of the vendor (where red arrows pointed to) can be clearly separated in YUV space, but it's difficult on HSV space, which is both "bright green" color for recycling bin and the vendor. The same phenomena appeared in lots of training images, as a result, YUV space was preferred in this project and it also gave the best result out of these spaces.

Fig. 3. Different color spaces: (a) RGB, (b) YUV, and (c) HSV.

Another issue researcher in this project faced is light variance, especially darkness like shaded area or gray color in the object. Even though YUV space can separate brightness, it's still not easy to get bounding boxes with correct shapes if the objects located within shadow. To fix this problem, contrast stretching, as Fig. 1 shown, was applied before color space transformation and the overall accuracy was improved. For example, there are gray color on the cover of recycling boxes in 0063.jpg in validation set and this caused generated bounding box with unwanted aspect ratio and therefore cause incorrect prediction. But this problem was fixed after contrast histogram with the same labelling procedure and algorithm, and improvements were observed in other images with the same issue since the contrast was enlarged. The result was shown in Fig. 4.

Fig. 4. Contrast stretching effect. The first row was original/mask/bounding box without contrast hitogram, and the shape of bounding box was wrong. The second result was preprocessed with contrast stretching, which produced more reasonable bounding box shape.

One interesting fact can be observed in mask result of 0070.jpg, as Fig. 7. shown. The left side of image, including swimming pool was blue, and this color seems to be very similar to our interested one. As a result, the mask in this area was predicted as positive, even there's no recycling bin there. Similar phenomena happened in the case of 0068.jpg. This shows our model prediction depends heavily on single pixel intensities rather than shape of the positive area. This could be a problem if applied this model on real-life situation. Imagine there is an object with similar blue color and the rectangular shape, the model might totally go wrong in this case. Hence, future research on "shape" effect should be taken into account, for example, extract sub-area into model in training instead of single pixel only might give better result in this case.

One failed predicted case is 0067.jpg in validation set. The main reason in this failure is the "layout" of objects. Since GDA model in this experiment learned distribution in color and brightness, the shape can only be completed by using image processing method – erosion and dilation. Unlike the regional features extracted in convolutional neural network, it's hard to classified "each" pixel into positive or negative by GDA model and considered the "expected whole shape" of recycling bin at the same time. They were done by different steps, therefore, two objects closed to each other, as fig. were easily fused into one object by dilation, which was not a correct prediction in this case. Such problem can be issued by manipulating parameters in dilation and erosion, but this is not a stable solution for all the cases.

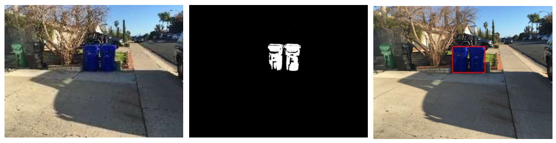

Fig. 5. 0065.jpg. There are two recycling bin in the (a) original image, and GDA generated reasonable mask (b), however, after dilation/erosion bounding boxes searching, they were clustered into one object.

In summary, the overall accuracy in this part is 90% (9 correct predictions out of 10 validation images). There are several parts should be improved, including model design, bounding box searching method, and labelling. As mention above, multiple color classes in generative models should improve the performance because of more complicated decision boundary to handle outliers. Due to morphological operation, the morphology in mask would be different from the original, this would cause the intersection-of-union index changed, like Fig. 6. showed. And the most important part – labelling, would directly influence how training data distributed, and better image processing methods, e.g., color space transformation, and criteria of extracting ROIs must be reconsidered to improve the model.

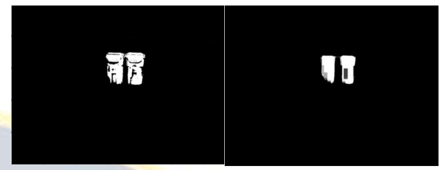

Fig. 6. 0065.jpg mask. (a) Mask generated by GDA model and (b) after morphological operation

REFERENCES

[1] UCSD ECE276A: Sensing & Estimation in Robotics (Winter 2022) https://natanaso.github.io/ece276a/

Fig. 7. Results on validation set: 0061.jpg ~ 0070.jpg. The first row presented masks generated by GDA model and the second row shows its corresponding bounding box images. Only 0067.jpg was reported as failure prediction.