HW1 Report

Guosong Li(A53277848)

1. Introduction

In the field of image processing, great importance is attached to the techniques used for color segmentation since it can simplify the vision problem by assuming that objects are colored distinctively. The reason for doing so is that the resulting simplified (and impoverished) image can be processed very rapidly, which can be important in mobile robot applications or system like target identification.

This report focuses on a barrel-detection problem, which is to identify a blue barrel from background with some blue but non-barrel noises. The results will be its corresponding binary color segmentation and figure with blue barrel being boxed.

This report will implement Mixture Gaussian model with k-means initialization strategy.

2. Problem Formulation

The object is to train a classifier to detect the barrel in test example. Given the labeled training data $\mathrm{D} = \{(\mathbf{x_i}, y_i)^n\} = (X, \boldsymbol{y})$ with $\mathrm{y_i} = (1, \dots, K)$ and $\mathbf{x_i} \in R^d$, a mixture Gaussian model with **J** components can be expressed as

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

where

$$\mathrm{p}(\mathrm{y} \mid \theta) = \prod_{i=1}^{n} \prod_{k=1}^{K} \theta_k^{1\{y_i = k\}}$$

$$\mathrm{p}(\mathrm{x_i} \mid y_i = k, \omega) = \sum_{j=1}^{J} \alpha_{kj} \phi(\boldsymbol{x_i}\,; \boldsymbol{\mu}_{kj}, \boldsymbol{\Sigma}_{kj})$$

And using EM algorithm to commit unsupervised learning and obtain **MLE** parameters, then for a given test example $\mathbf{x}^* \in R^d$, the classifier can be written as:

$$\mathrm{y}^* = \underset{y \in \{1, \dots, K\}}{\arg\max}\ \ log\theta_y^{MLE} + \log\left(\sum_{j=1}^{J} \alpha_{yj}^{MLE} \phi(\boldsymbol{x}^*\,; \boldsymbol{\mu}_{yj}^{MLE}, \boldsymbol{\Sigma}_{yj}^{MLE}) \right)$$

3. Technical Approach

First use function ‘roipoly’ to hand-label the training image. To improve the ability of the classifier to identify a blue object is barrel or not, the training set is split into three classes. Before labeling, a chart of label information is created which contains the number of blue barrels and blue non-barrels to be labeled through observation on each image. Thus, the training set has been divided into 3 classes: $\{\mathrm{k} = 1 = \mathrm{background},\ \mathrm{k} = 2 = \mathrm{blue\ but\ not\ barrel},\ \mathrm{k} = 3 = \mathrm{blue\ barrel}\}$. Then a ‘for’ loop is implemented to label each image in training set and obtain its corresponding mask. After labeling, the labeled training data is stored in the file ‘label_K_class.data’.

Then it comes to the training step. First thing to do is to read each pixel of each training image and combine them into matrix form. This solution uses the default RGB color space, which makes each training data a 3-dimensional vector $\mathbf{x_i}$ that contains the features of R, G, B value. To prevent the presence of overfitting, 6 out of 46 training images is randomly picked as the validation set, and the rest 40 training images is converted to a **n-by-3** matrix $\boldsymbol{X}$ such that

$$\boldsymbol{X} = [\boldsymbol{x_1}; \boldsymbol{x_2}; \dots; \boldsymbol{x_n}]$$

For the number of clusters J, several different numbers are being tested and finally $J = 4$ is picked as the cluster number. Then the K-means strategy is used to for initialization for parameters:

$$\omega_{\mathrm{kj}}^{(0)} = \{\alpha_{kj}^{(0)}, \boldsymbol{\mu}_{kj}^{(0)}, \boldsymbol{\Sigma}_{kj}^{(0)}\}$$

Then through the EM algorithm, the MLE of each parameter can be obtained as

$E\ step$:

$$\boldsymbol{r}_{kj} = \frac{\alpha_{kj}^{(t)} \cdot \phi(\boldsymbol{X}; \boldsymbol{\mu}_{kj}^{(t)}, \boldsymbol{\Sigma}_{kj}^{(t)})}{\sum_{m=1}^{J} \alpha_{km}^{(t)} \cdot \phi(\boldsymbol{X}; \boldsymbol{\mu}_{km}^{(t)}, \boldsymbol{\Sigma}_{km}^{(t)})}$$

$M\ step$:

$$\alpha_{kj}^{(t+1)} = \frac{(\boldsymbol{1^T} \cdot \boldsymbol{r}_{kj})\{y_i = k\}}{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} 1\{y_i = k\}}$$

$$\boldsymbol{\mu}_{kj}^{(t+1)} = \frac{(\boldsymbol{1^T} \cdot \boldsymbol{r}_{kj} \cdot \boldsymbol{X})\{y_i = k\}}{(\boldsymbol{1^T} \cdot \boldsymbol{r}_{kj})\{y_i = k\}}$$

$$\boldsymbol{\Sigma}_{kj}^{(t+1)} = \frac{\left(\boldsymbol{1^T} \cdot \boldsymbol{r}_{kj} \cdot \left(\boldsymbol{X} - \boldsymbol{\mu}_{kj}^{(t+1)}\right)^T \cdot \left(\boldsymbol{X} - \boldsymbol{\mu}_{kj}^{(t+1)}\right)\right)\{y_i = k\}}{(\boldsymbol{1^T} \cdot \boldsymbol{r}_{kj})\{y_i = k\}}$$

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

Thus, by iterating EM algorithm for certain amount of times until convergence (which in this case a certain number is picked to reduce computation cost), the local maximum likelihood estimate for each parameter can be obtained and they are stored in file like 'mu_rgb.data'.

Then, using the classifier in problem formulation part to classify each pixel of test image to K different labels. For segmentation mask, the label other than the blue barrel is considered as **0** and blue barrel label is set to **1**.

For bounding box task, the function 'dilate' and 'erode' are implemented to get rid of the noise around on the segmentation mask. Then through implementation of function 'label' and 'regionprops', the region of interests, which is the blue barrel, are labeled and each bounding box coordinates are returned. One small trick is that the area of region of interest is being monitored to discard those noises whose area is not large enough to become a barrel.

It turns out that the performance of classifier on images with bad lightness is not desiring. So, tries to convert training image to different color space (to be specific, LUV, YCrCb and HSV) to learn a new classifier has been committed, but the result is even worse which misclassify pixels in background as barrels. One reasonable explanation of this result is that the label is not enough for classify difference between different kinds of background and barrel.

Also, the idea of combing RGB and other color space features together to achieve better performance on testing dark image, i.e. makes the dimension of **x** becomes 3*n, also failed during update on parameters. The reason might due to the different value range in different color spaces (normalization doesn't work either).

However, mixture Gaussian model using EM algorithm is computationally expensive because of the difficulty to convergence especially with high number of clusters, it might not be the best approach for this problem since the number of data points is large enough.

4. Results

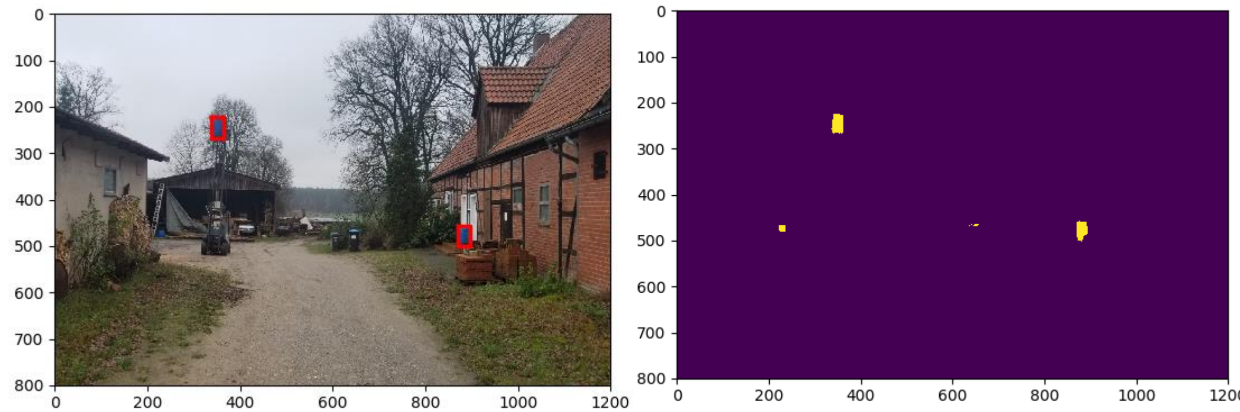

Figure **1**: test example of perfectlly boxed and its corresponding segmentation mask

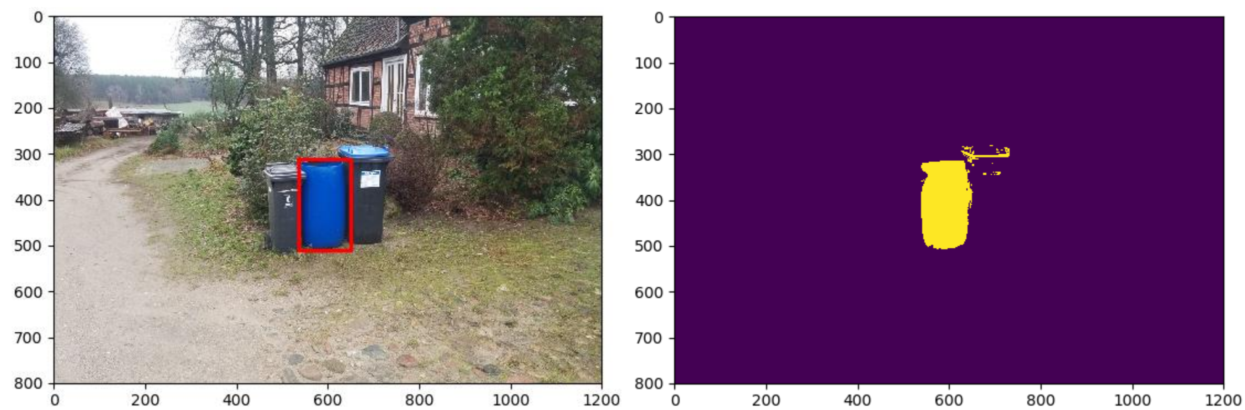

Figure **2**: test example of perfectlly boxed and its corresponding segmentation mask

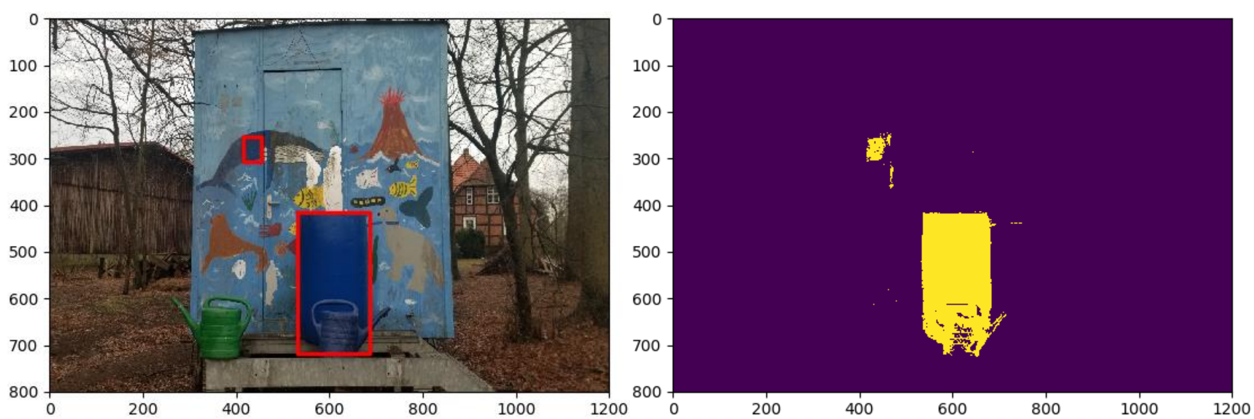

Figure **3**: test example of overly-boxed and its corresponding segmentation mask

From figure 1 and 2 it can be observed that despite the existence of some noises in segmentation mask, the bounding box can handle noises perfectlly as long as the light conditions is acceptable.

Figure 3 shows a test example that cannot handle the noises decently because area of noise is too large.

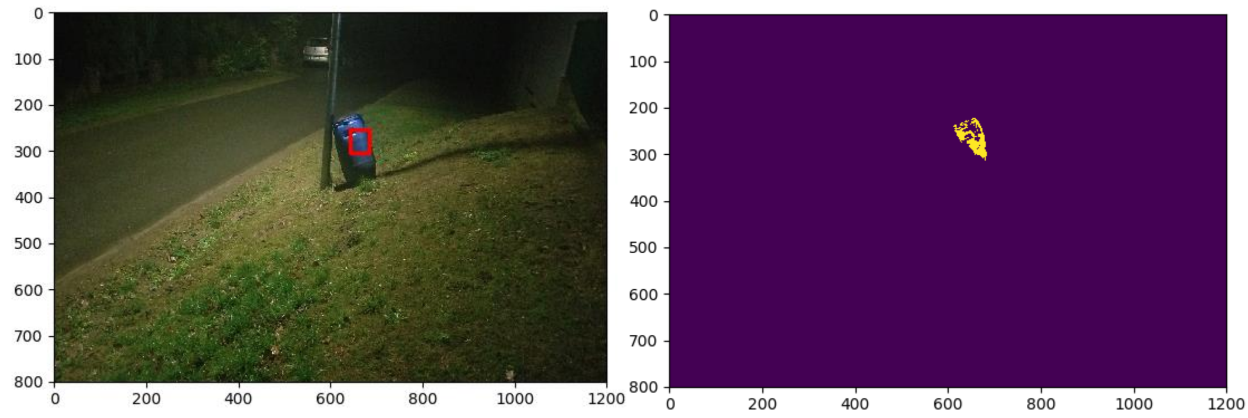

Figure 4: test example of missing-boxed

Figure 4 illustrates an example that due to the fact that the barrel in the image is not in a good light condition, some of the features are not properly classified. So even with the help of function like 'dilate', the bounding box is still not desirable.

A possible approach to improve performance is to train the classifier on those images without proper light conditions.

Below is the statistical histogram of the training set as the test image. The majority of the missing-boxed is those image without peoper light conditions.