Project 3 Report

Adam Wendler

5/11/2016

1 Abstract

Contained is the python code for the Project 3 assignment. This code uses a support vector machine to identify one of 5 image types. To achieve this, the code is trained with labeled images, and the resulting weights are preserved to create a probability that an image is of one of 5 types. The code is specifically designed to identify 100 x 100 pixel jpg format images, and only these should be entered. The images are also generally simple drawings.

2 Requirements

2.1 Libraries

The code relies on several python libraries, some of which may need to be downloaded. The Required libraries are:

- PIL, sometimes called Pillow
- Numpy
- sklearn.svm

Both of these Packages should be included with a download of the Anaconda python distribution.

2.2 Additional Concerns

Also included with this code is a directory named "Data". The Python script is hard coded to draw from this directory during the training process, so please make sure the Data directory is downloaded and in the same directory as the Python script before running. In addition, do not rename or Data or any of it's subdirectories, or any of the file contained with in. For the best possible performance, it would be best to leave the directory as it is at the time of download.

3 Algorithm

The Algorithm the code follows can be split into two major phases, Training and User Mode

3.1 Training Phase

The object of the training phase is to train the support vector machines to recognize and classify images based on a set of features. In this phase, the Image is loaded, flattened, and then used to train the support vector machines. First, the Image is loaded into a numpy array using the PIL library. The array represents each pixel as an array of RBG values. The algorithm then iterates through each of these RGB arrays and averages them. The averages for the whole image are then averaged again. This average is used as a cut-off point in an algorithm to convert the image into a bitmap. After this, the image is ready to be flattened. Finally, the flattened images are given to a support vector machine for training. The SVM has 5 outputs, 0-4, and each image is assigned the correct output.

3.2 User Phase

During the User Phase, the user is allowed to enter a file path to an image. The image is converted to black and white and flattened. Then, the flattened image is given to the SVM for it to produce a label. Typing stop during the user phase will end the program.

4 Accuracy

The algorithm is most accurate with the Face, Hat, and Dollar Sign shapes. The algorithm recognizes these shapes around 80-90 percent of the time. The Hash and Heart shapes are less likely to be recognized. This may be due to their shapes being relatively similar. However, 50-70 percent of the time the SVM will still produce the correct label for the Hash and Heart shapes.

5 Additional

In addition, the code for creating a bitmap is borrowed from the following source:

https://pythonprogramming.net/automated-image-thresholding-python/?completed=/thresholding-python-function/