Visual and Facial Recognition Algorithms: Origins and Applications

Donn Puckle
Dec 9th, 2001
Version 2.0

Introduction. This paper is a brief discussion of the advances made in the field of visual and facial recognition. What's discussed is background into a few of the types of visual recognition systems available commercially, the algorithms that are used to drive those systems, how these algorithms are constructed, information on the adaptation of biological facial recognition models in current facial recognition systems, and how those systems are being put to use to aid in security.

Background. How do you get a computer to see? Really all that is necessary is to connect a camera to a computer and point it out at the world. What's returned is an array of pixels, two dimensional for gray scale from 0 (black) to 255 (white) and three dimensional for color (red, green, blue). Simple and straight forward really. The computer takes the picture and transfers into a language (mathematics) that it can make sense of. But, how do we instruct our computer to take those numbers and interpret them so that they represent something other than pixels? Where we can look at a cow and know it's a cow (or chicken if our parents had a cruel sense of humor) the computer, without instruction, looks at a cow and sees 10110010 11000101 etc. What translates these numbers into something recognizable is an embodiment of a set of rules that operate on a collection of columns of numbers (1) more commonly known as an algorithm. The more sophisticated the algorithm the better ability the computer has to differentiate a cow from a horse. The problem of telling a cow from a horse makes up most of what is called pattern recognition and it's a very complex and difficult thing to teach a computer. Even more difficult for the computer are the problems of telling one cow from another or determining where the cow ends and the background begins. This last concept is known as "edge definition." This unique problem falls into the category of shape recognition.

A brief discussion of shape recognition algorithms. One of the most common problems of pattern recognition is shape similarity. Consider a child's picture book and the way that children learn. The child reaches a point where she can identify the pictures in a book with those in the real world, two-dimensional as compared to three-dimensional. This is generally the same way that a system that does optical character recognition (OCR) works. The system maintains a known library of shape models that represent ASCII characters and obtains unknown shapes from a scanned document. It then attempts to match the unknown shape to its closest representation from the library (2). The primary problem of shape similarity is that defining the term "similar" is application specific. Even after one decides on an algorithm that applies to particular problem, it's still difficult to fine-tune that algorithm to achieve peak performance. Comparing characters to characters, faces to faces, or fingerprints to fingerprints all require unique algorithmic approaches. No one shape-matching algorithm can solve all of these problems. Sometimes, it's better to use an existing algorithm than to reinvent the wheel. Some shape similarity algorithms that are freely or commercially available for use are:

- The Hamming Distance Algorithm. Measures the area of symmetric difference between two overlaying polygons. This is accomplished by finding the union or intersection of the two and then computing the areas. If the two polygons are identical the Hamming distance will be zero. This algorithm works best with applications used to compare well-defined shapes of identical size. It's not very effective on anything else because of its crude notion of what a shape is.

- Thinning Algorithms. These algorithms use a method known as medial-Axis transformation or thinning (2) to extract a skeleton of the objects to be compared. Then it's just a matter of comparing tree topology and edge length/slope of the two skeletons. This can work well with basic facial recognition applications and other applications requiring three-dimensional shape comparisons.

- Neural Networks. While not exactly an algorithm in and of themselves, neural networks are also used for shape comparisons. This method has a number of unknown variables and doesn't always return the results expected. Meta-data of the object shape (size, area, number of edges, etc.) is collected and used as training data for the neural network to produce what's called a classification function. The classification function accepts these values as input data and returns an educated 'guess' of what the shape might be. The results are deterministic and based upon the complexity of the neural net.

Okay, so we now have the means available to train our computer to interpret what it's seeing, or at least the ability to compare the shape of what it sees with what it has in a database. In terms of actual visual recognition we are still in the primitive stage. It would be like wearing sunglasses at dusk and being able to tell from the shape of a cow that it's a cow. You don't know what color the cow is, whether it has one good eye or two, if it has spots, or even if it's male or female. But, the human brain and the cells that are actually used to identify faces are so sophisticated that you can view the picture of a person whom you've never met before and identify that individual in a crowd. How is this mapped as an algorithm? How did the FBI manage to identify individual criminals out of the 50,000 or so fans that attended the superbowl this year?

<u>Facial Recognition and Biologically Inspired Approaches.</u> How are we able to recognize one person from another even after having seen them only once or after only viewing an image of that person? The human brain possesses specialized hardware referred to as "face detector cells" to detect and remember faces. These face detector cells, located in the inferotemporal cortex and regions in the frontal right hemisphere of the human and primate brain are used specifically to identify faces **(3)**. Even more interesting is that these cells seem to be divided into cells that are used to interpret facial expressions and cells whose primary responsibility is to establish facial recognition. Findings of research conducted by J. Konorski as far back as 1968 established that certain groups of cells are able to detect edges and lines and then send this data to a pyramid of higher level cells **(4)** that use it to form other more complex cells. These newly created cells called gnostic units respond very selectively to a small range of inputs. Some of these cells responded to simple line drawings of faces, but did not respond well to faces with rearranged facial features, even if all of the features were present. At the very top of this pyramid was a cell that would only fire when, for example, you saw your grandmother. This is a somewhat simplified explanation in that this cell may only fire when it sees a frontal view of your grandmother's face and not when a profile view was presented. There may be, and this has not yet been verified, other cells that respond to non-frontal views of the same face. So much of the human brain is uncharted that these issues are still unresolved. The hard question for facial algorithm developers is "how do we model cell behavior when we know so little about it"? There have been several different approaches to modeling facial recognition on a computer based on what is known of human facial recognition. While some of these algorithms work better than others they all have as their common origin, research on human and primate behavior. Two of these algorithms are outlined below.

The Eigenfaces Algorithm. The Eigenfaces algorithm is the work of Matthew Turk and Alex Pentland who applied the use of a method called "principle component analysis" to facial recognition. In principal component analysis, faces are first aligned with each other and then treated as high dimensional pixel vectors from which eigenvectors (a.k.a. Eigenfaces) are computed, together with the corresponding eigenvalues **(5)**. These in turn are submitted as training data to a neural network as described previously. In simpler terms what this means is that the peaks and valleys of a person's facial features are translated into a mathematical vector and then compared with those vectors already known to the neural network. Originally, the principle component analysis treated an entire face as one vector value but when fine-tuning was done on the algorithm, the facial representation was broken into much smaller areas and vector values were assigned to each providing for a higher success rate. Results that returned non-recognition values were again submitted as training data to the neural network so that they could be used for future comparisons. While having advantages such as high recognition speed and ease of implementation, the Eigenfaces algorithm does have its disadvantages. For example the altering of facial features by a change in hairstyle, addition of subtraction of facial hair, or long-term facial changes due to aging can often cause the algorithm to return false results. These shortfalls can be overcome however by introducing new training data images to the neural network. Research using the Eigenfaces algorithm is ongoing and is becoming more and more refined. Several internet-based Eigenfaces and Eigenfeatures (an additional layer of description in terms of facial features), projects are on-going. The Photobook experiment/demonstration uses a relatively large database of 7562 images of approximately 3000 people (6) along with the eigenvector meta-data describing the image. The principal components analysis for this experiment was done on a sampling of 128 facial images representing the 3000 people contained in the database. An interactive demo is available at the web site referenced in this paragraph (see reference 6). The Eigenfaces Group website provides an excellent and very understandable explanation of how they constructed their Eigenfaces database, and how they taught their computer to learn a face (7). They've even provided the source code, written in the Python programming language for both study and expansion. If you're interested in studying and playing with facial recognition software, the site referenced in reference 7 is a good place to start.

Feature-Based Recognition. Another popular method often used in facial recognition systems is feature matching: deriving distance and position features from the placement of internal facial elements **(3)**. This is one of the earliest facial recognition algorithms and is based on automatic feature detection and has been championed by Peter Yianilos of Princeton University. The algorithm localizes the corners of the eyes, nostrils, etc. in frontal views, and then compares the computed parameters for each face against the parameters for known faces. The author of the cited source **(8)** reports that the initial grid-matching phase now (1999) takes about 30 seconds on a Sparc10 and that the actual recognition phase takes less than a second. This algorithm has been shown to be quite accurate, recognizing 98% of all frontal images presented, 84% of all profile images and 57% of half-rotated faces. These last numbers represent a very good performance considering how difficult profile and half-rotated cases are for facial detection and recognition systems. Other feature-based recognition algorithms produced similar results. The mixture-distance algorithm, a feature-based recognition technique, produced recognition results of 95% accuracy on a database 685 people where each face is represented by 30 measured distances. This is currently the best-recorded recognition rate for a feature-based system applied to a database of this size. The one disadvantage to this method is that grid matching for the first 70 images must be done manually before the automatic graph matching can be done reliably.

Humans vs. Algorithms. Case studies conducted by Hancock, Bruce and Burden to see how effective the Eigenfaces and feature-based recognition algorithms were when compared with human subjects turned up interesting results **(9)**. The Eigenfaces algorithm turned out to be much more sensitive to changes in facial features than originally predicted. The system using the Eigenfaces algorithm completely failed to recognize one of the faces presented to it. Upon analysis, it was found that a mirror image of the face had been presented. Neither the human subject nor the feature based recognition system detected this error and both were able to readily recognize the face. To give it some credit though, the Eigenfaces algorithm performed equally as well as the human when presented with frontal facial features, its strongest suit. While not performing quite as well in the frontal facial features test case, the feature-based recognition system actually outperfomed the human subject in the facial hair to non-facial hair test case. The conclusion drawn from these test cases was that both the Eigenfaces and feature-based systems would do equally as well when used for grouping images in criminal identification applications.

Conclusion. Visual recognition, especially facial recognition is a difficult problem. While the researchers involved with these algorithms publish high success rates, most of the results are from static, high quality, frontal facial features tests such as the Photobook and Eigenfaces Group demonstrations. The developed algorithms need to be improved upon to overcome the problems of profile images, lighting variations, and natural changes to a face. While researchers derive inspiration from studies of human facial recognition, results are still inconclusive as to whether they are good models for computer algorithms. For one thing, there's still not enough known about how human facial recognition cells work, whether there's a natural hierarchy of cells which build up to recognition or even where in the brain all of these cells might reside. Perhaps the correct answer is to be inspired by research into human facial recognition but not to try and do a direct translation to the computer. There are many examples throughout technology where man has attempted to imitate nature and has failed. Flight experiments are one. Whereas the Bernoulli principal applies to both man-made flight and the flight of birds, you will not find any aircraft with flapping wings. Visual recognition and other biometric studies may soon give the security world powerful tools to protect systems, facilities and people. The use of facial recognition algorithms coupled with fingerprinting analysis and retina scans may someday make our facilities less prone to illegal entry. But, we must be prudent in our use of these systems lest we end up in a society such as that portrayed by George Orwell in his novel 1984.

References:

1. Adler, Michael D., "An Introduction to Pattern Recognition: Statistical, Neural Net and Syntactic Methods of Getting Robots to See and Hear", Sep 19, 1997

2. Skiena, Steven S., "The Algorithm Design Manual", Pp. 386 - 387, 1998, Springer-Verlag

3. Sukthankar, Gita, "Face Recognition: A Critical Look at Biologically-Inspired Approaches", May 7, 1999
http://www.cs.cmu.edu/~gitars/16-721/final/final.html

4. Konorski, J., "Integrative Activity of the Brain. An Interdisciplinary Approach", University of Chicago Press, 1967.

5. Turk, M. and Pentland, A., "Eigenfaces for recognition", Journal of Cognitive Neuroscience, 3(1):71—86, 1991

6. http://www-white.media.mit.edu/vismod/demos/facerec/basic.html

7. http://www.owlnet.rice.edu/~elec301/Projects99/faces/algo.html

8. Kanade, T. "Picture Processing by Computer Complex and Recognition of Human Faces", PhD thesis, Kyoto University, 1973

9. Hancock, P., Bruce, V., and Burton, M., "A comparison of two computer-based face identification systems with human perceptions of faces", Vision Research, 38(15-16), 1998.

Last Updated: October 24th, 2017

Upcoming Training

SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
San Diego Fall 2017 - SEC401: Security Essentials Bootcamp Style	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Gulf Region 2017	Dubai, United Arab Emirates	Nov 04, 2017 - Nov 16, 2017	Live Event
Community SANS Vancouver SEC401^	Vancouver, BC	Nov 06, 2017 - Nov 11, 2017	Community SANS
Community SANS Colorado Springs SEC401~	Colorado Springs, CO	Nov 06, 2017 - Nov 11, 2017	Community SANS
SANS Miami 2017	Miami, FL	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Sydney 2017	Sydney, Australia	Nov 13, 2017 - Nov 25, 2017	Live Event
SANS Paris November 2017	Paris, France	Nov 13, 2017 - Nov 18, 2017	Live Event
SANS San Francisco Winter 2017	San Francisco, CA	Nov 27, 2017 - Dec 02, 2017	Live Event
SANS London November 2017	London, United Kingdom	Nov 27, 2017 - Dec 02, 2017	Live Event
Community SANS St. Louis SEC401	St Louis, MO	Nov 27, 2017 - Dec 02, 2017	Community SANS
Community SANS Portland SEC401	Portland, OR	Nov 27, 2017 - Dec 02, 2017	Community SANS
SANS Khobar 2017	Khobar, Saudi Arabia	Dec 02, 2017 - Dec 07, 2017	Live Event
SANS Munich December 2017	Munich, Germany	Dec 04, 2017 - Dec 09, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Dec 04, 2017 - Dec 09, 2017	Community SANS
SANS Austin Winter 2017	Austin, TX	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS vLive - SEC401: Security Essentials Bootcamp Style	SEC401 - 201712,	Dec 11, 2017 - Jan 24, 2018	vLive
SANS Bangalore 2017	Bangalore, India	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Cyber Defense Initiative 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Dec 14, 2017 - Dec 19, 2017	vLive
SANS Security East 2018	New Orleans, LA	Jan 08, 2018 - Jan 13, 2018	Live Event
Community SANS Hawaii SEC401	Honolulu, HI	Jan 08, 2018 - Jan 13, 2018	Community SANS
Community SANS Nashville SEC401^	Nashville, TN	Jan 08, 2018 - Jan 13, 2018	Community SANS
Mentor Session - SEC401	Memphis, TN	Jan 09, 2018 - Mar 13, 2018	Mentor
SANS Amsterdam January 2018	Amsterdam, Netherlands	Jan 15, 2018 - Jan 20, 2018	Live Event
Northern VA Winter - Reston 2018	Reston, VA	Jan 15, 2018 - Jan 20, 2018	Live Event
Mentor Session - SEC401	Minneapolis, MN	Jan 16, 2018 - Feb 27, 2018	Mentor
SANS Las Vegas 2018	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	Live Event
Las Vegas 2018 - SEC401: Security Essentials Bootcamp Style	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	vLive
SANS Miami 2018	Miami, FL	Jan 29, 2018 - Feb 03, 2018	Live Event