

Project 1

Algorithmically determine if a picture contains people

Jones, Jonathan

jonathan.jones@gatech.edu

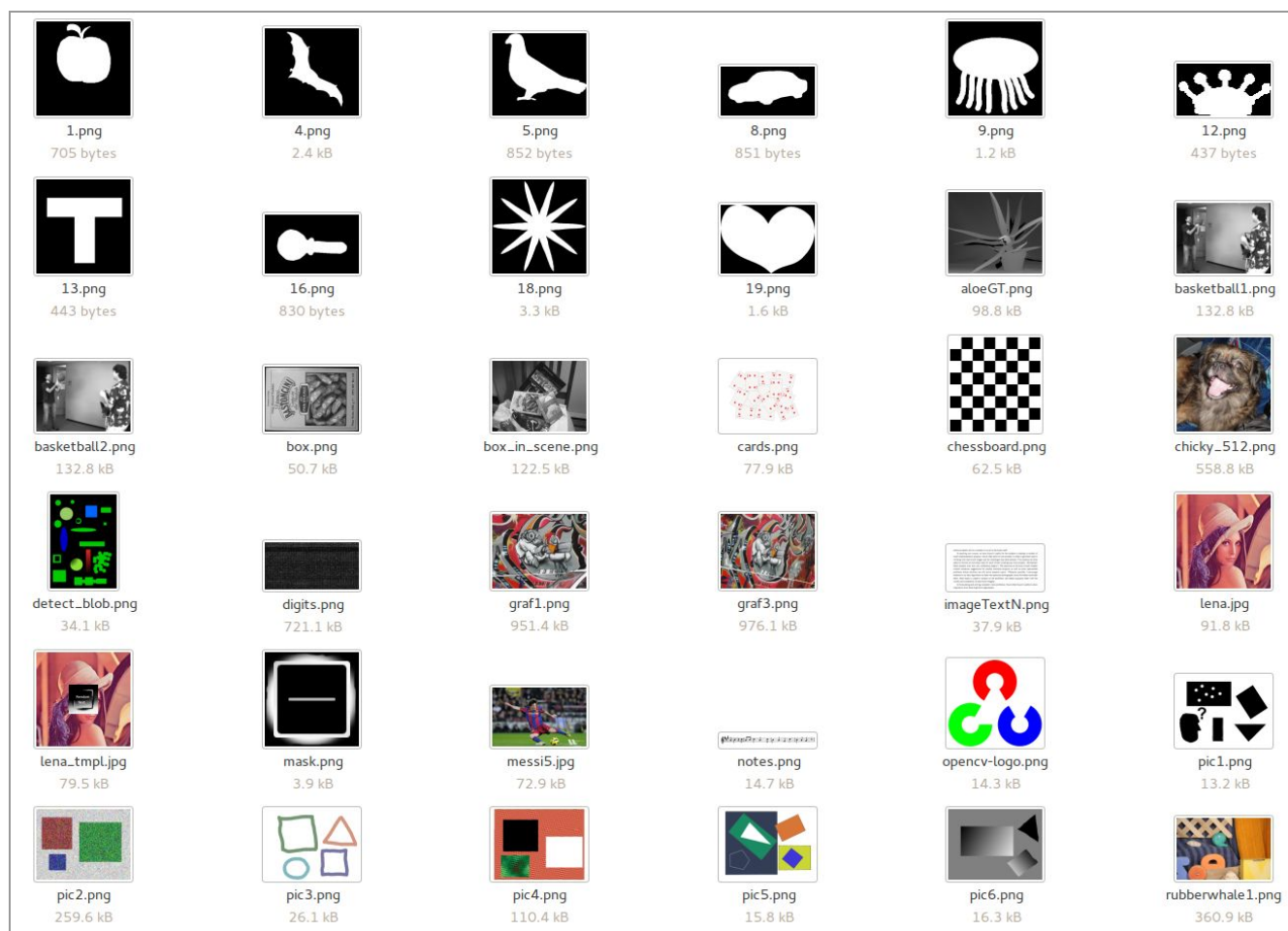
jjones646@gmail.com

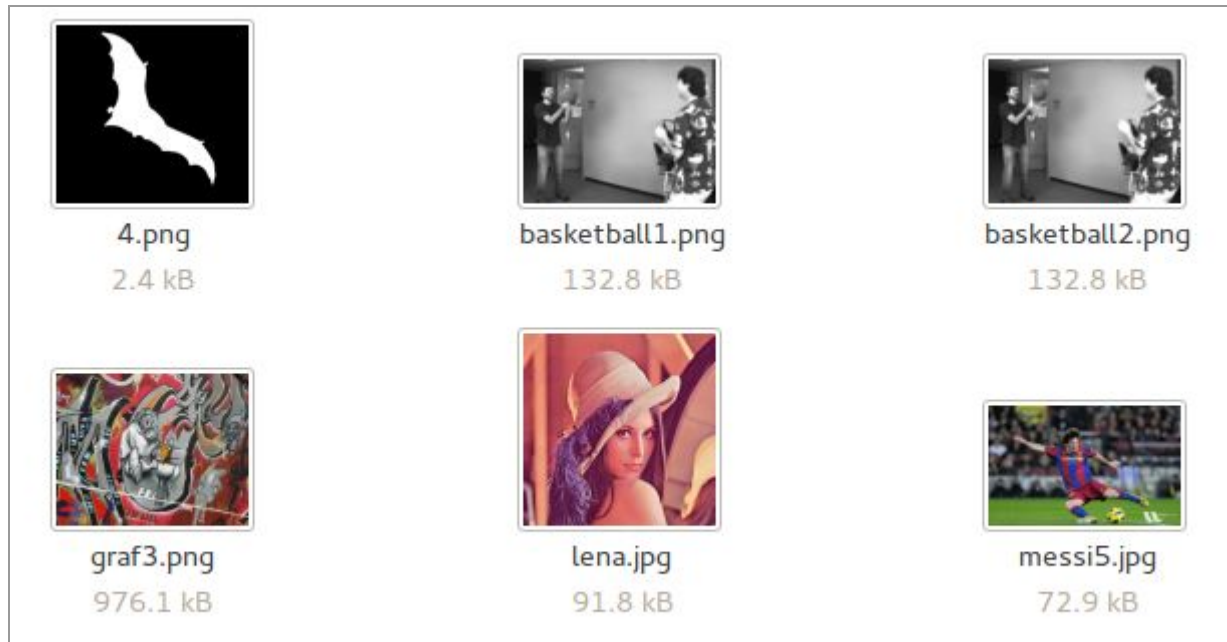
Introduction

This program performs facial feature detection on any given directory of images. You can optionally copy the detected images into a new location if desired. If you've ever heard of Google's [Vision API](#), this program can be thought of as a mini version of that.

The original idea to implement a program like this came about when I was filtering through old hard drives searching for some family photos. Using this program for the 100+ thousand images I came across was the only reasonable approach to finding the pictures I was looking for.

After running the program with the directory of example images, the final results are shown to be acceptable. Shown below is a listing of the example images followed by a listing of images that the program determined to contain a person.





We can see that the program is not 100% accurate, and it's unreasonable to expect that kind of accuracy. However, it detected 4 images correctly and 2 incorrectly (0% false negatives; 6% false positives).

The source code for this program is also hosted on [GitHub](https://github.com/jjones646/humfilt) at the following link:
<https://github.com/jjones646/humfilt>

Workflow

The face detection is performed using OpenCV's [cascade classifier](#) class. Since this is a very computationally intensive task to perform on a large set of images, I added multicore support for increased parallelism.

While the program could estimate how many individuals are in a given image, the algorithm is simplified on purpose so that it can only check if an images has 1 or more individuals in it or not.

There are parameters in the code that can be tweaked to either raise or lower the threshold for determining if a face is detected. These parameters start on line 44 in [humfilt.py](#).

Credits

The example images were taken from the source code for [OpenCV](#), which is licensed under a [BSD license](#).

The [classifier tree](#) used for feature detection was also taken from the OpenCV source code.

Another notable area of credit is the [official OpenCV tutorial about facial detection](#).

Artistic/Technical Emphasis

Artistic: 10%

Technical: 40%

This program is much more technical than artistic. For that reason, most of the weight is shifted towards the technical side of everything. The OpenCV API calls that were used implement areas of feature detection that have been discussed in class. The artistic part of the program is that it can be used to find lost artistic family photos.