

## Face Recognition

Deepika Garg<sup>1</sup>, Anubhav Kumar Sharma<sup>2</sup>

*M.Tech Scholar, Dissertation Guide Department Of Information and Technology Lingaya's University,  
Nachauli, Faridabad, INDIA.*

---

**Abstract:** - Images play an important role in today's information age. A single image represents a thousand words. They are produced at a high rate, and can be found everywhere. A reason for all this is because cameras and other photo equipment has become cheap and easily accessible. Today we have image retrieval systems like Google's image search, where we can easily search for images using keywords. But getting the computer to understand the semantics inside of images isn't easy. The reason for this is simply because the computer isn't able to understand the context. In this paper we will talk about a problem that is very complex and common, namely face recognition. Identifying and comparing faces in images is a very complex task, this is probably why it has attracted so many researchers in the latest years. We will also describe some of the problems you will meet when designing a face recognition system, and we will take a closer look on how real-life systems solves these issues. Common methods like feature extraction, holistic matching and hybrid methods will be discussed. We will also take a closer look on a promising approach that uses 3D modeling.

**Keywords:** - Face Recognition

---

### I. INTRODUCTION

Research on automatic face recognition systems have been conducted now for almost 60 years. The first paper talking about face recognition can be traced back to the 1950's in psychology [3]. The first work concerning automatic face recognition was done in 1970 by Kelly [7]. His thesis describes a computer program which performs a complex image processing task? The task was to find the same person in a set of images taken by a TV camera.

Interest concerning face recognition is spread among different research environments, not only computer scientists. It all started by the psychologists, but over the last 40 years, extensive research has been conducted by psychophysicists, neuroscientists, and various types of engineers [9]. Psychophysicists and neuroscientists have been concerned with issues such as whether face perception is a dedicated process or not. This issue is still being debated in the psychology community.

During the 1980's, work on face recognition had no progress. But the interest grew rapidly again from the beginning of the 1990's. [9] gives some reasons why the research interest increased: real-time hardware became more available, and the importance of surveillance-related applications increased.

This paper contains four sections. The first section describes problems designers meet when creating a face recognition system. The second section describes various approaches taken in designing a face recognition system; it also describes some real-life systems. The third section tries to describe the future. The fourth section concludes this paper.

### II. PREVIOUS RELATED WORK

In identifying a face we usually give an image as input to a face recognition system. The process in recognizing the face is done in three key steps: (1) Face detection, locating the face in the image. (2) Feature/component extraction, a feature/component may be the eyes, the nose or the chins. (3) Recognition, comparing the input image with the ones in the database [9]. The result of this process will hopefully be a set of images that are similar to the input image. The result will be returned to the user of the system.

A face recognition system needs to compare its input images to a set of known images. These images are often stored in databases. There are several problems that may occur when comparing a database image with an input image. The main concern is of course that all images of the same face are heterogeneous. When image databases are created they contain good scenario images. These images are often taken in good conditions. But this isn't always the case. Examples of bad scenarios are when the face area is unfocused and too small, this is often the case of input images. A face recognition system needs to solve the problem concerning different facial expressions as well. The system must be able to know that two images of the same person with different facial expressions actually is the same person. Other issues face recognition systems need to solve is Makeup, posing positions, illumination conditions, and comparing images of the same person with and without glasses.

### III. Different Approaches

In the beginning of the 1970's, face recognition was treated as a 2D pattern recognition problem [7, 2, and 6]. The distances between important points were used to recognize known faces. E.g. measuring the distance between the eyes or other important points or measuring different angles of facial components. But the main focus has been on making face recognition systems fully automatic.

#### III.a Feature extraction methods

Feature extraction is the task where we locate facial features, e.g. the eyes, the nose, and the chins etc.



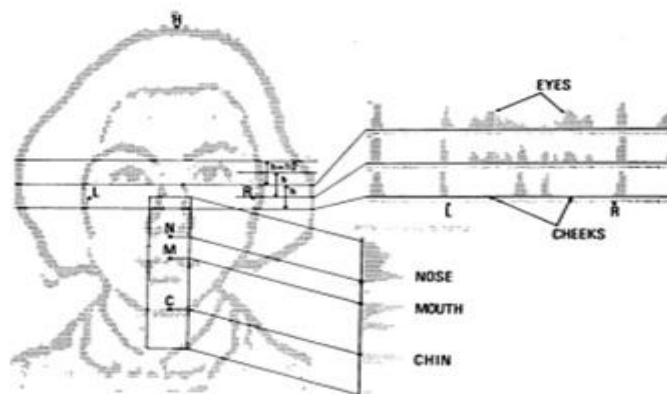
**Figure 1:** Kanade's approach to convert a regular image to a binary image.

This task may be performed after the face detection task. Or they may be performed at the same time, but that is if the face detection task locates features in order to find out where the face actually is inside the image. [9] Distinguishes between three different extraction methods: (1) generic methods based on edges, lines, and curves, (2) feature-template-based methods, (3) structural matching methods that take into consideration geometrical constraints on the features.

A big challenge for feature extraction methods is feature "restoration", this is when the system tries to recover features that are invisible due to large variations, e.g. Head pose when we are comparing a frontal image with a profile image.

#### III.b Feature extraction example

Feature extraction was the first approach developed to recognize faces in images. Kanade[5] was one of those who actually managed to extract features automatic in a quite simple manner and to recognize faces, previous systems had a manual extraction approach such as[2]. Kanade's approach was to convert a regular image into a binary image, as shown in Figure 1, which later was used as input to the extraction program. The extraction program made slits over the face to find the different features in the face.



**Figure 2:** How slits are used to locate features in the face

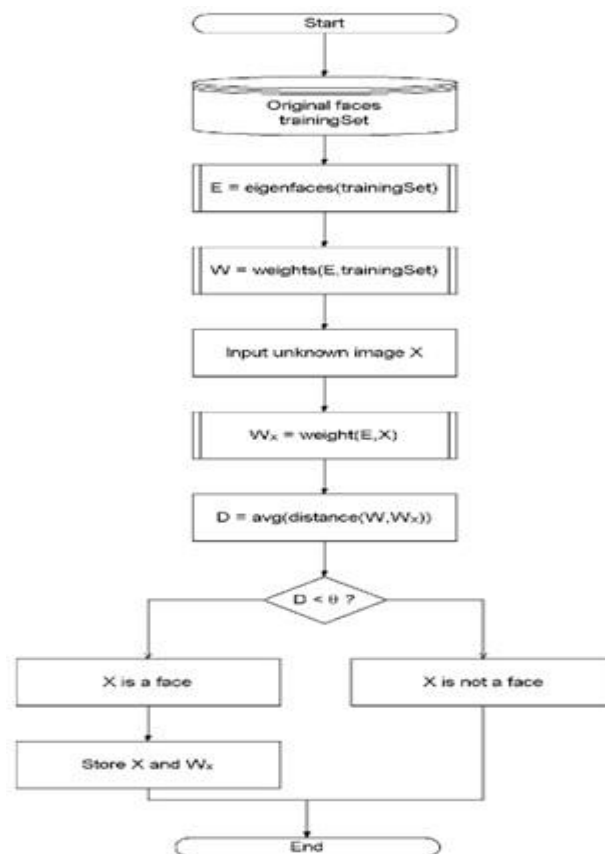
**Figure 2** illustrates how he is able to locate the eyes, cheeks, nose, mouth and the chins by using slits. The results Kanade had with his system been quite impressive. In a database with 670 images he managed to extract features from 608 images correctly when the faces had no glasses and no beard. When the faces had glasses he only managed to get 2 out of 77 correct. But with different turn or tilt on the face he managed to get 63 out of 79 correct.

### III.c Holistic methods

Holistic methods use the whole face region as the input to a recognition system [9]. One of the best examples of where this approach is used is in Eigen faces [8].

### III.d Holistic example

The first successful demonstration of machine recognition of faces was made by Turk and Pentland[8] in 1991 using eigenfaces. Their approach treats face recognition as a two-dimensional recognition problem. The flowchart in Figure 3 describes the different stages in an eigenface based recognition system.



**Figure 3:** High-level flow chart of the eigenface-based algorithm.

(1) The first stage is to insert a set of images into a database, these images are called the training set, this is because they will be used when we compare images and when we create the eigenfaces. (2) The second stage is to create the eigenfaces. Eigenfaces are created by extracting characteristic features from the faces. The input images are normalized to line up the eyes and mouths. They are then resized so that they have the same size. Eigenfaces can now be extracted from the image data by using a mathematical tool called Principal Component Analysis (PCA). (3) When the eigenfaces have been created, each image will be represented as a vector of weights. (4) The system is now ready to accept incoming queries. (5) The weight of the incoming unknown image is found and then compared to the weights of those already in the system. If the input image's weight is over a given threshold it is considered to be unknown. The identification of the input image is done by finding the image in the database whose weights are the closest to the weights of the input image. The image in the database with the closest weight will be returned as a hit to the user of the system[9].

The following steps summarize the recognition process:

1. Initialization: Acquire the training set of face images and calculate the eigenfaces, which define the face space.
2. When a new face image is encountered, calculate a set of weights based on input image and the M Eigen faces by projecting the input image on to each of Eigen faces.
3. Determine if the image is a face at all (whether known or unknown) by checking to see if the image is sufficiently close to “face space”.
4. If it is a face, classify the weight pattern as either a known person or as unknown.
5. (Optional) if the same unknown face is seen several times, calculate its characteristic weight pattern and incorporate into known faces (i.e. learn to recognize it).

#### IV. OpenCv: Free Source Library used for face Recognition

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real time computer vision, developed by Intel. It is free for use. The library is cross-platform. It focuses mainly on real-time image processing. The OpenCV Library is a collection of low-overhead, high-performance operations performed on images. The OpenCV implements a wide variety of tools for image interpretation. It is compatible with Intel® Image Processing Library (IPL) that implements low-level operations on digital images. In spite of primitives such as linearization, filtering, image statistics, pyramids, OpenCV is mostly a high-level library implementing algorithms for calibration techniques (Camera Calibration), feature detection (Feature) and tracking (Optical Flow), shape analysis (Geometry, Contour Processing), motion. Analysis (Motion Templates, Estimators), 3D reconstruction (View Morphing), object segmentation and recognition (Histogram, Embedded Hidden Markov Models, Eigen Objects). The essential feature of the library along with functionality and quality is performance. OpenCV has so many capabilities it can seem overwhelming at first. Some of them are as follows:

- a) General computer-vision and image-processing algorithms (mid-and low-level APIs).
- b) High-level computer-vision modules.
- c) AI and machine-learning methods.
- d) Image sampling and view transformations.
- e) Methods for creating and analyzing binary images.
- f) Methods for computing 3D information.
- g) Math routines for image processing, computer vision, and image interpretation.
- h) GUI methods.

##### IV.a Advantages of OpenCv over MatLab for Face Detection

###### (i) Ease of Use

OpenCv is more easy to use than MatLab. OpenCv is can run cross platform and it can run on almost all the systems irrespective of OS while Matlab is also cross platform but it has some issues while running on apple and linux based computers.

###### (ii) Speed

Matlab is a pretty high-level scripting language. Matlab is built on Java, and Java is built upon C. So when we run a Matlab program, your computer is busy trying to interpret all that Matlab code. Then it turns it into Java, and then finally executes the code. OpenCV is basically a library of functions written in C. We are closer to directly provide machine language code to the computer to get executed. So ultimately we get more image processing done for our computers processing cycles, and not more interpreting. As a result of this, programs written in OpenCV run much faster than similar programs written in Matlab.

###### (iii) Resources Needed

Due to the high level nature of Matlab, it is a resource hog. Matlab code requires over a gig of RAM to run through video. In comparison, a typical OpenCV programs only require ~70mb of RAM to run in real-time. Similarly, a full installation of Matlab plus all the toolboxes will take up a couple of gigs on computer. In comparison, OpenCV only requires a gig at most.

**(iv) Cost**

List price for the base (no toolboxes) MatLab (commercial, single user license) is around 1500 euro while OpenCv is free for use.

**(v) Development Environment**

Matlab comes with its own development environment. So when we use Matlab, we have to use its programming environment and IDE as well. For OpenCv, there is no particular IDE that we have to use. Instead we have a choice of any C programming IDE depending upon whether we are using Windows, Linux or OS X.

**(vi) Memory Management**

OpenCV is based on C. As such, every time you allocate a chunk of memory you will have to release it again. While due to the high-level nature of Matlab, it is “smart” enough to automatically allocate and release memory in the background.

**(vii) Portability**

Matlab is having the feature of portability but it causes some problems while running on Linux and apple OS while OpenCv has no such issues. OpenCv can run without any issues on any platform but it Windows, Linux or OS X. Based on some other criterias, we can conclude that OpenCv is much better than MatLab.

	Matlab	OpenCV
Ease of Use	9	3
Speed	2	9
Resources Needed	4	9
Cost	4	10
Development Environment	8	6
Memory Management	9	4
Portability	3	8
Development of usefull programming skills	3	8
Help and Sample Code	8	9
Debugging	9	5
<b>Total:</b>	<b>59</b>	<b>71</b>

## V. CONCLUSIONS

In this paper we have looked at three different approaches on how to design a face recognition system. The first approach was the feature extraction method. This method is widely used to create individual vectors for each person in a system, the vectors are matched when an input image is being recognized.

The second approach was the holistic extraction method. This method used the whole face as input to the system. The eigenface method by Turk and Pentland is considered to be the most successful approach.

An intelligent system should have an ability to adopt over time. Reasoning about images in face space provides a means to learn and subsequently recognize new faces in unsupervised manner. When an image is sufficiently close to face space (i.e. it is face like), but it is not classified as one of the familiar faces, it is initially labeled as “unknown”. The computer stores the pattern vector and the corresponding unknown image. If a collection of “unknown” pattern vectors cluster in the pattern vectors cluster in the pattern space, the presence of a new but unidentified face is postulated. A noisy image or partially occluded face should cause recognition performance to degrade gracefully, since the system essentially implements an auto associative memory for the known faces. This is evidenced by the projection of the occluded face image.

The Eigen face approach to face recognition was motivated by information theory, leading to the idea of basing face recognition on a small set of image features that best approximate the set of known face images, without requiring that they correspond to our intuitive notions of facial parts and features.

## VI. FUTURE WORK

Since the beginning of year 2000 there have been produced numerous papers concerning face recognition. The website [www.face-rec.org](http://www.face-rec.org) holds a list over recently released work and also a list with links to different research groups..

The future of face recognition systems looks bright, there is over 50 groups spread around the world working on the issue today.

## VII. ACKNOWLEDGEMENT

The authors would like to thank Honorable HOD Of Information Technology, Dr. Brijesh Kumar and Mr. Anubhav Kumar (Asst. professor), Lingaya's University, Faridabad, Haryana, INDIA, for providing us the entire necessary infrastructure for higher technical education and valuable support for research & development in the college. Without the facilities at the college, this work would have not been possible.

## REFERENCES

- [1] V. Blanz and T. Vetter. Face recognition based on a 3d morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9):1063-1074, 2003.
- [2] W. Bledsoe. Man-machine facial recognition. 1966.
- [3] I. Bruner and R. Tagiuri. The perception of people, pages 634-654. Addison-Wesley, second edition, 1954.
- [4] J. Huang, B. Heisele, and V. Blanz. Component-based face recognition with 3d morphable models. In J. Kittler and M. S. Nixon, editors, *International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA-3)*, volume 2688 of *Lecture Notes in Computer Science*, pages 27-34, Surrey, UK, 2003. Springer.
- [5] T. Kanade. *Picture Processing System by Computer Complex and Recognition of Human Faces*. PhD thesis, doctoral dissertation, Kyoto University, November 1973.
- [6] T. Kanade. *Computer Recognition of Human Faces*, 47, 1977.
- [7] M. D. Kelly. *Visual identification of people by computer*. PhD thesis, Stanford University, Stanford, CA, USA, 1971.
- [8] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91.*, IEEE Computer Society Conference on, pages 586-591, 1991.
- [9] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld.