Blue Recycling Bin Detection Using Naive Bayes

Benjamin Chang
Department of Electrical and Computer Engineering
University of California, San Diego
bmc011@ucsd.edu

***Abstract*—Machine-learning models aim to enable machines to emulate human tasks. Being able to recognize objects in images is one such task. This paper investigates the use of the Naive Bayes machine-learning model to create a model which recognizes blue recycling bins in images, creates a segmentation mask, and displays bounding boxes around detected bins.**

I. Introduction

The ability for robots or cars to recognize objects in their surroundings is highly important and has use in projects such as autonomous vehicles, emergency response robots, and medical diagnostics. There has been a lot of work done in recent decades to improve recognition algorithms, many of which involve training some sort of model which represents the object to be recognized. Naive Bayes is one such method used to create probabilistic models which are used to classify data. It has a strong assumption of feature independence which surprisingly helps in some classification tasks.

In this project, several Naive Bayes blue recycling bin detection models are trained and their accuracy on validation data and autograder test data are evaluated. Each trained model varies in one or more of the following: number of classes (i.e. blue, brown, etc.), color space (i.e. RGB, HSV, LAB), or filtering methods (i.e. segmentation mask filtering, bounding box area filtering, erosion/dilation filtering).

This project is split into two sections. The first is ”Pixel_Classification” which classifies test pixels as red, green, or blue. The second is ”Bin_Detection” which classifies image pixels into color classes, segments pixels of the same classes together, and then predicts if there is a blue recycling bin in an input image.

II. Problem Formulation

We train a Naive Bayes model to recognize blue recycling bin pixels. Naive Bayes has input data $X \in \mathcal{R}^{n\times d}$ and output $y \in \mathcal{R}^{n\times 1}$. For our problem, each X row represents a pixel and each X column represents pixel features. For our problem, the features represented by each X column is dependent on the colorspace of the input image. For example, for the HSV colorspace, the X coumns would represent H, S, and V pixel values.

Naive Bayes needs to train 3 parameters to represent a model. These parameters are $theta$, mu, and $sigma$ and are defined by the following equations.

$$\theta_k^{MLE} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{y_i = k\} = \frac{1}{2}$$

$$\mu_{kl}^{MLE} = \frac{\sum_{i=1}^{n} x_{il}\mathbb{1}\{y_i = k\}}{\sum_{i=1}^{n} \mathbb{1}\{y_i = k\}}$$

$$\sigma_{kl}^{MLE} = \sqrt{\frac{\sum_{i=1}^{n}(x_{il} - \mu_{kl}^{MLE})^2\mathbb{1}\{y_i = k\}}{\sum_{i=1}^{n} \mathbb{1}\{y_i = k\}}}$$

In the above equations, n represents the number of rows in X, k represents various classes (i.e. 1-8 for 8 class case), and l represents the number of columns in X.

Once the parameters are calculated, the model can be tested using the following equation.

$$y_* = \operatorname*{arg\,min}_{k\in\{-1,+1\}} \left\{ \log\frac{1}{\theta_k^2} + \sum_{l=1}^{d} \log\sigma_{kl}^2 + \frac{(x_{*l} - \mu_{kl})^2}{\sigma_{kl}^2} \right\}$$

In the above equation, x_*l represents an entire column of X. The output y is a vector which contains class values for each pixel in input data X. The output equation takes the $argmin$ of the output vector and chooses the class with the lowest score as the predicted class for each pixel in X.

To create the segmentation mask, the output of y are all set to value 0 except for pixels with the ’recycling bin blue’ class. Bounding boxes are then obtained by checking for rectangular boxes which include segmented areas. The bounding boxes must meet certain filtering criteria involving bounding box area and bounding box height/width ratio. Once obtained, estimated bounding boxes are compared to ground truth bounding boxes to determine the accuracy of a trained model.

III. Technical Approach

”Pixel_Classification” uses training and validation datasets. The training set contains red, green, and blue single-colored pixel images with roughly 3700 images per color. The validation set contains 60-80 images of each RGB color. To train the Naive Bayes model to classify pixels as red, green, or blue, the training pixels are used to calculate 3 parameters: θ, μ, and σ.

To test the accuracy of the resulting model, pixels from the validation dataset are input to the model. The model labels each pixel as 1, 2, or 3 depending on if it estimates the given pixel to be red, green, or blue respectively. The estimated results are compared with the ground truth color of each pixel to calculate the model’s accuracy.

”Bin_Detection” uses training and validation datasets which contain 60 and 10 images respectively. The testing dataset also contain the ground truth pixel locations of the recycling bins in each of its images. We label the training data into various numbers of color classes such as recycling bin blue and green using the ’roipoly’ python package. The labelled data is stored with class labels and then used to train the parameters of the Naive Bayes classification model using steps described in the ”Problem Formulation” section above.

Once the Naive Bayes model is trained, images from the validation dataset are input into the model which in turn predicts a color class for each pixel. A binary image mask is created from this data. This is done by segmenting out all pixels predicted to belong to the recycling bin blue class as white with value 255 and all other pixels as black with value 0. Various filtering criteria are used to improve the image mask result. These include a bounding box filter which filters out bounding boxes which contain less than a certain number of pixels, a height-to-width ratio filter which filters out bounding boxes with height not between 1x to 2.5x of its width, and an erosion/dilation filter which removes estimations without other estimations near, essentially avoiding having lots of small bounding boxes.

Once the segmentation image mask is processed by the filters, it is estimated which pixel groups likely form an actual blue recycling bin. This is done using the Python’s opencv function ’findContours’ which locates contours (large segmented sections of the same value). Using these locations, we draw bounding-boxes which encompass the discovered contour areas. The bounding box coordinates are recorded and the original input image with its estimated and ground truth bounding boxes are displayed for comparison.

IV. Results

For ”Pixel_Classification”, our model achieves an accuracy of 92.7711% on the validation set. Some examples of the test images and validation images are shown in Fig. 1.

Fig. 1. Pixel Classification Training Data Examples

Before training a Naive Bayes model with multiple classes, 3 test models were trained in the RGB, HSV, and LAB color spaces to see which colorspace offers the best performance. Segmentation masks and bounding box images from testing the models on validation images are shown below as well as the number of images they classified correctly. The validation images and their estimated segmentation masks and bounding boxes are shown in Fig. 2.

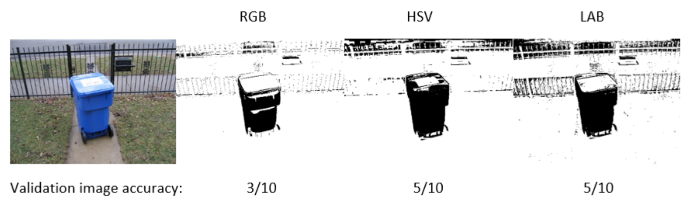

Fig. 2. 2 Class RGB, HSV, LAB Colorspace - Results and Accuracy on Validation Data

From these results, it is seen that the LAB colorspace offers the worst performance with poor recycling bin and background segmentation. The HSV result on the other hand has good recycling bin segmentation, but poor background segmentation. The background segmentation errors can be fixed later by training with more classes though so shouldn’t be too big of an issue. The RGB result has good background segmentation, but poor recycling bin segmentation which implies it doesn’t differentiate recycling bin blue from other colors as well as the HSV or LAB colorspaces. This is easily seen by comparing the segmentation mask results between the colorspaces. Since the HSV results seem to be the best based on its segmentation mask accuracy, we use it for for training a more complex multi-class model.

For ”Bin_Detection”, 3 multi-class models were trained and 4 tests run on these models total. The tests were run on 10 validation images and 10 autograder test images. The 10 validation images are shown in Fig. 3.

Fig. 3. 10 Validation Images

Model 1 is a Naive Bayes model trained in the LAB colorspace. It has 6 classes which are recycling bin blue, sky blue, brown, light gray, dark gray, and green. It achieves an accuracy of 9/10 on the validation data, but a low score of 3/10 on the autograder test data. The validation images and their estimated segmentation masks and bounding boxes are shown in Fig. 4 and Fig. 5.

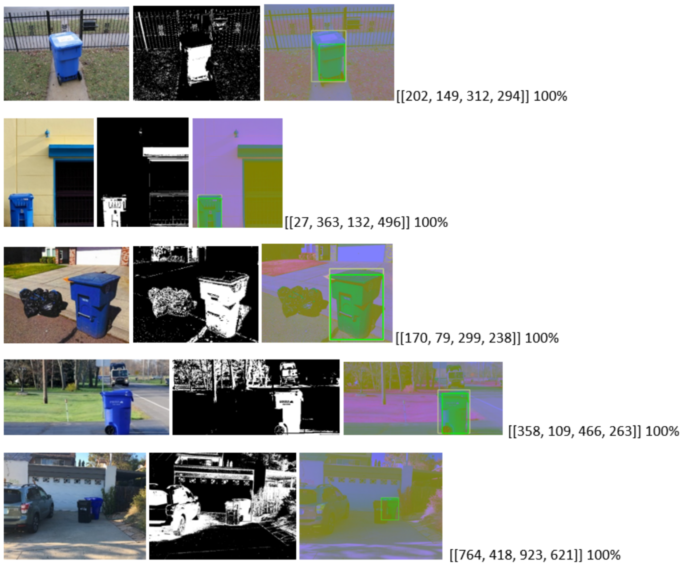

Fig. 4. 6 Class LAB Colorspace - Results and Bounding Box Coordinates

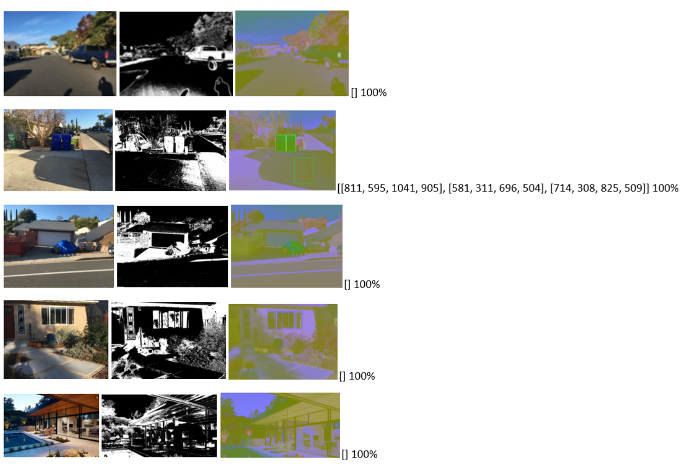

Fig. 5. 6 Class LAB Colorspace - Results and Bounding Box Coordinates

From the results of Model 1, it seems black was often misclassified as recycling bin blue. To counteract this, a 'black' class is added for Model 2.

Model 2 is a Naive Bayes model trained in the LAB colorspace. It has 7 classes which are recycling bin blue, sky blue, brown, light gray, dark gray, green, and black. It achieves an accuracy of 9/10 on the validation data, but a low score of 3.25/10 on the autograder test data. The validation images and their estimated segmentation masks and bounding boxes are shown in Fig. 6 and Fig. 7.

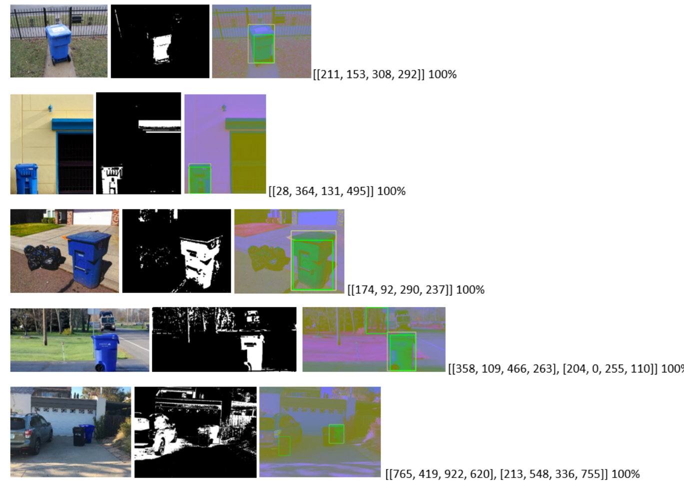

Fig. 6. 7 Class LAB Colorspace - Results and Bounding Box Coordinates

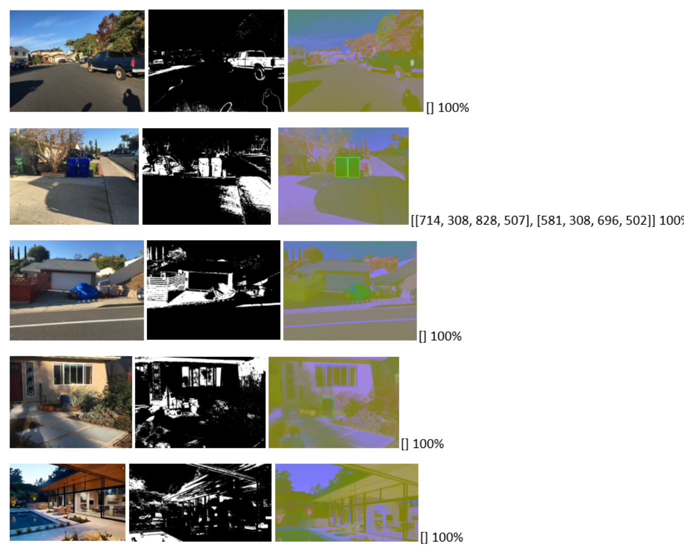

Fig. 7. 7 Class LAB Colorspace and Bounding Box Coordinates - Results

At this point, it was noted that we were using the LAB color space the whole time when intending to use the HSV colorspace. So, a switch was made to the HSV color space for Model 3 as well as the addition of a 'tan' class since it was noted from the Model 2 results that there were a lot of tan areas misclassified as recycling bin blue.

Model 3 is a Naive Bayes model trained in the HSV color space. It has 8 classes which are recycling bin blue, sky blue, brown, light gray, dark gray, green, black, and tan. It achieves an accuracy of 10/10 on validation data and 7.75/10 on autograder test data. The validation images and their estimated segmentation masks and bounding boxes are shown in Fig. 8 and Fig. 9.

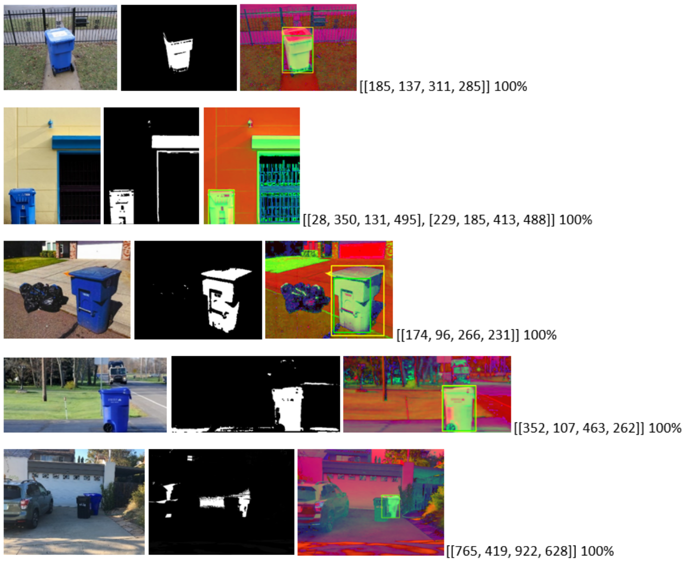

Fig. 8. 8 Class HSV Colorspace - Results and Bounding Box Coordinates

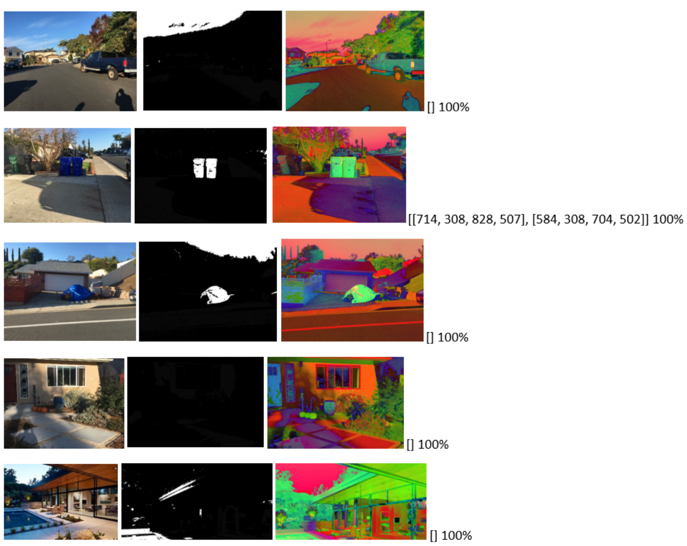

Fig. 9. 8 Class HSV Colorspace - Results and Bounding Box Coordinates

During data labelling, pixels labelled for the 'recycling bin blue' class were mostly pretty saturated blue hues. Since there was the possibility there was a lot of sky blue recycling bins in the autograder test data Model 3 was trained with pixels predicted to be 'sky blue' set to 'recycling bin blue' from. This variational model was tested to see how it would perform. The validation images and their estimated segmentation masks and bounding boxes are shown in Fig. 10 and Fig. 11.

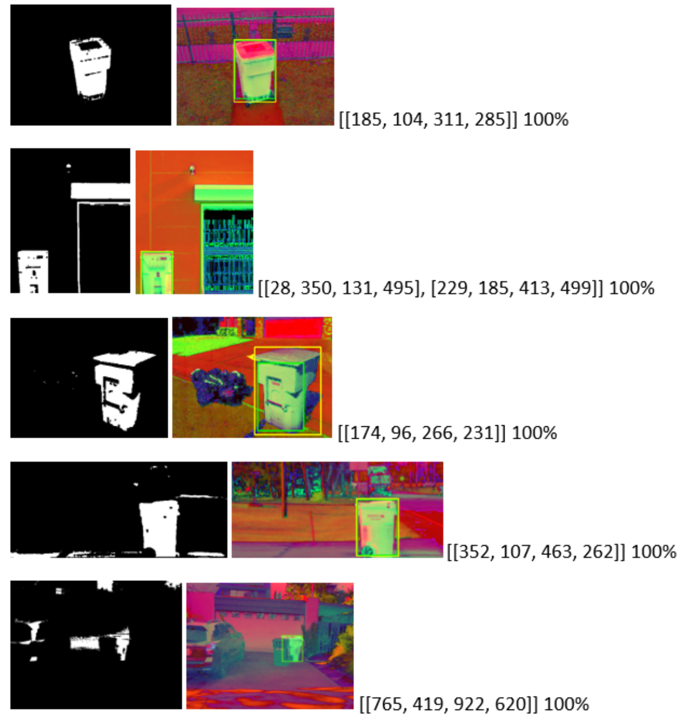

Fig. 10. 8 Class HSV Colorspace - Count SkyBlue Same As Bin Blue - Results and Bounding Box Coordinates

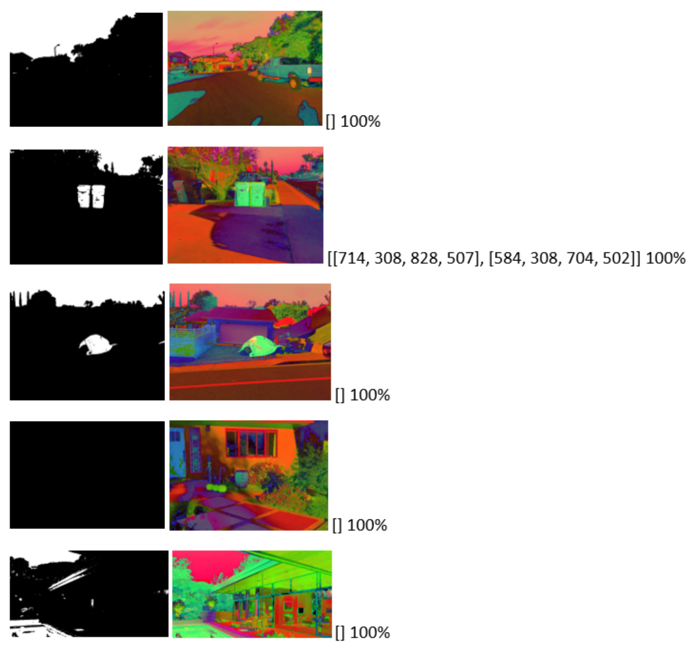

Fig. 11. 8 Class HSV Colorspace - Count SkyBlue Same As Bin Blue - Results and Bounding Box Coordinates

This model achieved an accuracy of 10/10 on the validation data and 6.25 on autograder test data. As such, it can be concluded that Model 3 is the best performing model.

It seems there is a significant improvement in using the HSV colorspace vs the LAB colorspace which is expected since

HSV represents pixels according to their hue, saturation, and value. This enables the recycling bins to stand out more since their saturation levels are much higher than that of most other pixels in the validation images. This is in contrast to the LAB colorspace where there is greater emphasis on lightness (black to white) than color. It also contrasts to the RGB colorspace which emphasizes hue.

The $theta$, mu, and $sigma$ parameters for all of the above models are shown in Fig. 12 and Fig. 13.

PIXEL CLASSIFICATION

theta = [[0.38224147]
[0.33540877]
[0.33676232]]
mu = [[0.38224147]
[0.33540877]
[0.33676232]]
sigma = [[0.38224147]
[0.33540877]
[0.33676232]]

2 Class RGB Colorspace Model

theta = [[0.08014936]
[0.91985064]]
mu = [[61.86209932 92.60885268 169.05600458]
[21.15841053 21.75027146 19.72919568]]
sigma = [[38.85126821 43.71959677 45.17878854]
[115.88640259 114.23272819 114.84949014]]

6 Class LAB Colorspace Model

theta = [[0.24619695]
[0.08063391]
[0.17774382]
[0.32886092]
[0.14503262]
[0.02153176]
[0.]]
mu = [[113.32762926 114.19887196 159.42867358]
[17.58585908 100.17997925 150.22865035]
[66.64687477 37.39505019 173.56690356]
[80.83475803 40.88389668 104.29700185]
[48.1071132 100.83215803 98.92584608]
[98.46708367 108.54739373 30.44854907]
[nan nan nan]]
sigma = [[19.15868462 73.49901755 38.32630443]
[22.98342388 41.23808893 44.58619031]
[46.06637338 26.8051262 32.46812202]
[41.24061771 38.8071515 40.2650335]
[24.3392386 55.62608449 50.7089971]
[35.68368768 47.24093114 11.09916548]
[nan nan nan]]
[24.3392386 55.62608449 50.7089971]
[35.68368768 47.24093114 11.09916548]
[nan nan nan]]

Fig. 12. Model Parameters

2 Class HSV Colorspace Model

theta = [[0.07092489] [0.92907511]]
mu = [[61.86209932 92.60885268 169.05600458]
[21.15841053 21.75027146 19.72919568]]
sigma2: [[6.90524904 49.77531337 42.13665871]
[105.43957841 107.14618106 114.90285563]]

2 Class LAB Colorspace Model

theta = [[0.08023809]
[0.91976191]]
mu = [[105.45675137 143.50879915 84.4059345
[27.61327666 32.21109464 33.0569627]]
sigma = [[40.41495745 14.5961051 16.454574
[114.98113462 96.28731436 97.12426081]]

7 Class LAB Colorspace Model

theta = [[0.2259114]
[0.08239562]
[0.07399003]
[0.16309851]
[0.30176422]
[0.13308257]
[0.01975764]]
mu = [[113.32762926 114.19887196 159.42867358]
[105.85997519 127.8649378 217.39612955]
[17.58585908 100.17997925 150.22865035]
[66.64687477 37.39505019 173.56690356]
[80.83475803 40.88389668 104.29700185]
[48.1071132 100.83215803 98.92584608]
[98.46708367 108.54739373 30.44854907]]
sigma = [[19.15868462 73.49901755 38.32630443]
[4.99943642 43.90189523 27.93679889]
[22.98342388 41.23808893 44.58619031]
[46.06637338 26.8051262 32.46812202]
[41.24061771 38.8071515 40.2650335]
[24.3392386 55.62608449 50.7089971]
[35.68368768 47.24093114 11.09916548]]

8 Class HSV Colorspace Model (BEST MODEL)

theta = [[0.0684561]
[0.02201082]
[0.02579361]
[0.09055232]
[0.55357144]
[0.04042055]
[0.08026328]
[0.11893189]]
mu = [[113.26631748 208.051578 179.8161967]
[101.24026912 82.57533606 249.80005779]
[11.92207533 123.13661767 170.61049406]
[96.24633038 31.0805862 191.48105575]
[80.40401603 65.15727341 68.76643407]
[38.4167572 136.02476193 104.35232373]
[93.44434299 112.85956141 30.30619468]
[18.02220411 64.26717254 171.42640049]]
sigma = [[4.52345825 36.76521552 37.11821204]
[4.33849854 30.69547856 7.57755628]
[2.659631 40.64367083 31.83310551]
[37.40997855 22.48371559 42.64111071]
[37.64502114 39.05752975 17.01460787]
[8.86090256 49.50310332 46.33672447]
[39.03508071 53.54916508 12.41015824]
[3.95535934 15.10284068 30.86356117]]

8 Class HSV Colorspace Model – Sky Blue same class as Recyclin

- Sam parameters as '8 Class HSV Colorspace'

Fig. 13. Model Parameters

V. Discussion

From the above results, the best model implemented was the 8 class HSV colorspace model. It seems that the more classes are added to the model, the better the segmentation mask become and the more accurate the predicted bounding boxes are. Some color classes which made a larger difference in increasing model accuracy were dark gray, black, green, and brown. This is expected since these colors are very similar to the blue recycling bin colors in the training data.

The filtering methods significantly increased model accuracy as well. The erosion/dilation filters helped denoise the predicted segmentation mask and got rid of small random pixels which were misclassified as 'recycling bin blue'. The area and height-to-width ratio filters got rid of small bounding boxes which were misclassified as recycling bins. These are acceptable ways to handle misclassfied bounding boxes since they filter out small bounding boxes only. If a recycling bin is small in an image, it is not very consequential whether it is detected or not. During training iterations, the parameters for these filters were iteratively tested.

VI. Conclusion

This paper explores the Naive Bayes model and provides evidence that this model is able to achieve good performance. There are many other more complex methods to classify blue recycling bin pixels, but the simple effectiveness of Naive Bayes is certainly something to be amazed at. In the future, more classes can be added to increase the accuracy of the model as well as testing more filtering methods.