Facial Recognition Using PCA

David Huie
Harvey Mudd College

Dustin Rodrigues and **Alex Ruch**
Pomona College

Abstract

Facial recognition is an important area of research of computer science with obvious implications. As one would imagine, it is not a simple task. A large number of variables come into play including facial expressions, hairstyles, camera angle, and lighting. Our paper in particular examines Principal Component Analysis (PCA). We used databases that contained faces with different lighting and with different face angles. The PCA algorithm worked exceptionally well on classifying these data sets.

Introduction

Facial recognition is a varied area of study with applications to security, biometrics, and personal use. In terms of security, one could imagine a situation where an authority has a database of images of faces and security footage of a criminal. It is rarely feasible for a human to manually comb through a database, but a sophisticated enough facial recognition system could possibly determine the identity of the criminal. In biometrics, using a camera to confirm a subject's identity is cheaper than a fingerprint reader or an iris scanner, and is more convenient than having to manually enter a password. The personal uses of facial recognition include training some images of a photo album based on user-based tags and then automatically tagging any new photos.

However, all of these applications have clear deficiencies, some of which come from variations of facial expressions, hairstyles, camera angles, and lighting. These factors can contribute to a significant amount of inaccuracy in facial recognition. For personal uses such photo tagging, a wrongly tagged photo is not a very large issue. However, if an authorized user cannot log into their own system because they recently got a haircut, or even worse, an unauthorized user was able to log in, this is a significant security deficiency.

One of the challenges of facial recognition is the extraordinarily large space of possible images. For instance, a 100×100 image will have 10 000 pixels. A naïve method would be to take an image we want classified and compare it against known images using some notion of distance. However, given the aforementioned size of the data, this will be a very computationally intensive task. Instead, Principal Component Analysis (PCA) is used to extract the features with the most amount of variance. Using PCA, one can reduce the space from a dimension of 10 000 to something much more manageable, such as 10 or 20 dimensions. Once these principal components are found, we can project an image to be classified onto the principal components and determine which face it most closely resembles.

Algorithm

The PCA Algorithm extracts a set of eigenvectors from a set of vectors P such that the eigenvectors decreasingly account for the variance in P. In more detail, consider a set of mean centered vectors V. That is, the mean vector is calculated from V and is subtracted from each $v_i \in V$. The PCA Algorithm then returns a set E of M eigenvectors e_i such that the quantity $\lambda_i = \frac{1}{M}\sum_{n=1}^{M}(e_i^T v_n)^2$ is maximized. These eigenvectors correspond to the principal components for the v_i's. The eigenvalues that maximize this equation are found from a covariance matrix created from V. The covariance matrix for V would be $C = WW^T$, where W contains the vectors $v \in V$ as column vectors. It is then known that the eigenvalues of C are equal to the e_i's specified above. In practice, it is not very feasible to calculate the eigenvalues of C directly because it often is of an enormous size. It is known, equivalently, that the matrix $W^T W$, which is much smaller than C, shares the same eigenvalues and eigenvectors as C. This different representation is usually used by the PCA Algorithm when computing principal components.

In the context of images, the PCA Algorithm would take in as input a matrix W, which is composed of a training set of mean centered images $V = \{v_1, v_2, \dots v_m\}$ encoded as its column vectors. This is done by creating a one-dimensional vector out of the matrix representation of each image. The set of principal components, or eigenvectors, e_i of the covariance matrix for W represents the training data for a classifier built on PCA. Given an image vector $v_t \notin V$, the image most similar to v_t, $v_i \in V$, is found by first computing $\Omega_i = [d_1 d_2 \dots d_M]$ with $d_s = e_s^T v_i$ for $i \in \{t\} \cup \{1, 2, \dots, M\}$. Then, if we minimize $||\Omega_t - \Omega_i||$ for all i, the v_i corresponding to the minimum belongs to the most similar image to v_t. For facial recognition purposes, the person represented by v_t would be identified as the person in the image v_i.

Table 1: Results (average of 10 trials)

	Yale	Yale (modified)	Yale B
Average Accuracy	.773	.955	.998
Faces tested	20	20	130

Results

To conduct our tests, we randomly partitioned the images into 80% training data and 20% testing data. The unmodified Yale Database contained images with varying facial expressions as well as lighting that created shadows that significantly obscured the images. When trying to classify these images with the shadow, accuracy was low because the region of the shadow made it more similar to other images with the shadow as opposed to an image of an actual face, in particular, the correct person's face. Once we removed the the images with the shadows from the first database, our system classified the face correctly with $> .95$ accuracy during repeated trials with random partiions.

Yale Face Database B contained various images of a person under a strobe light. Once again, our system classified the face correectly with $> .99$ accuracy in repeated trials. Even though there were lighting variations within the set of images for each person, a test image would be sufficiently close to an image with similar enough lighting that classification was for the most part very successful.

Precision, recall, and f1 were initially calculated for each of the 15 people in the Yale Database and each of the 10 people in Yale Database B, but this data was statistically insignificant because most entries were 1 with different faces each trial having values that represented one misclassification.

Conclusion

As our results showed, facial recognition on a controlled data set can work exceedingly well. Possible areas for further research include testing how well the algorithm performs on other variations of images. In the real world, images will not all be so similar, so PCA is not suitable for general-purpose use, but it still can be used in controlled environments. There are other facial recognition systems that can better handle these variations in images, yet even the best ones have some deficiencies, whether it be accuracy, time, or space. Facial recognition is a continually-evolving field facing numerous challenges, but it has continued to improve over the years. In fact, advanced facial recognition systems using mathematical texture analysis can differentiate between identical twins according to Ralph Gross, a researcher at the Carnegie Mellon Robotics Institute. This is one of the many possible directions in which this growing field may go.

Acknowledgements

We used Yale Face Database and Yale Face Database B.

References

Georghiades, A.; Belhumeur, P.; and Kriegman, D. 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence* 23(6):643–660.

Kim, K. 2003. Face recognition using principle component analysis. *DCS, University of Maryland, College Park, USA*.