Color Detection Under Supervised Learning

Cyrus Anderson
University of Michigan
andersct@umich.edu

Jocelyn Bohr
University of Michigan
bjocelyn@umich.edu

Michael Lu
University of Michigan
lumike@umich.edu

Nghia Vo
University of Michigan
thnghia@umich.edu

Abstract

1 Introduction

Each summer the autonomous robotic boat team UM:Autonomy at the University of Michigan competes in the Association for Unmanned Vehicle Systems International's RoboBoat competition held in Virginia Beach, VA against various robotics labs and other clubs. Since a pond serves as the site for the various challenges used to judge the robots, buoys act as the primary landmarks. Solidifying buoys' importance, there are often few other objects usable as landmarks so buoys most often serve as the source of loop closures in the robot's long term path planning, allowing the robot to build an accurate internal map and minimize drift in its state. Knowing additional features such as color will further reduce contradictory information that can produce incorrect mapping. Buoys may also serve as boundaries or goals for certain challenges [competition specs] with determining the buoy color being in some cases part of the challenge. Thus there is substantial need to accurately identify buoy color.

Currently the UM:Autonomy robotic boat relies on color filters to determine color. This involves specifying bounds in RGB values for each color of buoy, and classifying buoys according to their match with the closest color template. These filters are faulty because the tuning is highly susceptible to bias on current lighting conditions, resulting in detections that work well for several hours before deteriorating in accuracy. This is especially apparent in Virginia where the sun will be high in the sky for most of the day, casting glare on the buoys and water, with the occasional cloud passing in front to cast a shadow the entire competition ground. Rainy and overcast days present yet another set of lighting conditions, and even on days when the sun stays high in the sky, backlit green buoys can appear black *(fig.1 left)*, yellow can appear green, and washed out red may look orange. Additionally white and black are difficult to filter because filters that do so generally have high false positive rates, ending up including the very light or very dark patches of water produced with the noon sun's lighting *(fig.1 right)*.

In this paper we examine the results of various supervised learning techniques applied to classifying buoy color, aiming to create a procedure to accurately classify buoy colors in a variety of lighting conditions, creating a robust system that requires minimal maintenance.

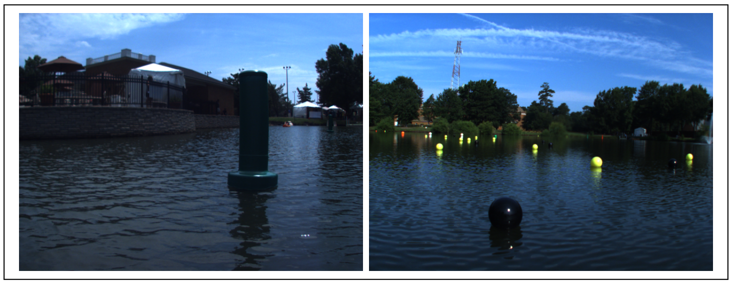

Figure 1: Difficult buoys

2 Proposed Method

2.1 Preprocessing

The robotic boat has two Point Grey Flea 2, 1.3MP cameras that publish images over the robot's Lightweight Communications and Marshalling (LCM) framework at 10Hz. Logging utilities use LCM to capture and store the various messages, and the datasets are crafted from the different logs of the robot's runs. Each boat photo we label by hand, marking the color and rectangle around each buoy within about 30 feet from the boat. From each rectangle, we extract the associated color histogram into a 256x256x256 feature vector, where each entry represents the frequency of the RGB color associated with the entry index. We then reduce this feature vector to a vector of size 64x64x64, to make feature vectors less sparse, and to decrease classification time as we are building this for a real time application.

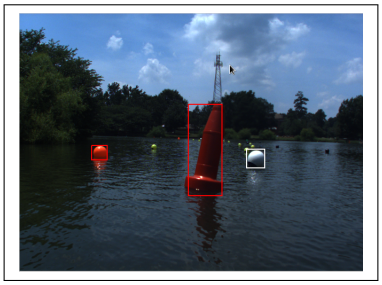

Figure 2: Labelled photo from boat

2.2 Classification

In this project, we run our training data with multiclass SVM and multinomial Naive Bayes, and have excellent results.

We began with SVM beacuse it is a parametric algorithm, thus we do not need to store entire training data set to make predictions on the boat. It is also kernelized, which provides us with the flexibility to create arbitrarily complex boundaries. SVM prediction can be used to calculate confidence values for the decision by taking the distance from the separating hyperplane, which is a useful measurement. SVM does not make any assumptions about the underlying distribution of the data. SVM is a very popular algorithm, so it is very good as a baseline classification algorithm. We used LIBLINEAR for our SVM classification, which utilizes a one-vs-rest multi-class classification scheme [1]. We achieved very good results with LIBLINEAR, as described later, so we decided not to use any kind of SVM kernelization. Using a linear kernel with SVM is preferrable, as it is faster to train and test. Making decisions quickly is essential to this application, as we want to classify colors at the same rate photos are captured (10 Hz).

We also used Naive Bayes because it is simple to implement, and performs well even on small sets of training data. Like SVM, Naive Bayes is parametric, so we do not need to store the entire training data set to make decisions during the real time application. A parametric classifier is essential, as it is impractical to store all training data on the boat to make decisions in real time. Naive Bayes is also very fast in training and testing, which is ideal for this application. The multinomial Naive Bayes classifier works well for classification with discrete features, such as color frequencies of pixels in a certain section of a photo, or word frequencies of a text document (the classic application). We used MATLAB's multinomial Naive Bayes classifier, which performs very well, as described later.

3 Related Work

// tommy majority vote and average color algorithm performance
// cyrus other related work

4 Experimental Results

Using both linear SVM and multinomial Naive Bayes, we achieve excellent accuracy rates, especially compared to current methods discussed in related works. With enough training examples, both methods achieve over 98% accuracy. Naive Bayes performs marginally better, the advantage of Naive Bayes is emphasized when fewer training examples are used. Each average accuracy is calculated by selecting a random training set of size n, then selecting a random testing set of size 130 from the remaining data, then train and test classifiers accordingly. The average accuracy is taken over 10 classification results. Although these results are excellent, we believe one contributing factor to such great results can be attributed to a fairly homogeneous data set. We do not have enough variation in our data set, so classification is easier. This problem, and how we intend to solve it, will be discussed in the future milestones section. Experimental results are shown below.

Table 1: Linear SVM Average Accuracy

EXAMPLES	**ACCURACY**
50	89.08
100	93.77
200	96.92
400	97.85
700	98.38

Table 2: Multinomial Naive Bayes Average Accuracy

EXAMPLES	**ACCURACY**
50	92.85
100	96.54
200	97.31
400	98.38
700	98.31

5 Future Milestones

One problem with our method we have identified is that the data set we are using is from one log, resulting in a very homogenous data set. Within one week from the due date of the progress report, we plan to label at least another 1000-5000 buoys, including many more blue and green buoys, and more buoys in varied lighting conditions. We hope that after running our algorithms on this data set with variation, we have a more realistic classification accuracy, which will perform better during the real application. We also want to try to add a feature which encodes the direction of lighting, to be completed within two weeks. One idea we have as an extension to this color classification project is implementing a segmentation algorithm to identify buoy boundaries under unsupervised learning, then do color classification with or current supervised learning algorithms. We are considering an extension to this project as we have achieved excellent results so far, thus we don't have much room for further improvement for the rest of the semester. We will discuss what we should pursue as a possible extension with the EECS445 staff.

6 Conclusion

// michael do conclusion

References

// michael do references

References follow the acknowledgments. Use unnumbered third level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to 'small' (9-point) when listing the references. **Remember that this year you can use a ninth page as long as it contains *only* cited references.**

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D. S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609-616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer-Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.