Pixel Color Classification for Recycling Bin Detection

Allen Zeng
UCSD
azeng@ucsd.edu

Abstract

This paper considers the use of a K-ary logistic regression model for predicting which pixels in an image correspond to a blue recycling bin. After pixels are classified into certain classes of colors, including *recycling-bin blue* and *non-recycling-bin blue*, a segmentation mask of the recycling bins are produced for each image. While the initial segmentation mask is noisy, morphological image processing techniques can be applied to refine the mask. The efficacy of the K-ary logistic regression model is tested on the RGB, HSV, and LAB colorspaces.

1 Introduction

Autonomous vehicles developed in the near future offer many benefits to society, one of which is the automated collection of trash, recycling, etc. An autonomous vehicle must be capable of identifying which objects in its environment such a trash or recycling bin. These bins can be detected through the use of cameras and computer vision techniques. In this paper, one such technique for identifying blue recycling bins in images is addressed. Although blue recycling bins are specifically targeted here, this technique may be extended to detect other colors of bins as well.

Using a dataset of images containing and excluding blue recycling bins, K-ary logistic regression is utilized in order to segment the blue pixels in the images. The noisy segmentation masks are then refined using morphological image processing techniques, and bounding boxes corresponding to the bins are extracted.

2 Problem Formulation

The method of logistic regression with K-classes can be formulated as follows, using the notation from [1]. For a dataset $\{x_1, x_2, ..., x_n\}, x_i \in \mathbb{R}^d$ of n input examples each with dimensional d, the logistic regression model approximates an unknown label-generating probability density function (pdf) for $y \in \{1, ..., k\}^n$ with a softmax function

$$p(y|X, W) = \prod_{i=1}^{n} e_{y_i}^T S(Wx_i) = \prod_{i=1}^{n} e_{y_i}^T \frac{\exp(Wx_i)}{\sum_{j=1}^{n} \exp(Wx_j)}$$

where $W \in \mathbb{R}^{k \times d}$ are weight parameters, and e_i is the i-th standard basis vector.

For the problem of classifying whether a pixel belongs to a blue recycling bin, $d = 3$ corresponds to the number of channels in a pixel. Later in this paper, results are shown for the RGB, HSV, and LAB colorspaces. The number of classes K is greater than or equal to 2, and depends on the availability of labeled data. For initial results, the colors red, green, and blue are the classification targets so $k = 3$. For later results, a value of $k = 5$ is chosen to correspond with "colors": *bin blue* (BB), *non-bin blue* (NBB), green, red, and *ground*. The choice of these colors are further discussed in the technical approach section.

First using a training dataset with input examples and corresponding labels, the maximum likelihood estimate of weights W is performed. Subsequently using the weights and an unlabeled test dataset, the class probabilities of each pixel are predicted. The class with the maximum probability for each example is chosen as the predicted label.

3 Technical Approach

Given a dataset with corresponding labels, the maximum likelihood estimate of weights W can be estimated via gradient ascent [1]. The update rule for the weights can be summarized as

$$W \leftarrow W + \alpha \left(\sum_{i=1}^{n} \left(e_{y_i} - S(W x_i) \right) x_i^T \right)$$

where α is the learning rate, a tunable hyperparameter.

This weight update can be made more computationally efficient by reformulating the problem and taking advantage of matrix operations. First, for each data point x_i, a bias term of 1 is appended so the separating hyperplane in the model is not fixed to the origin. The dimension of the data d is redefined to contain this bias term. The x_i are then collected into a matrix $X \in \mathbb{R}^{n \times d}$. The weight matrix is still $W \in \mathbb{R}^{k \times d}$. The labels y are one-hot encoded in a matrix $Y \in \mathbb{R}^{n \times k}$.

The gradient update rule is now

$$W \leftarrow W + \alpha \left(Y - S_1(X W^\top) \right)^\top$$

where $S_1(\cdot)$ is the softmax function performed along the second dimension (the class dimension). And in order to predict class probabilities for the n input examples, the equation $p(Y|X, W) = S_1(XW^\top)$ is used. For each of the n examples, the class with the maximum probability for that specific example is chosen as the predicted label.

For the case of simple color classification, the model has parameters $k = 3$ corresponding to red, green, and blue classes; and $d = 3$ corresponding to the RGB colorspace.

For the case of blue bin detection, the model has parameters $k = 5$ corresponding to "colors" *bin blue* (BB), *non-bin blue* (NBB), green, red, and *ground*. The BB color is utilized so that the model can detect that corresponding color. The NBB color is utilized so that the model can distinguish between blues corresponding to bins and blues corresponding to non-bin surfaces. The other colors were chosen so the model could be more expressive, such that it can distinguish colors from both BB and NBB. If the model was trained on only the BB color versus all other colors, with parameter $k = 2$, the NBB color would be a very small subset of the set of all other colors. This imbalance would lead to the model classifying all blues pixels as the BB class, and not only blues belonging to blue bins. Although $d = 3$ in the case as well, the model is trained and tested with the input images encoded in three different colorspaces: RGB, LAB, and HSV. (Each colorspace has corresponding model weights.)

For blue bin (binary) pixel classification using the $k = 5$ model, the 5 class probabilities are first predicted for each pixel. Then only if the class with the maximum probability is the BB class, the binary output label is positive. Otherwise, the binary label is negative.

To support training the blue bin pixel classifier, masks corresponding to the 5 colors were manually labeled. The BB masks were labeled such that they only contain pixels belonging to blue bins. The NBB masks were labeled such that they only contain blue pixels that do not belong to bins. For example, NBB pixels can correspond to the sky or the ocean. The green class contains masks of objects such as trees and grass. The red class contains masks of red objects such as red bricks or red bins. The ground class contains masks of the ground, which can be yellowish or gray, such as concrete and asphalt. All of the pixels within the masks are selected and concatenated to the construct the input matrix X.

Once the K-ary logistic regression model has been trained, it can be used to segment image data. For blue bin detection, the input image is first converted to the colorspace corresponding to the colorspace used to train the model, and then segmentation is run to obtain a mask of pixels that are predicted to belong to a blue bin. Once the mask is found, a morphological "opening" is performed to

Table 1: Pixel Classification Model Weights, W

2.99962169e+01	-1.51493054e+01	-1.48694708e+01	-1.29330492e+00
-1.63318574e+01	2.96608841e+01	-1.41929216e+01	-5.74049416e-01
-1.40626541e+01	-1.27295371e+01	2.62398056e+01	-1.62561966e-01

remove speckle noise from the mask. This morphological operation consists of an erosion followed by a dilation of the mask. A disk of radius 2 was empirically found to be most effective for these pixel neighborhood operations. Then regions with connected positive pixels are identified. These regions are filtered using a few simple heuristics, and finally bounding boxes are extracted.

The heuristics used to filter proposed blue bin regions are as follows: (1) the bounding box of the regions must have an area greater than 10,000 pixels. Detected bins should be not smaller than roughly a 100x100 pixel patch; they are not too far away relative to the camera. (2) the number of pixels in the connected-region area divided by the number of pixels in the bounding box must be greater than 0.25. The region must not be "hollow" relative to the bounding box; the region should be filled in. (3) The height of the bounding box must be greater than the width. Bins are assumed to be standing upright, and the camera is viewing the bin from a side angle. (4) The height of the bounding box must be less than 2x the width. Again, bins are assumed to be standing upright and there is prior knowledge that bins are not overly tall.

4 Results

Prior to implementing blue bin detection, the model was trained on a simpler case of color classification. Using a training dataset of example red, green, and blue pixels, the model was trained for 50 epochs with a learning rate of 10^{-2}. Because of the simplicity of the dataset, 100% training and validation accuracy was achieved. This is shown in Figure 1. The model weights are summarized in Table 1.

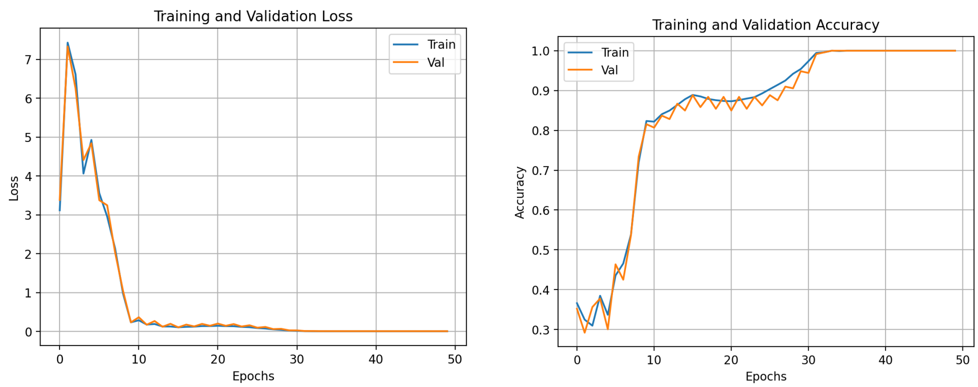

Figure 1: Pixel classifier training and validation results.

Table 2: Bin Detection Pixel Classification Model Weights, LAB colorspace, W.

3.093942e+00	2.119417e+01	-2.058168e+01	-1.928670e+00
-5.393489e+00	-2.453670e+01	2.859471e+01	1.041987e+00
-4.093516e-01	-6.704203e+00	6.981555e+00	2.387108e+00
7.355332e+00	1.406218e+00	-2.966961e+01	8.701452e+00
-1.347408e+00	1.197043e+01	1.241252e+01	-1.227995e+01

Table 3: Bounding box coordinates, in [x1, y1, x2, y2] order, where (x1, y1) and (x2, y2) are the top left and bottom right coordinate respectively.

1.	$[[189, 158, 311, 283]]$
2.	$[[28, 358, 129, 496]]$
3.	$[[173, 95, 264, 234]]$
4.	$[[351, 106, 465, 263]]$
5.	$[[778, 418, 923, 621]]$
6.	$[]$
7.	$[[580, 307, 704, 503], [713, 307, 829, 508]]$
8.	$[]$
9.	$[]$
10.	$[]$

For blue bin detection, the model was trained on hand-labeled data. The images and masked regions contained many pixels, so the training data has a size in the order of $n \approx 10^6$. Training was performed with a learning rate of 10^{-5} for 50 epochs. Three separate models were trained corresponding to the RGB, LAB, and HSV color spaces. Training accuracies and losses for each model are plotted in Figure 2.

The HSV model had the best training accuracy, but qualitatively the LAB model segmentation results were better on the validation dataset. Qualitative segmentation results are shown in Figure 3. Because the LAB results on the validation images were found to be qualitatively better for bounding box detection, the corresponding weights are listed in Table 2. The results of bounding box detection for the LAB model are shown in Figure .

Each of the estimated bounding box coordinates for the 10 validation images are listed in Table 3.

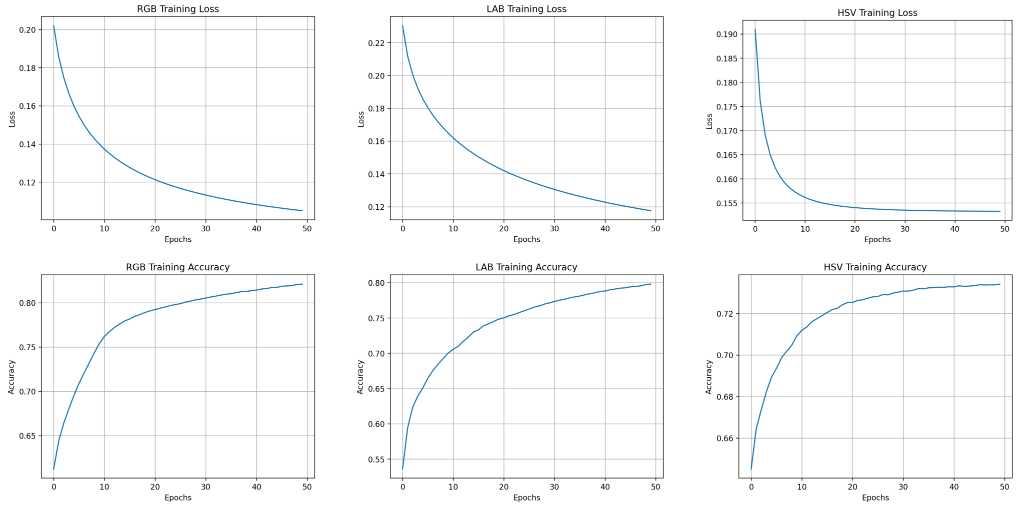

Figure 2: RGB, LAB, and HSV training losses and accuracies respectively.

Figure 3: Comparison of blue-bin segmentation in different colorspaces on the validation dataset. From left to right: input image, RGB segmentation, LAB segmentation, HSV segmentation.

Figure 4: Bounding box results. From left to right: input image, LAB segmentation, LAB segmentation after morphological opening, resulting bounding box after applying heuristics. (Note that row 5 and 7 has no blue bins, and row 6 has two blue bins.)

References

[1] Atanasov, N. (2020) *ECE276A: Sensing & Estimation in Robotics Lecture 5: Logistic Regression*. pp. 2-15