Blue Recycling Bin Detection using Multi-class Logistic Regression

1st Kai Chuen Tan
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, United States
kctan@ucsd.edu, A59011493

***Abstract*—In the field of computer vision, color is one of the most important features of a digital image. Color not only provides crucial information that helps computers to gain high-level understanding from digital images and videos but also enables computers to emulate repetitive human tasks, for instance, identifying and blocking inappropriate image and video content on Facebook. This paper presents a supervised machine learning model to detect and locate one or more blue recycling bins in images. To recognize blue recycling bins, the One-vs-All Logistic Regression approach is implemented to classify recycling-bin blue and other classes of colors of an image pixel for the image segmentation process. With the segmented image, recycling bin shape statistics are applied using Python's sci-kit image package functions to detect and locate blue recycling bins in the image after filtering the segmented image using Python's OpenCV package morphological operations. If a blue recycling bin is recognized, a red bounding box will be drawn around it in the image. The effectiveness of the multi-class logistic regression approach for pixel colors classification and blue recycling bin detection is verified with the Gradescope Autograder.**

***Index Terms*—Machine Learning, Supervised Learning, Multi-class Logistic Regression, Robotics, Computer Vision, Object Recognition.**

I. INTRODUCTION

The recent COVID-19 pandemic reveals how the labor shortage issue not only badly disrupted the logistic and global supply chain but also negatively impacted the economy of the United States. For instance, according to the American Trucking Association, approximately 71% of the U.S. economy's products are transported by truck drivers across the United States, and the United States (U.S.) Trucking and Freight Transportation industry that worth USD 791.7 billion is facing a driver shortage problem; it is also estimated to reach a historic high of just over 80,000 drivers in the year 2021 due to the COVID-19 pandemic [1]. To solve both labor shortage and global supply chain issues, fully autonomous semi-trucks and intelligent robots are developed and introduced to the market by robotics companies like Tesla, TuSimple, and Boston Dynamics to replace the labor workforce and automate human tasks. To allow autonomous vehicles, drones, and robots to navigate in an open space, it is crucial to have the most fundamental ability to recognize and locate objects in their surroundings using cameras and light detection and ranging (LIDAR) sensors.

Using deep learning models to detect objects has been developed over the years, and the models' accuracy of recognizing an object has improved significantly too. The work in [2] discusses the use of deformable and multi-scaled convolution neural networks to detect tiny and deformed target objects accurately in still images. However, there is a trade-off between object detection accuracy and computational efficiency using deep learning models due to the huge amount of training data required. Many real-world applications require robots to detect and locate objects accurately in real-time. For instance, autonomous vehicles and drones must be able to recognize obstacles, pedestrians, and other objects in real-time and react instantly to prevent any collision. Hence, both computational efficiency and accuracy are equally important for the object detection algorithm in order to apply it in real-world applications safely.

This project aims to implement an algorithm that can detect blue recycling bins accurately in still images without the need for gigantic amounts of data. One-vs-All Logistic Regression approach is proposed to classify pixel colors, which are recycling bin blue, green, black, white, etc. for the image segmentation process. Image filtering is also added to the algorithm to filter out noise or misclassified pixel labels in the segmented image using Python's OpenCV package morphological operations including erosion and dilation. Then, recycling bin shape statistics are applied using Python's sci-kit image package functions to search and locate blue recycling bins in an image. If a blue recycling bin is detected, a red bounding box will be drawn around it in the original image. Overall, the developed algorithm with the supervised machine learning model is invariant to lighting; hence, it is able to detect blue recycling bins in numerous environmental conditions.

The paper is organized as follows. §II presents the problem formulation. §III describes the image pixel classification method and bin detection algorithm Lastly, §IV presents the training, validation, test results, and the discussion of the results.

II. PROBLEM FORMULATION

The main objective of the object detection algorithm is to detect and locate blue recycling bins in a test image that is in BGR color space format, denoted as $\boldsymbol{M}_{\boldsymbol{test}} \in \mathbb{N}_0^{H\times W\times C}$, where H is the height of the image, W is the width of the image, and C is the number of channels in the image, which is $C = 3$. The values of each channel are a non-negative integer ranging from 0 to 255. The object detection algorithm is split into three main processes, which are pixel color classification, image segmentation, and recycling bin shape recognition.

A. Pixel Classification

The goal of the pixel classification is to differentiate recycling-bin blue from other color classes including green, black, and white. The pixel color classification requires three main inputs to train and predict each pixel color of the test image, which are the color pixels training dataset, denoted as $\boldsymbol{X_{train}} \in \mathbb{N}_0^{T\times N}$, where T and N denoted as number of pixels in a training image and number of features, respectively, the pixel color's training label, denoted as $\boldsymbol{y_{train}} \in \mathbb{N}^{T\times 1}$, and the validation dataset that is converted to a feature matrix from $\boldsymbol{M_{test}}$, denoted as $\boldsymbol{X_{test}} \in \mathbb{N}_0^{(H\cdot W)\times N}$. The rows of $\boldsymbol{X_{train}}$, denoted as $\boldsymbol{x_{train}}$, and $\boldsymbol{X_{test}}$, denoted as $\boldsymbol{x_{test}}$, represent the feature vector of each pixel value, and the columns of $\boldsymbol{X_{train}}$ and $\boldsymbol{X_{test}}$ represent each channel's values. The multi-class logistic regression classifier model parameters (i.e., the weight matrix with a bias term), denoted as $\boldsymbol{\Theta} = \{B, G\} \in \mathbb{R}^{L\times(N+1)}$, where $B \in \mathbb{R}^{L\times 1}$ is the bias term vector, $G \in \mathbb{R}^{L\times N}$, and L is the total number of color pixel classes, are needed to be train to output the predicted color classes of the $H \cdot W$ number of pixels in the test image, denoted as $\boldsymbol{y_{predict}} \in \mathbb{N}^{(H\cdot W)\times 1}$.

B. Image Segmentation and Recycling Bin Shape Recognition

The image segmentation process aims to segment out the recycling-bin blue class from other color classes based on the input $\boldsymbol{y_{predict}}$ by outputting a binary mask image, denoted as $\boldsymbol{K} \in \mathbb{N}_0^{H\times W}$. The elements 1 and 0 in the matrix $\boldsymbol{K}$ represent white and black pixels, respectively. Element 1 also indicates that the pixel color belongs to the recycling-bin blue color class, and element 0 indicates that the pixel color belongs to other color classes. Next, the objective of the recycling bin shape recognition is to search, locate blue recycling bins, and draw a red bounding box around each blue recycling bin in the segmented image $\boldsymbol{K}$ input. The final output of the bin detection is a list of lists of bounding boxes' top left x-y coordinates, and bottom-right x-y coordinates, for instance, $[x_{Top_Left}, y_{Top_Left}, x_{Bottom_Right}, y_{Bottom_Right}] \in \mathbb{Z}^{1\times 4}$ if there is at least one blue recycling bin; otherwise, an empty list is returned.

III. Technical Approach

A. Color Pixel Classification

1) Data Pre-processing

In this project, a given set of color images is split into two subsets, which are the training dataset, denoted as $\boldsymbol{D_{train}} = \boldsymbol{D_{train}}_{\{1,2,\ldots,q\}} = \{\boldsymbol{X_{train,\ i}}, \boldsymbol{y_{train,\ i}}\}_{i=1}^{q} \in \mathbb{N}_0^{T\times(N+1)}$, where q is the total number of training images, and the validation dataset, $\boldsymbol{M_{test}} = \boldsymbol{M_{test}}_{\{1,2,\ldots,r\}}$, where r is the total number of test images. The training dataset is used to train the implemented color pixel classifier that uses the One-vs-All Logistic Regression model and to optimize the model parameters, $\boldsymbol{\Theta}$, and the validation dataset is used to check the accuracy of the prediction from the trained multi-class logistic regression model.

Before training the multi-class logistic regression model, $\boldsymbol{M_{test}}$ 3D-array must be converted to a 2D-array, which is $\boldsymbol{X_{test}}$. To convert from $\boldsymbol{M_{test}}$ to $\boldsymbol{X_{test}}$, first, $\boldsymbol{M_{test}}$ image must be converted from BGR format to four different color space formats, which are RGB, HSV, LAB, and YCrCb, denoted as $\boldsymbol{M_{test,\ RGB}} \in \mathbb{N}_0^{H\times W\times C}$, $\boldsymbol{M_{test,\ HSV}} \in \mathbb{N}_0^{H\times W\times C}$, $\boldsymbol{M_{test,\ LAB}} \in \mathbb{N}_0^{H\times W\times C}$, and $\boldsymbol{M_{test,\ YCrCb}} \in \mathbb{N}_0^{H\times W\times C}$, respectively. Then, $\boldsymbol{M_{test,\ RGB}}$, $\boldsymbol{M_{test,\ HSV}}$, $\boldsymbol{M_{test,\ LAB}}$, and $\boldsymbol{M_{test,\ YCrCb}}$ are reshaped to 2D-arrays, which are $\boldsymbol{X_{test,\ RGB}} \in \mathbb{N}_0^{(H\cdot W)\times N}$, $\boldsymbol{X_{test,\ HSV}} \in \mathbb{N}_0^{(H\cdot W)\times N}$, $\boldsymbol{X_{test,\ LAB}} \in \mathbb{N}_0^{(H\cdot W)\times N}$, and $\boldsymbol{X_{test,\ YCrCb}} \in \mathbb{N}_0^{(H\cdot W)\times N}$, respectively. Next, $\boldsymbol{X_{test,\ RGB}}, \boldsymbol{X_{test,\ HSV}}, \boldsymbol{X_{test,\ LAB}}$, and $\boldsymbol{X_{test,\ YCrCb}}$ 2D-arrays are stacked horizontally to the right to form $\boldsymbol{X_{test}}$. $\boldsymbol{X_{train}}$ and $\boldsymbol{y_{train}}$ must be extracted from $\boldsymbol{D_{train}}$ before passing them to the pixel classifier function. Both $\boldsymbol{X_{train}}$ and $\boldsymbol{X_{test}}$ have 12 features (i.e., $N = 12$), which are Red, Green, Blue, Hue, Saturation, Value, Lightness, Red/Green Value, Blue/Yellow Value, Brightness (Luma), Blue Minus Luma, and Red Minus Luma. According to the work in [3], adding more different color spaces to the features of a color pixel value significantly improves the color pixel classification accuracy. Hence, in this project, four different color spaces are used, which are RGB, HSV, LAB, and YCrCb color spaces.

2) One-vs-All Logistic Regression

One-vs-All Logistic Regression model is considered as one of the supervised machine learning algorithms, and it is a discriminative model that made up of L different binary classifiers to classify L distinct classes. In this project, the multi-class logistic regression model is used to classify multiple classes of color pixels in test images by using a given training dataset, $\boldsymbol{D_{train}}$ to optimize the $\boldsymbol{\Theta}$ parameters. For each color label, denoted as $l \in \{1, 2, ..., L\}$ in the $\boldsymbol{y_{train}}$, the binary logistic classifier relabels $\boldsymbol{y_{train}}$ to binary labels, $\boldsymbol{y_{bin}}$ as presented below:

$$\boldsymbol{y_{bin}} = \begin{cases} 1 & \text{if } \boldsymbol{y_{train,\ i}} = l \\ 0 & \text{if } \boldsymbol{y_{train,\ i}} \neq l \end{cases} \tag{1}$$

The binary logistic regression is also a discriminative model that selects model $p(\boldsymbol{y_{bin}}|\boldsymbol{X_{bias,\ train}}, \theta)$ with parameters $\boldsymbol{\Theta}$ and the independent and identically distributed **(iid)** training dataset, $\boldsymbol{D_{train}}$ to approximate the unknown label-generating probability density function (p.d.f) for binary labels, $\boldsymbol{y_{bin}} \in \{0, 1\}^T$ with a logistic sigmoid function as formulated below:

$$p(\boldsymbol{y_{bin}}|\boldsymbol{X_{bias,\ train}}, \theta) = \prod_{i=1}^{T} \sigma(\boldsymbol{y_{bin,\ i}} \boldsymbol{x}^T_{\boldsymbol{bias,\ train,\ i}} \theta) \tag{2}$$

$$p(\boldsymbol{y_{bin}}|\boldsymbol{X_{bias,\ train}}, \theta) = \prod_{i=1}^{T} \frac{1}{1 + exp(-\boldsymbol{y_{bin,\ i}} \boldsymbol{x}^T_{\boldsymbol{bias,\ train,\ i}} \theta)} \tag{3}$$

where $\boldsymbol{X_{bias,\ train}} \in \mathbb{N}_0^{T\times(N+1)}$ is the $\boldsymbol{X_{train}}$ with the bias of one vector added to the first column of $\boldsymbol{X_{train}}$, $\boldsymbol{x_{bias,\ train}} \in \mathbb{N}_0^{(N+1)}$ is the rows of $\boldsymbol{X_{bias,\ train}}$, $\theta \in \mathbb{R}^{(N+1)}$ is the rows of $\boldsymbol{\Theta}$, and σ is the sigmoid function. The gradient descent method is applied for the training step to optimize the parameters θ with the maximum likelihood estimation (MLE) as shown in the following equation:

$$\theta_{MLE}^{(t+1)} = \theta_{MLE}^{(t)} + \alpha^{(t)} \sum_{i=1}^{T} (1 - \sigma(\boldsymbol{y_{bin,\ i}} \boldsymbol{x_{bias,\ train,\ i}^T} \theta_{MLE}^{(t)})) \boldsymbol{y_{bin,\ i}} \boldsymbol{x_{bias,\ train,\ i}} \tag{4}$$

where α is the learning rate. To predict the color pixel label of the given test example with an added bias of one, $\boldsymbol{x_*} \in \mathbb{R}^{(N+1)}$, the optimized parameters $\theta_*^l \in \mathbb{R}^{(N+1)}$ in the color pixel class l are used as shown in the equation below:

$$\boldsymbol{y_{predict,\ i}} = \arg\max_{l \in 1,\ldots,L} \boldsymbol{x_*^T} \theta_*^l \tag{5}$$

where $\boldsymbol{y_{predict}} = \{\boldsymbol{y_{predict,\ i}}\}_{i=1}^T$. The multi-class logistic regression model's classifier function $h : \mathbb{R}^{N+1} \to \mathbb{R}$ will be able to assign a pixel color label to a given data point, $\boldsymbol{x_{bias,\ train}}$ and minimize loss, $min_h Loss_{0-1}(h)$.

Algorithm 1 Color Pixel Classification using One-vs-All Logistic Regression Model

```
Function Pixel_Classifier(X_train, y_train, X_test):
    // Initialize classifier parameters
    MLR.initialize(α, max_iters, err_tol, bias, Θ)
    X_bias,train ← Add_Bias_Term(X_train, bias)
    T, (N + 1) ← Get_Size(X_bias,train)
    color_class_list ← Get_Unique_Labels(y_train)
    // Train the model
    for l in color_class_list do
        // Only the current color class labels as 1;
           otherwise, 0
        y_bin ← Convert2Binary_Labels(y_train)
        θ ← Initialize_Vector_Zeros_Elements(N + 1)
        for idx = 0 to max_iters do
            y_pred ← Sigmoid(X_train, θ)
            G ← Gradient_Descent(T, y_bin, X_bias,train, y_pred)
            θ_previous ← θ
            θ ← Update_Weight(α, G)
            // Early Stopping to Prevent Over-fitting
            if ||θ_previous − θ|| ≤ err_tol then
                break
        Θ ← Append2List(θ^l_*)
    // Predict the validation set
    X_bias,test ← Add_Bias_Term(bias, X_test)
    y_predict ← Predict_Labels(X_bias,test, Θ)
    // Return final predicted color pixels
    return y_predict
```

B. Image Segmentation and Blue Recycling Bin Detection

1) Image Segmentation

After classifying the color pixel labels in the test image, the image segmentation can be done by inputting $\boldsymbol{y_{predict}}$, and the test image, $\boldsymbol{M_{test}}$ to the image segmentation function. The mask image, $\boldsymbol{K}$ is created at the beginning of the image segmentation process, and the dimensions of the mask image, $\boldsymbol{K}$ is the same as the dimensions of the test image, $\boldsymbol{M_{test}}$. The predicted labels, $\boldsymbol{y_{predict}}$ is reshaped to a $H \times W$ 2D-array, denoted as $\boldsymbol{y\_2D_{predict}}$. Then, each pixel of the test image is looped through with $\boldsymbol{y\_2D_{predict}}$ to check which pixel color is belongs to the recycling-bin blue color. If a particular pixel color in the test image is a recycling-bin blue, the mask image's pixel value at the same coordinates as the test image' coordinate will be assigned as a white pixel. Finally, the white region in the mask image is a blue recycling bin, and the rest is black; hence, the final binary mask, $\boldsymbol{K}$ is returned by the image segmentation as show in the **Algorithm 2** below:

Algorithm 2 Image Segmentation Algorithm

```
Function Image_Segmentation(y_predict, M_test):
    // Create a binary mask
    H, W, C ← Get_Size(M_test)
    K ← Initialize_3D_Arrays_Zeros(H, W, C)
    // Reshape y_predict to 2D array
    y_2D_predict ← Reshape_Array(H, W)
    // Loop through each pixel of the test image
    for h = 0 to H do
        for w = 0 to W do
            // If the color pixel is a recycling bin
               blue
            if y_2D_predict[h, w] == 1 then
                // The mask pixel turns white at the
                   same position as the test image
                K[h, w, :] = 1
    return K
```

2) Blue Recycling Bin Detection

Once the image segmentation process is completed, the mask image, $\boldsymbol{K}$ is passed to the bin detection function to search and locate the shape of a recycling bin. However, before recognizing the shape of a recycling bin, the binary mask image has noises or a small amount of misclassified recycling-bin blue that are needed to be filtered out. To eliminate noises in the binary mask image, morphological operations are implemented using the dilation and erosion built-in function from Python's OpenCV library. After the image filtering process, the built-in "labels" and "regionprops" functions from Python's Scikit-Image Package are applied to locate connected white pixels in the mask image and to get the dimensions of the white regions in the mask image. With the dimensions of the white regions, the blue recycling bin can be detected by matching the height-to-width ratio, r_{bin}, of the recycling bin's similarity score and the area of the recycling bin's, A_{bin} similarity score. If both white regions' height-to-width ratio and area match the recycling bin's shape, the bin detection function will output a list of bounding box's top-left and bottom-right coordinates lists; if there is none, an empty list will be the output. With the bounding box list, bb_{list}. Finally, with the bounding box list, the bounding box can be drawn around the blue recycling bin in the test image.

Algorithm 3 Blue Recycling Bin Detection

Function Bin_Detection(K):

// Filter out the noises in the binary mask image

$K \leftarrow cv2.erode(K, kernel\_shape, iteration)$

$K \leftarrow cv2.dilate(K, kernel\_shape, iteration)$

// Get the mask image area.

$A_{mask} \leftarrow Get\_Area(K)$

// Labeled connected regions in the mask image.

$label\_img \leftarrow skimage.measure.label(K)$

// Obtain the white region shape properties from the label_img.

$regions \leftarrow skimage.measure.regionprops(label\_img)$

// Initialize bounding box list

$bb_{list} \leftarrow Initialize\_Empty\_List()$

// Loop through white regions and extract the shape properties

for *props* ***in*** *regions* **do**

$A_{bin} \leftarrow props.area$

if $0.55{*}A_{mask} > A_{bin} > 0.006{*}A_{mask}$ **then**

// Get bounding box top-left and bottom-right coordinates

$bb\_coordinates \leftarrow props.bbox$

$r_{bin} \leftarrow Calculate\_Ratio(bb\_coordinates)$

// Check the height-to-width ratio

if $1.0 < r_{bin} < 2.55$ **then**

// Get the bounding box coordinates

$bb_{list}.append(bb\_coordinates)$

return bb_{list}

3) Color Pixel Data Collection

In this project, only 38051 color pixels are collected, and the color pixels are collected from the first two training images and one validation image, which is "0068.jpg". Recycling-bin blue color pixel label is 1, green color pixel label is 12, black color pixel label is 14, and white color pixel label is 15. To better train the multi-class logistic regression model, it is important to collect the optimum amount of data and to minimize human error during the data collection process. Therefore, collecting a huge amount of data could train the machine learning model to misclassify color pixels due to larger human error of mislabeling colors during the data collection process. The data collection methodology is to collect samples from one image, train with the collected samples, and predict the current image and other images to check the accuracy performance of the model. If the model misclassifies recycling-bin blue, the data collection process is repeated with other images until the model can classify recycling-bin blue in most of the training images.

IV. Results and Discussion

A. Pixel Classification Accuracy Performance

The pixel classifier uses the One-vs-All Logistic Regression model to predict a color pixel label from 3 different classes of color, which are red, green, and blue. The red color is labeled as 1; the green color is labeled as 2; the blue color is labeled as 3.

TABLE I: Color Pixel Classification Performance using One-vs-All Logistic Regression Model

Color	*Accuracy [%]*	*Execution Time [s]*	*Number of Predicted Pixels*
Red	100.0	2.134	82
Green	100.0	2.028	68
Blue	100.0	2.072	83

Table I above presents that the implemented One-Vs-All Logistic Regression model achieves an accuracy of 100% on the validation solid color images. The examples of the pixel classification's validation dataset are shown in Figure 1. The multi-class logistic regression model took approximately 2 seconds to predict 68 to 83 color pixels without misclassifying any color pixels.

Fig. 1: Pixel Classification Validation Dataset Samples

TABLE II: Optimized Weight and Bias Term Parameters

Color Class	*Bias Weight*	*Red Weight*	*Green Weight*	*Blue Weight*
Red	-1.414	8.366	-4.408	-4.232
Green	-1.1557	-4.808	8.143	-4.277
Blue	-1.178	-4.802	-4.291	8.095

Table II shows the optimized weights for each class, θ_*^l after the multinomial logistic regression model is trained in the RGB color space with the pixel classification's training dataset. The examples of the pixel classification's training dataset are shown in Figure 2. According to the optimized weight parameters, $\boldsymbol{\Theta}$ as presented in Table II, the weights of its color class have a significant higher value than the weight of the other color classes. Hence, when the supervised machine learning model encounters a color pixel that is similar to a certain color pixel class, the model will generate a higher probability value for that particular color class.

Fig. 2: Pixel Classification Training Dataset Samples

B. Image Segmentation and Blue Recycling Bin Detection

Since using the One-vs-All Logistic Regression model as a color pixel classifier gives 100 % accuracy on all the different color classes' validation dataset, the model is selected to classify multiple color pixel classes including recycling-bin blue, green, black, and white on ten validation images. Recycling-bin blue is labeled as 1; green is labeled as 12; black is labeled as 14; white is labeled as 15. The Multinomial Logistic Regression model is trained in a combination of 4 different color spaces, which are RGB, HSV, LAB, and YCrCb color spaces to obtain the best accuracy performance; as mentioned in §III, the work in [3] verified that adding more color space features to the training dataset significantly improves the color pixel classification accuracy.

The segmented mask images and validation images with red bounding boxes are presented as follows:

Fig. 3: Validation Image 61

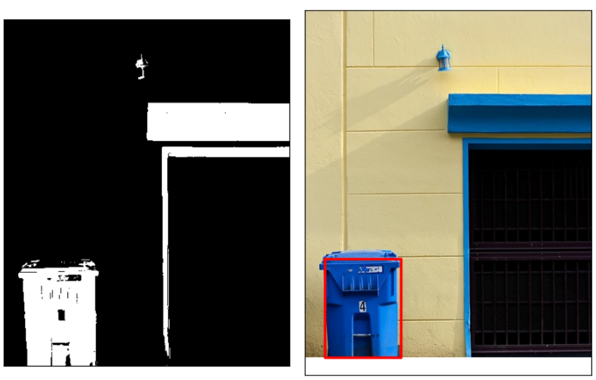

Fig. 4: Validation Image 62

Fig. 5: Validation Image 63

Fig. 6: Validation Image 64

Fig. 7: Validation Image 65

Fig. 8: Validation Image 66

Fig. 9: Validation Image 67

Fig. 10: Validation Image 68

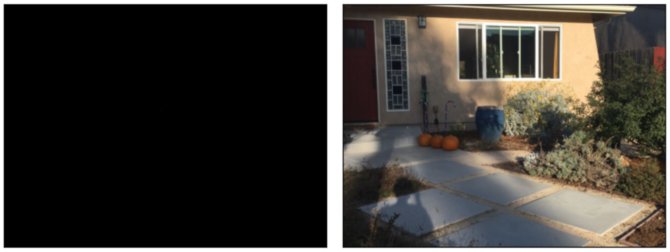

Fig. 11: Validation Image 69

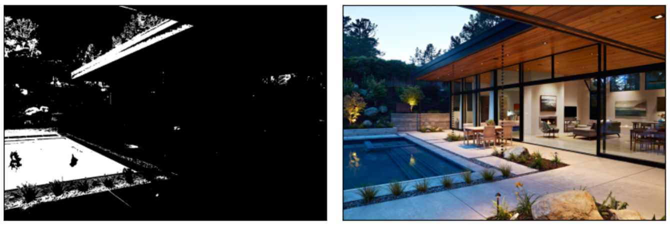

Fig. 12: Validation Image 70

TABLE III: Comparison between Validation Bounding Boxes and Predicted Bounding Boxes

Validation Image Number	***Validation Bounding Boxes***	***Predicted Bounding Boxes***
61	[[182, 101, 313, 295]]	[[199, 154, 314, 288]]
62	[[25, 347, 133, 497]]	[[28, 357, 138, 499]]
63	[[168, 64, 300, 239]]	[[173, 96, 277, 233]]
64	[[349, 104, 467, 264]]	[[361, 110, 466, 272]]
65	[[762, 416, 924, 622]]	[[802, 415, 937, 629]]
66	[]	[]
67	[[578, 305, 706, 504], [711, 305, 830, 509]]	[[586, 306, 703, 512], [704, 306, 830, 515]]
68	[]	[]
69	[]	[]
70	[]	[]

Figure 3 to Figure 12 above proves that the One-Vs-All Logistic Regression model can classify recycling-bin blue pixel and other color pixels accurately on the ten validation images. As a result, the segmented mask images clearly show the recycling bin blues' location without much noise in the background. However, several mask images show some white regions although there is no blue recycling bin in the images; for example, in Figure 10, the dark blue shade that is similar to the recycling-bin blue color is classified as the recycling-bin blue color, and in Figure 12, the swimming pool's blue color is also misclassified as the recycling-bin blue.

Table III presents that using morphological operations (e.g., dilation, and erosion) to filter out segmented mask images' noise and the recycling bin shape statistics (i.e., recycling bin shape area, and recycling bin height-to-width ratio) to decide if there are any recycling bins in the segmented mask images is an effective algorithm implementation to detect and locate blue recycling bins because the algorithm manages to achieve 100 % accuracy on all the local validation images and score an average of 97.5 % accuracy in the Gradescope Autograder's validation images.

The final parameters used by the One-vs-All Logistic Regression classification model are listed in Table IV.

References

[1] N. Goodkind, "The supply chain issue is really a truck driver shortage," Fortune, 28-Oct-2021. [Online]. Available: https://fortune.com/2021/10/28/truck-driver-shortage-supply-chain-80000/. [Accessed: 04-Feb-2022].

[2] Cao, D., Chen, Z. Gao, L. An improved object detection algorithm based on multi-scaled and deformable convolutional neural networks. Hum. Cent. Comput. Inf. Sci. 10, 14 (2020). https://doi.org/10.1186/s13673-020-00219-9

[3] S. N. Gowda and C. Yuan, "Colornet: Investigating the importance of color spaces for image classification," Computer Vision – ACCV 2018, pp. 581–596, 2019.

TABLE IV: Optimized Weights and Bias Parameters for the Blue Recycling Bin Detection

Color Class	θ^*_{Bias}	θ^*_{Red}	θ^*_{Green}	θ^*_{Blue}	θ^*_H	θ^*_S	θ^*_V	θ^*_L	θ^*_A	θ^*_B	θ^*_Y	θ^*_{Cr}	θ^*_{Cb}
Recycling-bin Blue	-2.168	-3.470	-2.363	2.880	0.5582	3.612	1.900	-1.626	0.6361	-3.843	-2.099	-2.071	1.719
Green	0.2812	0.6201	2.119	-5.679	-1.733	3.491	-1.554	1.279	-2.462	4.146	0.7758	0.02785	-3.515
Black	1.508	-0.518	-1.932	-1.678	-0.843	-4.054	-2.743	-1.930	1.148	0.6973	-1.478	1.444	0.6494
White	-3.044	3.279	2.514	2.941	1.414	-4.172	1.550	2.345	-1.769	-1.782	2.794	-1.179	-1.441

Name: Kai Chuen Tan
PID: A59011493
Course: ECE 276A
Assignment #: HW1
Date: 19th Jan 2021
Professor: Dr. Atanasov

Problem 4

$X \sim \mathcal{N}(\underbrace{\mu}_{\text{unknown}}, \overbrace{\sigma^2}^{\text{known}})$

$\{x_i\}_{i=1}^{n}$, where n is an independent samples obtained from X.

a) Formulate a maximum likelihood estimation (MLE) problem to determine the unknown mean of X.

Apply Likelihood Function: $L(x_1,\dots,x_n|\mu,\sigma^2) = \prod_{i=1}^{n} f_x(x_i|\mu,\sigma^2)$; $f_x(x_i|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x_i-\mu)^2}{2\sigma^2}}$

$$= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x_i-\mu)^2}{2\sigma^2}}$$

$$L(x_1,\dots,x_n|\mu,\sigma^2) = (2\pi\sigma^2)^{-\frac{n}{2}} e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2}$$

Take the natural logarithm of the likelihood function,

$$l(x_1,\dots,x_n|\mu,\sigma^2) = \ln(L(x_1,\dots,x_n|\mu,\sigma^2))$$

$$= \ln\left((2\pi\sigma^2)^{-\frac{n}{2}} e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2}\right)$$

$$= -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2$$

$$= -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2$$

To $\max_{\mu,\sigma^2} l(x_1,\dots,x_n|\mu,\sigma^2)$, $\frac{\partial}{\partial\sigma^2} l(x_1,\dots,x_n|\mu,\sigma^2) = 0$

$$\frac{\partial}{\partial\sigma^2}\left(-\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2\right) = 0$$

$$= -\frac{n}{2\sigma^2} + \left(\frac{1}{2}\sum_{i=1}^{n}(x_i-\mu)^2\right)\left(\frac{1}{\sigma^4}\right)$$

$$\boxed{\frac{\partial l}{\partial\sigma^2} = \frac{1}{2\sigma^2}\left[\frac{1}{\sigma^2}\left[\sum_{i=1}^{n}(x_i-\mu)^2\right] - n\right]} = 0$$

$$\boxed{\hat{\sigma}^2_{MLE} = \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2}$$

$$\frac{\partial}{\partial\mu} l(\mu,\sigma^2|x_1,\dots,x_n) = 0 \Rightarrow \frac{\partial}{\partial\mu}\left(-\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2\right)$$

$$\boxed{\frac{\partial l}{\partial\mu} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i-\mu)} = 0$$

Name: Kai Chuen Tan
PID: A59011493
Course: ECE 276A
Assignment #: HW 1
Date: 19th Jan 2021
Professor: Dr. Atanasov

Problem 4

b.) Solve the problem in part (a) to obtain maximum likelihood estimate $\hat{\mu}_{MLE}$

$$\frac{\partial l}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i - \mu) = 0$$

$$\left(\sum_{i=1}^{n} x_i - n\mu\right) = 0$$

$$\boxed{\hat{\mu}_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i}$$

c.) Formulate a maximum a posteriori (MAP) problem to determine the unknown mean of X. Suppose that a prior Gaussian distribution $\mathcal{N}(\mu_o, \sigma_o^2)$ with known μ_o and σ_o^2 is available.

Given that $f(\mu) = \frac{1}{\sqrt{2\pi\sigma_o^2}} e^{-\frac{(\mu-\mu_o)^2}{2\sigma_o^2}}$ → prior probability function, apply Bayes' Rule:

$$f(\mu \mid x_1, \dots, x_n) = \frac{f(x_1, \dots, x_n \mid \mu)\, f(\mu)}{h(x_1, \dots, x_n)} \quad ; \quad H = h(x_1, \dots, x_n)$$

$$= \frac{\left[(2\pi\sigma^2)^{-\frac{n}{2}} e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2}\right](2\pi\sigma_o^2)^{-\frac{1}{2}} e^{-\frac{(\mu-\mu_o)^2}{2\sigma_o^2}}}{H}$$

To $\max_{\mu}$ $\ln(f(\mu \mid x_1, \dots, x_n)) = \left(\sum_{i=1}^{n} -\ln(\sqrt{2\pi\sigma^2}) - \frac{(x_i-\mu)^2}{2\sigma^2}\right) - \ln(\sqrt{2\pi\sigma_o^2}) - \frac{(\mu-\mu_o)^2}{2\sigma_o^2}$

$$\boxed{\frac{\partial \ln(f(\mu \mid x_1, \dots, x_n))}{\partial \mu} = \left(\sum_{i=1}^{n}\frac{x_i - \mu}{\sigma^2}\right) - \frac{\mu - \mu_o}{\sigma_o^2}} = 0$$

Name: Kai Chuen Tan
PID: A59011493
Course: ECE 276A
Assignment # : HW1
Date : 20th Jan 2021
Professor: Dr. Atanasov

Problem 4

d.) Solve the problem in part (c) to obtain the MAP estimate $\hat{\mu}_{MAP}$.

$$\frac{\partial \ln(f(\mu \mid x_1, \dots, x_n))}{\partial \mu} = 0$$

$$\left(\sum_{i=1}^{n} \frac{x_i - \mu}{\sigma^2}\right) - \frac{\mu - \mu_0}{\sigma_0^2} = 0$$

$$\frac{\mu - \mu_0}{\sigma_0^2} = \sum_{i=1}^{n} \frac{x_i - \mu}{\sigma^2}$$

$$= \frac{1}{\sigma^2}\left[\left(\sum_{i=1}^{n} x_i\right) - n\mu\right]$$

$$\frac{(\sigma^2 + n\sigma_0^2)\mu}{\cancel{\sigma^2\sigma_0^2}} = \frac{\left(\sigma_0^2 \sum_{i=1}^{n} x_i\right) + \sigma^2\mu}{\cancel{\sigma^2\sigma_0^2}}$$

$$\boxed{\hat{\mu}_{MAP} = \frac{\left(\sigma_0^2 \sum_{i=1}^{n} x_i\right) + \sigma^2\mu}{\sigma^2 + n\sigma_0^2}}$$