ECE 276A: Colour Classification and Object Detection: Project Review

Varun Vupparige
Mechanical and Aerospace Engineering
University of California San Diego
La Jolla, USA
vvuppari@ucsd.edu

With the Internet of Things (IoT) and Artificial Intelligence (AI) becoming ubiquitous technologies, we now have huge volumes of data being generated. Since the start of last decade, there has been considerable increase in advent of image processing techniques, especially in robotics and industrial automation. These techniques have been widely applied in medical imaging, facial recognition, surveillance and many other applications. Since the vast amount of image data we obtain from cameras and sensors is unstructured, we depend on advanced techniques such as machine learning algorithms to analyze the images efficiently. Furthermore, classifying pixels forms a fundamental part of a SLAM problem in robotics. It is a method in which a digital image is broken down into various subgroups called segments which helps in reducing the complexity of the image and thus extract necessary information required to meet a certain objective. This process is generally diving in following categories: Image pre-processing, detection, feature extraction, training and classification. The field of computer vision includes a set of main problems such as image classification, localization, image segmentation, and object detection. Among those, image classification can be considered as the fundamental problem. It forms the basis for other computer vision problems.

In this particular project, a supervised pixel classifier based on logistic regression is modelled which is capable of classifying between recycling bin and not a recycling bin, and construct a bounding box around it based on certain rules. The parameters for predicting the labels of new test image are estimated by minimizing the cross entropy loss. Furthermore, using few morphological operations and apply heuristics, a bounding box is created for the test image. Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. To model the classifier for three classes, we construct the model in a way that it solves three different binary logistic regression problems. The parameters for predicting the labels of new test image are estimated by minimizing the cross entropy loss. Furthermore, using few morphological operations and apply heuristics, a bounding box is created for the test image.

I. Problem Formulation

Consider X be a matrix representing the pixel values of an image and Y be a vector representing the corresponding labels. Since the pixel values are distorted, it's not a deterministic model and hence we model it as a probabilistic model. Consider $K = 1, 2, 3$ representing the labels for red, blue and green respectively. Let $D = xi, yi$ be the training dataset used to train the probabilistic model. Logistic regression is a discriminative model, and hence it is modelled as the conditional probability of labels Y given X and θ.

In order to convert regression values into a probability mass function, we use the logistic sigmoid function. Thus, the probability mass function can be represented as $p(y|x; \theta)$. Based on the training data, our aim is to find the parameters which maximises the probability density function. We find the parameters using maximum likelihood estimation, and solving the optimization problem using the gradient descent method. The labels for new test image is computed using the optimized parameter.

II. Technical Approach

A. Color Classification

In the part 1 of this project, we are expected to build and train a probabilistic model to distinguish among red, green and blue. Using the generate$_rgb_data.py, thetrainingdatasetisprepared.$

The training data set needs to be loaded into the model to generate parameters which can be used for predict labels of test images. Since the problem requires us to define three classes, it is approached as one vs all classifier, where in it solves three independent binary classifier problems. The parameters of this probability distribution is estimated using MLE approach. The MLE approach, which is abbreviated as Maximum Likelihood Estimation looks to solve the optimization problem of finding the parameter which maximised a given distribution based on the training data set.

For logistic regression, we use a function called as logistic sigmoid function to convert regression values to probability mass function ranging between zero and one. But the jacobian of this function does not produce a closed form solution. Due to this we employ a gradient descent algorithm to solve the optimization problem posed by MLE. After carefully choosing the step size and number of iterations, we obtain a set of optimised parameters for each class. These three parameters are loaded into to the probabilistic model to obtain the labels for a given test image.

In the second part of the project, we are required to build a probabilistic model to classify between recycling bin blue and the rest. Using this classifier, a bounding box needs to be constructed which identifies the recycling bin. Using the roipoly function, the training dataset is prepared by carefully selecting region of interest in both the classes. This training dataset is loaded into the probabilistic model and we obtain the parameters. In this case, since we are using only two classes, we solve a binary logsitic regression problem.

The calculated parameters are loaded into the predict function, given a new test image. This is done using the segmentimage function which outputs a grayscale image. Furthermore, this grayscale image is given as an input argument to the bounding box function which outputs the diagonal coordinated of the rectangular box.

III. Results and Discussion

This section discusses results and analysis of the classifier and segment image functions. For Part 1, Fig 1 depicts the behaviour of cost function when gradient descent is employed for 20000 iterations. Furthermore, fig 2 represents the validation report of pixel classifier for red, green and blue pixels.

Fig 2 and 6 shows the masked images obtained from the classifier for scene 1 and 2 respectively. Fig 2,3,4 shows the images with bounding box. Major reasons for inefficient segmentation is lack of enough training data and solving in the RGB space instead of a space which is more sensitive to lighting and brightness.

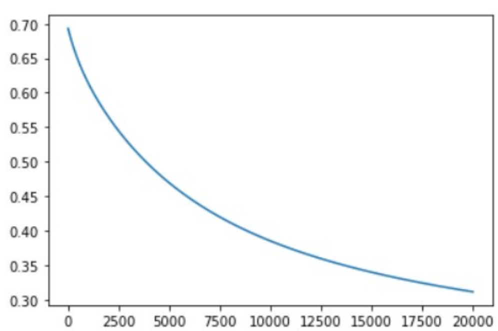

Fig. 1. Cross Entropy loss (Cost vs No of Iterations

IV. Conclusion

In all, we employed logistic regression to classify pixels and used various morphological and shape transformation functions to detect recycling bin and construct a bounding box around it.

```
y_pred = predict(np.array(ws), X)
np.count_nonzero(y==1),np.count_nonzero(y==2),np.count_nonzero(y==3)

from sklearn.metrics import classification_report

print(classification_report(y, y_pred, labels=[1,2,3], target_names=["Red", "Green", "Blue"]))
```

```
[[ 3.91709052 -2.8699295  -2.75985246]
 [-3.00431196  3.7958878  -2.73177523]
 [-3.0082238  -2.75857792  3.72503659]]
              precision    recall  f1-score   support

         Red       1.00      1.00      1.00      1352
       Green       1.00      1.00      1.00      1199
        Blue       1.00      1.00      1.00      1143

    accuracy                           1.00      3694
   macro avg       1.00      1.00      1.00      3694
weighted avg       1.00      1.00      1.00      3694
```

Fig. 2. Classifier validation report

Fig. 3. Masked image of scene 1

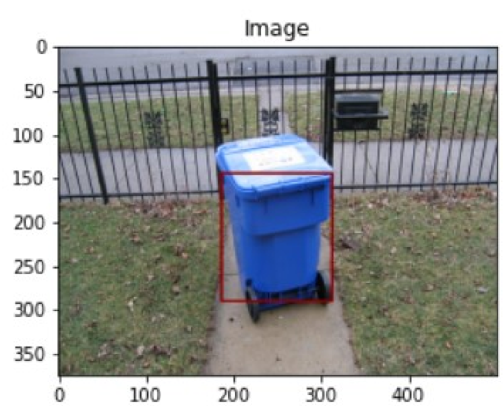

Fig. 4. Bounding box for scene 1

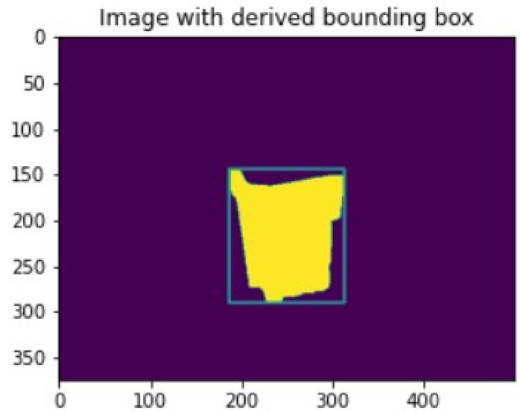

Fig. 5. Derived bounding box for scene 1

Fig. 6. Masked image of scene 2

Fig. 7. Bounding box for scene 2

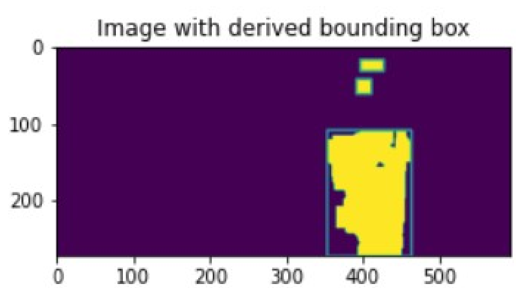

Fig. 8. Derived bounding box for scene 2

Fig. 9. Bounding box for scene 3

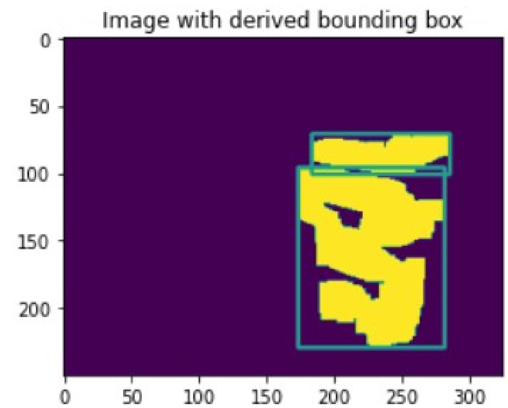

Fig. 10. Derived bounding box for scene 3

```
thetas = [-0.51719884, -0.25125233,  0.4795692]
w = np.array(thetas)
```

Fig. 11. Parameters for Part 2

```
[[ 3.91709052 -2.8699295  -2.75985246]
 [-3.00431196  3.7958878  -2.73177523]
 [-3.0082238  -2.75857792  3.72503659]]
```

Fig. 12. Parameters for Part 1