ECE 276A Project 1: Color Classification and Recycling Bin Detection

1st Yifan Wu

UCSD ECE
yiw084@ucsd.edu

***Abstract*—The goal of this project is to design an object detection algorithm based on color segmentation with a probabilistic pixel color classifier and deterministic bounding box algorithm with object shape detection. If there exists a blue recycling bin, a bounding box will be drawn around the specific blue bin**

I. Introduction

Object detection has applications in many areas of computer vision and image processing. Every object has its own special features such as size, dimension and color. Our object detection algorithm combined color segmentation with a pixel color classifier and bounding box algorithm with object dimension detection.

A machine learning approach was used to classify and extract blue color features. Three one-vs-all binary logistic regression models were developed to distinguish among red, green, and blue pixels. In order to improve the accuracy, the binary logistic regression model was retrained with additional hand-labeled pixel color data among bin-blue, non-bin-blue, black, red, yellow and green. A sufficient amount of pixel color data allowed the classifier to detect bin-blue region in an image. A Region of Interests classification algorithm was developed to compare the similarity between the blue region of interest and blue recycling bin. This simple algorithm helped to detect the existence and location of blue recycling bin in the image.

II. Problem Formulation

A blue-bin detection problem consists of three problems: pixel classification, image segmentation and Region of Interests classification. Three binary logistic regression models are combined into an one-vs-all logistic regression model. In each binary logistic regression, given a sample pixel $x \in \mathbb{R}^3$, a logistic sigmoid function maps the continuous value $\omega^T x$ to a Bernoulli probability mass function for a binary label $y \in \{-1, 1\}$. Three parameters (ω^*_{red}, ω^*_{green}, ω^*_{blue}) were optimized individually with respect to 3 target colors (red, green, blue).

Image segmentation segments the 3-D color space into a set of volumes associated with different colors. A million hand-labeled training pixel data with both positive examples and negative examples is collected with respect to 6 color class categories (bin-blue, non=bin-blue, black, red, yellow, green). Each pixel is a 3-D vector $x = (R, G, B)$. The model takes entire original RGB color images as input which is a 3 dimensional matrix $M \in \mathbb{Z}_0^{m \times n \times 3}$, and uses probabilistic pixel color classifier to classified all pixels in the image. The output is a binary image mask, $M \in \mathbb{Z}_0^{m \times n}$, which indicates region of interests based on the bin-blueness. Each entry of matrix, $M_{i,j} \in \{0, 1\}$ is 1 indicates the pixel has bin-blue color and 0 otherwise.

Fig. 1. *Blue-bin Detector Sample Output*

Fig. 2. *Dimension of Blue Bin*

Region of Interests classification evaluates similarity between region of interests by size, dimension orientation etc. Figure 2 shows the dimension of three blue recycling bins with different sizes respectively [1]. In each size of blue bin, H denoted the height, and, W denoted the width of a regular

recycling bin. Similarity is largely weighted on shape ratio of blue bin, $H/W \in \mathbb{R}$, Other criteria such as ratio between region area and bounding box area, ratio between region area and entire area of image, and orientations of the bin were evaluated as well. The input is a prepossessed image mask, $M \in \mathbb{Z}_0^{m \times n}$, and output is a list of bounding box coordinates[(x_1, y_1), (x_2, y_2)] corresponding to coordinates of upper-left corner and lower-right corner of output bounding box. Figure 1 displays a sample result of blue-bin detector with a red rectangular bounding box around a blue bin.

III. Technical Approach

A. *Supervised Learning*

Given a set $D := \{x_i, y_i\}_{i=1}^n$ of iid examples $x_i \in \mathbb{R}^d$ with associated scalar labels $y_i \in \mathbb{R}$ from unknown joint probability density function $p_*(y, x)$, supervised learning defines a function $h : \mathbb{R}^d \to \mathbb{R}$ that can assign a label y to a given data point x, either from the training dataset D or from an unseen test set generated from the same unknown probability density function $p_*(y, x)$.

The loss function measures number of times h is wrong about the labels is defined as:

$$h = min_h Loss_{0-1}(h) := \frac{1}{n} \sum_{i=1}^{n} 1_{h(x_i) \neq y_i} \tag{1}$$

In pixel classification, a set $D := \{x_i, y_i\}_{i=1}^n$ of iid pixel data is given where, x_i is the i^{th} pixel, y_i is the label for the i^{th} pixel, n is the number of number of pixel samples, and d is the number of feature in RGB color space. Each pixel is normalized by 255. The training data can be written in matrix from $D = (X, y)$, where $X \in \mathbb{R}^{n \times d}$, $y \in \mathbb{R}^n$

B. *Logistic Regression*

Logistic Regression is a discriminative model which choose to model $p(y|x; \omega)$ with parameters ω to approximate the unknown label-generating pdf with a logistic sigmoid function. A logistic sigmoid function maps the continuous value $\omega^T x$ to a Bernoulli probability mass function for a binary label $y \in \{-1, 1\}$

$$\sigma(z) := \frac{1}{1 + \exp(z)} \tag{2}$$

Since the data $D = (X, y)$ are iid, the conditional probability of label $y \in \{-1, 1\}$ given training data $X \in \mathbb{R}^d$ and learning parameter $\omega \in \mathbb{R}^d$ is:

$$p(y|X, \omega) = \prod_{i=1}^{n} \sigma(y_i x_i^T \omega) = \prod_{i=1}^{n} \frac{1}{1 + \exp(-y_i x_i^T \omega)} \tag{3}$$

In order to maximize the joint likelihood $p(y|X, \omega)$, maximum likelihood estimation(MLE) is used to optimize the learning parameter ω.

$$\omega MLE = \arg\min_{\omega} \sum_{i=1}^{n} \log\left(1 + \exp(-y_i x_i^T \omega)\right) \tag{4}$$

Since $\nabla_\omega(-\log p(y|X, \omega)) = 0$ does not have a closed-form solution and negative log-likelihood $-\log p(y|X, \omega)$ is convex in w, iterative optimization algorithm like gradient descent is used to approximate the global minimum:

$$\omega_{MLE}^{(t+1)} = \omega_{MLE}^{(t)} - \alpha \sum_{i=1}^{n} y_i x_i (1 - \sigma(y_i x_i^T \omega_{MLE}^{(t)})) \tag{5}$$

where α is the learning rate and n is number of iteration.

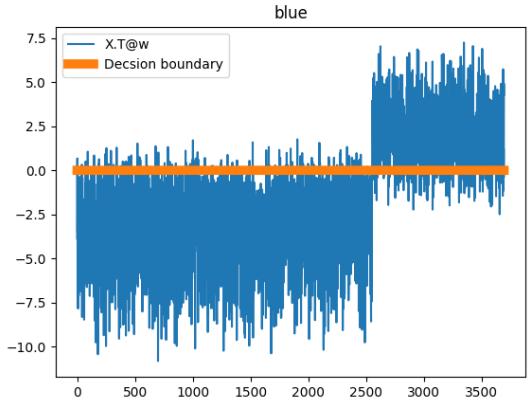

Fig. 3. *Decision Boundary of Logistic Regression*

Logistic Regression generates a linear decision boundary: $x^T \omega = 0$. The prediction $y^* \in \{-1, 1\}$ corresponding to test data $x_* \in \mathbb{R}^d$ and optimized learning parameter $\omega \in \mathbb{R}^d$ is defined as:

$$y_* = \begin{cases} 1 & x^T \omega \geq 0 \\ -1 & x^T \omega < 0 \end{cases} \tag{6}$$

From figure 3 above, a line boundary separate blue pixels and non-blue pixels. Fluctuation exists due to the property of Logistic Regression that it has lower bias but higher variance.

C. *One-vs-all Logistic Regression*

One-vs-all Logistic Regression is a model consisted of $K \in \mathbb{Z}^+$ distinct binary Logistic Regression classifier approximates the unknown label-generating pdf for $y \in \{1, ..., K\}^n$. For each class label $l \in \{1, ..., K\}$, a binary logistic regression classifier is trained by corresponding label $y_i' = l_j$ for $j \in \{1, ..., K\}$.

$$y_i' = \begin{cases} 1 & y_i = l_j \\ -1 & y_i \neq l_j \end{cases} \tag{7}$$

Given a test sample x_*, the prediction of one-vs-all Logistic Regression y^* is perform by finding the argmax of all of the individual prediction $y_{\{-1,1\}}^*$ in each binary Logistic Regression.

$$\begin{aligned} y^* &= argmax_{l \in \{1, ...K\}} p(y = 1 | x_*, \omega_l) \\ &= argmax_{l \in \{1, ...K\}} x_*^T \omega_l^* \end{aligned} \tag{8}$$

where K is number of classes, l is the class label, and ω_l is the learning parameter with respect to particular class l

D. *Image Segmentation*

Image segmentation utilizes Logistic Regression pixel classifier to classify the color class of each pixel in an image. A million hand-labeled training pixel data is collected through "roipoly.py" from the same probability distribution. Both positive examples and negative examples are included to form a balanced dataset with respect to six color categories (bin-blue, non-bin-blue, black, red, yellow, green). We are only interested in bin-blue color pixels and not bin-blue color pixels. Therefore, the training data can be written in matrix from $D = (X, y)$, where $X \in \mathbb{R}^{n \times d}$, $y \in \mathbb{R}^n$, $K \in \{-1, 1\}$, 1 indicates pixel is bin-blue and -1 otherwise.

The output of image segmentation is a binary image mask, $M \in \mathbb{Z}_0^{m \times n}$, which indicates region of interests based on the bin-blueness. Each entry of matrix, $M_{i,j} \in \{0,\ 1\}$ is 1 indicates pixel has bin-blue color and 0 otherwise. An image mask sample is shown in the second plot of Figure 4.

E. *Image Processing*

A large portion of blue recycling bin is recognizable in the image mask, but the model is not perfect due to some low bias and high variance. It misclassifies some pixel $x_j \in \mathbb{R}^3$ with similar value in the RGB color space. In order to filter out noisy objects in an image mask, morphological operations (e.g., dilation or erosion) was performed on the output binary image mask.

The erosion operator takes two pieces of data as inputs, binary image to be eroded and a kernel. Kernel determines the precise effect of the erosion on the binary mask image. The 3×3 square kernel is probably the most common kernel used in erosion. A larger kernel produces more extreme erosion effect. The trade-off of erosion is that erosion deletes some bin-blue pixels on the edge and shrinks the area region of interest. Other ensemble techniques of image processing such as opening (erosion followed by a dilation) and closing(dilation followed by an erosion) can be utilized to attain a clean and accurate image mask as well.

Removing small objects and connecting large regions of interests improved clarity and accuracy of blue-bin detector with acceptable trad-offs. A clean prepossessed binary image mask is shown in the third plot of Figure 4.

F. *Region of Interests Extraction*

1) ***Region Labeling****:* In order to identity and group pixels in each region of interests. Label function in skimage.measure takes the binary mask as input, and labels connected pixels in region of interests. The connectivity between pixels can be 1-, 2- and ndim-connected senses. Two pixels are connected when they are considered as neighbors by connectivity and have the same value. The output is a labeled integer image mask $P \in \mathbb{Z}_0^{m \times n}$. Each entry of matrix, $P_{i,j} \in \{0,...,L\}$ is labeled as integer, where L is the number of non-zero labels. Each value in the mask is the region label of the corresponding pixel. 0-valued pixels are considered as background pixels.

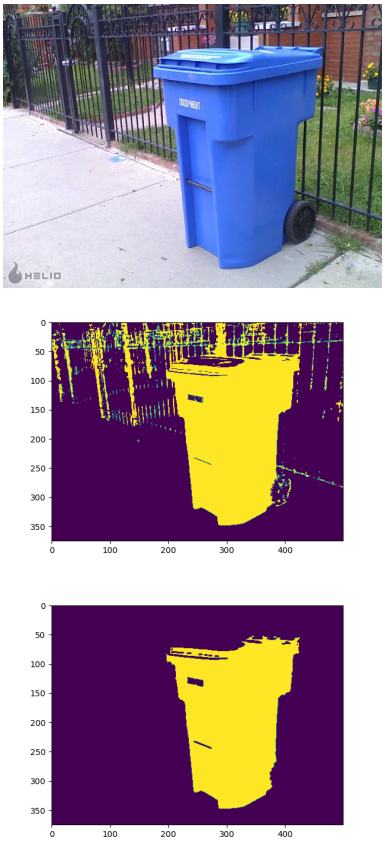

Fig. 4. *Comparison between original and processed Binary Image Mask*

2) ***Properties of Region of Interests****:* With the help of regionprops in skimage.measure, properties in each labeled region of interests are measured such as number of pixels of the region, denoted as $Area$, bounding box coordinate, denoted as (y_1, x_1), (y_2, x_2), number of pixels in bounding box, denoted as $Area_{bbox}$.

G. *Region of Interests Classification*

After labeling and extracting properties, region of Interests classification algorithm compares the similarity between each region of interests and a real blue recycling bin based on following assumptions and criteria.

Assumptions:

- Input image has at least a VGA (640×480) image size with basic quantity
- The blue bin presents in input image cannot be occluded more than 90%
- The blue bin presents in input image should have a similar dimension illustrated in Figure 2
- Input image cannot be extremely deformed which might distort the shape of blue bin in input image

Criteria:

1) ***Area of Region vs Area of Image****:* Ratio of pixels of the region to pixels in entire image should be at least larger or equal to 1 percent of image's total size. If the region of

interests is too small, even humans cannot recognize it in a normal-sized image. This criteria can filter out region of interest with extremely small numbers of pixels compared to number of pixels in entire image.

$$A/(m \times n) \geq ImageAreaThreshold = 1\% \tag{9}$$

*2) **Area of Region vs Area of Bounding Box**:* Ratio of pixels in the region to pixels in the total bounding box denoted as E. Some of region of interests contains a small number of pixels in labeled region, but they are spread out and form a large bounding box area with a large portion of 0-valued background pixels. Area of region of interests should fill approximately half of the bounding box area, in order to be considered as a blue bin candidate. Certain level of tolerance is allowed during detection.

$$\begin{aligned} E &= Area/Area_{bbox} \\ &\geq bboxAreaThreshold \\ &= 0.45 \end{aligned} \tag{10}$$

*3) **Shape ratio**:* Shape ratio of bounding box denoted as $S = h/w$, where $h \in \mathbb{Z}^+$ is height of bounding box, and $w \in \mathbb{Z}^+$ is width of bounding box. From Figure 2, shape ratio $S = h/w$ is approximately equals to 2. Other orientations of blue bin are considered as well. If the blue bin lays down, the ratio h/w is approximately equals to 0.5. Therefore a valid region of interests need to have a shape ratio $S \in [0.5, 2]$ This criteria can filter out region of interest which are tall and narrow, or short and wide.

$$\begin{aligned} h &= y_2 - y_1 \\ w &= x_2 - x_1 \\ \tfrac{1}{\epsilon_s} \leq S &\leq \epsilon_s = 2 \end{aligned} \tag{11}$$

where (x_1, y_1) is the upper-left corner of bounding box, (x_2, y_2) is the lower-right corner of bounding box, shape ratio threshold $\epsilon_s = 2$

Region of interests fulfills assumptions and criteria above will reward a rectangular bounding box and be classified as a blue bin. The coordinates of upper-left corner and lower-right corner of bounding box are record in to a list [x_1, y_1, x_2, y_2]

IV. Results

A. Pixel Classification Performance

Each example in the training or validation sets is a 28 × 28 image with a single RGB value at all of its pixels. Labeled training data was collected in the standard format $X \in \mathbb{R}^{n \times 3}$and $y \in \{1, 2, 3\}^n$, where 1 = red, 2 = green, 3 = blue and n are the number of examples.

The hyper-parameters for each Logistic Regression and one-vs-all Logistic Regression are:

- Learning rate α: {red : $5e-3$, green : $5e-2$, blue : $5e-3$ }
- Number of iteration: $10,000$
- Batch size: $3,694$
- ω_{red}: $[1.24744119, -1.05895242, -0.54773663]^T$
- ω_{green}: $[-5.89283605, 7.98480435, -5.30636785]^T$
- ω_{blue}: $[-5.92267304, -5.36443759, 7.95462632]^T$

TABLE I
Binary Logistic Regression Performance

Binary Logistic Regression	Metrics			
	Accuracy	*Precision*	*Recall*	*F1 Score*
Red	0.904	0.919	0.875	0.875
Green	0.925	0.884	0.883	0.884
Blue	0.927	0.879	0.882	0.824

TABLE II
One-vs-all Logistic Regression Performance

One-vs-all Logistic Regression	Metrics			
	Accuracy	*Precision*	*Recall*	*F1 Score*
3-class RGB	0.906	0.920	0.906	0.907

In pixel classification three binary logistic regression models are combined into an one-vs-all logistic regression model. Among Logistic Regression, Gaussian Discriminant Analysis(GDA) and Gaussian Naive Bayes, Logistic Regression is chosen, because Logistic regression has lower bias than Gaussian Naive Bayes and its implementation is easier than GDA Although this approach works well enough under simple environment. The decision between three classes might raise some ambiguity. Since conditional probability $p(y|x, w_l)$ does not sum to 1 A better approach to classify multiple color is to use K-ary Logistic Regression which ensures the probability will sum to 1.

Deep convolution neural networks (CNN) are able to significantly improve the accuracy of image classification and object detection [2] [3].There are certain advantages offered by convolution layers when working with image data:

- Fewer parameters: A small set of parameters (the kernel) is used to calculate outputs of the entire image
- Sparsity of connections: Each output element only depends on a small number of input elements, which makes the forward and backward passes more efficient.
- Parameter sharing and spatial invariance: The features learned by a kernel in one part of the image can be used to detect similar pattern in a different part of another image.

However, the disadvantage of CNN is it requires a much larger amount of training data. Training on CNN takes a longer time than supervised learning, and it is preferred to train on an expensive and powerful GPU in order to speed up training.

B. Color Segmentation Performance

In order to improve the accuracy, the binary logistic regression model was retrained with additional hand-labeled pixel color data. A million hand-labeled training pixel data is collected through "roipoly.py" from the same probability distribution. Both positive examples and negative examples are included to form a balanced dataset with respect to six color categories (bin-blue, non-bin-blue, black, red, yellow, green). A sufficient amount of pixel color data allowed the classifier

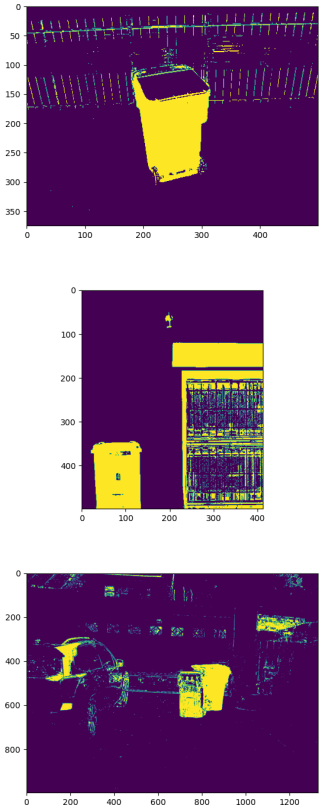

Fig. 5. *Color Segmentation Sample Result*

to detect the bin-blue region in an image. 6 binary logistic Regression were trained with collected data. The learning parameter is the following:

- $\omega_{bin-blue}$: $[-1.4130434, -6.53439706, 5.78082191]^T$
- ω_{blue}: $[-5.14103945, 3.58349541, -0.98399058]^T$
- ω_{black}: $[-1.3991432, -2.46383269, -2.96282062]^T$
- ω_{red}: $[1.66143423, -2.95040415, -5.70373092]]^T$
- ω_{yellow}: $[3.51266549, 2.58091188, -8.54103648]^T$
- ω_{green}: $[-2.3281738, 1.6028809, -5.0699202]]^T$

Although training data are collected from various color categories, our classifier does not perform well on color such as sky blue, dark purple and shadow black. Figure 5 displays several image masks of images in validation data. Although the shape and location of the blue bin were captured in each mask image after color segmentation, a lot of noisy features or misclassified pixels remained due to high variance. From a supervised learning point of view, collecting more training data might help to reduce the variance. Other generative models such as Gaussian Discriminant Analysis, Gaussian Naive Bayes are supposed to have lower variance. However, these models highly depend on a balanced dataset. One advantage of using Logistic Regression is that the decision boundary of Logistic Regression is linear. There is not too much degree of freedom of linear decision boundary. Therefore, Logistic Regression is less likely to suffer from an imbalanced dataset.

C. Image Processing Performance

Another approach to filter out unwanted features or objects in a binary mask is through image processing. Image processing techniques such as erosion and dilation were applied to the mask image in order to diminish small bright spots and enhance a large bright region of interests. Removing small object and connecting large region of interests improved clarity and accuracy of blue-bin detector with acceptable trade-offs. Image possessed binary image masks are shown in Figure 6.

Image processing could be performed before generating the image mask. If the input image is in RGB color space, the result of the binary mask is highly varied to brightness, occlusions, and other variations in the environment. By converting RGB color space to YCrCb or HSV and perform brightness equalization or saturation enhancement. Note that Logistic Regression might not perform well, since Logistic Regression has a linear decision boundary and H channel in HSV is not linear. Therefore, other color classified models with non-linear decision boundaries such as GDA should be chosen.

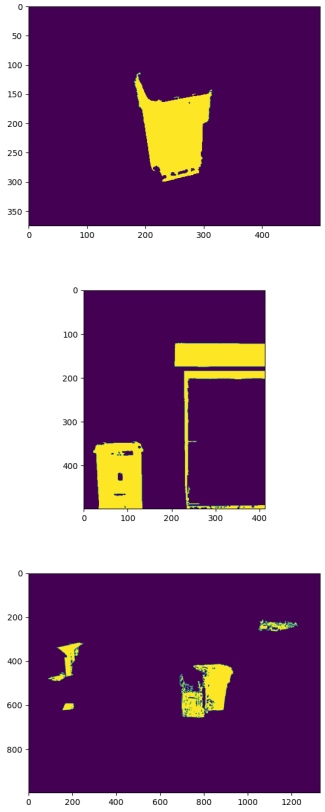

Fig. 6. *Image Processing Sample Result*

D. Bounding Box Performance

Region of interests fulfills assumptions and criteria described in Section G of Technical Approach above will reward a rectangular bounding box and be classified as blue bin. The coordinate of upper-left corner and lower-right corner

of the bounding box will be recorded into a list [x_1, y_1, x_2, y_2]. If there is no blue bin detected, it returns an empty list. Following are bounding box coordinates detected on each validation image:

$$
\begin{aligned}
&bbox1 : [(181, 114), (314, 300)] \\
&bbox2 : [(22, 346), (136, 499)] \\
&bbox3 : [(174, 95), (289, 235)] \\
&bbox4 : [] \\
&bbox5 : (751, 414), (934, 639) \\
&bbox6 : [] \\
&bbox7 : [] \\
&bbox8 : [] \\
&bbox9 : [] \\
&bbox10 : []
\end{aligned} \tag{12}
$$

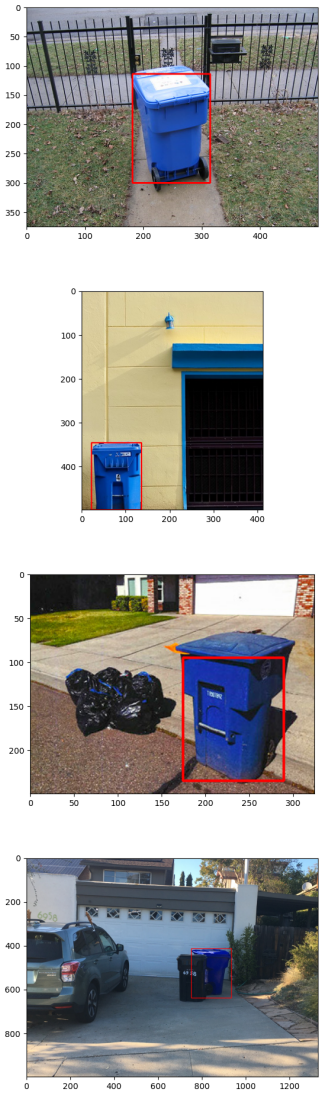

Fig. 7. *Bounding Box Result*

It is hard to evaluate the performance only based on the euclidean distance between the True bounding box coordinate and detected bounding box coordinate. A combination of reward and manual detection is used to evaluate the performance of bounding box. The idea of reward comes from reinforcement learning, each image has a reward equals to number of blue bins present in the image. If there is no blue bin in the image, reward equals to 1. Each correct detection of blue bin gets a reward $+1$ and each missing or wrong detection gets a reward -1.

Training set and validation set contain images with one blue bin, several blue bins and no blue bin. In training set the maximum reward, $\max R_{train} = 77$. Bounding box detection received $R = 57$, and it misclassified 20 Region of interest. The total percentage of reward in training set $= 74.3\%$ In validation set, the maximum reward $\max R_{test} = 11$. Bounding box detection received $R = 8$. The total percentage of reward in training set $= 72.72\%$. Summing maximum reward in both training and validation set, and calculate the average reward received $\bar{R} = 73.86\%$

Applying restrictions according to the number of pixels and dimension, a blue-bin detector is able to detect blue bin in an image with 70% accuracy. A large portion of detection failures are due to some shadow or dark spot between two or several near blue bins. This causes bin detector to classify two blue bin into one blue bin or no blue bin at all. Morphological operations such as dilation indeed enhances the bright region, but it might connect two separate blue bin together as shown in the first and third plot of Figure 9. A sophisticated shape-based algorithm can further improve accuracy and precision. The number of edges of the approximated polygon of a region of interests should be between 4 to 6, since a blue bin is approximately a rectangle or a trapezoid.

REFERENCES

[1] N. Atanasov, "Color classification and recycling bin detection."

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[3] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, and Y. Yu, "Hd-cnn: hierarchical deep convolutional neural networks for large scale visual recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2740–2748.

Fig. 8. *Bounding Box Result: Bounding Box not detect*

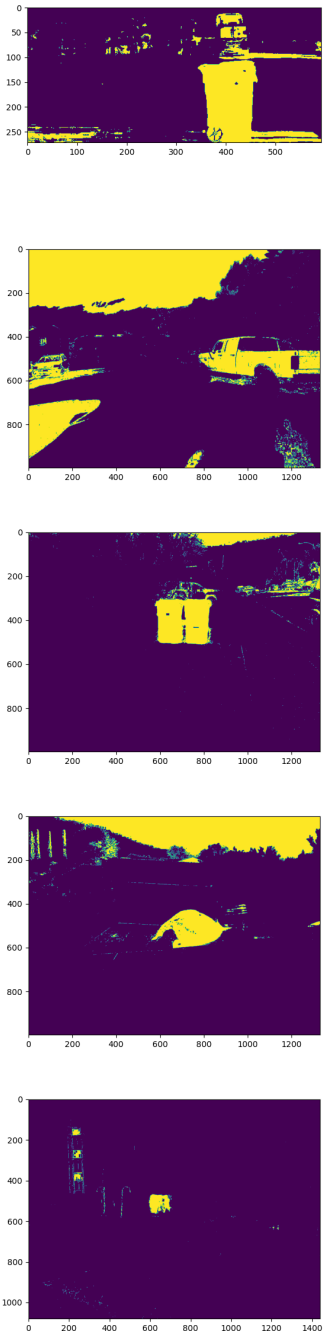

Fig. 9. *Image Mask from Color Segmentation*