

CMSC 471: Project 3

Lee Brown

May 25, 2016



Figure 1: A sample image.

1 Introduction

The purpose of this project was to use a support vector machine to make a basic image recognition software. The packages, language, and versions used for this project are as follows:

- Python 3.5.1
- Anaconda 2.5.0
- numpy 1.10.4
- sklearn 0.17
- Python Image Library

2 Method

```
Terminal
lee@littlewilhelm ~/Documents/471/Proj3 $ python proj3.py 11.jpg
/home/lee/anaconda3/lib/python3.5/site-packages/sklearn/utils/class_weight.py:62: DeprecationWarning: The class_weight='auto' heuristic is deprecated in 0.17 in favor of a new heuristic class_weight='balanced'. 'auto' will be removed in 0.19
  " 0.19", DeprecationWarning)
smile
lee@littlewilhelm ~/Documents/471/Proj3 $
```

Figure 2: The program predicting the above smiley face.

The process used for this project is a multi-step process. The first step is loading and processing the training set. The program opens the location of the training set and processes the files one by one. By using the Python Imaging Library's (PIL) functions for images and list comprehensions, it was easy to make the transition from an image to the images normalized bitmap. The steps for completely processing the image were

- Open the image.
- Get the list of tuples containing the pixel data using PIL's `getdata()`.
- Flatten the list of tuples into just a 1-dimensional list of integers.
- Normalize the list by dividing each entry by 255.

The normalized list of pixel values were stored in the main function in a large, 2-dimensional list named `tempList`. Alongside of that list, the items were also being classified by storing a value in another list called `classifierList`, that saved a string describing the image based on which directory it was opened from.

Once all of the images were processed, the Sci-kit Learn SVM was created. By using the class weight setting of `auto`, I increased the accuracy of the SVM significantly. The two lists are passed in together for it to learn from the training set, and prepare it for the test.

The test image is then loaded and processed like the training images, and the SVM is used to predict what type of image it was. It then prints out the results.