TJHSST Computer Systems Lab Senior Research Project
Biometric Security- Face Recognition
2009-2010

Kyle Ferris

June 14, 2010

Abstract

In the modern world, sensitive data or access to buildings can be protected by more than just a key or a password. Biometric data unique to every human can be used to allow or deny access. The purpose of this project is to be able to create a "key" for any person who wishes to use the program. An image of the client's face will be taken and used as the base biometric key. When the client wishes authorization, a new picture of their face will be taken and compared to the base image. The program should be able to recognize the client and authorize him or her, while denying access to those not recognized.
Keywords: computer vision, image recognition, principal component analysis

1 Introduction

1.1 Scope of Study

This project is focused on reducing error involved in determining the similarity of two faces.

1.2 Expected results

This project should be able to fulfill my goals of creating a successful facial recognition algorithm. The expected application of the finished program is to act as a security program. The user will be able to store a base image. Whenever the user wishes authorization, a new image of their face will be taken and used as a comparison image. The program will then evaluate whether, based on the differences between the two images, the user asking for authorization is the same as the one who stored the base image.

1.3 Type of research

My research will be use-inspired basic research. I wish to fundamentally understand numerous computer vision techniques, but I am not seeking knowledge for it's own sake. I am researching this topic in an attempt to create a reliable facial recognition program.

2 Background and review of current literature and research

There are many different approaches to this type of problem. One of the most effective is Principal Component Analysis (PCA). PCA involves identifying the principal features of a face (aka, eyes, mouth, jaw structure) and comparing these features for different images.

3 Procedures and Methodology

There are several tasks that need to be completed for this project. The first is developing a neural network program that can be trained to recognize faces. The second phase is training the network to recognize a specific face. The final phase is testing, redesigning and refining the network until it can achieve fast, accurate results. Not much specialized equipment is needed for this project. However, a web cam might be a useful tool in demonstrating the capabilities of the program and creating image files for use in training. The code will be written entirely in python.

Input data will be a training set of images of the user's face. These images will probably be collected with a web cam taking frames of a video.

Alternatively, these images could be obtained with use of a standard digital camera, although many pictures would need to be taken in order to ensure a sufficiently large training set. One output of my program is a line graph of the data obtained from the mean pixel difference algorith.

To test how accurate my results are, I have created an algorithm that determines the "interference level" between two images. This calculates how viable the given results are, and whether there is another object in the image or a lighting difference that is causing interference. It calculates the standard deviation over subsets of the data, and returns which percentage of these sets have a standard deviation that exceeds some set amount.

4 Primary algorithms and their interaction

Two different algorithms form the bulk of this project. The Mean Pixel Area Comparison, or MPAC, algorithm, is the primary comparison and evaluation method. Given two grayscale images, it will in its internal processing split the image into different groups of pixels. It will then average the pixel intensities in those areas, and calculate the difference between average pixel intensity in equivalent areas in the two images. By default, these groups that the images are split into are 3x3 groups of pixels. However, this is not optimal, as in most images the primary components are not squares.

That is where the second main algorithm comes in. Connected Component Labeling, or CCL, is a method of determining what the primary areas of interest in an image are. It will find groups of pixels of similiar intensity that are considered part of one component of the image, and then "label" them as one component. The output is a list of groups of connected pixels- the components. Then, instead of running the MPAC method with 3x3 pixel groups, the MPAC method can be run with these components instead. This will ensure a more optimal evaluation. Together, these algorithms form a rudimentary Principal Component Analysis method.

5 Results

I ran into problems incorporating the CCL method. While I coded it correctly, the recursion process used presented problems. The code involves recursively going over almost every pixel in an image. This means that there is a new level of recursion for each pixel. However, python has a hard cap of 100 recursions and the images I am using are hundred of thousands of pixels. Therefore this could not be practically incorporated. It was still worthwhile as I used all the techniques involved in CCL, even if the finish was not quite there.

However, the program overall works well. Incorporating CCL successfully would have increased the reliabilty of the results. However, running the simple function without CCL still presents results comparable to the expectations I laid out at the beginning. Therefore, this project can be considered a success.