**Chapter 1**

**INTRODUCTION**

A face recognition system is a computer application capable of [identifying](https://en.wikipedia.org/wiki/Identification_of_human_individuals) or [verifying](https://en.wikipedia.org/wiki/Authentication) a person from a [digital image](https://en.wikipedia.org/wiki/Digital_image) or a [video frame](https://en.wikipedia.org/wiki/Film_frame) from a [video](https://en.wikipedia.org/wiki/Video) source. One of the ways to do this is by comparing selected [facial features](https://en.wikipedia.org/wiki/Face) from the image and a face [database](https://en.wikipedia.org/wiki/Database_management_system). It is typically used in [security systems](https://en.wikipedia.org/wiki/Burglar_alarm) and can be compared to other [biometrics](https://en.wikipedia.org/wiki/Biometrics) such as [fingerprint](https://en.wikipedia.org/wiki/Fingerprint) or eye [iris recognition](https://en.wikipedia.org/wiki/Iris_recognition) systems. Recently, it has also become popular as a commercial identification and marketing tool.

Human face detection and recognition has attracted much attention ,it is an active area of research spanning several disciplines such as computer vision and automatic access control system. Especially, face detection is an important part of face recognition as the first step of automatic face recognition. A complete face recognition system presents two stages.

First detect the human faces in a cluttered background and extract pertinent features, second involve classification of facial images obtained in the previous stage. A robust face recognition system is a system based on good feature extraction method and good classifier. Neural network have been successfully applied to many pattern classification problems. And among techniques used in the literature for skin detection based on different color spaces include HSV, normalized RGB, YCrCb, YIQ and CIELAB.

Among several recognition subjects, face recognition has drawn extensive interest and a focus from several researchers for the last twenty years owing to its potential applications, like within the areas of surveillance, secure trading terminals, closed circuit tv (CCTV) control, user authentication, HCI Human computer Interface, intelligent automation & so on. Varieties of face recognition strategies are projected and a few related face recognition systems are developed.

In this paper we tend to propose a computational model of face recognition, that is quick, moderately easy, and correct in constrained environments like any workplace or a home. The proposed approaches have benefits over the other face recognition schemes in its speed and ease, learning capacity and relative inability to little or gradual changes within the face image.

* 1. **Techniques for Face Recognition**

Some face recognition [algorithms](https://en.wikipedia.org/wiki/Algorithms) identify facial features by extracting landmarks, or features, from an image of the subject's face. For example, an algorithm may analyze the relative position, size, and/or shape of the eyes, nose, cheekbones, and jaw. These features are then used to search for other images with matching features.

Other algorithms [normalize](https://en.wikipedia.org/wiki/Normalization_(image_processing)) a gallery of face images and then compress the face data, only saving the data in the image that is useful for face recognition. A probe image is then compared with the face data. One of the earliest successful systems is based on template matching techniques applied to a set of salient facial features, providing a sort of compressed face representation.

* + 1. **3-dimensional recognition**

A newly emerging trend, claimed to achieve improved accuracy, is [three-dimensional face recognition](https://en.wikipedia.org/wiki/Three-dimensional_face_recognition). This technique uses 3D sensors to capture information about the shape of a face. This information is then used to identify distinctive features on the surface of a face, such as the contour of the eye sockets, nose, and chin.



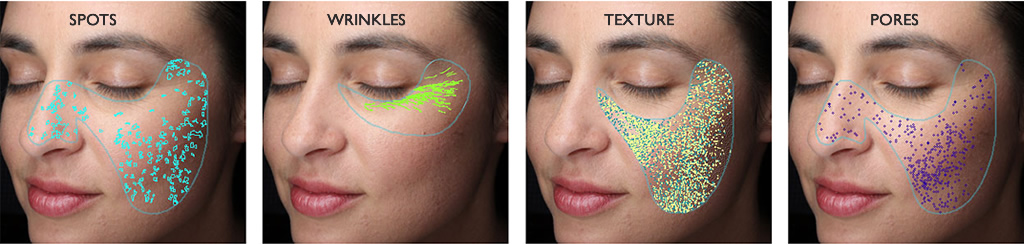
**Fig.1.1: 3D Recognition**

One advantage of 3D face recognition is that it is not affected by changes in lighting like other techniques. It can also identify a face from a range of viewing angles, including a profile view. Three-dimensional data points from a face vastly improve the precision of face recognition. 3D research is enhanced by the development of sophisticated sensors that do a better job of capturing 3D face imagery. The sensors work by projecting structured light onto the face. Up to a dozen or more of these image sensors can be placed on the same CMOS chip—each sensor captures a different part of the spectrum....

Even a perfect 3D matching technique could be sensitive to expressions. For that goal a group at the [Technion](https://en.wikipedia.org/wiki/Technion" \o "Technion) applied tools from [metric geometry](https://en.wikipedia.org/wiki/Metric_geometry) to treat expressions as [isometries](https://en.wikipedia.org/wiki/Isometries" \o "Isometries) A company called Vision Access created a firm solution for 3D face recognition. The company was later acquired by the biometric access company [Bioscrypt Inc.](https://en.wikipedia.org/wiki/Bioscrypt_Inc." \o "Bioscrypt Inc.) which developed a version known as 3D FastPass.

A new method is to introduce a way to capture a 3D picture by using three tracking cameras that point at different angles; one camera will be pointing at the front of the subject, second one to the side, and third one at an angle. All these cameras will work together so it can track a subject’s face in real time and be able to face detect and recognize.

* + 1. **Skin texture analysis**



**Fig.1.2: Skin Texture Analysis**

Another emerging trend uses the visual details of the skin, as captured in standard digital or scanned images. This technique, called skin texture analysis, turns the unique lines, patterns, and spots apparent in a person’s skin into a mathematical space.

Tests have shown that with the addition of skin texture analysis, performance in recognizing faces can increase 20 to 25 percent.

* + 1. **Thermal cameras**



**Fig.1.3: Thermal Imaging**

A different form of taking input data for face recognition is by using thermal cameras, by this procedure the cameras will only detect the shape of the head and it will ignore the subject accessories such as glasses, hats, or make up. A problem with using thermal pictures for face recognition is that the databases for face recognition are limited. Diego Socolinsky, and Andrea Selinger (2004) research the use of thermal face recognition in real life, and operation sceneries, and at the same time build a new database of thermal face images.

The research uses low-sensitive, low-resolution ferro-electric electrics sensors that are capable of acquire long wave thermal infrared (LWIR). The results show that a fusion of LWIR and regular visual cameras has the greater results in outdoor probes. Indoor results show that visual has 97.05% accuracy, while LWIR has 93.93%, and the Fusion has 98.40%, however on the outdoor proves visual has 67.06%, LWIR 83.03%, and fusion has 89.02%.

* 1. **Advantages and disadvantages**
     1. **Compared to other technologies**

Among the different biometric techniques, face recognition may not be most reliable and efficient. However, one key advantage is that it does not require the cooperation of the test subject to work. Properly designed systems installed in airports, multiplexes, and other public places can identify individuals among the crowd, without passers-by even being aware of the system. Other biometrics like fingerprints, iris scans, and speech recognition cannot perform this kind of mass identification.

* + 1. **Weaknesses**

Face recognition is far from perfect and struggles to perform under certain conditions. Face recognition has been getting pretty good at full frontal faces and 20 degrees off, but as soon as you go towards profile, there've been problems. Current face recognition still often misidentifies people which can sometimes lead to controversy. Face recognition software generally doesn't do as well in identifying minorities when most of the subjects used in testing the technology were from the majority group.

Other conditions where face recognition does not work well include poor lighting, sunglasses, hats, scarves, beards, long hair, makeup or other objects partially covering the subject’s face, and low resolution images.

Another serious disadvantage is that many systems are less effective if facial expressions vary. Even a big smile can render the system less effective. There is also inconstancy in the datasets used by researchers. Researchers may use anywhere from several subjects to scores of subjects, and a few hundred images to thousands of images. It is important for researchers to make available the datasets they used to each other, or have at least a standard dataset.

* + 1. **Effectiveness**

It can be explained by the notion that when the public is regularly told that they are under constant video surveillance with advanced face recognition technology, this fear alone can reduce the crime rate, whether the face recognition system technically works or does not. This has been the basis for several other face recognition based security systems, where the technology itself does not work particularly well but the user's perception of the technology does.

An experiment in 2002 by the local [police](https://en.wikipedia.org/wiki/Police) department in [Tampa](https://en.wikipedia.org/wiki/Tampa,_Florida), [Florida](https://en.wikipedia.org/wiki/Florida), had similarly disappointing results. A system at Boston's [Logan Airport](https://en.wikipedia.org/wiki/Logan_International_Airport) was shut down in 2003 after failing to make any matches during a two-year test period.

As of 2016, facial recognition is still not effective for most applications even though the accuracy has been substantially improved. Although systems are often advertised as having accuracy near 100%, this is misleading as the studies often uses much smaller sample sizes than would be necessary for large scale applications.

Because facial recognition is not completely accurate, it creates a list of potential matches. A human operator must then look through these potential matches and studies show the operators pick the correct match out of the list only about half the time.

* + 1. **Privacy issues**

Civil rights right organizations and privacy campaigners express concern that [privacy](https://en.wikipedia.org/wiki/Privacy) is being compromised by the use of surveillance technologies. Some fear that it could lead to a “total [surveillance society](https://en.wikipedia.org/wiki/Surveillance_society),” with the government and other authorities having the ability to know the whereabouts and activities of all citizens around the clock.

This knowledge has been, is being, and could continue to be deployed to prevent the lawful exercise of rights of citizens to criticize those in office, specific government policies or corporate practices. Many centralized power structures with such surveillance capabilities have abused their privileged access to maintain control of the political and economic apparatus, and to curtail populist reforms.

Face recognition can be used not just to identify an individual, but also to unearth other personal data associated with an individual – such as other photos featuring the individual, blog posts, social networking profiles, Internet behavior, travel patterns, etc. all through facial features alone. Moreover, individuals have limited ability to avoid or thwart face recognition tracking unless they hide their faces. This fundamentally changes the dynamic of day-to-day privacy by enabling any marketer, government agency, or random stranger to secretly collect the identities and associated personal information of any individual captured by the face recognition system.

* 1. **Proposed Solution Using Surf Features**

Speed-up robust features (SURF) are a scale and in-plane rotation invariant feature. It contains interest point detector and descriptor. The detector locates the interest points in the image, and the descriptor describes the features of the interest points and constructs the feature vectors of the interest points.

SURF is a scale and in-plane rotation invariant detector and descriptor. In SURF, detectors are first employed to find the interest points in an image, and then the descriptors are used to extract the feature vectors at each interest point. SURF uses Hessian-matrix approximation operating on the integral image to locate the interest points, which reduces the computation time drastically.

As for the descriptor, the first-order Haar wavelet responses in x and y directions are used in SURF to describe the intensity distribution within the neighborhood of an interest point. In addition, only 64 dimensions are usually used in SURF to reduce the time cost for both feature computation and matching. Because each of SURF feature has only 64 dimensions in general and an indexing scheme is built by using the sign of the Laplacian, SURF is much faster at the matching step.

* 1. **Speed-Up Robust Features**
     1. **Interest point detection**

SURF uses the determinant of the approximate Hessian matrix as the base of the detector. To locate the interest point, we detect blob-like structures at locations where the determinant is at maximum. Integral images are used in Hessian matrix approximation, which reduce computation time drastically. Given a point x = (x, y) in an image I, the Hessian matrix H(x;σ ) in x at scale σ is defined as follows:

Capture.JPG

where (x,σ ) Lxx , (x,σ ) Lxy and (x,σ ) Lyy are the convolutions of the Gaussian second order partial derivatives with the image I in point x respectively.

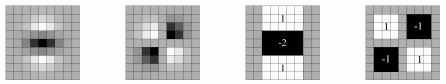
To reduce the computation time, a set of 9×9 box filters (Fig.1.4) is used as the approximations of a Gaussian with σ = 1.2 and represents the lowest scale (i.e. highest spatial resolution) for computing the blob response maps. We will denote them by (x,σ ) Dxx , (x,σ ) Dyy , and (x,σ ) Dxy . The weights applied to the rectangular regions are kept simple for computational efficiency. This yields:

Capture.JPG

where ω is a weight for the energy conservation between the Gaussian kernels and the approximated Gaussian kernels, and

Capture.JPG

|X|F is the Frobenius norm. For scale invariant, the SURF constructs a pyramid scale space. SURF directly changes the scale of box filters to implement the scale space due to the use of the box filter and integral image.



**Fig. 1.4: The box filters of approximations of Gaussian second order partial derivative. The figure shows (x,σ ) Lyy , (x,σ ) Lxy , (x,σ ) Dyy and (x,σ ) Dxy from left to right.**

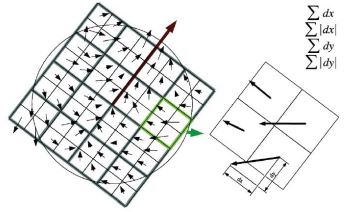
* + 1. **Interest point description**

In SURF the used sum of the Haar wavelet responses to describe the feature of an interest point. Fig.1.5 shows the Haar wavelet filters used to compute the responses at x and y directions. For the extraction of the descriptor, the first step consists of constructing a square region centered at the interest point and oriented along the orientation decided by the orientation selection method.



**Fig. 1.5: The Haar wavelet filters used to descript the interest points.**

The region is split up equally into smaller 4×4 square sub-regions (as shown in Fig. 1.6). This preserves important spatial information. For each sub-region, we compute Haar wavelet responses at 5×5 equally spaced sample points. For simplicity, we call dx the Haar wavelet response in horizontal direction and dy the Haar wavelet response in vertical direction. To increase the robustness towards geometric deformations and localization errors, the responses dx and dy are first weighted with a Gaussian centered at the interest point.



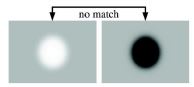
**Fig. 1.6: The demonstration of descriptor building.**

Then, the wavelet responses dx and dy are summed up over each sub-region and form a first set of entries in the feature vector. In order to bring in information about the polarity of the intensity changes, we also extract the sum of the absolute values of the responses, | dx | and | dy |.

Hence, each sub-region has a four-dimensional descriptor vector v for its underlying intensity structure v = (∑ ∑∑ dx, dy | dx |,∑| dy |). Concatenating this for all 4×4 sub-regions, this results a descriptor vector of length 64. The wavelet responses are invariant to a bias in illumination (offset). Invariance to contrast (a scale factor) is achieved by turning the descriptor into a unit vector.

* + 1. **Fast index for matching**

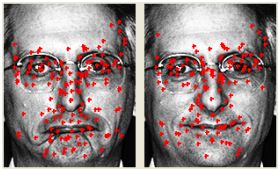
To speed up matching step, the sign of the Laplacian (i.e, the trace of the Hessian matrix) for the interest point is used. Only the point-pair with the same sign will be matched with the features. Fig.1.7 shows the example blobs of the sign.

****

**Fig. 1.7: The fast index for matching.**

* 1. **SURF For Face Recognition**
     1. **SURF feature extraction**

SURF features should be extracted from images through SURF detectors and descriptors. Interest points are first extracted from each face image after pre-processing, such as normalization and histogram equalization (shown in Fig.1.8). This turns out to obtain about 30-100 interest points per image.



**Fig. 1.8: Interest points in face image.**

The SURF feature vectors of the set of interest points are then computed to describe the image and these feature vectors are normalized to 1. These features are person-specific, since the number and the positions of points selected by SURF detector as well as the features around these points computed by SURF descriptor are different in each person’s image.

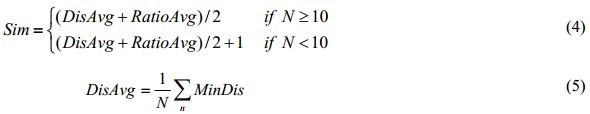
* + 1. **SURF Feature Matching**

Point matching is commonly employed in face recognition with SIFT. In order to effectively match two face images, M. Bicego and Jun Luo suggested different sub-region matching strategies. But both methods have to compute the sub-region similarities and global similarities, with increase of computation cost in matching. They all used the  
point matching method as a part of their evaluation of matching.

In this paper, based on point matching method, we introduce geometric constraints into point-matching based on SURF features to increase the matching speed and robustness. Because in face recognition, face images are usually upright and normalized, the matching points in two images must have the similar locations on the two faces.

Thus for an interest point (x,y) in the probe image the search area for its mate is limited within a rectangular window centered at (x,y) of the gallery image. The point-pair with the minimum distance between descriptors will be considered as a candidate matching pair. To verify the validity of the candidate point-pair, the next minimal distance of point-pair, which contains the same point of the probe image, is then searched over the whole area of the gallery image. If the ratio of these two distances is smaller than a pre-defined threshold, the point-pair with the minimum distance is confirmed as a matched pair.

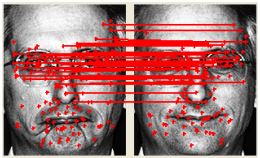
Since location information is introduced in search of the minimum-distance point-pair, and the ratio of the minimum distance and next to minimum distance measures the matching reliability of two interest points in some degree, the above method can avoid mismatching effectively. Finally, based on the result of the point-matching, we define a similarity measure in Eq.(4), which contains the number of matched points, the average value of the Euclidean distance, and the average distance ratio of all matched points, for face recognition.



Capture.JPG

where N is the number of matched points of two face images, MinDis is the Euclidean distance between two matched points, and DisRatio is the distance ratio of the matched points. Here, n = 1,…,N.

When the number of the matched points of two images is smaller than the predefined threshold (10 in our experiment), it is thought that the matching result is not reliable even if the similarity measure Sim is small. Therefore, in Eq. (4), 1 is  
added to minish the similarity in this case. Fig.1.9 shows an example of the point matching result. The red lines indicate the corresponding matched interest points.

****

**Fig. 1.9: Example of point matching result.**

**Chapter 2**

**FACIAL FEATURES DETECTION**

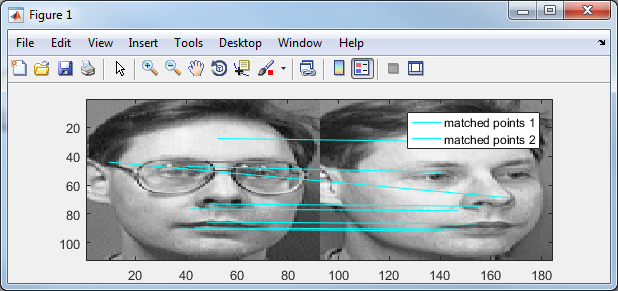
* 1. **Method adopted for Facial Features Detection**

There are some traditional algorithms for face recognition, which have been widely used in object detection and recognition. Among them the detector and descriptor, named Speed-Up Robust Features (SURF) suggested by Herbert Bay, attracts people’s attentions. SURF is a scale and in-plane rotation invariant detector and descriptor. In SURF, detectors are first employed to find the interest points in an image, and then the descriptors are used to extract the feature vectors at each interest point.

SURF uses Hessian-matrix approximation operating on the integral image to locate the interest points, which reduces the computation time drastically. As for the descriptor, the first-order Haar wavelet responses in x and y directions are used in SURF to describe the intensity distribution within the neighborhood of an interest point. Only 64 dimensions are usually used in SURF to reduce the time cost for both feature computation and matching. Because each of SURF feature has only 64 dimensions in general and an indexing scheme is built by using the sign of the Laplacian, SURF is much faster.

* 1. **SURF Algorithm**
* Start Step 1: Input two images and extract the faces by using any face detection algorithm (e.g. Voila Zones).
* Step 2: Detect SURF feature points from the corresponding faces.
* Step 3: Extract SURF features.
* Step 4: Compare the SURF features of two images, find the matching points and calculate the matching percentage.
* Step 5: If the percentage is greater than a threshold the take the decision that the faces are matched. Otherwise reject the matching. Stop.
  1. **SURF Feature Extraction**

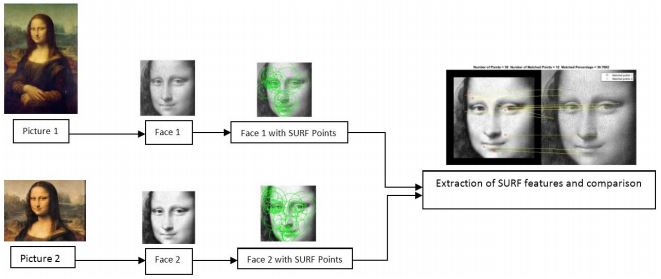
SURF features can be extracted from images through SURF detectors and descriptors. Interest points are first extracted from each face image after pre-processing, such as normalization and histogram equalization. This turns out to obtain about 30-100 interest points per image.



**Fig 2.1: Surf matching Points.**

The SURF feature vectors of the set of interest points are then computed to describe the image. These features are person-specific, since the number and the positions of points selected by SURF detector as well as the features around these points computed by SURF descriptor are different in each person’s image.

* 1. **Proposed Schemes**

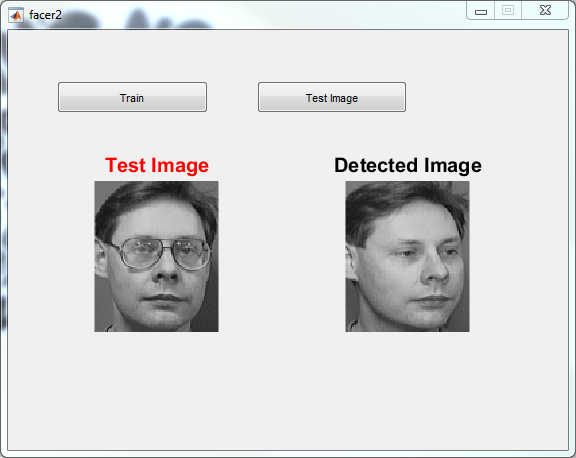


**Fig 2.2: Steps for face matching using SURF feature points**

In our proposed scheme we have compared two faces for matching. Firstly we have used Voila-Zones face detecting algorithm to detect faces from a picture. Then we have calculated the SURF feature points on the two faces, which are also the ROI points of the same faces. Then we have extracted different features from the both face regions by using the feature points.

* 1. **SURF Feature Matching**

After SURF features have been extracted from images through SURF detectors and descriptors the features are matched against each other. We have calculated the matching percentage by using the following formula. Number of Matched Points Matching Percentage = x 100 Number of Interest points to take the decision if the two images are matched one threshold must be selected. From our experimental result the threshold is taken as 30%.

****

**Fig 2.3: Surf Detected Image Result**

* 1. **Algorithm and features**
* Detection
* Scale-space representation and location of points of interest
* Descriptor
* Orientation assignment
* Descriptor based on the sum of Haar wavelet responses
  1. **Testing**

When testing the algorithm there are several key factors to take into account. Gossow et al has published a report comparing several open source Implementations of the SURF algorithm, they introduce 3 different performance criteria. Repeatability, precision and recall. Repeatability is the how good the detector is at detecting the same features under different transformations of the images.

As some transformations, for example a rotation may move interest points out of the image it must take this into account. Thus repeatability is defined as = #correspondences min (n1, n2) , where n1,2 is the number of interest point in the images to be compared and correspondence whether the points are close to the correct result.

Precision measures how many correct matches are found, and recall is the relation between correct matches and correspondences. Gossow et al uses a fixed pool of images and they have calculated the transformation between them, so they are able to calculate correspondence and verify if a match is correct.

As I have implemented the U-SURF variant I have mainly focused the testing invariance to scale. This also removed the need to calculate the position of interest points after the transformations, as the position is stored as relative coordinates, and correspondence is linear in with respect to the distance between the match and the original point, when there is no skewing involved.

* 1. **Outline and Algorithm Overview**
* **Multi-scale analysis:** Similarly too many other approaches, such as the SIFT method; the detection of features in SURF relies on a scale-space representation, combined with first and second order differential operators. The originality of the SURF algorithm (Speeded Up Robust Features) is that these operations are speeded up by the use of box filters techniques. For this reason, we will use the term box-space to distinguish it from the usual Gaussian scale-space. While the Gaussian scale space is obtained by convolution of the initial images with Gaussian kernels, the discrete box-space is also obtained by convolving the original image with box filters at various scales.
* **Feature detection:** During the detection step, the local maxima in the box-space of the “determinant of Hessian” operator are used to select interest point candidates. These candidates are then validated if the response is above a given threshold. Both the scale and location of these candidates are then refined using quadratic fitting. Typically, a few hundred interest points are detected in a megapixel image.
* **Feature description:** The purpose of the next step is to build a descriptor of the neighborhood of each point of interest that is invariant to view-point changes. Thanks to multi-scale analysis, the selection of these points in the box-space provides scale and translation invariance. To achieve rotation invariance, a dominant orientation is defined by considering the local gradient orientation distribution, estimated from Haar wavelets. Using a spatial localization grid, a 64-dimensional descriptor is then built, based on first order statistics of Haar wavelets coefficients.
* **Feature matching:** Finally, when considering the image matching task (e.g. for image registration, object detection, or image indexation), the local descriptors from several images are matched. Exhaustive comparison is performed by computing the Euclidean distance between all potential matching pairs.
  1. **Quick Mask Using Edge Detection Method**

The edge detector is an [edge detection](https://en.wikipedia.org/wiki/Edge_detection) operator that uses a multi-stage [algorithm](https://en.wikipedia.org/wiki/Algorithm) to detect a wide range of edges in images. It was developed by [John F. Canny](https://en.wikipedia.org/wiki/John_F._Canny) in 1986. [Edge detection](https://www.mathworks.com/products/image/features.html) is an image processing technique for finding the boundaries of objects within images. It works by detecting discontinuities in brightness. Edge detection is used for [image segmentation](https://www.mathworks.com/discovery/image-segmentation.html) and data extraction in areas such as image processing, computer vision, and machine vision.

Common [edge detection algorithms](https://www.mathworks.com/help/images/ref/edge.html) include Sobel, Canny, Prewitt, Roberts, and [fuzzy logic](https://www.mathworks.com/products/fuzzy-logic.html) methods. A Figure is shown below 2.4. Edge detection includes a variety of mathematical methods that aim at identifying points in a [digital image](https://en.wikipedia.org/wiki/Digital_image) at which the [image brightness](https://en.wikipedia.org/wiki/Luminous_intensity) changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges.

The same problem of finding discontinuities in one-dimensional signals is known as [step detection](https://en.wikipedia.org/wiki/Step_detection) and the problem of finding signal discontinuities over time is known as [change detection](https://en.wikipedia.org/wiki/Change_detection). Edge detection is a fundamental tool in [image processing](https://en.wikipedia.org/wiki/Image_processing), [machine vision](https://en.wikipedia.org/wiki/Machine_vision) and [computer vision](https://en.wikipedia.org/wiki/Computer_vision), particularly in the areas of [feature detection](https://en.wikipedia.org/wiki/Feature_detection_(computer_vision)) and [feature extraction](https://en.wikipedia.org/wiki/Feature_extraction).



**Fig 2.4: Image segmentation using the**[**Edge method**](https://www.mathworks.com/help/images/object-analysis.html#f11-12512)**.**

Edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. It has been widely applied in various computer vision systems. Canny has found that the requirements for the application of edge detection on diverse vision systems are relatively similar. Thus, an edge detection solution to address these requirements can be implemented in a wide range of situations.

The general criteria for edge detection includes:

1. Detection of edge with low error rate, which means that the detection should accurately catch as many edges shown in the image as possible.
2. The edge point detected from the operator should accurately localize on the center of the edge.
3. A given edge in the image should only be marked once, and where possible, image noise should not create false edges.
   1. **Types of edges**

Generally edges are of three types:

* Horizontal edges
* Vertical Edges
* Diagonal Edges
  1. **Why detect edges**

Most of the shape information of an image is enclosed in edges. So first we detect these edges in an image and by using these filters and then by enhancing those areas of image which contains edges, sharpness of the image will increase and image will become clearer.

Here are some of the masks for edge detection:

* Prewitt Operator
* Sobel Operator
* Robinson Compass Masks
* Krisch Compass Masks
* Laplacian Operator

**Prewitt Operator**

Prewitt operator is used for detecting edges horizontally and vertically.

**Sobel Operator**

The sobel operator is very similar to Prewitt operator. It is also a derivate mask and is used for edge detection. It also calculates edges in both horizontal and vertical direction.

**Robinson Compass Masks**

This operator is also known as direction mask. In this operator we take one mask and rotate it in all the 8 compass major directions to calculate edges of each direction.

**Kirsch Compass Masks**

Kirsch Compass Mask is also a derivative mask which is used for finding edges. Kirsch mask is also used for calculating edges in all the directions.

**Laplacian Operator**

Laplacian Operator is also a derivative operator which is used to find edges in an image. Laplacian is a second order derivative mask. It can be further divided into positive laplacian and negative laplacian.

All these masks find edges. Some find horizontally and vertically, some find in one direction only and some find in all the directions. The next concept that comes after this is sharpening which can be done once the edges are extracted from the image.

* 1. **Edge Detection Algorithm**

The Process of Edge detection algorithm can be broken down to 5 different steps:

1. Apply Gaussian filter to smooth the image in order to remove the noise
2. Find the intensity gradients of the image
3. Apply non-maximum suppression to get rid of spurious response to edge detection
4. Apply double threshold to determine potential edges
5. Track edge by [hysteresis](https://en.wikipedia.org/wiki/Hysteresis)
   1. **Steps in Edge Detection**

Algorithms for edge detection contain three steps:

**Filtering:** Since gradient computation based on intensity values of only two  
points are susceptible to noise and other vagaries in discrete computations, filtering is commonly used to improve the performance of an edge detector with respect to noise. However, there is a trade-off between edge strength and noise reduction. More filtering to reduce noise results in a loss of edge strength.

**Enhancement:** In order to facilitate the detection of edges, it is essential to determine changes in intensity in the neighborhood of a point. Enhancement emphasizes pixels where there is a significant change in local intensity values and is usually performed by computing the gradient magnitude.  
**Detection:** We only want points with strong edge content. However, many points in an image have a nonzero value for the gradient, and not all of these points are edges for a particular application. Therefore, some method should be used to determine which points are edge points. Frequently, thresholding provides the criterion used for detection.

**Localization:** The location of the edge can be estimated with subpixel resolution if required for the application. The edge orientation can also be estimated. It is important to note that detection merely indicates that an edge is present near a pixel in an image, but does not necessarily provide an accurate estimate of edge location or orientation. The errors in edge detection are errors of misclassification: false edges and missing edges. The errors in edge estimation are modeled by probability distributions for the location and orientation estimates. We distinguish between edge detection and estimation because these steps are performed by different calculations and have different error models. Many edge detectors have been developed in the last two decades.

**Chapter 3**

**Back Propagation Neural Network (BPNN)**

* 1. **Introduction**

Humans are adept at recognizing faces and can do so with ease even under a range of difficult physical conditions. However, developing artificial systems to mimic the human ability has proven to be a very difficult and computationally complex task. There have been numerous studies exploiting various concepts and problems in the face recognition process and many steps in designing human face recognition system.

Some of the systems have employed artificial neural networks while the others have a variety of approaches such as template matching of iso-density lines of subject faces. Comparison of sizes/relative distances of facial features (nose, eyes, mouth) of subjects of facial images.

Face recognition, although a trivial task for the human brain has proved to be extremely  
difficult to imitate artificially. It is commonly used in applications such as human-machine interfaces and automatic access control systems. Face recognition involves comparing an image with a database of stored faces in order to identify the individual in that input  
image. The related task of face detection has direct relevance to face recognition because images must be analyzed and faces identified, before they can be recognized.

Detecting faces in an image can also help to focus the computational resources of the face  
recognition system, optimizing the systems speed and performance. Face detection involves separating image windows into two classes; one containing faces (targets), and one containing the background (clutter). It is difficult because although commonalities exist between faces, they can vary considerably in terms of age, skin color and facial expression.

The problem is further complicated by differing lighting conditions, image qualities and geometries, as well as the possibility of partial occlusion and disguise. An ideal face detector would therefore be able to detect the presence of any face under any set of lighting conditions, upon any background. For basic pattern recognition systems, some of these effects can be avoided by assuming and ensuring a uniform background and fixed uniform lighting  
conditions.

This assumption is acceptable for some applications, where lighting conditions can be controlled, and the image background will be uniform. For many applications however, this is  
unsuitable, and systems must be designed to accurately classify images subject to a variety of  
unpredictable conditions.

* 1. **Methodology**

Face recognition process can be decomposed into two major tasks.

* + 1. Finding a face in an image.
    2. Recognizing or identification of that face.

Task 1: Finding a face in an image is also known as face registration or face localization.

* The control one has lighting and background conditions.
* Whether the image is color or monochrome and whether the images are still or video.

If the background lighting can be controlled then it might be possible to extract the head very simply as one of the main source of brightness in the image. Hence depending on the control, one has to cover these factors. Face recognition can be a hard or easy problem.

Task 2: Face recognition and verification

There are two versions of the face recognition problems. The recognition and the verification problem. In the face of the verification one has to test the likelihood. It involves testing the quality of match of an image against a single model. In the case of  
face recognition the problem is to find best match for an unknown image against a database of face models or to determine whether it does not match any of them well.

The practical importance of this distinction is the speed required. Generally if there are N subjects in the database then the recognition process will be N times slower than the verification process. This may place practical limits on the algorithm used. Although the ability to infer intelligence or character from facial appearance is suspect, the human ability to recognize face is remarkable.

We can recognize thousands of faces learned throughout our lifetime and identify familiar faces at a glance even after years of separation. This skill is quite robust, despite  
large changes in the visual stimulus due to viewing conditions, expression, aging, and distractions such as glasses or changes in hairstyle or facial hair. As a consequence the visual processing of human faces has fascinated philosophers and scientists for  
centuries.

Computational models of face recognition, in particular are interesting because they can contribute not only to theoretical insights but also to practical applications, Computers that recognize faces could be applied to a wide variety of problems,  
including criminal identification, security systems, image and film processing, and human-computer interaction. Detecting faces in photographs, for instance, is an important problem in automating color film development, since the effect of many enhancement and noise reduction techniques depends on the picture content (e.g., faces should not be tinted green, while perhaps grass should). Unfortunately, developing a computational model of face recognition developing is a quite difficult, because faces are complex, multidimensional, and meaningful visual stimuli.

We therefore focused our research toward developing a sort of early, pre-attentive  
pattern recognition capability that does not depend on having three-dimensional information or detailed geometry. Our goal, which we believe we have reached, was to develop a computational model of face recognition that is fast, reasonably simple, and accurate in constrained environments such as an office or a household. In addition the approach is biologically implement able and is in concrete with preliminary findings in the physiology and psychology of face recognition.

The scheme is based on an information theory approach that decomposes face images into a small set of characteristics feature images called “eigen faces”, which may be thought of as the principle components of the initial training set of face images.  
Recognition is performed by projecting a new image into the subspace spanned by the eigen faces (“face space”) and then classifying the face by comparing its position in face space with the positions of known individuals.

Automatically learning and later recognizing new faces is practical within this framework. Recognition under widely varying conditions is achieved by training on a limited number of characteristics views (e.g., a “straight on” view, a 45º view, and a profile view). The approach has advantages over other face recognition schemes in its speed and simplicity, learning capacity, and insensitivity to small or gradual changes in the face image.

* 1. **Neural Networks**

How do you recognize a face in a crowd? When faced with problems like this the human brain uses a web of inter connected processing elements called neurons to process information. Each neuron is autonomous and independent. It does its work asynchronously i.e., without any synchronization to other even taking place. The two problems Pascal i.e., recognize a face and forecasting an interest rate have two important characteristics that distinguish them from other problem.

The problem are complex i.e., you can’t devise a simple step by step algorithm  
or precise formula to give you an answer. Resolve the problem is equally complex and may be noisy or incomplete. A neural network is a computational structure inspired by the study of biological neural processing. There are many different types of neural networks from relatively simple to very complex just as there are many theories on how biological neural net work and branch out to other paradigm later.

A layered feed forward neural network has layers of sub-groups of processing elements. A layer of processing elements makes independent computation or data that  
it receives and passes the result to another layer. The next layer may in turn make it independent computations and pass on the results to yet another layer. Finally a sub-group of one is more processing elements determine the output from the network. Each processing element makes its computation based upon a weighted sum of its inputs. The first layer is the input layer and the last is the output layer.

* 1. **Face Recognition Algorithm**

Proposed by Paul Viola and Michael Jones, it proves to be a one of its kind by providing competitive results for real time situations .It not only aces the process of detecting frontal faces but can also be modeled for detecting variety of other object classes. It’s robust nature and its adaptability for practical situation makes its popular among its users. It follows a four stage procedure for the entire face detection which is: first is Haar feature selection, and then comes the integral image creation followed by ada boost training and cascading of amplifiers.

Haar Feature selection matches the commonalities found in human faces. The integral image calculates the rectangular features in fixed time which benefits it over other sophisticated features. Integral image at (x, y) coordinates gives the pixel sum of the coordinates above and on to the left of the (x,y). Ada boost training algorithm boosts the performance of cascading amplifiers by training them appropriately so as to form a strong classifier.

The strong classifiers are arranged in a cascade in order of complexity, where each successive classifier is trained only on those selected samples which pass through the preceding classifiers. If at any stage in the cascade a classifier rejects the sub-window under inspection, no further processing is performed and continues on searching the next sub-window.

The cascade therefore has the form of a degenerate tree. In the case of faces, the first classifier in the cascade – called the attentional operator – uses only two features to achieve a false negative rate of approximately 0% and a false positive rate of 40%.The effect of this single classifier is to reduce by roughly half the number of times the entire cascade is evaluated. In cascading, each stage consists of a strong classifier. So all the features are grouped into several stages where each stage has certain number of features.

The job of each stage is to determine whether a given sub-window is definitely not a face or may be a face. A given sub-window is immediately discarded as not a face if it fails in any of the stages.

* 1. **Back Propagation Algorithm**

The Back Propagation algorithm was originally introduced in the 1970s, but its importance wasn't fully appreciated until a [famous 1986 paper](http://www.nature.com/nature/journal/v323/n6088/pdf/323533a0.pdf) by [David Rumelhart](http://en.wikipedia.org/wiki/David_Rumelhart), [Geoffrey Hinton](http://www.cs.toronto.edu/~hinton/), and [Ronald Williams](http://en.wikipedia.org/wiki/Ronald_J._Williams). That paper describes several neural networks where Back Propagation works far faster than earlier approaches to learning, making it possible to use neural nets to solve problems which had previously been insoluble. Today, the Back Propagation algorithm is the workhorse of learning in neural networks.

As in the case with most neural networks, the aim is to train the network to achieve a balance between the network’s ability to respond and the ability to give a reasonable response to the input that is similar, but not identical to the one used in the training. The training of a back propagation network involves the three stages.

The feed forward of the input training pattern, the calculation and the back propagation of the associated error and the weighted adjustment. After the network has been trained, its application involves only the feed forward phase. A multi layer network can learn only input patterns to an arbitrary accuracy. A weight in a neural network is a segment of the information about the input signal that has to be stored.

* 1. **Implementation**

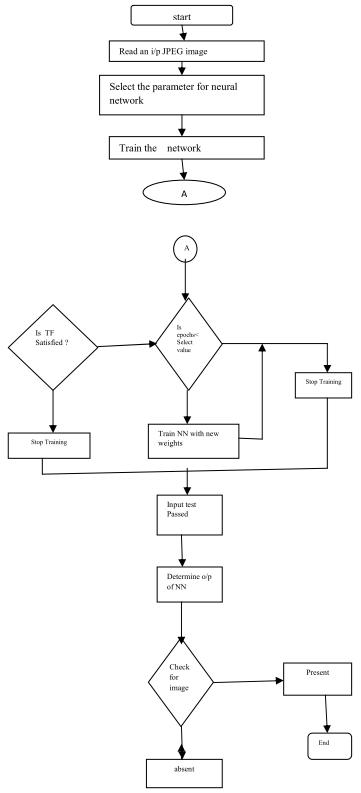
Back propagation training takes place in 3 stages.

1. Feed forward of the input training pattern.
2. Back propagation of the associated error.
3. Weight adjustment.

During feed forward, each input neuron receives an input a signal and broad casts it to the each hidden neuron, which in turn computes the activation and passes it on to its output unit, which again computes the activation to obtain the net output. During training, the net output is compared with the target value and the appropriate error is calculated, from the error, the error factorbk (delta K) is obtained which is used to distribute the error back to the hidden layer.

The weights are updated accordingly. In a similar manner, the error factorb (delta j) is calculated for units Zi. After the error factors are obtained, the weights are updated simultaneously (Fig. 3.1). The Feed forward Back propagation network is a  
very popular model in neural network. It does not have feedback connections, but errors are back propagated during training. Least mean squared error is used. Many applications can be formulated for using a feed forward back propagation network and the methodology had ls been a model for most multi layer neural networks.

Errors in the output determine measures of hidden layer output errors, which are  
used as a basis for adjustment for connection weights, forward the pairs of layer and recalculating the outputs is an iterative process i.e. carried on until the  
errors fall below a tolerance level. Learning rate parameters scale the adjustments

****

**Figure 3.1: Flow Chart of Implementation**

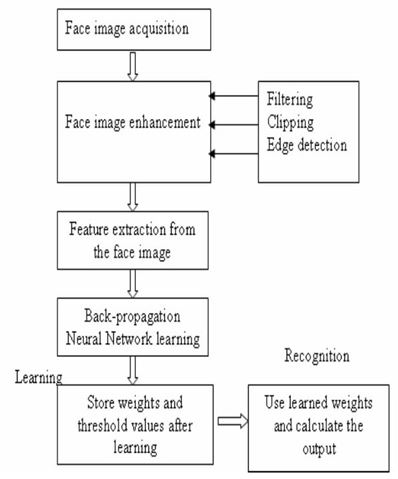
from a previous iteration and adding to the adjustments in the current iteration. The design flow of the feed forward back propagation network is as shown in figure. Here, first we read the input image which is in the format jpeg, tiff, png etc. Then by selecting the parameters required for neural network i.e., number of input layers, hidden layers and epochs, the network is trained. If the transfer function values of the input are similar to that of the transfer function values of  
known face, then the training is stopped. Otherwise increase the number of epochs. By comparing the values of the log-sigmoidal transfer function, the system decides whether the input is present or not.

* 1. **Outline of the System**

The issues of the design and implementation of the Face Recognition System (FRS) may be divided into 2 main parts. The primary half is image processing and therefore the second half is recognition techniques. The image process half consists of Face image acquisition through scanning, Image enhancement, Image clipping, Filtering, Edge detection and Feature extraction. The second half consists of the factitious intelligence that consists of Back Propagation Neural Network.

The primary part of FRS consists of many image processing techniques. The face’s image acquisition is achieved by a webcam, digital camera or using a scanner. Then the image clipping is performed using the start-point and end-point detection algorithm. The sides are detected using high-pass filter, high-boost filter, median filter or many edge detection techniques. The features are extracted; these extracted features of image are then fed into Back-propagation Neural Network.

In the second half, Technique used relies on Back propagation neural network. Within the initial techniques, the extracted features are saved into memory and using Back-propagation Neural Network; the recognition of unknown face image is performed by comparison this special pattern to the pattern for which an image module is already designed. A special advantage of the projected technique is that there's no further learning process enclosed here, solely by saving the face data of the person and appending the person’s name within the learned database completes the learning method. Within the second, extracted options are fed into the input of the Back-propagation Neural Network and the network is trained to make a knowledge data base for recognition which is then used for recognition process.



**Figure 3.2: Outline of Face Recognition System Using Back Propagation Neural Network**

As the recognition machine of the system; a 3 layer neural network has been used that was trained with Error Back-propagation learning technique with a blunder tolerance of 0.001. Outline of the entire system is given in fig.3.2 shown above.

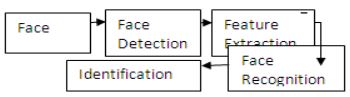
**Chapter 4**

**Back Propagation Neural Network with Surf Features**

**4.1 Face Recognition**

We focus on image support face recognition. Prearranged a picture taken from a digital camera, we’d like to know if there is any person inside, where his or her face locate at, and who he or she is. Towards this goal, we usually divide the face recognition process into three steps:

1. Face Detection
2. Feature Extraction and
3. Face Recognition



**Fig.4.1: Face Recognition Structure**

* 1. **Face Detection:** The major function of this step is to decide whether human faces come into view in a given image, and where these face are situated at. The expected outputs of this step are patches contain each face in the input image. In order to make additional face recognition system more healthy and easy to design, face position are perform to give reason for the scales and orientations of these patches. In addition serving as the pre-processing for face recognition, face detection could be used for region-of interest detection and image classification, etc.
  2. **Feature Extraction:** The following the face detection step, human-face patches are extract from images. Directly with this patch for face detection have some disadvantage, first, each patch more often than not contain over 1000 pixels, which are too large to build a robust recognition system. Second, face patch may be taken from different camera alignments, with different face expressions, illuminations, and may suffer from occlusion and clutter. To overcome these  
     drawbacks, feature extraction is performed to do information packing, dimension reduction, salience extraction, and noise attack.
  3. **Face Recognition:** The following formulizing the symbol of each face, the last step is to recognize the concept of “curse of dimensionality”. Identity of the faces. In order to achieve routine recognition, a face file is required to build. For each being, several imagery is taken and  
     their features are extracting and stored in the database. Then when an input face image comes in, we perform face detection and feature extraction, and compare its feature to each face class store in the database.

**4.2 Advantages of Face Recognition**

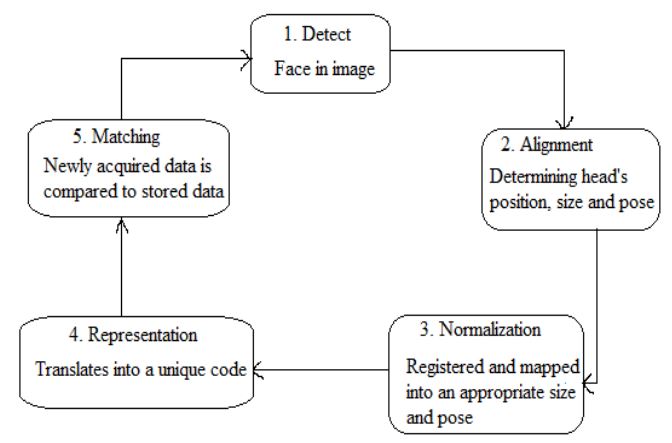
1. Non Interfering.
2. Cheap Equipment.
3. Can stop card counters, etc from entering casinos.
4. Can recognize terrorists, criminals, etc.
5. Prevent voter fraud.
6. Targets shoppers.

**4.3 Disadvantages of Face Recognition**

2D acknowledgment is exaggerated by changes in illumination, the person’s hair, the age, and if the person wear glasses. Require camera gear for user identification; thus, it is not probable to become popular until most PCs include cameras as standard equipment. Measured an invasion of solitude to be watched.

**4.4 Facial Recognition Methods**

* **Detection:** The recognition software searches the faces using the video camera when the system is attached to a video surveillance system. And if there is a face in the view, it is detected within a fraction of a second.
* **Alignment:** Once the system has detected the face, then it determines the head's position, size and pose. For the system to register the face it needs to be turned at least 35 degrees towards the camera.
* **Normalization:** For the image of the head to be registered and mapped into an appropriate size and pose, it is scaled and rotated. Normalization is performed despite of the head's location and distance from the camera. Light is not an issue in the normalization process.
* **Representation:** After the normalization has been done, the system converts the facial data into a unique code. This coding process allows for easier representation and comparison of the newly acquired facial data to facial data which is already stored.
* **Matching:** The newly acquired facial data is compared to the stored data and linked to at least one stored facial representation. The system decides if the features extracted from the newly acquired facial data are a match or not. If a score is above a predetermined threshold, a match is declared.



**Fig. 4.2: Basic process of Face Recognition system**

**4.5 Outline of the System**

The issues of the design and implementation of the Face Recognition System (FRS) can be subdivided into two main parts. The first part is image processing and the second part  
is recognition techniques. The image processing part consists of Face image acquisition and feature extraction.

The SURF features are extracted from all the faces in the database. Then, given new face, the features extracted from that face are compared against the features from each face in  
the database. The face in the database with the largest number of features matched is included as nearby face. A feature is considered matched with another feature when the distance to that feature is less than a specific fraction of the distance to the next nearest feature.

This guarantees that we reduce the number of false matches. This is because in case  
of a false match, there will be a number of other near features with close distances, due to the high dimensionality of the features. On the other hand, in case of a correct match, it is unlikely to find another feature that is too close due to the highly distinctive nature of SURF features.

The second part consists of the artificial intelligence which is Back Propagation Neural Network. The first part of FRS consists of several image processing techniques. When, the features are extracted, these extracted features of image are then fed into Back-propagation Neural Network.

In the second part two techniques are used one is based on Genetic algorithm and another one is based on Back propagation neural network. In this method, extracted features are fed into the input of the Back Propagation Neural Network and the network is trained to create a knowledge base for recognition which is then used for recognition.

**4.6 Neural Networks**

How do you recognize a face in a crowd? When faced with problems like this the human brain uses a web of inter connected processing elements called neurons to process information. Each neuron is autonomous and independent. It does its work asynchronously i.e., without any synchronization to other even taking place.

The two problems Pascal i.e., recognize a face and forecasting an interest rate have two important characteristics that distinguish them from other problem. The problem are complex  
i.e., you can’t devise a simple step by step algorithm or precise formula to give you an answer. Resolve the problem is equally complex and may be noisy or incomplete. A neural network is a computational structure inspired by the study of biological neural processing.

There are many different types of neural networks from relatively simple to very complex just as there are many theories on how biological neural net work and branch out to other paradigm later. A layered feed forward neural network has layers of sub-groups of processing elements. A layer of processing elements makes independent computation or data that it receives and passes the result to another layer.

The next layer may in turn make it independent computations and pass on the results to yet another layer. Finally a sub-group of one is more processing elements determine the output from the network. Each processing element makes its computation based upon a weighted sum of its inputs. The first layer is the input layer and the last is the output layer.

**4.7 Neural Network (Feed Forward Back Propagation Neural Network)**

A feed forward neural network is a biologically inspired classification algorithm. It consists of a (possibly large) number of simple neuron-like processing units, organized in layers. Every unit in a layer is connected with all the units in the previous layer. These connections are not all equal: each connection may have a different strength or weight. The weights on these connections encode the knowledge of a network. Often the units in a neural network are also called nodes.

Data enters at the inputs and passes through the network, layer by layer, until it arrives at the outputs. During normal operation, that is when it acts as a classifier, there is no feedback between layers. This is why they are called feed forward neural networks.

Neural Network is used for the classification of the Network. There are three layers in  
this section.

1. Input Layer: Input Layer concerns the SURF features followed by the BPNN (Back Propagation Neural Network) Optimization.
2. Hidden Layer: Hidden Layer is concerned with N number of hidden neurons and the input data is processed in the Hidden Layer.
3. Output Layer: The output layer provides the trained Neural Object which is used for the classification.

**4.8 Back Propagation -- learning in feed-forward networks**

Learning in feed-forward networks belongs to the realm of supervised learning, in which pairs of input and output values are fed into the network for many cycles, so that the network 'learns' the relationship between the input and output.

We provide the network with a number of training samples, which consists of an input vector i and its desired output o. For instance, in the classification problem, suppose we have points (1, 2) and (1, 3) belonging to group 0, points (2, 3) and (3, 4) belonging to group 1, (5, 6) and (6, 7) belonging to group 2, then for a feed-forward network with 2 input nodes and 2 output nodes, the training set would be:

|  |  |
| --- | --- |
| { | i = (1, 2) , o =( 0, 0) |
|  | i = (1, 3) , o = (0, 0) |
|  | i = (2, 3) , o = (1, 0) |
|  | i = (3, 4) , o = (1, 0) |
|  | i = (5, 6) , o = (0, 1) |
|  | i = (6, 7) , o = (0, 1) } |

The basic rule for choosing the number of output nodes depends on the number of different regions. It is advisable to use a unary notation to represent the different regions, i.e. for each output only one node can have value 1. Hence the number of output nodes = number of different regions -1.

In back propagation learning, every time an input vector of a training sample is presented, the output vector o is compared to the desired value d.

The comparison is done by calculating the squared difference of the two:

Error

The value of Err tells us how far away we are from the desired value for a particular input. The goal of back propagation is to minimize the sum of Err for all the training samples, so that the network behaves in the most "desirable" way.

|  |  |
| --- | --- |
| Minimize | Sum square error |

We can express Err in terms of the input vector (i), the weight vectors (w), and the threshold function of the perceptions. Using a continuous function (instead of the step function) as the threshold function, we can express the gradient of Err with respect to the w in terms of w and i.

Given the fact that decreasing the value of w in the direction of the gradient leads to the most rapid decrease in Err, we update the weight vectors every time a sample is presented using the following formula:

|  |  |
| --- | --- |
| Equation | where n is the learning rate (a small number ~ 0.1) |

Using this algorithm, the weight vectors are modified so that the value of Err for a particular input sample decreases a little bit every time the sample is presented. When all the samples are presented in turns for many cycles, the sum of Err gradually decreases to a minimum value.

**4.9 System Development Methodologies**

As discussed earlier, the system starts with acquisition of the image and then recognition is done at output end. Various step series flowed as image pre-processing, feature extraction, feature reduction and recognition method.

1. Face Image Acquisition

To collect the face images, phone has been used. After scanning the image can be saved into various formats such as Bitmap, JPEG, GIF & TIFF. This FRS can process face images of any format.

1. Feature Reduction

To extract features of a face at first the image is converted into a binary. E.g. From the binary image the centroid (X, Y) of the face image is calculated using eq. 1 & 2.

X = ∑ mx / m ……………. (1)  
Y = ∑ my / m ……………. (2)  
Where x, y is the co-ordinate values and m=f(x,y)=0 or 1

1. Recognition

Extracted features of the face images have been fed in to the Surf Features for features reduction & Back-propagation Neural Network for recognition. The unknown input face image has been recognized by Back Propagation Neural Network.

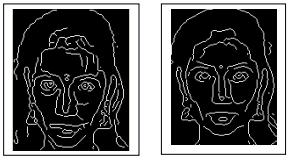
**Chapter 5**

**Comparison of Quick Mask & BPNN with SURF & BPNN**

**5.1 BPNN using Quick Mask**

The edge detector is an [edge detection](https://en.wikipedia.org/wiki/Edge_detection) operator that uses a multi-stage [algorithm](https://en.wikipedia.org/wiki/Algorithm) to detect a wide range of edges in images. [Edge detection](https://www.mathworks.com/products/image/features.html) is an image processing technique for finding the boundaries of objects within images. It works by detecting discontinuities in brightness. Edge detection is used for [image segmentation](https://www.mathworks.com/discovery/image-segmentation.html) and data extraction in areas such as image processing, computer vision, and machine vision.

Edge detection is a fundamental tool in [image processing](https://en.wikipedia.org/wiki/Image_processing), [machine vision](https://en.wikipedia.org/wiki/Machine_vision) and [computer vision](https://en.wikipedia.org/wiki/Computer_vision), particularly in the areas of [feature detection](https://en.wikipedia.org/wiki/Feature_detection_(computer_vision)) and [feature extraction](https://en.wikipedia.org/wiki/Feature_extraction). Several methods of edge detection exits in practical. The procedure for determining edges of an image is similar everywhere but only difference is the use of masks. Different types of masks can be applied such as Sobel, Prewitt, Kirsch, quick mask to obtain the edge of a face image. The performance of different masks has a negligible discrepancy. But here quick mask has been used as this is smaller than any others. It is also applied in only one direction for an image; on the other hand others are applied in eight direction of an image. So, the quick mask is eight times faster than other masks. The detected edge of a face after applying quick mask is shown in fig.5.1. Also the table for the Quick Mask.



**Fig.5.1: Edges of Face Images**

**Table I:** Results for BPNN using Quick Mask

|  |  |  |  |
| --- | --- | --- | --- |
| **No. of Face Image** | **Successfully recognized Face Image** | **Unrecognized Face Image** | **Efficiency %** |
| 5 | 3 | 2 | 60% |
| 13 | 9 | 4 | 69.23% |
| 20 | 15 | 5 | 75% |
| 22 | 16 | 6 | 72.72% |
| 25 | 18 | 7 | 72% |
| 30 | 25 | 5 | 83.33% |

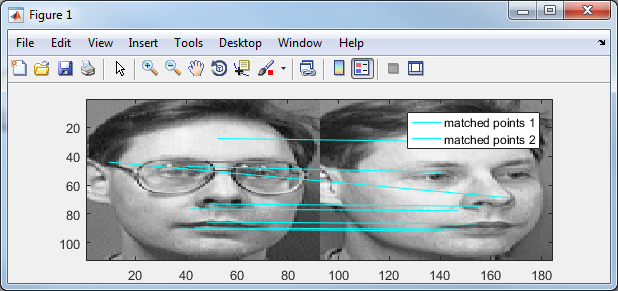
Hence the efficiency of the Face Recognition System by using the Back Propagation Neural Network is 74.78%.

**5.2 BPNN Using SURF**

There are some traditional algorithms for face recognition, which have been widely used in object detection and recognition. Among them the detector and descriptor, named Speed-Up Robust Features (SURF) suggested by Herbert Bay, attracts people’s attentions. SURF is a scale and in-plane rotation invariant detector and descriptor. In SURF, detectors are first employed to find the interest points in an image, and then the descriptors are used to extract the feature vectors at each interest point.

SURF uses Hessian-matrix approximation operating on the integral image to locate the interest points, which reduces the computation time drastically. As for the descriptor, the first-order Haar wavelet responses in x and y directions are used in SURF to describe the intensity distribution within the neighborhood of an interest point. Only 64 dimensions are usually used in SURF to reduce the time cost for both feature computation and matching. Because each of SURF feature has only 64 dimensions in general and an indexing scheme is built by using the sign of the Laplacian, SURF is much faster & efficient as compared to Quick Mask

SURF features can be extracted from images through SURF detectors and descriptors. Interest points are first extracted from each face image after pre-processing, such as normalization and histogram equalization. This turns out to obtain about 30-100 interest points per image. The Fig.5.2 & result table for the same is shown below.



**Fig 5.2: Surf matching Points.**

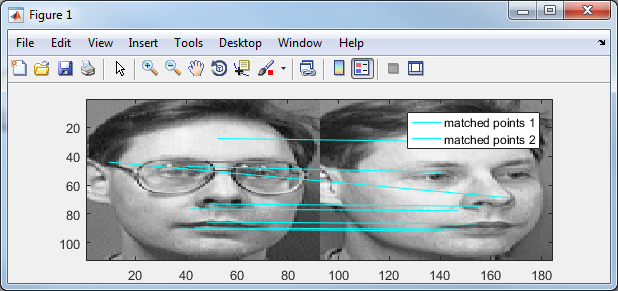
**Table II:** Results for BPNN using Surf Features

|  |  |  |  |
| --- | --- | --- | --- |
| **No. of Face Image** | **Successfully recognized Face Image** | **Unrecognized Face Image** | **Efficiency %** |
| 5 | 4 | 1 | 80% |
| 13 | 12 | 1 | 92.30% |
| 20 | 19 | 1 | 95% |
| 22 | 20 | 2 | 90.90% |
| 25 | 24 | 1 | 96% |
| 30 | 28 | 2 | 93.33% |

Therefore the potency of the Face Recognition System by using Back-propagation algorithm using SURF Features is 93.04% which is best.

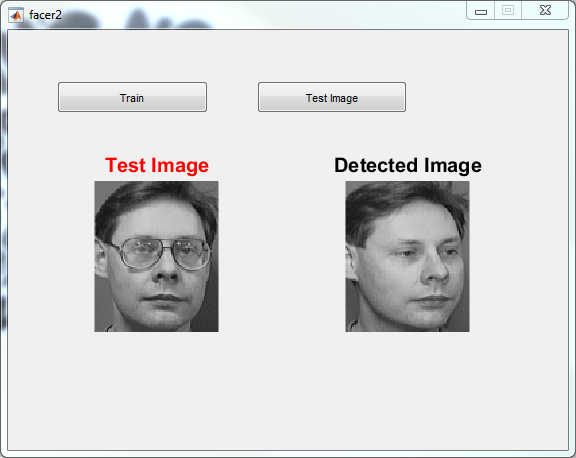
**Chapter 6**

**Images**

****

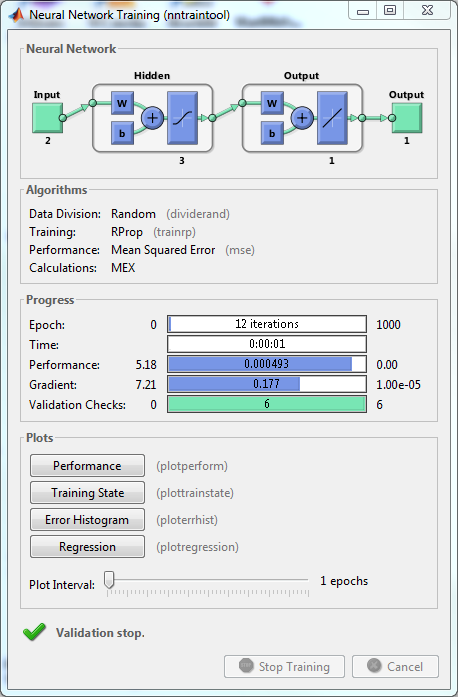
**Fig.6.1: Matching Points**

As you can see in the above fig. 6.1, the detectors are first employed to find the interest points in an image, and then the descriptors are used to extract the feature vectors at each interest point. As for the descriptor, the first-order Haar wavelet responses in x and y directions are used in SURF to describe the intensity distribution within the neighborhood of an interest point. Only 64 dimensions are usually used in SURF to reduce the time cost for both feature computation and matching.



**Fig.6.2: Image Result**

After SURF features have been extracted from images through SURF detectors and descriptors the features are matched against each other.

****

**Fig.6.3: Neural Network Training Tool**

When the training in Train and Apply Neural Networks is complete, you can check the network performance and determine if any changes need to be made to the training process, the network architecture, or the data sets. First check the training record, tr, which was the second argument returned from the training function.

tr =

struct with fields:

trainFcn: 'trainlm'

trainParam: [1×1 struct]

performFcn: 'mse'

performParam: [1×1 struct]

derivFcn: 'defaultderiv'

divideFcn: 'dividerand'

divideMode: 'sample'

divideParam: [1×1 struct]

trainInd: [1×176 double]

valInd: [1×38 double]

testInd: [1×38 double]

stop: 'Validation stop.'

num\_epochs: 9

trainMask: {[1×252 double]}

valMask: {[1×252 double]}

testMask: {[1×252 double]}

best\_epoch: 3

goal: 0

states: {1×8 cell}

epoch: [0 1 2 3 4 5 6 7 8 9]

time: [1×10 double]

perf: [1×10 double]

vperf: [1×10 double]

tperf: [1×10 double]

mu: [1×10 double]

gradient: [1×10 double]

val\_fail: [0 0 0 0 1 2 3 4 5 6]

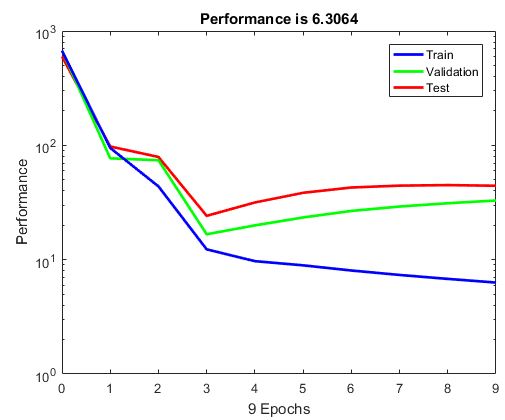
best\_perf: 12.3078

best\_vperf: 16.6857

best\_tperf: 24.1796

The tr structure also keeps track of several variables during the course of training, such as the value of the performance function, the magnitude of the gradient, etc. You can use the training record to plot the performance progress by using the plotperf command:

plotperf(tr)



**Fig.6.4: Performance**

The next step in validating the network is to create a regression plot, which shows the relationship between the outputs of the network and the targets. If the training were perfect, the network outputs and the targets would be exactly equal, but the relationship is rarely perfect in practice. For the body fat example, we can create a regression plot with the following commands. The first command calculates the trained network response to all of the inputs in the data set. The following six commands extract the outputs and targets that belong to the training, validation and test subsets. The final command creates three regression plots for training, testing and validation.

bodyfatOutputs = net(bodyfatInputs);

trOut = bodyfatOutputs(tr.trainInd);

vOut = bodyfatOutputs(tr.valInd);

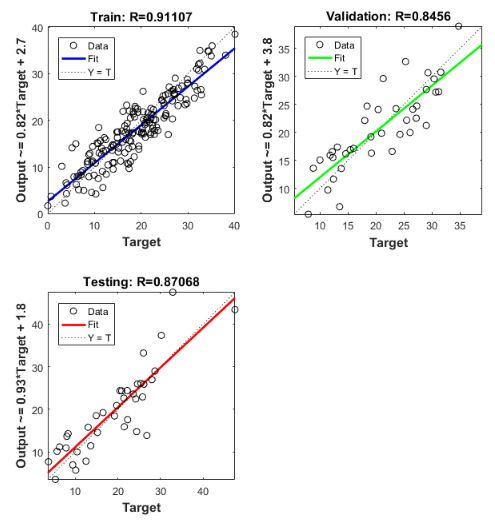
tsOut = bodyfatOutputs(tr.testInd);

trTarg = bodyfatTargets(tr.trainInd);

vTarg = bodyfatTargets(tr.valInd);

tsTarg = bodyfatTargets(tr.testInd);

plotregression(trTarg, trOut, 'Train', vTarg, vOut, 'Validation', tsTarg, tsOut, 'Testing')



**Fig.6.5: Output**

The three plots represent the training, validation, and testing data. The dashed line in each plot represents the perfect result – outputs = targets. The solid line represents the best fit linear regression line between outputs and targets. The R value is an indication of the relationship between the outputs and targets. If R = 1, this indicates that there is an exact linear relationship between outputs and targets. If R is close to zero, then there is no linear relationship between outputs and targets.

**Chapter 7**

**Conclusion**

**7.1 Conclusion**

The face recognition system is used widely in verification and identification for the security. Various image processing technique and methods are used to improve the face data to identify. This system performs human face recognition at a very high degree of accuracy. This paper deals with using SURF features in face recognition and gives the detailed comparisons with Quick Mask features.

Experimental results show that the SURF features perform only better in recognition rate than Quick Mask, but there is an obvious improvement on matching speed. Therefore, SURF features are proven to be suitable for face recognition.

The most important improvement, however, is the speed of the detector. Even without any dedicated optimizations, an almost real-time computation without loss in performance is possible, which represents an important advantage for many on-line computer vision applications. Our descriptor, based on sums of Haar wavelet components, outperforms the state-of-the-art methods.

It seems that the description of the nature of the underlying image intensity pattern is more distinctive than histogram based approaches. The simplicity and again the use of integral images make our descriptor competitive in terms of speed. Moreover, the Laplacian-based indexing strategy makes the matching step faster without any loss in terms of performance.

But the proposed system works well in spite of the lighting variations. The work can be readily used in biometric applications like access control and verification systems. In proposed work SURF and BPNN has been used because unlike rule based systems or programmed systems neural networks are flexible in changing environment. If the situation changes they are unable to operate in changed environment. Though neural may take time to learn to a sudden change, they are good at adapting to changing situations.

**References**

[1] http://www.signspeak.eu/publications/8-dreuw-SURF-Face Face Recognition Under View point Consistency Constraints bmvc 2009.pdf

[2] S. Biswas and A. Biswas,2012,Face Recognition Algorithms Based on Transformed Shape Features, IJCSI, International Journal of Computer Science Issues, IJCSI, Vol. 9, Issue. 3, No. 3, pg-445-451

[3] GuoshenYu,Jean M Morel, ASIFT: An Algorithm for fully Affine Invariant Comparison, Image Processing On Line, Vol. 2011,http://dx. doi. org/10. 5201/ipol. 2011. my-asift

[4] Face Recognition using Principle Component Analysis‟, Kyungnam Kim Department of Computer Science University of Maryland.

[5] K. R. Singh, Roshni S Khedgaonkar, Swati P Gawande,” A New Approach to Local Plastic Surgery Face Recognition Using Near Sets”, International Journal of Engineering Science and Technology (IJEST) Feb 2011

[6] <http://www.cs.ubc.ca/~lowe/keypoints/>

[7] <http://en.wikipedia.org/wiki/Artificial_neural_network>

[8] Wei-Lun Chao,” Face Recognition”, International Journal of Engineering Science and Technology, vol. 83, no. 5, pp. 705-74, 2006.

[9] A. F. Abate, M. Nappi, D. Riccio, and G. Sabatino, “face recognition advantages,” Pattern Recognition Letter, vol. 28, no. 14, pp. 1885–1906, 2007.

[10]. Mohamed Aly,” Face Recognition disadvantages”, International Journal of Engineering Science and Technology, vol. 80, no. 4, pp. 715-74,2008.

[11] J.Kirkland Reynolds and et.al,”Face recognition memory: The effects of exposure duration and encoding instruction”, Applied Congnitive, vol. 6, 279-(2009).

[12] Richard Russellô, Pawan Sinha,” Real-world face recognition: The importance of surface reflectance properties”, Perception, 2007, volume 36, pages 1368 - 1374.

[13] Dr R. JENKINS, Professor A. M. BURTON,” Limitations in Facial Identification: The Evidence”, feature of the week, 5-12 January 2008.

[14] Helms, E. and Low, B., Face Detection: A survey, Computer Vision and Image Understanding, 83, 2001, 236-274.

[15] Sakai, T., Nagao, M. and Kanade, T., Computer Analysis and Classification of Photographs of Human Face, Proc. First USA – Japan Computer Conference,1972, 2-7.

[16] Ching-Liang Su Chidchanok Lursinsap, Face Recognition by Feature Orientation and Feature Geometry Matching, IEEE Proceedings of ICPR ’96, 1996, 401-405.

[17] Zdravko Liposcak, Sven Loncaric, Face Recognition from Profiles Using Morphological Operations.

[18] S.Arivazhagan, J.Mumtaj, L.Ganesan, Face Recognition using Multi-Resolution Transform,  
International Conference on Computational Intelligence and Multimedia Applications, 2007, 301-306

[19] [US 2009238460](https://worldwide.espacenet.com/textdoc?DB=EPODOC&IDX=US2009238460), Ryuji Funayama, Hiromichi Yanagihara, Luc Van Gool, Tinne Tuytelaars, Herbert Bay, "ROBUST INTEREST POINT DETECTOR AND DESCRIPTOR", published 2009-09-24

[20] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "[Speeded Up Robust Features](http://www.vision.ee.ethz.ch/en/publications/papers/articles/eth_biwi_00517.pdf)", ETH Zurich, Katholieke Universiteit Leuven

[21] S. U. R. F. (SURF), “Herbet bay, andreas ess, tinne tuytelaars, luc van gool,” Elsevier preprint, 2008.

[22] Wikipedia, “Feature detection — Wikipedia, the free encyclopedia,” 2011, [Online; accessed 14-July-2010]. [Online]. Available: https://secure.wikimedia.org/wikipedia/ en/wiki/Feature detection %28computer vision%29

[23] P. Getreuer, A Survey of Gaussian Convolution Algorithms, Image Processing On Line, 3 (2013), pp. 286–310. http://dx.doi.org/10.5201/ipol.2013.87.

[24] Feature detection with automatic scale selection, International Journal of Computer Vision, 30 (1998), pp. 79–116. http://dx.doi.org/10.1023/A:1008045108935.

[25] K.C. Yow and R. Cipolla, “Feature-based human face detection,” Image and Vision Computing, vol. 25, no. 9, pp. 713–735, Sept. 1997.

[26] D. Maio and D. Maltoni, “Real-time face location on gray-scale static images,” Pattern Recognition, vol. 33, no. 9, pp. 1525–1539, Sept. 2000.

[27] M.S. Lew and N. Huijsmans, “Information theory and face detection,” Proc.IEEE Int’l Conf. Pattern Recognition, pp. 601- 605, Aug. 1996.

[28] A.J. Colmenarez and T.S. Huang, “Face detection with information based maximum discrimination,” Proc. IEEE Int’l Conf. Computer Vision and Pattern Recognition, pp. 782–787, June 1997.

[29] “Artificial Neural Networks”, https://en.wikipedia.org/wiki/Artificial\_neural\_network, accessed online 2 October 2001.

[30] Michael Nielsen Jan 2016, “Neural networks and deep learning”, http://neuralnetworksanddeeplearning.com/chap2.html.

[31] Tyrell turing, 7 April 2005, “Feed forward neural network”, https://en.wikipedia.org/wiki/Feedforward\_neural\_network.

[32] Priyanka, Dr. Yashpal Singh “A Study on Facial Feature Extraction and Facial Recognition Approaches” International Journal of computer Science and Mobile Computing, Vol. 4, pp 166-174, 2015.

[33] Geng Du, Fei Su, Anni Cai “Face Recognition using SURF features” Pattern Recognition and Computer Vision, Vol. 7496, 2009.

[34] Ambika Ramchandra, Ravindra Kumar “Overview of Face Recognition System Challenges” International Journal of Scientific and Technology Research, Vol. 2, pp 234-236, 2013.