Color Classification and Recycling Bin Detection

Nathan Cusson-Nadeau

I. Introduction

When developing a robotic system, it is paramount to provide accurate sensing information for the robot to use to make decisions. One conduit of such information is through the use of computer vision. In this sensing technique, some kind of visual receptor like a camera is used to observe the robot's environment visually to produce an image, and then equipped with this new information the robot can react based on data extracted from this image. For instance, an autonomous vehicle may be interested in identifying a stop sign or red light through the use of cameras so that it may initiate braking. While this sounds like a trivial task to a human, it is a very difficult one to a computer. In order to accomplish this task, attributes from the object of interest must be used by the robot to identify with high accuracy the object. For the case of the stop sign, the octagonal shape and red coloring would be good attributes to use to discriminate the sign from its surroundings. However, even this simple identification provides a problem to a computer when interpreting an image. How can it recognize only the red parts of an image? How does it know if these red parts of the image are a stop sign and not something else like a red fire hydrant?

In this paper, we are presented with a similar problem to detecting the stop sign. We had two goals. First, create a color classifier which could predict with high accuracy the class of color of a never before seen set of pixels. Second, given an unseen set of images that could contain a blue recycling bin, successfully identify and draw bounding boxes around these bins and return the coordinates of the boxes.

To accomplish the first goal, we developed a pixel classifier computer program employing a logistic regression predictive model. For the second goal, a similar logistic regression predictive model was developed which tried to predict with high accuracy what pixels in an image were recycling bin blue. Then after processing the image and augmenting it into a binary one, a program was developed to draw bounding boxes around bin blue regions with similar shapes to a recycling bin.

II. Problem Formulation

A. Pixel Classification

Provided an unseen set of normalized pixel values $X^* \in \mathbb{R}^{n\times 3}$ where $X^*_n = [R_n, G_n, B_n]$ and associated RGB color space label $y_n \in \{1, 2, 3\}$ (representing red, green and blue respectfully), predict with high accuracy the corresponding label y^*_n of each pixel in X^*.

B. Recycling Bin Detection

1) Image Segmentation: Provided an unseen RGB image in the form of a pixel array, $Img \in \mathbb{Z}^{m\times n}_{\geq 0}$, where m and n denote the height and width of the image in pixels respectively, and the coordinates $(0, 0)$, (m, n) represent the top-leftmost and bottom-rightmost pixels respectively, with pixel values $X \in \mathbb{R}^{n\times 3}$, create a binary mask $Mask \in \mathbb{Z}^{m\times n}$ with binary pixel elements $Mask_{(m,n)} \in \{0, 1\}$.

2) Bounding Box Creation: Given a binary mask image $Mask$ generated from the previous color segmentation step, generate rectangular bounding boxes $B = [(x, y), (x + w, y + h)]$ with coordinates (x, y) and $(x + w, y + h)$, where $x, y \in \mathbb{Z}_{\geq 0}$ are coordinates in the $Mask$ image, $w, h \in \mathbb{Z}_{\geq 0}$ are the width and height of B in pixels respectively.

III. Technical Approach

A. Creating a Pixel Classifier

In order to classify pixels in X^*, three discriminative prediction models $p_r(y_r|x; \omega_r)$, $p_g(y_g|x; \omega_g)$, $p_b(y_b|x; \omega_b)$ (one for red, green and blue pixels respectively) were trained using a training data set $D := (X, y)$. Here, a logistic regression model was chosen for our discriminative model (1):

$$p(y|X; \omega) = \prod_{i=1}^{n} \sigma(-y_i x_i \omega) \tag{1}$$

Where $\sigma(z)$ represents the sigmoid function (2):

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

and $\omega^{\perp} \in \mathbb{R}^3$ is the trained Maximum Likelihood Estimation (MLE) weight parameter defined as:

$$\omega_c := [\omega_{c,r}, \omega_{c,g}, \omega_{c,b}]$$

where $c \in \{r, g, b\}$ defines which probabilistic model the weight corresponds. x_i is the row vector associated with row i of pixel training set X. To obtain ω, a gradient descent (2) algorithm was used with learning rate $\alpha = 0.05$ and iteration count $k = 50,000$.

$$\omega^{(k+1)}_{MLE} = \omega^{(k)}_{MLE} + \alpha \sum_{i=1}^{n} y_i x_i (1 - \sigma(y_i x_i \omega^{(k)}_{MLE})) \tag{3}$$

After training, each ω_c was used to compute the most probable label $y_{*,c} \in \{1, -1\}$ for each pixel in the unseen pixel set X^* using the linear decision boundary (4):

$$y_{*,c} = \begin{cases} 1 & x_* \omega_c \geq 0 \\ -1 & x_* \omega_c < 0 \end{cases} \tag{4}$$

Lastly, to account for conflicting pixel decisions between models, the probability $P_{c,i}$ of each index's pixel labels

$(y_{*,r,i}, y_{*,g,i}, y_{*,b,i})_{i=1}^{n}$ were computed using a modified equation (1).

$$P_{c,i}(y_{*,c,i}|x_{*,i};\omega_c) = \sigma(-y_{*,c,i}x_{*,i}\omega_c) \tag{5}$$

Setting the probability of negative labels y_* to 0, an argmax function was then used to compute which label $y_{*,c,i}$ provides the highest probability given $x_{*,i}$ and ω_c.

$$y_{*,i} \in \arg\max_{y_*,c,i} \mathbf{P}(y_{*,c,i}|x_{*,i};\omega_c) \tag{6}$$

Using these final computed labels, the vector $\mathbf{y}_* \in \mathbb{R}^n$, $y_{*,i} \in \{1,2,3\}$ our end goal, was constructed with the following algorithm:

Algorithm 1 $\mathbf{y}_*$ Label Assignment

```
for i=1 to n do
    if p_{i,max} == red then
        y_{*,i} = 1
    else if p_{i,max} == green then
        y_{*,i} = 2
    else if p_{i,max} == blue then
        y_{*,i} = 3
    end if
end for
```

(Note: In the extremely unlikely event of a probability tie between colors, the algorithm will decide it is the lowest indexed color by order of operations)

B. Detecting Recycling Bins

In order to detect recycling bins in an image, our approach was separated into two steps. For the first step, the image was converted into a binary mask. In the second step, this binary mask was then augmented to make finding the contour of a bin more consistent to ultimately create a bounding box around all bins in the image. The technical approach of both steps will now be covered.

1) Step 1 - Image Segmentation Using Logistic Regression:

Image segmentation is a color classification problem. As such, it was approached in a very similar manner to Part A. of this report. However in this case, the problem is much simpler because when using the same binary logistic regression model only one probability model for the pixel color combination of bin-blue is required.

Using the same model from (1), it was necessary to collect pixel data D to train ω. In order to do this, a combination of two data sets was collected. For the first dataset, the same collection of red and green pixels from image files in Part A was used. For the second data set, a tool called roipoly was used in the programming language Python. This tool collects pixel values from a selected image's regions of interest. By selecting various pixels from both bin-blue regions and not bin-blue regions, 2 more datasets, bin-blue and not bin-blue, were collected. Combining all 3 datasets and providing the associated labels ($y_i = 1$ for bin-blue, $y_i = -1$ for not bin-blue) we finally construct our training data set $D := (X, y)$.

Utilizing equation (3) to train the weights using learning rate $\alpha = 0.1$ and $k = 1000$ iterations, ω was obtained.

With ω, given an unseen image Img a binary mask image $Mask$ could be created using the linear decision boundary similar to (4).

$$Mask_{(m,n)} = \begin{cases} 1 & Img_{(m,n)}\omega_c \geq 0 \\ 0 & Img_{(m,n)}\omega_c < 0 \end{cases} \tag{7}$$

Thus, providing the desired binary mask.

2) Step 2 - Bounding Box Creation Using OpenCV:

To isolate the *high* pixels in $Mask$ we used erosion and dilation image processing techniques using the erode and dilate methods from the OpenCV library. Through erosion, noisy small pixel contours could be removed from the image. By then dilating the image, some of the original size of larger contours could be preserved. Kernel sizes for erosion were greater than those of dilation to attempt to remove overlap between contours to account for cases where bins were adjacent or placed near regions of similar pixel color.

Next, the image was padded with a 1 pixel width boundary of 0's to prevent contours intersecting with the borders of the image from being misinterpreted.

Now with the image processed, contours of pixel value 1 could be created and labeled using the OpenCV method findContours. This method returns an object called contours which contains each labeled contour and the coordinates of all of its associated pixels.

Using the newly created contours another OpenCV method was used called boundingRect, which using the centroid and extremities of each contours returns the coordinates, width, and height $((x, y), w, h)$of each contour's associated bounding rectangle.

Due to misclassified regions of value 1 pixels, it was necessary to create a similarity score and area filter to remove contours which were unlikely to be recycling bins. The similarity score S was based on how close the ratio of a contour's bounding rectangle's width w and height h was to a standard sized recycling bins ratio of width to height. The ratio of the contours rectangle was defined as:

$$r := \frac{h}{w}$$

and the ratio of height to width of a standard recycling bin r_{bin} was computed by averaging length l_{avg}, width w_{avg}, and height h_{avg} dimensions of varying standard volume bins, then averaging the average length and width aw_{avg} (to attempt to account for orientation of bin facing camera). Defined as:

$$r_{bin} := \frac{h_{avg}}{aw_{avg}}$$

See Table 1 for metrics used.

TABLE I: Standard Recycling Bin Dimensions

Bin Type	Dimensions (in)		
	w	l	h
32 Gallon	19	26	36.75
64 Gallon	22	26.5	44
95 Gallon	23	31.5	46

Using these ratios the similarity score was defined as a percentage given by:

$$S := (\frac{r}{r_{bin}}) \times 100$$

During testing $S = 0.60$ was used.

Next to account for small contours that could not be bins that could pass the similarity score test, an area filter was put in place. If the $\frac{A_{rect}}{A_{img}} \leq p_A$ where A_{rect} represents the area of a contour's bounding rectangle in pixels, A_{img} is the area of the image in pixels, and p_A is a arbitrary threshold percentage. The idea here being that if a rectangle is similar enough but does not take up sufficient room in the image, it is assumed to not be a bin. For testing, $p_A = 0.015$

The following algorithm was used to return bounding box coordinates:

Algorithm 2 Bounding Rectangle Creation

Given S, p_A, r_{bin}
boxes = []
for i=1 to n **do**
  x, y, w, h = Bounding Rect. Coordinates of Contour i
  $A_{rect} = w \times h$
  $A_{img} = w_{img} \times h_{img}$
  $r = \frac{h}{w}$
  if $\frac{A_{rect}}{A_{img}} \leq p_A$ **and** $\frac{r}{r_{bin}} \geq S$ **then**
    boxes = boxes + $[x, y, x + w, y + h]$
  end if
end for
return boxes

IV. Results

A. Pixel Classification Accuracy

Using the trained parameters $\omega_r, \omega_g, \omega_b$, (see Table II for trained values) Equation (6) and Algorithm 1, on an unseen validation test set of 3D RGB pixel values, our predictive model yielded 100% accuracy. Accuracy was determined by tallying how many correct predictions of labels y_* there over total possible labels reported as a percentage.

However, when testing this set of values against another different unseen validation set only 81% accuracy was received. It was found that all mistaken labeling events occurred when the predictive model attempted to classify blue. This does not seem to be caused by overfitting as good accuracy was seen in the validation set. Of the blue's that were misclassified, many were classified as red. This could imply that both red and blue were good predictions for these pixels but because red's weight in the ω_r value is larger than blue's weight in the ω_b it would always be chosen as it's confidence in it's prediction would on average be larger. Further investigation and comparison would be necessary between validation sets to find the root causes behind this inconsistency.

TABLE II: Logistic Regression Trained Weights

Trained Weights*	Colors		
	Red	***Green***	***Blue***
ω_r	67.42	-37.77	-36.44
ω_g	-46.05	51.93	-41.01
ω_b	-44.37	-39.83	49.83

*Values rounded to second decimal place.

B. Recycling Bin Detection Accuracy

Using a discriminative model (7) and trained parameter ω (see Table 3) to create a binary mask and a similarity score of $S = 0.60$ and area filter of $p_A = 0.015$, Algorithm 2 was able to create bounding rectangles with 100% accuracy when given a set of 10 images with and without recycling bins. Coordinates of the bounding rectangles can be seen in Table 4.

TABLE III: Image Segmentation Trained Parameter ω

Trained Weight*	Colors		
	Red	***Green***	***Blue***
ω	-4823.12	956.66	2040.85

*Values rounded to second decimal place.

TABLE IV: Bounding Box Coordinates of Validation Set

Image	Bounding Box Coordinates			
	x	***y***	***x+w***	***y+h***
0061.jpg	188	140	312	286
0062.jpg(1)	0	0	415	501
0062.jpg(2)	30	354	135	501
0063.jpg	177	102	277	223
0064.jpg	357	115	461	274
0065.jpg	793	422	930	623
0066.jpg	-	-	-	-
0067.jpg(1)	695	312	825	507
0067.jpg(2)	589	312	702	504
0068.jpg	-	-	-	-
0069.jpg	-	-	-	-
0070.jpg	-	-	-	-

Accuracy was determined by if manually drawn bounding box around a recycling bin overlapped by at least 50% with Algorithm 2's bounding box. However, this did not account for excess bounding boxes. See subsection "Observations and Room for Improvement" for further elaboration.

Images of binary masks compared to their original image can be found in Figure 1. Examples of eroded and dilated images can be found in Figure 2. Finally, all 10 validation images are shown with contours drawn in red and bounding boxes drawn in light-green in Figures 3 and 4.

1) Observations and Room for Improvement: As mentioned above, when multiple bounding boxes were provided in excess to the actual number of bins, an accuracy of 100% was still rewarded. In events where this happens, it's not valid to say that the algorithm has performed correctly. An incident of this case occurred in image 0062.jpg. As shown in Table 4, 2 bounding boxes were returned by Algorithm 2. However, only 1 bin was present in the photo. Upon closer examination, it appears this arose because of a combination of the image padding and aspect ratio of the photo. Due to the padding, the algorithm thought that there was a contour surrounding the edge of every photo. Because of the similarity score, this contour was generally ignored. However because the aspect ratio of 0062.jpg was of close enough similarity, this contour was mistakenly thought to be a very large recycling bin. To avoid this issue in the future, the area filter could simply be adjusted to also exclude area ratios equal to 1 (i.e. the entire image).

A weak point in the trained color segmentation parameter ω was apparent in many images. Certain shades of black and lighter shades of blue were often mistakenly thought to be bin-blue. In image 0065.jpg this is very apparent. The adjacent black trashcan was thought to be bin-blue when in the shade. Additionally, in image 0066.jpg the shadows on the asphalt were misinterpreted as bin-blue. Finally, the biggest mistaken color (although ultimately of little relevance) was the daytime sky. However because of it's contour shape it was very difficult to be mistaken as a recycling bin. To circumvent these misclassifications, a new parameter ω should be trained with more shades of black and light blue added to the dataset with associated -1 labels. This should in theory filter out these problem colors. However, more testing would be necessary to see if this new weight would also struggle to recognize darker blue-bin shades due to light invariance in photos.

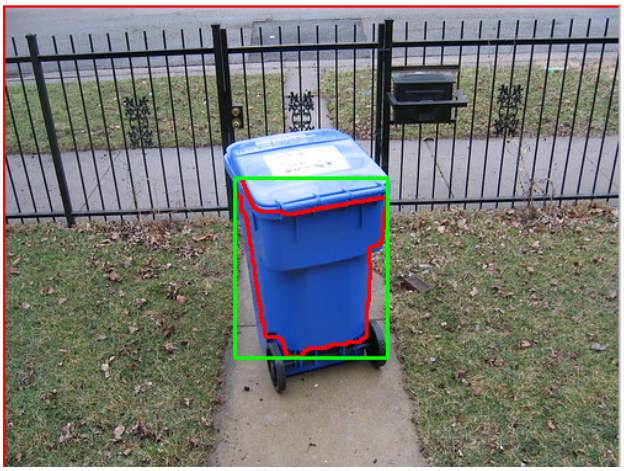

Fig. 1: 0061.jpg Bounding Box

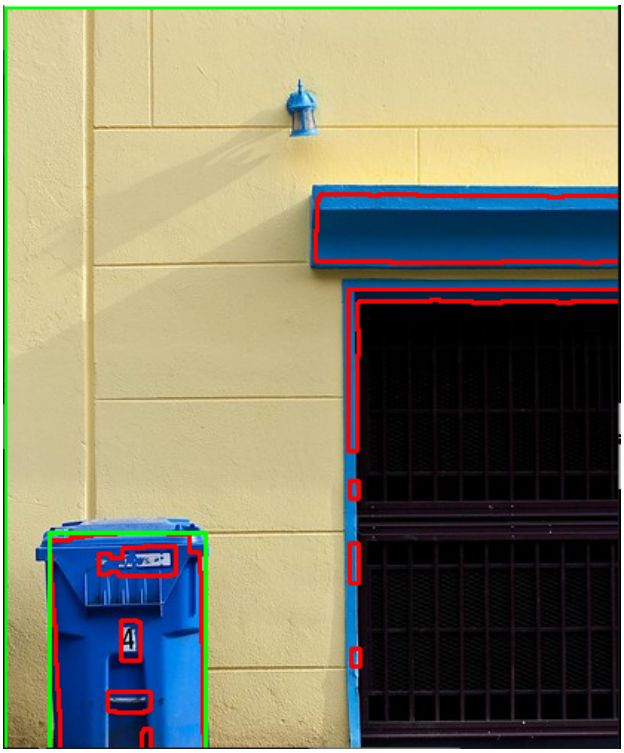

Fig. 2: 0062.jpg Bounding Box

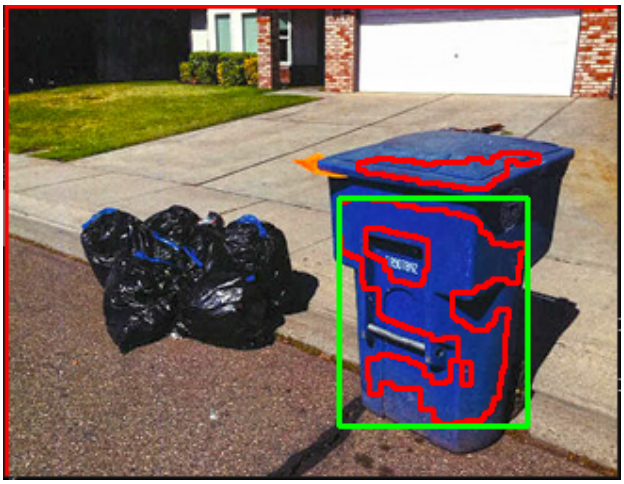

Fig. 3: 0063.jpg Bounding Box

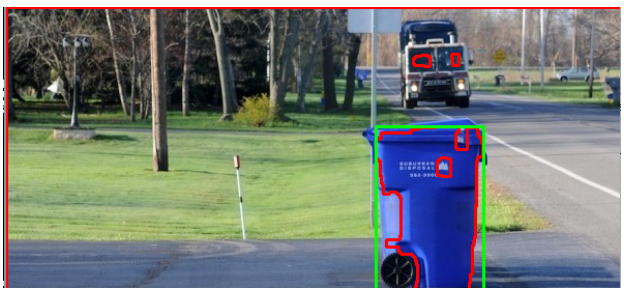

Fig. 4: 0064.jpg Bounding Box

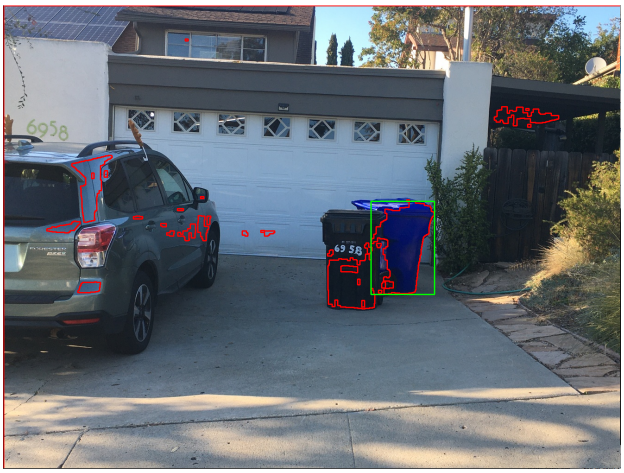

Fig. 5: 0065.jpg Bounding Box

Fig. 6: 0066.jpg Bounding Box

Fig. 7: 0067.jpg Bounding Box

Fig. 8: 0068.jpg Bounding Box

Fig. 9: 0069.jpg Bounding Box

Fig. 10: 0070.jpg Bounding Box

Fig. 11: 0061.jpg Binary Mask

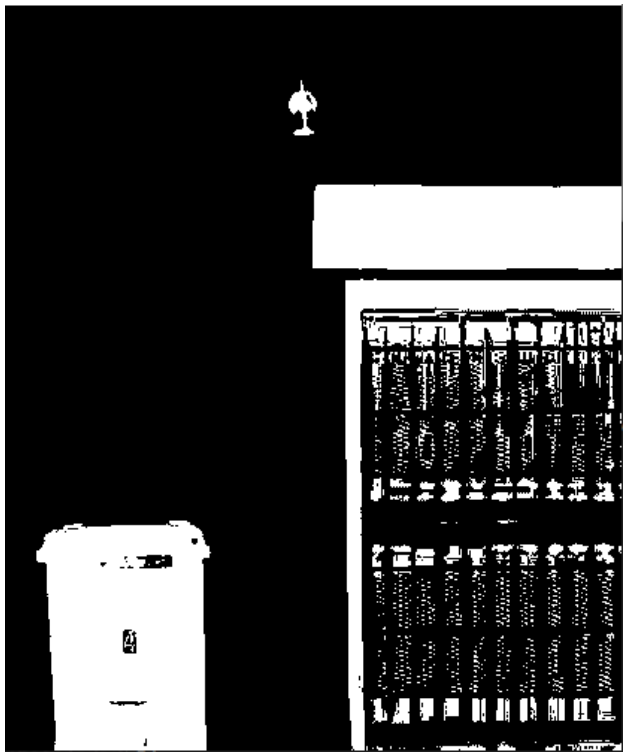

Fig. 12: 0062.jpg Binary Mask

Fig. 13: 0063.jpg Binary Mask

Fig. 14: 0064.jpg Binary Mask

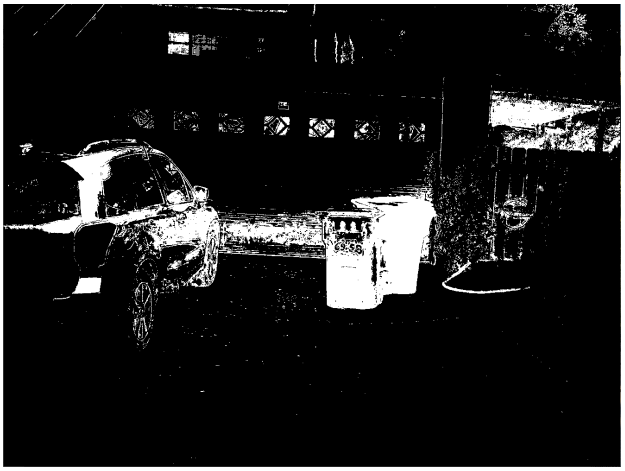

Fig. 15: 0065.jpg Binary Mask

Fig. 16: 0066.jpg Binary Mask

Fig. 17: 0067.jpg Binary Mask

Fig. 18: 0068.jpg Binary Mask

Fig. 19: 0061.jpg Erosion

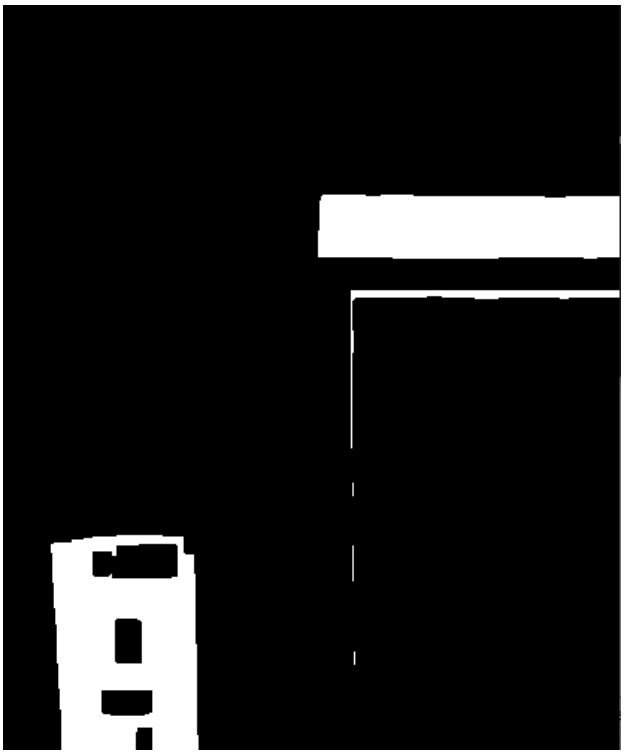

Fig. 20: 0062.jpg Erosion

Fig. 21: 0061.jpg Dilation

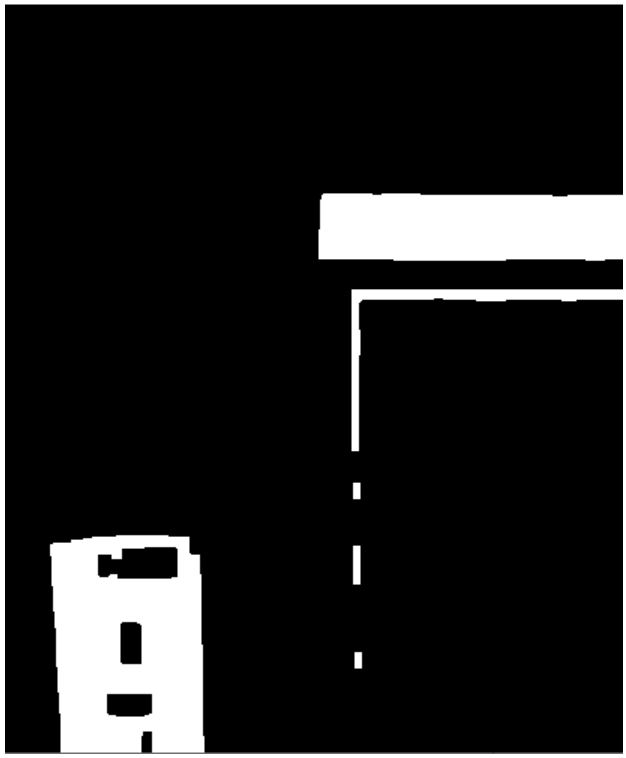

Fig. 22: 0062.jpg Dilation