Face Detection and Feature Recognition

Collection Editor:
Isiana Rendon Escarcega

Face Detection and Feature Recognition

Collection Editor:

Isiana Rendon Escarcega

Authors:

Subhash Doshi
Junjun Huang
Michelle Jin
Isiana Rendon Escarcega

Online:

< http://cnx.org/content/col11250/1.1/ >

C O N N E X I O N S

Rice University, Houston, Texas

Collection structure revised: December 14, 2010
PDF generated: February 6, 2011
For copyright and attribution information for the modules contained in this collection, see p. 21.

Table of Contents

Chapter 1

Starting Direction[1]

There are many features that we can use to locate faces in a picture, like movement, color, and shapes. While in real time systems or video clips motion is used more often, color and shape are more useful in portraits. Many different algorithms are developed based on this idea, such as Schneiderman and Kanade, Rowley, Baluja and Kanade. The one we chose as our base is Viola-Jones, which was motivated primarily by the problem of face detection. Its main benefit comes from its reduced computational requirements so much so that it can be used in real time. The way it works is as follows. The features employed by the detection framework involve the sums of image pixels within rectangular areas, that is the value of any given feature is always just the sum of the pixels within clear rectangles subtracted from the sum of the pixels within shaded rectangles.

[1]This content is available online at <http://cnx.org/content/m36383/1.1/>.

Chapter 2

Results[1]

Depending on the parameters we set, the algorithm works well to detect most fully frontal facial portraits in pictures. To detect faces more accurately and not skip actual faces, we decrease the step size of the detector as it runs through the image. While this generally detects more faces that are actually faces, it also increases the calculation time. The calculation time varies depending on the types of faces in the image, the lighting of the picture, the angles of the faces, the resolution of the image, the surrounding objects, the step sizes, the minimum and maximum scaling of the detector and the number of faces.

[1] This content is available online at <http://cnx.org/content/m36382/1.1/>.

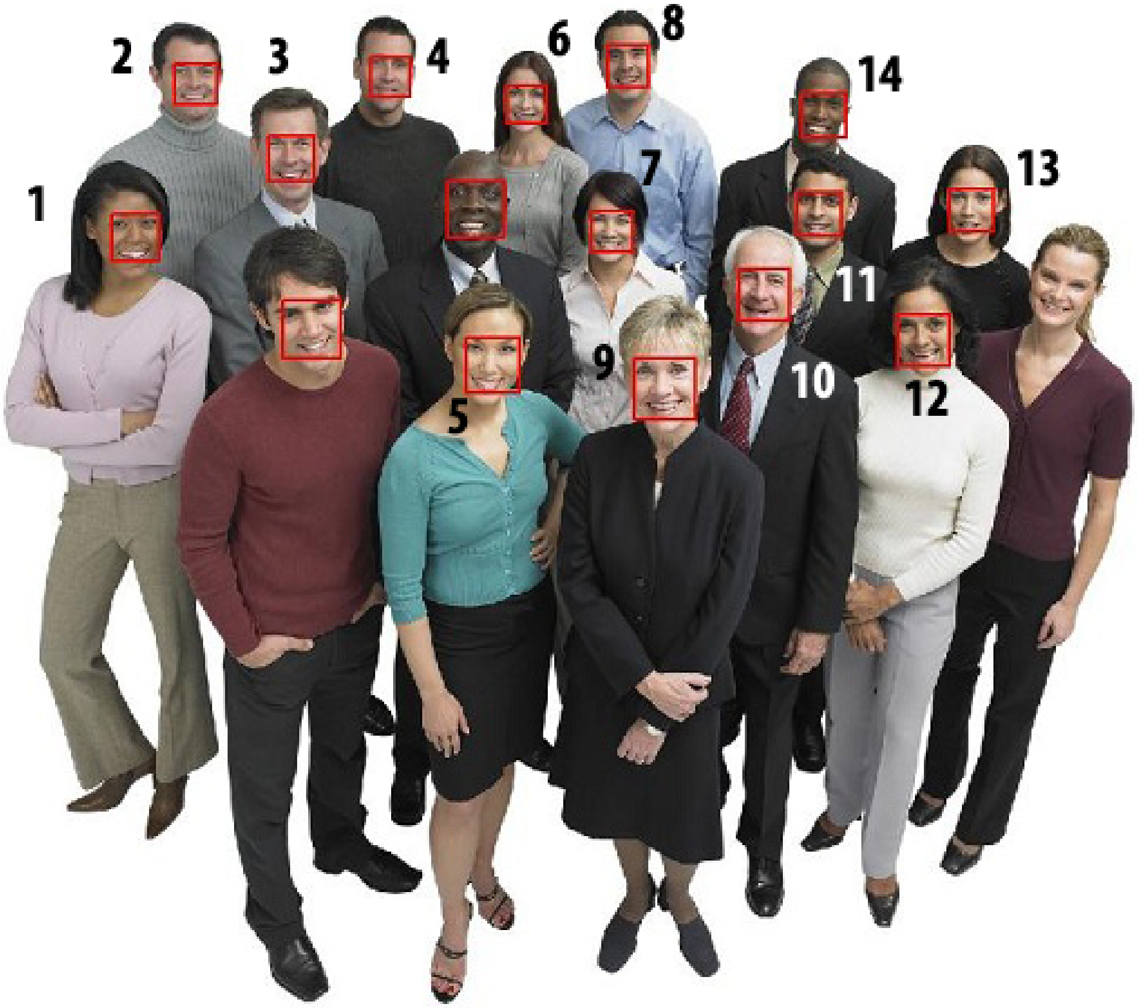

Figure 2.1: Numbers match face intensity values below.

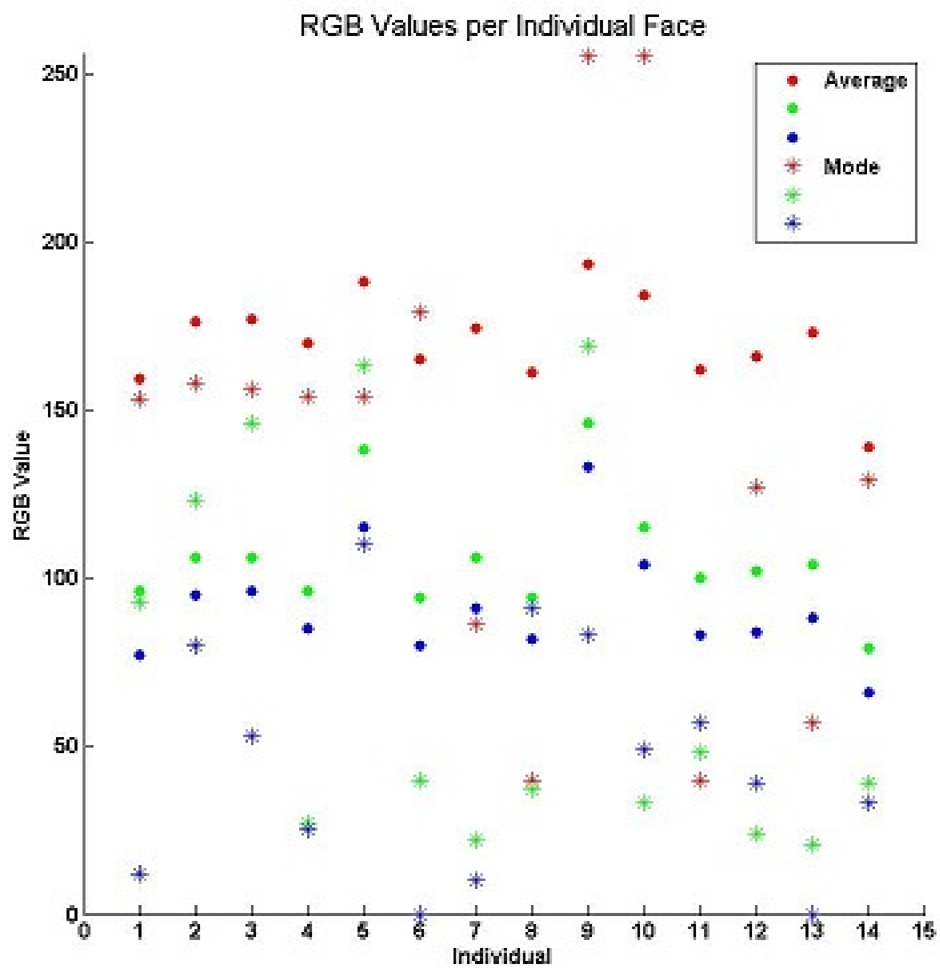

Figure 2.2: Face intensity. Results match numbered picture above.

The algorithm does not work to detect faces that are angled or tilted, as most of the faces it detects are either straight on or only very slightly tilted. Furthermore, it sometimes detects non-faces as faces as well, including objects like clasped hands, certain areas of denim jeans, and other non-face objects. For the facial color characterization, we decided to use an average of each of the RGB values to characterize the color and shading of each face. This works to determine the general color of each face, however, it also depends on the lighting of the picture and the lighting of each face. This does not give the "true average" of each face either, as it takes features such as mouths, eyes, and noses into account. However, it does give the relative color of each face according the RGB values. People with significantly different face colorings have very different RGB values. For hair color, using the average of the RGB values works to an extent, but not as well as it does on the face. This is partly due to the fact that we can only get a determinant rectangle around the area where the hair is. As mentioned before, we approximated the location of hair to the top of the head, as this is generally where most people who have hair on their heads will have some hair). We scaled the box according to the size of the face it is associated with, and set the hair rectangle determinant to 0.8 times the height of the face above the facial rectangle. While this gets a fairly accurate general location, it does not work exactly on different faces, as people have different amounts of hair at the top of their heads, different forehead lengths relative to their face sizes, and other varying factors. Thus, the rectangle will usually include some features that are not hair, such as parts of a person's forehead, the background behind the person's head, etc. This changes the average of a person's hair color, giving them a non-exact hair color average. To better detect hair color, we also implemented the mode method, by finding the modes of the RGB values of each rectangle of hair. This would eliminate background "noise", or objects that are not hair, since in general most of the rectangle contains hair. However, the problem with the mode method is that oftentimes the shading of the hair is very different that the modes found do not accurately represent the hair color. Furthermore, if multiple modes exists, then Matlab will automatically output the smallest valued mode as the actual mode, and the RGB values from this output is not entirely accurate for hair color determination.

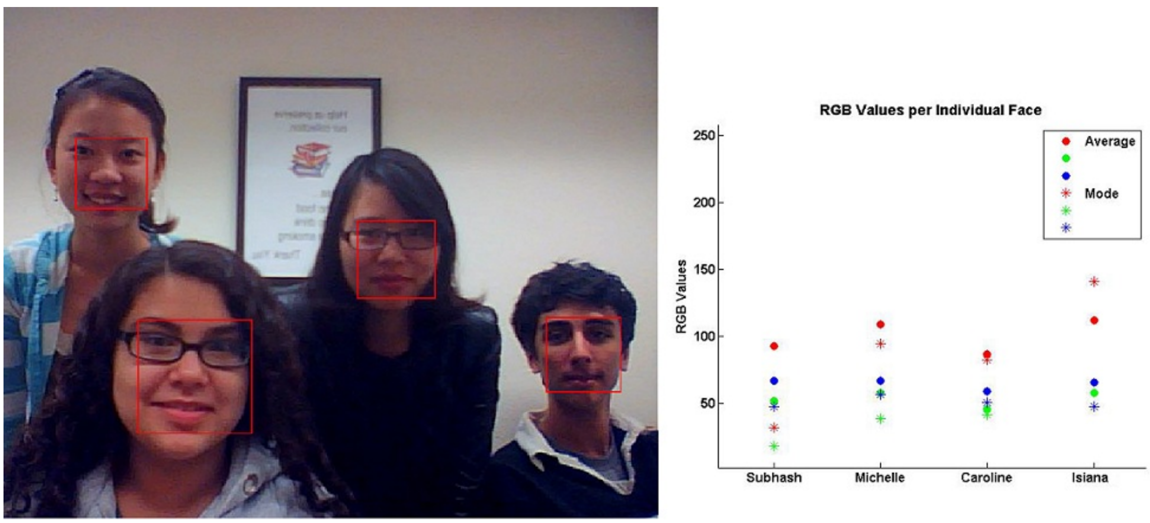

Figure 2.3: Side by side comparison of our group's faces and their intensities.

Chapter 3

Motivation and Applications[1]

The motivation behind this project is that facial detection has an amplitude of possible applications. From common household objects like digital cameras that automatically focus on human faces to security cameras that actually match a face to a person's identity. Webcams are often used as a security measure for locking a personal computer. The webcam's facial recognition technology allows for the computer to be accessible to the user only if it recognizes their face.Cameras can also use this technology to track human faces and keep a count of the number of people in a shot or in a certain location or even coming in through an entrance. This technology can be further narrowed down to the recognition and tracking of eyes. This would save power by dimming a screen if viewer is not looking. For this project, we hope to use an already existing algorithm as a basis for face detection and build upon it to create improvements and explore more data.

[1] This content is available online at <http://cnx.org/content/m36381/1.1/>.

Chapter 4

Implementation & Solutions[1]

Our starting base for the Viola-Jones algorithm was the Matlab code by Vahid Kazemi available here[2] . While Kazemi's code provided a strong basis for detection of front-face potraits, we made several additions and adjustments to implement better and more feature detection techniques. We adjusted several parameters to optimize facial detection. In our case, timing was less of a priority, and we decided we could afford a longer calculation time in order to gain more accuracy in facial detection. To do this, we decreased the step size of the detector moving across an image to match it against the training data. We also made changes to several other parameters, such as scaling size of the picture, and the minimum and maximum tolerances we had for scaling to optimize the data. To build upon the Viola-Jones algorithm, we also wrote code for greater feature detection. We wrote a program for the detection of the colors of faces. The Viola-Jones algorithm detects the locations of each face and forms a rectangular box around the face based on its size. This box stretches from the top of the eyes to the bottom of the mouth and across the width of the face. By finding the matrix of Red, Green, and Blue (RGB) values across each rectangle, we can find the average color and shade of each face. We can then average the RGB values for each face and plot them separately to characterize the different facial colors and shades. We have also used this color recognition for hair color detection. Unlike face color though, the Viola-Jones algorithm does not form a box around the hair for each person. Moreover, the locations of hair for each individual varies. For example, while those with long hair will have much of their hair on the two sides of their head, while those with shorter hair will not. In this case, we determined that most people with hair have some hair directly at the top of their heads. We then used formed determinants of rectangles above the face locations to where the top of the head hair locations were, and averaged the RGB colors in those determinants.

[1] This content is available online at <http://cnx.org/content/m36379/1.1/>.
[2] http://www.mathworks.com/matlabcentral/fileexchange/27150-face-detector-boosting-haar-features

Figure 4.1: Example of hair detection.

Chapter 5

Futher Improvements & Conclusion[1]

Our current program has a difficult time recognising faces that are not completely or mostly facing forward. An improvement to this would be most beneficial since you can't always guarantee a forward, un-angled portrait. Lighting is also an issue for our program. If a face or even half of a face is darkened by shadow the algorithm might not recognize it due to the lack of difference between the intensities of certain facial features, such as eyes and cheeks. Blinking is also an issue for this program. If a person is blinking or squinting the program does not recognize the face because it can not identify its eyes. As always, computation time can also be improved. Going beyond merely detecting faces and heading into recognition, this technology can be used as added security. By placing a camera at each entrance of a building, you can use facial recognition to identify a person walking in and store this identity to be able to recognize the same person walking out. With this you can calculate the amount of time spent inside the building and can use this to correlate time spent with shifty behavior.

[1] This content is available online at <http://cnx.org/content/m36378/1.1/>.

Chapter 6

Difficulties[1]

Some of the difficulties we face heading in to this project deal mainly with the picture quality. If a person is covered by shadow or if lighting is uneven, the algorithm won't work properly since it uses differences in light to detect a face. Additionally, if a person's face is tilted or if part of their face is obstructed by an object detection will not take place. Movement such as turning or blinking is also an issue.

[1]This content is available online at <http://cnx.org/content/m36377/1.1/>.

Chapter 7

Algorithm[1]

In a standard 19x19 pixel sub-window, there are a lot of possible features, in this case we are only using 500 of them since it would be prohibitively expensive to evaluate them all. Thus, the object detection framework employs a variant of the learning algorithm adaptive boosting to both select the best features and to train classifiers that use them.

Example

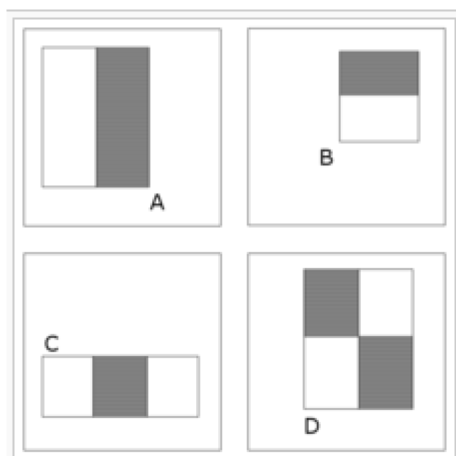

Figure 7.1: Example features in Viola-Jones

Viola-Jones closely resembles Haar basis function because it operates by summing the pixel values in rectangular areas of the image and then comparing the difference between the summations in the black and white triangles to the training data. It allows for possible detection of different sizes of faces by having a rectangular window detector that moves across the picture. The detector is capable of detecting faces, and its size changes with each scan. For the algorithm we use in the project, we can manually adjust the scaling depending on the size of the image. Unlike other facial recognition algorithms, which scale the picture each time and runs it through the detector, Viola-Jones scales the detector, allowing for faster calculation. The Viola-Jones calculation also contains a training data of faces and non-faces. For the algorithm that we use in this project, we have 2000 faces and 4000 non-faces in the training data.

[1]This content is available online at <http://cnx.org/content/m36375/1.3/>.

Chapter 8

The Team[1]

Subhash Doshi is a junior ECE student at Rice with a focus on signals and systems. Michelle Jin is a junior ECE student at Rice with a focus on computer engineering. Caroline Huang is a junior exchange student from Fudan University, Shanghai, China. Isiana Rendón is a junor ECE student at Rice with a focus on computer engineering.

[1]This content is available online at <http://cnx.org/content/m36384/1.1/>.

The World's Cutest Dog

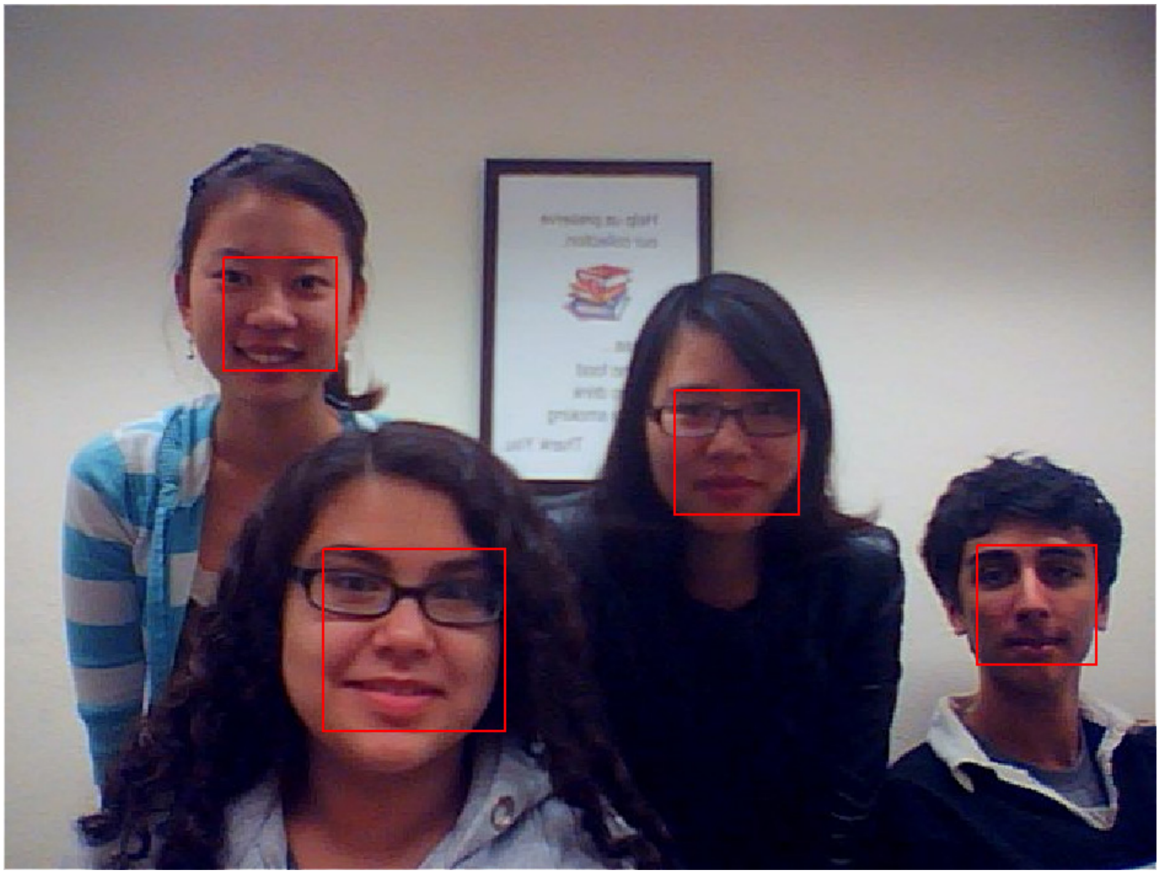

Figure 8.1: Successful face recognition!

Chapter 9

Introduction & Background[1]

Face detection is a computer technology that determines the locations and sizes of human faces in images. It detects facial features and ignores anything else, such as buildings, trees and bodies. There are many ways to accomplish this. For example, it can be done by color,motion or combination of these. For this project, we are trying to accomplish this in a model-based way using Matlab. Face models usually contain the appearance or shape of faces. They work together as a training base and teach the system how to classify the portions of an image, which in our case is rectangular, at all locations and scales, as either faces or non-faces. Hence it can calculate how many people are there in a specific picture. Since the face base we are using is composed of frontal human faces, it works best for portraits of a group of people. Furthermore after we locate faces, we can tell people's races by their skin and hair color.

[1]This content is available online at <http://cnx.org/content/m36380/1.1/>.

Index of Keywords and Terms

Keywords are listed by the section with that keyword (page numbers are in parentheses). Keywords do not necessarily appear in the text of the page. They are merely associated with that section. *Ex.* apples, § 1.1 (1) **Terms** are referenced by the page they appear on. *Ex.* apples, 1

Attributions

Module: "Algorithm"
By: Isiana Rendon Escarcega, Michelle Jin, Junjun Huang, Subhash Doshi
URL: http://cnx.org/content/m36375/1.3/
Page: 15

Module: "The Team"
By: Isiana Rendon Escarcega, Michelle Jin, Junjun Huang, Subhash Doshi
URL: http://cnx.org/content/m36384/1.1/
Pages: 17-18

Module: "Introduction & Background"
By: Isiana Rendon Escarcega, Michelle Jin, Junjun Huang, Subhash Doshi
URL: http://cnx.org/content/m36380/1.1/
Page: 19

Face Detection and Feature Recognition
Collection of modules for ELEC301 Viola-Jones-based facial detection and feature recognition project.

About Connexions
Since 1999, Connexions has been pioneering a global system where anyone can create course materials and make them fully accessible and easily reusable free of charge. We are a Web-based authoring, teaching and learning environment open to anyone interested in education, including students, teachers, professors and lifelong learners. We connect ideas and facilitate educational communities.

Connexions's modular, interactive courses are in use worldwide by universities, community colleges, K-12 schools, distance learners, and lifelong learners. Connexions materials are in many languages, including English, Spanish, Chinese, Japanese, Italian, Vietnamese, French, Portuguese, and Thai. Connexions is part of an exciting new information distribution system that allows for **Print on Demand Books**. Connexions has partnered with innovative on-demand publisher QOOP to accelerate the delivery of printed course materials and textbooks into classrooms worldwide at lower prices than traditional academic publishers.