CMSC 471 Artificial Intelligence Image Classification Project

William Davies

05/04/2016

0.1 Approach

0.1.1 Labeling and Assumptions about the Training Set

I chose to approach this problem step by step, first determining the best way to assign labels onto my data. Since the given data for classification was (reasonably) assumed to be 100x100 pixel images, I chose to represent all of the images as RGB matrixes. The most difficult part after determining how to represent the data was how to label the data. I chose a simple approach which assumed that certain files in a Training directory would be a certain class of image (i.e. the first 50 images in the Training directory would be smiles, and so on). From this assumption, I simply had to classify all the images in my Traning directory based on steps of 50.

0.1.2 Machine Learning Algorithm Choice

I chose to use K-Nearest Neighbors(KNN) since I started at first with two sets of data, the Smiles and the Hats. Once I determined that I had a good handle on the sklearn library for KNN, I tried to extend onto the rest of the data. Unfortunately, I did not know how to apply additional classes with the code that I had already made, which was only good at determining two classes.

0.1.3 Implementation of Solution

I determined that the fastest way to overcome this shortcoming would be to classify five different sets of images, with each set being a boolean of each desired class (i.e. Smile, Smile, etc). From this, I simply had to apply a for loop within my given KNN classification programs, and I was able to quickly train 5 seperate sets. After this, I simply tried to use each seperate set to predict if the user supplied image would be in a particular class.

0.2 Accuracy

The overall accuracy for the given project varies, but the code was able to consistently guess correctly around 90of the time for most of the given

input files from the validation set. However, the KNN approach sometimes failed to classify images at all, and wouldn't output anything if it couldn't determine which set the image belonged in. Furthermore, the Dollar class was the most ambiguous for classification, and KNN was consistently fairly inaccurate when predicting if a given image was of the Dollar class compared to all the other classes.

0.3 Conclusion

Through this project there were many practical and real applications that would be simple extensions of this code. Furthermore, there was an initially steep learning curve with regards to machine learning as it is an ever growing field, but once I chose an algorithm and approach the open-endedness of the project was simplified and I was able to find a viable solution.