2001

Segmented face detection using clustering

Gunturi Srimanth

Segmented Face Detection using Clustering

by
Gunturi Srimanth
Dr. Roger S. Gaborski, Advisor

Advisor: _____

(Dr. Roger S. Gaborski)

Reader: _____

(Dr. Peter G. Anderson)

Observer: _____

(Dr. Edith Hemaspaandra)

Februaury 2001

Department of Computer Science
Rochester Institute of Technology

Segmented Face Detection using Clustering

Abstract

Perception forms a very important part of learning. The way we perceive things has a lot to do with how we understand. It forms a very crucial link in our build-up of knowledge. Living organisms have a remarkable ability of understanding spatial information. It is due to their inherent ability of generating native organizations, models, etc, and most importantly, their ability to generalize and infer - based upon symmetry, probability, familiarity, etc, that allows them to instantly adapt their knowledge to the given surroundings. It forms a very basic step in survival. When trying to make machines intelligent, one of the first hurdles faced is the problem of perception - questions like which data is important(light, color, texture, sound, etc.)? how much importance should each data be given? etc come up.

The purpose of this work is to observe the workings, and results, of trying to detect faces in images, by searching separately for the eyes, nose and mouth regions of a face. The regions are searched independently of one another, using clustering and Neural Networks. Broadly speaking Clustering is used to locate generalized face regions, and Neural Networks are used to map accurately the decision surface. The search for eyes, nose and mouth is done separately, in an attempt to reduce the complexity of the intensity map being searched, thus hoping to improve upon the accuracy and reliability of the detection process. Also it provides for simultaneous parallel search, which is of high importance for real-time tasks like face-detection.

To observe the effectiveness, and generalization capability of the process, a very small and localized dataset was used for training purposes. Also, no feature sets or such abstractions were used in the training and implementation of the work - only raw data was used for detection - this was done to reduce the effects of selection of a wrong feature set.

Contents

List of Tables

List of Figures

1 Introduction

Face detection forms the basic requisite for many higher level processes of artificial intelligence like face recognition, expression analysis, person tracking, eye tracking etc. Most of these applications assume the presence of the face or appropriate regions of the face such as the face, eyes, mouth etc for efficient working.

Generally speaking the face is an interesting object for pattern recognition and detection - It has fixed underlying structural shape with myriads of discernible variations in the external make-over due to varying organ sizes, skin color, lighting, angle of sight, and additional interferences (like beard, glasses etc.). Various approaches have been taken for solving the problem, most of which generally fall under the categories of Probabilistic approaches, Feature based approaches, hybrid mechanisms like Neural Networks and Support vector machines.

Functionality: The goal of the work is to detect face regions in a given image/photograph. The input image can be of any size. Once the image is given, a decision window is moved in incremental positions over the image trying to decide if the given window region contains a face or not. After covering the entire image, the process is repeated with a window of modified dimensions. Ultimately each window is resized into a $M \times N$ rectangle. In the present implementation M and N are 30 each. The window is then split up into horizontal regions with similar characteristics called *Sections* and each region is analyzed separately. Some amount of image processing like histogram equalization and noise removal is performed to lessen the effects of dim-lighting, poor contrast etc. These regions consisting of a fixed number of pixels are then converted to a point in high-dimensional space, where the dimensionality of the space is the number of intensity values in the region. In the learning phase these points are made to form clusters.

Care has to be taken in selecting appropriate non-face images, as the information deduced from them is absolutely crucial for deciding upon a region or window. Good choices for such faces are those which are near the boundary of a face cluster but which do not belong to it. In the testing phase a decision of whether a test region belongs to a face cluster or not is made by some clustering analysis algorithm. Distances of the test point from each of the clusters can be taken as a criterion for judgment. A decision can be taken by a simple Euclidean distance algorithm. The clusters can also provide information for a higher level analyzer like Neural Networks to make

the decision.

1.1 Challenges in Face detection

Detection of faces comes under the broad category of Pattern Matching and Identification. Faces in images form an interesting set of two dimensional patterns which are created by a fixed three dimensional solid with varying sized components, with the relative locations of those components fixed. Adding to the difficulties of having varying sized components are three other parameters: (1) Skin Tone and Color. (2) Lighting Conditions and shadowing affects. (3) Effects of additive components like glasses, mustaches, beards etc.

Skin Color is an important parameter in face detection approaches based on color information. In the present work it does not have a significant effect due to the usage of intensity/grayscale values for detection, and the application of histogram equalization. But even in intensity based approaches it helps to have a good initial contrast. In a typical face, if the complexion is dark, the contrast between the eye-sockets, lips, nostrils and skin tends to get hazy. Along with that even slight noise variations get magnified when correcting through histogram equalization. Though there are methods to improve on such situations, there is an increased risk of an anomaly. Lighting is possibly the single most important parameter which can have adverse effects on the decision process. Lighting can seriously effect how a three-dimensional solid is imagined, especially when looking at an intensity map like a grayscale image. Light can remove crucial edges, hide and distort vital feature-intensities - changing the three dimensional visualization of the solid resulting in incorrect decisions. Humans perform very much better in such situations due to the extensive memory, training, and the ability of the decision process to generalize things that have been learnt. Additive features like glasses, mustaches, and beards tend to change the landscape, overpowering the underlying vital components like lips, eyes etc. A solution could lie in thinking of such cases as a different set of face-patterns and including them in the training process. As many cases as possible should be considered for training.

Most of these difficulties are significant while considering unoccluded frontal faces. But in real life there are faces which are tilted, faces which give only profilic views, faces which are occluded by some masks, and faces which have varied expressions etc., which could easily confuse a decision process into taking a wrong decision. Work in detecting occluded faces, non-frontal views, and face patterns in adverse lighting conditions will go a long

way in understanding our throught process, and developing independent and more robust computer systems.

1.2 Previous work

Work in the field of face detection can broadly be classified under three main approaches, which are based on how the input data is treated - Approaches based on color, approaches based on plain intensity values (monochrome, grayscale), and Motion based approaches. Though not much work has been done in incorporating motion into the decision process for face detection, Motion based approaches could incorporate either color data or intensity values to increase their efficiency. Most of the color based approaches use the presence of skin tone in the data to hone in on the regions of faces. Color has very useful characteristics in the fields of face detection and surveillance - It provides the ability to quickly localize on hot-spots of possible human presence and also possesses orientation invariant information, which is very valuable in human tracking. In [25], a tracking mechanism is presented which identifies and tracks possible human faces. In [23] human bodies are tracked based on the presence skin color-tone information. Difficulties in color based approaches arise due to the nature of color itself. It is a sensation rather than a physical phenomenon. This makes the results very hardware-quality dependent, varying from camera to camera. The environment in which color data is taken also has significant impact on the values generated by the hardware motion, orientation changes, directional light, ambient light etc. have an impact. Though color based approaches are very sensitive to a lot of factors they provide a very efficient approach in localizing on possible facial regions an effective alternative to the computation intensive bounding box approach in face detection and tracking.

Compared to the above approaches of Motion and Color based detection, most of the work is generally done in the region of Monochrome/Intensity based data. This is done because of various reasons (1) Data collection hardware and networks with high capacity transmission capabilities are hard to manufacture and costly to build. Cheaper and affordable technology would go a long way in helping humans. (2) Getting the correct color tone can prove to be very difficult. It can be offset even with the slightest of environmental changes. Effects of color mixing, insufficient intensities in one of the color channels etc. can result in significant color changes which might confuse color-tone searching algorithms. Sufficient algorithms for such corrections exist in the grayscale domain. (3) Generally solving a problem with the least amount of information is a useful thing - Grayscale contains the needed features in the least possible visual data and thus provides deeper insight into

the organization of data and how our brain proceeds to classify patterns.

Approaches using monochrome information come under the general categories of 'Pure Pixel Intensity' and 'Feature Based' detection. Pure pixel intensity based approaches include Probabilistic [14] and Statistical based approaches. They also include Neural Networks [16] [2] [1], Support Vector Machines(SVM) [18] and other such classifiers. Much work in the Probabilistic and statistical approaches revolves around a decision making process involving the values of mathematical criteria on an image which are quintessential to solving the problem.

Feature based approaches extract some high-level information from the pixel intensities and then base their decisions on them. They include Correlation templates, Deformable templates, and Spatial image variant approaches.

Figure 1: Summary of Face Detection Approaches

- *Motion Based Approaches*
- *Color based approaches*
  - Detection based upon skin color. (An alternative to the window box approach).
- *Monochrome/Intensity based approaches*
  - Pure Pixel Intensities Based
    - Probabilistic and statistical Approaches
    - Neural Networks and other Classifiers
  - Feature Based
    - Correlation Templates
    - Deformable Templates
    - Spatial Image Variants

1.3 Current Approach

1.3.1 Introduction

The primary objective of the current work is to observe the effects and usefulness of clustering, and Neural Network based decision, in the area of face detection. The entire work is made up of a training phase and an implementation phase. The face detection process uses Clustering and Neural Networks in making the final decision of whether a window is a face or not. In the training phase, Cluster generation and training of the Neural Networks based on the clusters generated take place. In the implementation phase both the cluster information and trained Neural Nets are employed to decide on a particular window.

1.3.2 Training Process

The training process is the part which develops all the decision making algorithms to work within reasonable error constraints. It is in this section that the robustness of the method is decided. The general goal of the problem is to carve out the decision surface in high-dimensional space. The clustering part of the process helps in honing into those regions and confining the decision problem to those areas with high face probability. The effectiveness of the clustering process lies in the data presented in training. The factors involved are discussed in the Clustering section in detail. Briefly, the factors are properly distributed data, preprocessing operations, clustering algorithms and cluster distance measurements. Once the clusters are generated, various criteria can be used to judge directly whether a given window (point in high-dimensional space) is a face or not. They can range from simple Euclidean distance measurements from the generated clusters, to more complex probabilistic distance measurment approaches. Since a decision process by clustering alone requires a huge and exhaustive dataset, Neural Networks are trained to determine whether the given pattern is a face or not. A different set of data is provided for the networks' training process because it has to take into consideration non-face data - something that is not used in cluster generation and cluster decision processes. A method involving non-face clusters could also be used to detect faces. The bulk of the training process constitutes (1) Face/Data gathering for Clustering and Neural Networks. (2) Clustering and Cluster Centroid generation. (3) Training of Neural Networks for the decision surface.

1.3.3 Implementation Process

In the implementation process an image is taken in by the software as input and an output of Face-Coordinates is generated. Current limitations include inability to detect rotated faces, faces with a lot of sideways tilt (more profilic in nature), extreme lighting conditions, etc. This work tries to attack the problem of face occlusion by dividing the face into three horizontal regions of eyes, nose and mouth. The splitting of the window into three Sections is due to the observation that they form symmetrical patterns which are easily identifiable and distinguishable. Also it helps to isolate noises, obstructions and deviations of each of the three Sections from one another. The lower half of the face due to its flexibility is generally difficult to detect when considering pattern identification. Also adding to the problem is the presence of a huge number of occluding objects in images beards, mustaches, teeth effects, cigarettes/pipes, veils etc. This gives a simple intensity based approach a lot of difficulty.

In the implementation process an image is taken up in grayscale format and a window is made to pass over it. The region in the window is the candidate for the decision algorithm. Since faces come in different sizes and locations, the window is resized and moved in very small increments across the length and breadth of the image. The training process is performed so as to handle the effects of the shift on faces. Every window of all sizes are ultimately resized into a $30x30$ matrix. This is the basic matrix which will be tested for being a face. This data is then subjected to some image processing so as to improve the quality of the data and its features. This generated data is now given to the clustering and Neural Network algorithms so as to generate the appropriate decision mechanisms.

In the clustering process the given data is converted to a points in high-dimensional space, where the dimensionality is the number of pixels in the data($30 \times 30 = 900$). If by some means this point in the high-dimensional space is considered to be part of a face cluster(generated from the training process), then that point (or image window) is considered to be a face. In the present work a simple Euclidean decision measurement of sufficient distance (fixed empirically) is the check for a point to belong to a cluster. The purpose of this approach is to remove a vast majority of non-face points from the decision process. The data if accepted by the clustering part as a probable face pattern, is presented to the neural network, which would then decide if the given data(window) is a face or not.

2 Preprocessing and Data Preparation

2.1 Introduction

In nature, every form of energy is represented in a continuous fashion where infinite samples can be taken off any interval of time. There is a huge abundance of information which helps a lot in understanding the various aspects of life. Intelligent systems in nature learn by trial and error on how to cope and get used to the overload of information - to make use of only a small amount of it for survival. Millions of years of evolution has created a system with a very high level of understanding.

In the chain - of vision to perception, data is converted from a continuous fashion to electronic pulses, in stimuli, and neural interactions. But data here is not affected by *digitization* due to the presence of huge amount of hardware operating in a parallel fashion. This provides sufficient information to keep the organisms informed of their surroundings to survive.

In any system built, its ability to be effective is heavily dependent on the input, and the input will be as good as the hardware can get. In computing, major deficiencies in hardware can be made up for using corrective procedures in software. Image Processing contains a good deal of resources to handle the deviations of the physical world.

For training purposes, an effective set of faces have to be collected - they should be a fair representation of the general dataset. To get a good dataset, one of the requisites is that all the test patterns should be standardized to reduce the effects of external influences. Doing this helps to bring out the true characteristics of the various patterns - both for training and testing. In the present work images are blurred to reduce effects of noise, and histogram equalization is performed to generate the standardized images for training. Similar techniques are applied to the images being tested.

The success of the clustering process depends upon the quality of the dataset. For the purpose of cluster training we have to generate a database of faces which would be enough to represent the general concept of a face in high-dimensional space. The necessity for a huge database is that we have to get a pretty good distribution of face models, so that when implementing even if a new face is got, the clustering algorithm will have no trouble in deciding how deviant it is from a face cluster.

2.2 Data Representation

Data Representation is an important factor in the making of an efficient system. Sometimes it makes a lot of difference in understanding and successfully implementing an idea. The way in which data is understood and handled, as well as the way in which data is gathered are discussed in this section.

2.2.1 General Data Flow

The input to the program is an image for which face coordinates have to be detected. The image should be in PGM [1] (Portable GreyMap) format. Once the image is got, some image processing is performed on it and it is then checked for faces. A *face* is nothing but a square box containing the face from the eyebrows below and mid-chin upwards. Laterally it is wide enough

to enclose the extreme tips of the eyes. Generally faces fit in this description. A *window* of the image is taken up for probable presence of a face, and it is split up into three equal sized rows. The first row is for the eyes, the second row is for the nose, and the third row if for the mouth. This is done because generally changes in structure are localized to these regions. The three rows are from now on handled individually from one another and decisions are taken on each of them separately .

Each of the three strips or *Sections* are now subjected to some image processing. The process of having three regions helps because each of the three regions have distinct differences from the others - The eyes are depressive regions, while the cheeks and nose regions have protruding regions, the mouth has the most deformed and flexible characteristics compared to the other two. Image processing of Blurring, Histogram Equalization, and Gradient Correction performed on the sections are effective when done individually.

In the training phase, the three *Sections* are converted to points in high-dimensional space where the dimensionality equals the number of pixels in each of the strips. These points form the clusters from which the Centroids are taken for analysis purpose. They are used by the clustering algorithm and the Neural Network algorithm. In the Neural Network algorithm the distance

[1] Appendix(1)

of possible candidates form the centroids is used as the input. In the Neural Network training process, faces as well as non faces are used for the training. The Neural Networks create the decision surface in high-dimensional space. There are three Neural Networks created for each of the three *Sections* of the face being analyzed. Each of the Neural Network is trained with a separate set of face/non-face data. Due to huge amount of variations in the mouth zone training in that area is difficult.

After a particular window is accepted by the cluster decision process to belong to a possible cluster, its distance from one of the cluster centroids is given to an appropriate Neural Network (eyes/nose/mouth) for its decision on the presented strip. The Neural Networks produce the appropriate confidence values and depending on that a window is either accepted or rejected. This process is repeated for varying sizes of windows which are swept across the entire image.

2.2.2 Implementation Formats

For any system to be useful it must work in an efficient fashion. In developing and implementing efficient systems, care must be taken to select the right tools and formats for implementing. Most of the work done here has been done on SunOS running on UltraSparc machines, but has later been ported to run on Linux systems. A huge bulk of the programming has been done in C Language, though snippets of Matlab and shell programming have also been used to do image processing and automated unix tasks respectively. The Graphical User Interface (GUI) part of the work has been done using X Windows programming specifically using XLib. The code is available in both SunOS and Linux. It can be easily be ported to Win32 systems by replacing the GUI modules. The whole process of implementing the training and testing has been done in various individual steps.

- **Step 1:** Collection of images - Here the images are collected, from face databases, hand-picked images, images from NN training (bootstrap) process etc. The images are carefully selected to contain required characteristics. The images are of dimension 30x30, and are of PGM [2] format.

- **Step 2:** Image Splitting - As explained earlier, analysis is done on an image by splitting it up into three horizontal *Sections*. In this step the

[2] Appendix (1)

30×30 dimension images are taken up by the program and separated into three different directories one each for the *Eyes*, *Nose*, and the *Mouth*. The generated images are of the same PGM[3] format having the dimensions 30×10. This is done for **all** the faces of cluster training and Neural Network training as the horizontal strips now form the basis for judgment.

- **Step 3:** Image processing on the collected strips - This is a good point to perform image processing on each of the horizontal *Sections*, as it provides good isolation from the others. Image processing comprised of Blurring, Histogram Equalization and Gradient Correction are performed on the image strips. Blurring and Histogram Correction are performed by C programs whereas Gradient Correction is performed by Matlab. For the purpose of gradient correction the files are batch converted to *.JPG* format, and are later converted to *PGM* format. All the files retain their initial dimensions of 30×10.

- **Step 4:** Collect the strips into individual *.bin* [4] Files.

  - Cluster Training: In this process only face *Section* files are collected as only face clusters are desired for face detection. The face files collected are in PGM format with dimensions of 30×10. A Program takes input as a directory and generates a *.bin* file for all the faces present.

  - Neural Network Training: In this process both face and non-face file strips are needed. The faces are in PGM format of dimensions 30×10. All the files are placed in a directory and an *.out*[5] text file is created containing the Neural Network outputs for each of the images in that directory. The associated program will generate a *.bin* file containing all the face/ non-face images. Care must be taken to verify the correct correspondence of output in the *.out* text file to the images in the *.bin* file.

- **Step 5:** Apply the Clustering algorithm to get *.cls* [6] (cluster) files Once the *.bin* file for each of the horizontal sections is got, a clustering

[3] Appendix(1)
[4] Appendix(3)
[5] Appendix(5)
[6] Appendix(2)

algorithm is applied to group the data into a group of 50 possible clusters. Thus three *.cls* cluster files - one each for nose, eyes and mouth - are generated.

- **Step 6:** Generate the Neural Network Files - The training of the Neural Networks requires the generated cluster information from the previous step. The inputs to this step are the cluster files got from the previous step, the *.bin* files got from Step 4, the *.out* files containing the NN outputs for each of the images presented. The program would turn out a *.nn*[7] file which contains the trained Neural Network and the cluster point which will be used as a basis for judgement in high-dimensional space. The generated NN takes in a 30×10 window strip and generates a single floating point confidence value. The NN has a single hidden layer and has the dimensions of $300 \times 100 \times 1$.

- **Step 7:** Testing This is the final step in the entire process, which ultimately finds faces in images. Its input is an unconstrained image with any dimensions in PGM format. Its other inputs include the cluster files and the Neural Network files for each of the horizontal sections. Its output includes a graphical indication on the input image of the presence of faces, and a text file containing the coordinates of faces in the image.

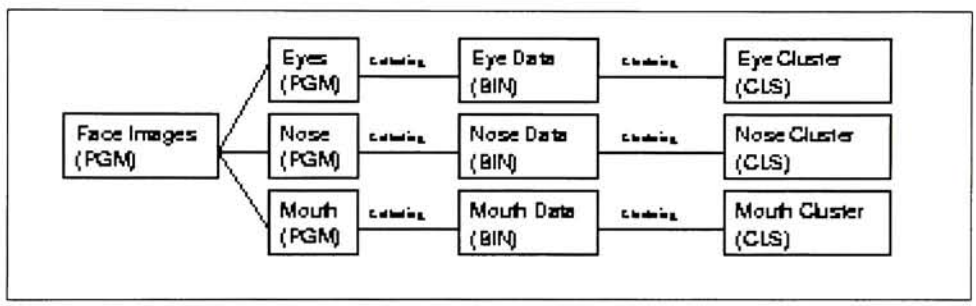

Figure 2: Clustering Implementation Format

Since the image window is of size 30×30 pixels, with each pixel having 256 intensity values, the search space has 900 dimensions with 256 units on each dimension. Using this representation, each possible face window could

[7] Appendix(4)

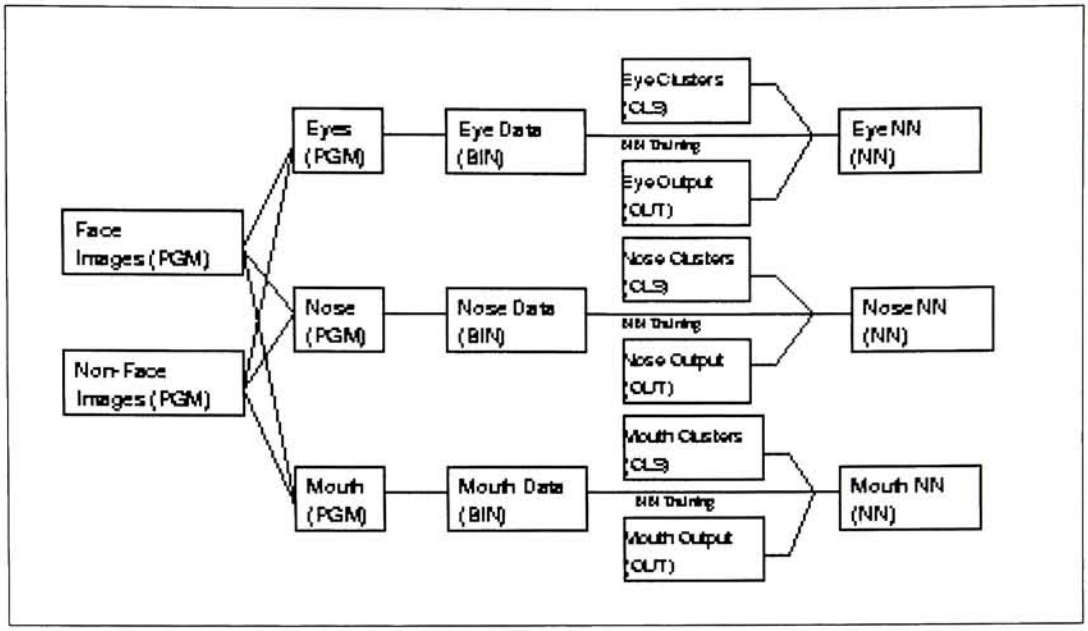

Figure 3: NN Implementation Format

be converted into a point in high-dimensional space. Due to the horizontal sectioning of the window into three regions, smaller spaces of 300 (30×10) dimensions are analyzed.

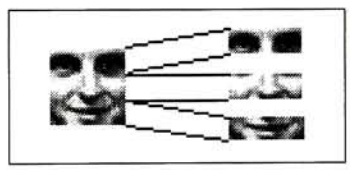

Figure 4: Sections

Table 1: Formats Used See Appendix

Acronym	Format/Purpose
PGM	Portable Grey Map
JPG	Joint Photographic experts Group
BIN	Collection file of images
CLS	Cluster file
OUT	Neural Network Output file
NN	Neural Network data file

2.3 Data Preparation - Training faces

2.3.1 General Process

By this time important and useful face-images for the purpose of training are collected by various means. The images having different characteristics (color, format, orientation, intensities) are standardized by the following approach.

Using a common image-processing software (PaintShop), the images of different formats are first converted to *ascii encoded* PGM image format. The images are then resized to a 30×30 dimensions. For the clustering face database we have only face images whereas for the Neural Network training database we have both face and non-face images. All the hand picked face-images are cropped manually in such a fashion such that only the pertinent and permanent features of the face are visible.

The pertinent features include the eyes, nose, and mouth. The smallest possible square which encompasses these parts is taken up as a bounding box and then resized to the dimensions of 30×30. Laterally the image encompasses one end of one eye to the other end of the other eye. Vertically it stretches from the eyebrow region to the region below the lower lip. Normally all faces will fit in such a square region, or a square which fits these regions is taken up.

The generated images are now given as input to a C program which breaks them up into three equal sized horizontal *Sections* This includes the eyes in one section, the nose in the next, and the mouth in the third section. The main input image is taken from a specified directory and the three child images that arise from it are placed in three separate sub-directories for the eyes, nose, and mouth respectively. The generated images are in the same

ascii encoded format of dimension 30×10.

This splitting of the image into horizontal sections allows us to isolate some of the structural features of the face. For example the section containing the eyes are in depressions and hence generally might be of lesser intensity than the rest of the face. Including this lower intensity region in image processing with the other sections results in less effective results in say histogram equalization as the range of grayscale is saturated and local features do not show up. For this reason image processing is done locally on each of the sections.

Once we have faces in the form of eyes, nose, and mouth separately we are ready to do further processing on them.

The images collected to this point are ones which are taken straight out of the natural environment. Various factors determine the quality of the image - dust and pollution in the atmosphere, shadows from nearby objects, limited resolution of the apparatus taking the image, light/vision path obstructing objects etc. Though deviants like visual obstructions must be corrected by physically removing the obstruction, things like shade correction, motion induced blurring correction etc., can be done by the help of image processing. Tools and algorithms exist to correct such deviant data. The subject of the next section is to observe in detail the behavior of such algorithms and their application to the present work.

2.3.2 Types of Input

One of the important factors in the successful working of the clustering and Neural Network algorithms is the proper type of input. In this section we shall discuss the types of faces to be considered for training. Also to be taken into account are the effects of facial expressions.

Having three regions helps because each of the three regions has distinct differences from the others - The eyes are depressive regions having darker qualities, while the cheeks and nose regions are protruding regions having very light shadings, the mouth has the most deformed and flexible characteristics compared to the other two. This was done to sort of isolate the effects of one section on the other two .

All different types of input are to be standardized so that the differences between inputs is only the spatial differences, rather than the effects of external environment or the deviations of input devices. The various processes involved are discussed in detail in the next few sections.

Lighting plays a very important part in the analysis of intensity based images. It has the ability to completely change the appearance of the surface just by the direction it comes from.

- Face Expressions: One of the main difficulties in face-detection is the myriad possible spatial permutations of the eyes, nose, mouth, the contours of the cheeks, forehead, chins, ears etc. Compounding this problem is the flexible structure between each of them. The ideal dataset would be a list of all the possible faces, in all the possible expressions. But due to its infinite data-space, some form of quantization has to be done to get the best representation and generalization of faces and expressions. The present approach of having three sections helps here as it allows the matching of an eye with the entire set of nose and mouth sections making it robust in terms of training all the possible facial expressions.

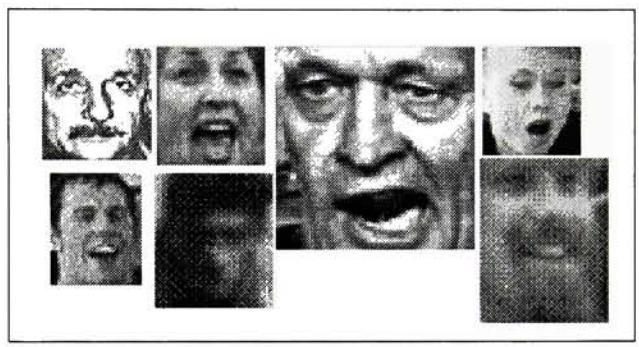

Figure 5: Sample Expressions

- Eye Section: This region is characterized by two dark patches in the rectangular region. Due to the surface contour of the eye region, lighting has an impact on the intensity values. The easiest of possible lighting is when it is generated from the top of the face - this generates perfect dark spots which are clearly distinguishable. In the presence of light from one of the sides the shadow causes asymmetry in the darkness leading to rejections. For training purposes such eye sections were taken in, as long as they seemed to resemble eyes in shadows.

  The intensity patterns generated by the eyes even depends on the resolution of images. Low resolution images are helpful because they gener-

alize the faces well and hide the discrepancies of having too much detail - which messes up the big picture. Details like the eye's white regions, teeth in the mouth region etc, can become more prominent enough to confuse the algorithm into rejecting good candidates.

The eye section starts from the top of the eye brows to the region just below the eyes. It is the first one-third part of the face region. The major components of the eye section are the eye-white, the cornea, the eye-lid, and the eye-brow. Of these the eye-brows and the eye-lids help in marking the region prominently.

Figure 6: Eye Sections

These images have been blurred to reduce some of the erratic effects of individual pixels. In most of the images we can see the effects of the eyelids when they are closed they reduce the darkness of the eye-regions, generating a thin bar at the lower regions in the eye section.

Regarding the amount of shading, the darker the patches of eyes the better it is for identification. But for training purposes, all the possible types of eye shutter positions are emplyoyed to make the process robust.

- Nose Section: The nose section consists of a tapering ridge like formation in the middle with cheeks on either side. In this section the width of the nose plays an important part of the analysis. Also affecting it is the shape of the mouth in the face. This is due to the interaction of the mouth shape with the contour of the cheeks.

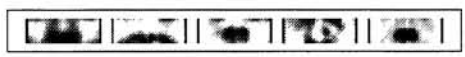

Figure 7: Nose Sections

- Mouth Section: This is where most of the deviation in the face occurs. This is because of the structure and shape of the mouth. It is a flexible orifice, which can be of varying shapes and sizes in expressions - expressions like smiling, laughing, talking, surprise, etc. It takes on shapes anywhere from flat line to a round or oval(surprise/astonishment). As can be seen in the images below mouth sections sometimes look very

similar to eye sections, with the tips of the dark eye patches forming the mouth tips. Care must be taken to train the NN to reject such sections for eyes and vice-versa.

Figure 8: Mouth Sections

These need to be represented in taking up the data set for training.

- **Dangers in Data Gathering:** Here the data set gathering for training should be done with utmost care because even in some meaningless section of the images there maybe eyes, nose or a mouth present. If care is not taken to identify them, we may run the risk of training a neural network of accepting non-faces and rejecting a face pattern type, making it unstable. For example, in the below image the seemingly plain image on the left when subjected to histogram-equalizitation produces the image on the right, which looks very much like the eyes. The nose

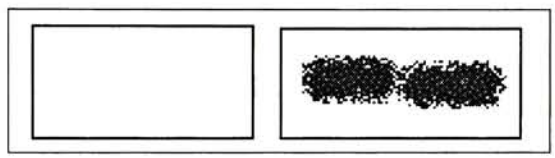

Figure 9: Effects of Histogram Equalization

section forms the biggest problem in low resolution images because the intensities showing the nose are lost, and all that is left is a plain white region which can easily be confused for a 'plain background' something which is not useful in training because plain backgrounds would be identified as face nose sections.

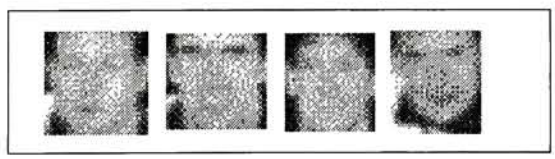

Figure 10: Low Resolution Features

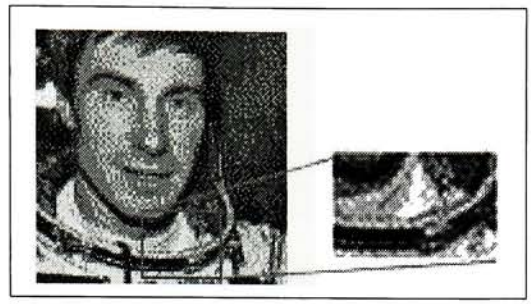

Figure 11: Hidden Face

2.4 Histogram functions

One of the problems that can have a significant effect on the decision process is the quality of the input data and images. In the real world the quality of the images depends on a variety of factors from the devices used to gather the data, to the random scenarios generated in the real world. Most of the problems dealing with the vagaries of devices can be controlled to a large extent by the understanding and workings of the devices, and their corrective solutions like image sharpening, etc. Other deviances can be corrected by applying corrective algorithms - Motion based blurring, high or low intensity images, loss of good focus etc. are some deviances whose effects can be lessened.

Since images here are of the intensity type most of them can be corrected by using histogram functions. Faces both for training and testing come in different colors and shades. The aim of this section is to lessen the effects of deviants. Skin color is normally constant on the face with the variation coming off from the angle of the surface of the skin to the line of sight. But apart from that a major portion of the skin is perpendicular to the line of sight. Extreme deviations with skin shades can be reduced by using histogram equalization method, as this spreads the shade evenly giving good contrast and feature characteristics.

2.4.1 Histogram Equalization

Histogram Equalization plays a very effective role in image processing. It attempts to bring out the features and contrast by spreading the histogram evenly from clustered and clumped up histogram distributions which are

characteristic of low/high intensity and low contrast images.

In the real world lighting is not always perfect - some random light source or object obstruction causes variations in intensities which can confuse the decision process. Such effects tend to make the image brighter or darker, decreasing the quality of features on the face pattern. Since quality of the features are the most important requisite of the decision process, the effects of such lighting should be lessened.

Histogram Equalization is based on the principal of achieving better contrast by increasing the dynamic range of the of the pixels in the histogram. Increasing the dynamic range of pixels results in better contrast of the image.

If r represents the Grey values of an image, then in the normalized form $0 \leq r \leq 1$ where 0 represents black and 1 represents white. Let $T(r)$ be a transform function which will produce a value s for every r. $s = T(r)$. The probability distribution function for r, $p_r(r)$ is the histogram of the image. Our goal through the transformation is to obtain an uniform distribution for the histogram so that the dynamic range of the image pixels is good. The probability density functions for r and s are related by the equation

$$p_s(s) = [p_r(r)\frac{dr}{ds}]_{r=T^{-1}(s)}$$

When the Transformation function selected is a **Cumulative Distribution Function**

$$T(r) = \int_0^r p_r(w)dw$$

the transformed variables' probability distribution function $p_s(s)$ has an uniform density of 1 ($0 \leq s \leq 1$). Since an uniform distribution indicates a good dynamic range for the pixels, the resultant image will have a good spread of intensities giving it good contrast - something very important for the present intensity based approach.

In the present work, histogram equalization is quintessential to the detection process. It gives a very good representation of the intensity based feature of the face. It reduces effects of shadows, variations of skin color, etc.

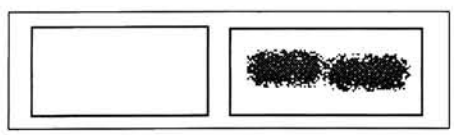

Figure 12: Histogram Equalization Effect

Shown below is how localized processing of each Section results in better features. On the left is the effect of Histogram Equalization on the entire image and on the right its effect on each of the individual sections.

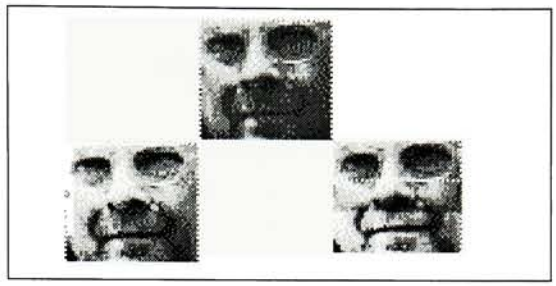

Figure 13: Comparison of Normal and Sectioned Face

2.4.2 Blurring

One of the other problems in images is the presence of noise. This is due to many reasons - effects of device capturing the image, the quality of the film holding the image, presence of physical obstructions like dirt, or simply the effect of pixelization, which is something common to digital images. Similar to Histogram equalization, in extreme cases it results in ineffective facial features.

The solution to this problem lies in identifying the charateristics of the noise present in the images and reducing their effects. One approach can be to reduce a noise pixels' effect by approximating it with surrounding pixels. This results in decreasing *local deviations* in intensities. This process of subjecting an image with a *low pass filter* is called Blurring.

Care must also be taken not to overdo blurring, as it can simply normalize the features, blending them into the surrounding. Determining the type of noise, and the effective blurring approach should be done taking into account the characteristics of the noise. There are many research works done for automating the process of noise type identification, and subsequent removal approaches. In the present work blurring is done on each of the three horizontal sections separately. This is done so as to give more attention to the correction process in the three rather distinctly shaded regions.

The effect that noise has on the clustering of data is that the *face point* is pushed away along some dimensions from the mean region of a cluster in high

Figure 14: Blurring

dimensional space. This in turn weakens the decision process of the clustering algorithm. Blurring tends to have some sort of a *gravitational effect* on the points in high-dimensional space. By correcting the position along some dimensions, taking into effect the positions on nearby dimensions.

2.4.3 Gradient Correction

Extreme lighting conditions is an all too common phenomenon in real-life images. They have a significant impact on the decision process due to huge deviations from the population means.

Normally in the case of extreme lighting, the face has disproportionate amount of dark and light regions, which tends to have a debilitating effect on the features present in the image. Features can be made insignificant, or in some cases even be removed from the picture.

Any form of corrections to recover the features should be done at regional areas of the face rather than globally due to the very nature of features themselves. Faces in nature form patterns with a solid underlying template but varying top level intensities. In the current work gradient correction was implemented by applying histogram equalization on each of the sections, based on some statistical decisions about contrast like deviation, variance.

The image is first split into three equal horizontal sections, after which the mean, standard deviation and variance are taken for left and right parts of each section. Using the the values from both parts of the section a decision is taken to correct a part if significant deviation is detected. In the presence of significant deviation the part with lesser contrast (variance and deviation in histogram) is taken and subjected to correction like histogram equalization.

In the present work, where intensities play a very critical role in clustering, awkward corrections can result in significant deviations in high-dimensional space, that tend to weaken the clustering algorithm.

The gradient correction process implemented, had a weakening effect on clustering. This was due to the nature of correction applied - the rectangular

Figure 15: Ineffective Gradient Corrections

shape of the corrected area produced regions of sharp intensity changes resulting in highly deviated cluster points in high dimensional space. As seen in the above pictures of the mouth Section, there is a thin abrupt band induced into each image - which adversely effects the clustering algorithm. Even perfect face intensities, when subject to this gradient correction, were revoked by the clustering process. To avoid such problems gradient correction was used in a very limited fashion. Faces with extreme lighting were included into the cluster and NN training process, in order to offset the inability of the gradient correction process.

3 Clustering

3.1 Introduction

Clustering has been an integral part of human understanding. It forms an important step in classifying information, so that higher level information can be extracted and observations can be made for further understanding. It is interesting to see that there can be a parallelism drawn to the current programming paradigm of 'Object Oriented Approach' - where software is re-used to form more complex modules.

Here the main aim of the clustering process is to generate some kind of a *generic face* from the many faces, and then test each presented image-region for acceptable proximity to the generated *generic face*, so that the region can be classified. The process involves a face database, which provides the population of data, from which data points are coagulated together to belong to some cluster. Each of the generated clusters represent a type or template of face. Against these generated clusters, statistical calculations are made for the input face point, to determine the acceptability of the point as a possible face region.

To make the process understandable from the clustering perspective, each of the face regions is converted into a *point* in high-dimensional space - where the dimensionality is the number of pixels in the region. This allows for proper application of the clustering principles, as clustering is heavily dependent on the notion of a dimensional space and points in it.

One of the main goals of this work is the use of a *minimal* dataset for training. In the next few sections, we look at the various possible ways of clustering data, and the applicability of each to the data being worked upon here. Then we look at how cluster data is to be interpreted once it has been generated, and how measurement decisions are to be taken, in the *Distance Measurements* section. In the decision process the use of distance measurements is taken up as opposed to Similarity measurements, as the distance information can be used more effectively in the next stage Neural Network training and implementation. Finally we have a look at the results of clustering and some interpretations.

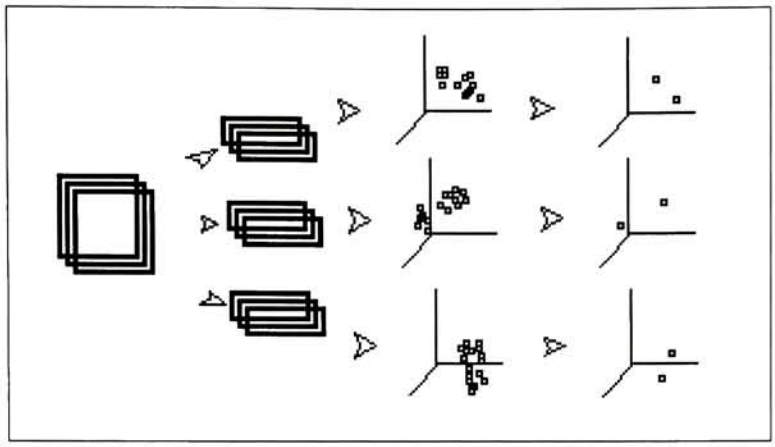

Figure 16: Clustering Process

3.2 Clustering Preview - Approaches

There are many different ways of how data can be gathered and made sense of, and most of the times a correct approach can help enormously in making sense of the data, and aid in developing better algorithms. In the present work, clustering has been used to locate acceptable face regions in high dimensional space. In this section we introduce the types of clustering available, and discuss their applicability to the present work. Some of the general categories of clustering are:

- Hierarchical techniques
  - Joining Approaches.
  - Splitting Approaches.
- Optimization-Partitioning techniques
  - Switching Approaches.
  - Adding Approaches.
- Clumping techniques

3.2.1 Hierarchical Techniques

- Joining Approaches. In these approaches, initially every element in the data is considered to be an unique cluster. Gradually clusters are

joined based on some measurement criterion. This allows the ability of information to grow naturally based on the decision function. It is not restrictive in nature, and the effectiveness of the approach depends on the accuracy of the decision algorithm.

One of the weakness of this approach is that there is no restriction on the number of clusters that can be generated. Theoretically all data can be grouped into a single cluster

- Splitting Approaches. Opposing in approach to the Joining ideology, here all data is considered to belong to a single cluster, and parts or individuals are broken off depending on how well they do not fit to that cluster. Difficulty lies in identifying, what portions/sizes of data are to be rejected. Much care has to be taken when splitting data, as they might belong to the same group, but might be viewed otherwise by the decision algorithm.

  Similar to the Joining Approach, this suffers from the question of 'How good is good enough?' Theoretically data can be broken down in each element as a single cluster.

In both the above cases, there needs to be a good cut off point for the algorithm. Whereas in the Switching Approach (in *Optimization-Partitioning Techniques*), this goes on till no more optimization is possible.

3.2.2 Optimization-Partitioning Techniques

- Switching Approaches These approaches are most useful, where there is a necessity of a fixed number of clusters. They start out by allocating all the elements into a fixed number of clusters based on some initialization process. Then the elements are moved around or switched between clusters, trying to get some optimal cluster arrangement, where the disturbance is least.

  The advantages of this approach is the development of a fixed number of clusters. Care has to be taken during the initialization of the clusters, as a good allocation of data points into initial clusters goes a long way in generating optimal cluster arrangement.

  An important method is the **K-means** method.

- Adding Approaches In this approach, each element is added to a cluster, based on the clusters already formed. A decision function would decide which cluster the presented element should join. The efficiency of the algorithm depends heavily on the order in which the data is presented. A further application of some *Switching Approach* clustering can be done to achive some kind of an optimal.

3.2.3 Clumping Techniques

In this approach, clusters are not considered disjoint groups of data, but rather have the possibility of overlapping. This need might arise, when we are classifying data into multi-tier hierarchies, and clusters can have other multi-clusters.

This approach is not a suitable option for the present work, as trying to get disjoint clusters with good inter cluster gap is much better suited. Also we might be dealing with the possibilities of including non-face clusters into the decision process, where disjoint clusters are more helpful.

3.3 Distance Measurement

Distance Measurement forms a very critical component in Clustering as it is the numerical element which should represent not only the quantitative distance, but must also come as close as possible to making the qualitative decision of whether a given point belongs to any of the given clusters.

This is a very important component both for training as well implementation of the clustering process, as the final outcome is dependent on the accuracy of both these levels.

Some of the distance metrics employed for distance measurements in clustering for the calculating distance between clusters, or between clusters and individual elements are:

Table 2: Distance Metric Equations

Metric	**Equation**
Euclidean	$d_{ij} = \{\sum_{k=1}^{p}(X_{ik} - X_{jk})^2\}^{1/2}$
Mahalanobis	$d_{ij} = (X_i - X_j)' \sum^{-1}(X_i - X_j)$
Minkowski	$d_{ij} = \{\sum_{k=1}^{p} \lvert X_{ik} - X_{jk}\rvert^r\}^{1/r}$

For this work the *Euclidean* distance metric was applied as each dimension was to have equal weight in the decision process. This was done to observe the effectiveness of pure raw unmanipulated data elements in the decision process. Other metrics ones which give increased weightage to certain dimensions - can be taken up provided there is a guarantee of the dimensions' significance in the decision process.

3.4 Implementation

In the implementation of the clustering process decisions have to be taken on how best the data is to be represented for processing. Some important decisions have to be made regarding the clustering process also.

Some of the crucial factors of the clustering process are:

- Number of clusters to be generated or used
- Type of clustering Algorithm
- Distance metric to be used for decision process

The number of clusters is an important criteria for both training and implementation. Too many clusters can result in heavy fragmentation of the data with too many possibilities causing problems in implementation and testing. Too many clusters also reduces the effectiveness of some cluster groups to produce good generalization. Too few clusters can result in joining of clusters which essentially should belong to different clusters, resulting in loss of feature information. This generally results in incorrect decisions due to the inclusion of non-face regions between clusters - areas which are crucial for getting accurate decision surfaces. So care must be taken to carefully select the number of clusters.

Any previous intuitive knowledge of how many classifications of the data might result in is extremely helpful. In [22], a fixed number of clusters is experimented upon - A constant value of six is fixed by empirical deduction - an increase or decrease of number of clusters results in an increase of the *energy* of the clusters. Also in [22] the decision surface is created with the help of face as well as non-face clusters. In the present work no such restrictions are imposed and the clustering algorithm is given the freedom to make clusters as long as there is a conformance to some threshold. No fixed number was selected as we have no definite clue to the general *types* of faces - confining to a fixed cluster number could remove a *type of face* cluster, making it difficult for the decision process.

One of the other important criteria for clustering is the type of algorithm used for clustering data. In this work the *Joining Approach* in the *Hierarchical techniques* was used. This is because we cannot restrict ourselves to a fixed cluster size, we have to use the hierarchical approaches. Partitioning techniques require a fixed number of clusters and hence are not employed. The Joining approach is selected because it is more efficient to form well-defined clusters from the data - since every face is its own type. In Splitting approaches, there is more chance that a wrong turn could be taken in the splitting process, and the clusters could be split inappropriately - this is due to the ambiguity of whether the mean of the cluster should be considered with or without the point included in it, and also it is computationally expensive to decide which point is to be branched off. In Joining approaches we start off from the best possible solution of *all faces being clusters themselves* and going up towards some generalization.

One of the main goals of this work is to use a minimal amount of dataset for training. This is to observe the robustness of the procedure. Other works on similar areas in [2] [1] [22] use a dataset of about 1000 faces gathered

from local databases, NIST mug-shot databases, etc. In this work a database of only 300 images are being used for both clustering and NN training. The same dataset is used for both the processes.

Once the image dataset for the clustering process is ready, the image files are read, and converted into cluster elements. Each cluster element has $M \times N$ variables. As the entire image comes from a single source, the variables are already *standardized.* Each variable is a value signifying intensity of that particular pixel. Thus with $M \times N$ variables, the clustering is to be done in a dimensional space of $M \times N$ dimensions where each dimension ranges from the value of 0 to 255.

Each image here is a *Section* or strip of the original face. Each face block considered for clustering is of dimension 30×30 pixels. So each image is a strip of 30×10 pixels, or 300 variables in each of the cluster elements.

The images are read, and a cluster point for each of the images is created initially. Then the joining algorithms joins the two closest points based on their Euclidean distances. Euclidean distance was chosen as an alternative compared to others because all pixels are of equal importance in the decision process, and giving a probabilistic advantage to some dimensions or variables could result in a system which could incorrectly classify faces that do not conform to the preference. No special weightage is given to specific variables in the cluster element, because the features are not restricted to any one dimension, but along a group of dimensions and these groups of variables are not the same in all the elements. It would be better to observe the effect of just having equally weighted variables, as they are of the same type and are standardized. Two clusters are joined together if the distance between both their means is less than a fixed threshold. Elements of the clustering process keep on clustering together, till no more clusters are nearer than the threshold. As a result of this process, many clusters are formed which represent various types of faces. Given below are the tables for the distribution of faces within clusters in the three sections of eyes, nose and mouth.

Table 3: Face Cluster Distributions

Section	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Eyes	161	15	10	8	6	6	3	3	3	3
Nose	154	30	16	12	3	3	3	2	2	2
Mouth	159	18	16	14	7	3	3	2	2	2

As seen in the tables above a disproportionate number of faces belong to a single cluster. the rest of the clusters form unique groups like people with mustaches etc. Also formed are clusters of just one face each - they vary so greatly from the others that they cannot be joined into other clusters.

The shape of the clusters in high-dimensional space is an important criterion in understanding the organization of the data, and in the effective interpretation of it. The shape of the cluster is generally determined by the distance metric involved in the process. It also depends upon the decision process which utilizes the generated distance metric to say whether a candidate belongs to a cluster or not. In the present case the distance metric is the Euclidean metric, and this gives simple geometric distances, and since we have a simple threshold as a criterion for clustering, the resultant clusters are generally spheres with the following equation:

$$(X_0^2 + X_1^2 + X_2^2 + \ldots + X_n^2) = (ClusterThreshold)^2$$
$$\text{Where } n = \text{Number of variables (300)}$$

Now since the face decision surface does not come under this equation, we employ the help Neural Networks to strengthen the decision surfaces around the clusters. Generating a decision surface based solely on clustering is possible, if one can gather the entire possible dataset for faces. Since this is a very huge dataset to consider and process, Neural Networks are employed to carve out the decision surface around the formed clusters.

4 Neural Network Training

4.1 Introduction

Neural Networks are widely used by variety of scientists for various reasons. All of which stem from the interest in understanding how the neural structure works and how they organize themselves to form such a powerful entity.

What makes a Neural Network a choice for computing something? Well, there are still problems, which are very inefficient using the normal *Von Neumann* machine approach (sequential logic). Areas like massive parallelism, Fault Tolerance, Adaptive Circumstances etc.. are areas where plain simple algorithmic approach would be highly wasteful. NN offers a significantly better package due to its inherent nature and operation.

4.2 Neural Network Architecture

A Neural Network is characterized by certain features:

- Connectionist architecture
- Training algorithm
- Activation Function

Using these unique features a Neural Network is able to adjust itself to the requirements of the training algorithm, to produce the required results. It is in short a malleable function, where the code inside the function is constantly changed so that desired outputs are got.

Neural Networks can in general learn a 'Basic rule' and fluctuations in it. They however fail to learn functions like generating random numbers, decrypting a good encryption algorithm etc.... They can do these, but at the cost of 'memorizing' the entire set.

How do Neural Networks work? Neural Nets are layered structures, where one layer acts as the input and another layer acts as an output. Inbetween these peripheral layers can be any number of layers. These layers are called as 'Hidden Layers'. Each layer contains a number of 'neurons'. Neurons in one layer are connected to the layers on either side. The type of architecture used throughout the experimentation and in the implementation is shown below:

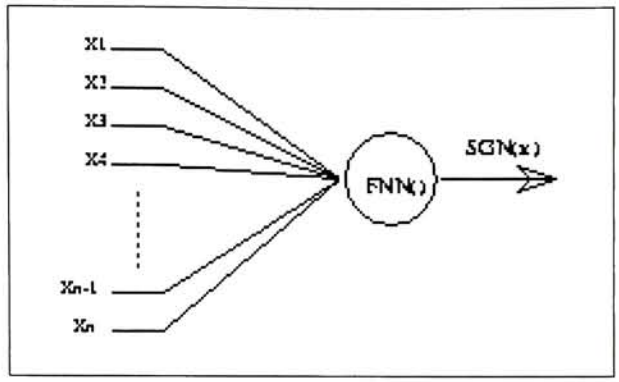

Figure 17: Neuron

Each neuron takes in the inputs X_1, X_2, X_3 $\ldots X_n$ and generates an output using the output function:

$$FNN() = SGN(\sum_{i=1}^{n} X_i w_i)$$

$$SGN(x) = \frac{2}{1+e^{-x}} - 1$$

(Bipolar Sigmoidal)

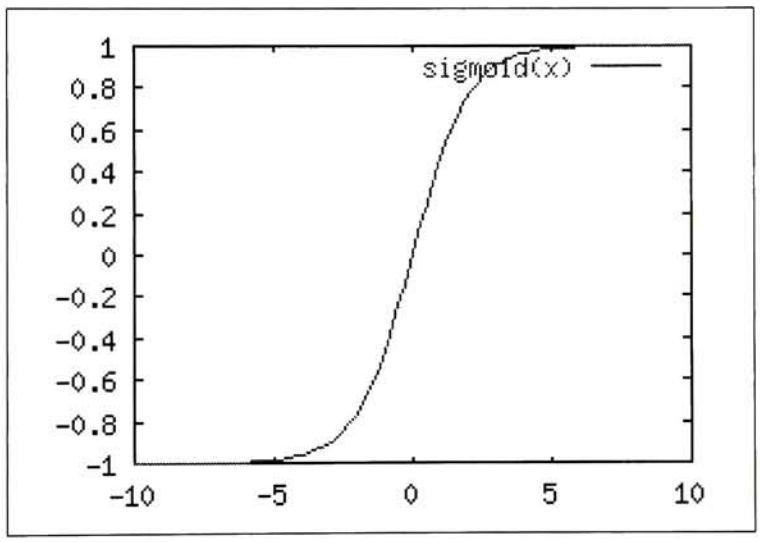

Figure 18: Sigmoidal Function

The neuron takes the input and responds appropriately to the sum of the product of the inputs and the weights. Thus the output can be restrictive(-1) or constructive(+1). This applies to all the neurons in the network, and thus the network corrects itself using the BackPropagation Algorithm.

The **BackPropagation** algorithm is the way a network corrects itself, such that the error is minimized. Weights are changed in a stepwise fashion, where first the output of the entire network is got, and then it is checked with the desired output to find the error. The weights between the first - second, and second - third layers are adjusted to decrease the final error. This is the *Back Propagation* of Error. Thus, using this method the internal weights are adjusted to reduce the final error. Various factors are influential in teaching the Network using the training algorithms, which are:

- Number of Hidden Layers
- Number of Nodes in each Hidden Layer
- Learning Rate
- Activation Function and Learning Algorithm
- Initial Values

4.3 Implementation

In the present work, a single hidden-layer Neural Network is implemented, to refine the decision surface around the clusters generated. The NN has 300 input nodes - to correspond to each of the variables in the cluster element, 100 hidden nodes, and 1 output node. The 300 input nodes correspond to the variables of the cluster element, and have their input ranges between -1.0 and 1.0 :- the intensity value is divided by the maximum possible intensity value (256 for a 8-bit grayscale image) to give a normalized fraction value, which is then subtracted from the normalized position of the largest cluster mean - giving us the position of the element relative to the cluster.

The connections which connect all the nodes together run from Layer1 to Layer2, and from Layer2 to Layer3. There are no connections from Layer1 to Layer3. Totally there are 30100 connections in the NeuralNet ($300 \times 100 + 100 \times 1$). The connections initially are randomly initialized to values between -0.5 and 0.5 [15].

A learning rate of 0.5 was implemented. This was fixed empirically, as too high a learning rate was having trouble reaching the global minimum for some face sections, which belonged to different smaller clusters. The activation function is the Bi-polar sigmoidal function which generates a value in the range of -1.0 to 1.0. The learning algorithm for the NN is the classic Backpropagation algorithm.

Similar to the clustering process where facial image-sections are transformed into cluster points, here also image-sections are taken in native image format and transformed into cluster points. But the difference here is that the training data not only includes the face-sections but also includes confusing non-face sections. This is done so that the decision surface can be constructed accurately. The dataset for the training consists of around 310 images containing face and non-face sections for each of the three regions of eyes, nose and mouth.

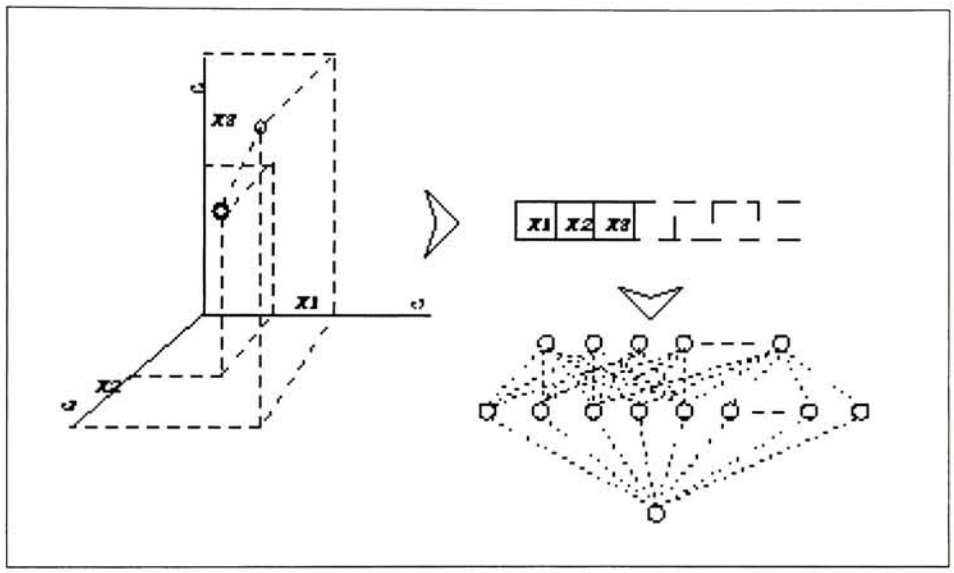

Figure 19: NN Data Gathering Process

The training process takes as input the training dataset of images and the cluster information for that particular section (eye/nose/mouth). The NN's are initialized as mentioned previously. Then the clusters information is read and the largest cluster is made to be the anchor of the NN training, from which the distances of the input training points will be measured. After

that, the training dataset containing the training images, and the required output is read, and their positions made relative to the main anchor cluster. After this the training process begins. The desired output for face sections is -1.0, and the desired output for non-face sections is 1.0. The error tolerance for completion is ± 0.1. To ensure greater accuracy of acceptance the NN's initial values are set so that the output of the NN tends heavily towards 0.1.

The number of iterations taken to train the Neural Network is in increasing order for the eyes, nose and mouth respectively. Most of the eye patterns were learnt quite quickly with a huge number of iterations being dedicated for eyes with different eye-brow orientations, and those with eye-lids closed etc. On the whole the eye-section was learnt most quickly compared to the other two sections of nose and mouth. The nose section training was expected to be easy because of the belief that that a lesser number of moving parts meant an easier target to be learnt and generalized. But the problem was that it had very few distinct high contrast features in it, and the only significant features were the tip of the nose and the shadow it formed the rest of the area was smooth without any abrupt features. This became especially prominent in low resolution images, where only a dark patch at the bottom of the section signified a nose. Such images were difficult to learn. Of the three sections the most difficult section was the mouth. This was due to the very physical nature of the mouth and the many possible deviations it could form, which made it the most difficult section to be learnt. There was no particular type of mouth section that was difficult to learn.

Once the input of the cluster file *.cls*, training images *.bin*, and the desired output file *.out* are input, an output file is generated for that particular section *.nn*. The Neural Network file generated for each of the sections is then sent to the testing stage.

5 Face Detection

5.1 Introduction

This section deals with the actual implementation of the face detection part where the clustering information and the Neural Network information is used to detect the presence of face patterns. We have three parallel searches going on for each of the eyes, nose and mouth sections. This parallel execution is advantageous for implementation purposes because it can take efficient use of the present hardware and network systems to finish each of the searches independent of others. The results from all the three sections are then joined together to generate the final face boxes, which is shown on the input image in the form of boxes.

The entire process of detection acts in two stages. In the first stage of *Sections Identification* the main program of detection is run for each of the three sections of eyes, nose and mouth over the entire image. This produces a list of possible regions for the given image. Then in the second stage of *Sections Collection* the list of regions is taken up and an analyzer groups ordered eyes, nose and mouth sections to form final face windows.

5.2 Details of working

5.2.1 Sections Identification

The process of detection starts off by the loading of the image file in which faces are to be detected into memory. The input image is in the **PGM** format. Then the clusters for each particular section (eyes/nose/mouth) are loaded. After that the third input the Neural Networks and the anchor cluster for each of the sections are loaded.

Once the mechanisms for detection have been loaded the detection of faces begins. This is done by moving a window over the entire image. The dimensions of the window being moved varies, but ultimately the contents of the window is resized to a 30×30 matrix. Smaller sized windows are enlarged and larger ones reduced to conform to the 30×30 dimension. The 30×30 image is then split into three *sections* of 30×10 pixels - one each for the eyes, nose and mouth. Then the next batch of processes are done on each of the sections individually.

Once the required dimension image section is got (30×10), it is subject to

the image processing algorithms, which tend to generalize the face - Blurring is done first by using a simple low-pass averaging filter.

Table 4: Blurring filter

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

A Gaussian filter was not used here because of the insufficient generalization it was able to effect to the central pixel. After blurring the image sections, some variance calculations are made on the processed section. The reason this is done is to eliminate sections with absolutely no kind of features or intensity variations in them - like background.

The blurred image sections, are then histogram equalized. As described in the previous section, it adjusts the histogram so that there is an uniform distribution of intensities in the entire section. The idea of splitting the face into three horizontal sections helps over here as it isolates each from the effects of others. The eyes are generally more darker than the nose section etc. Histogram Equalization can also have a very potent effect on the input sections It can change a normal background image into something with eyes, nose or mouth.

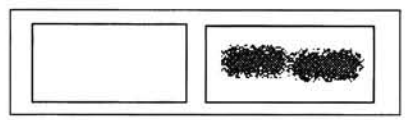

Figure 20: Effect of Histogram Equalization

So care must be taken to check for appreciable variance in intensity before attaching any significance to the image generated by the histogram equalized image. The contrast enhanced histogram equalized section is then used in conjunction with the clusters and the Neural Networks.

When the clusters are first loaded into memory, only the top 50 clusters are loaded because these represent almost all the main groups of faces. The rest are individual single face clusters. Next the position of the histogram equalized image section is recalculated relative to each of the clusters generated. If the distance between the cluster mean and the image section is

less than a certain threshold the image is considered to be a possible face-candidate either for eyes, nose or mouth. If it is considered as a possible candidate, the output of the respective Neural Network is observed. If the output is within acceptable range then that section is considered to belong to the eye, nose or mouth section. Some other statistical calculations are made on the image section before making a final decision on it. Apart from the mean, variance, and other such measurements, a measure of the symmetry of the sections an important characteristic in faces, is taken. The check for symmetry is taken, which is a simple calculation of the difference of intensities about the vertical half of the section. Lower this count, better the symmetry.

So with the values of Symmetry Count, Variance, Relative distance from nearest cluster and the Neural Network Output, a decision is made as to whether that image strip belongs to the respective *Section* type (eyes/nose/face) or not.

The eye section is the easiest one to detect, due to its prominent dark patches, with little interference from other sources. In testing almost all the cases of eye presence is easily detected with very accurate values. The nose Section is the one least detected even when compared to the mouth. The rate of detection was almost as low as 5%. This is due to the inherent nature of that particular section - with no abrupt features it is a smooth region with only a ridge like nose. The only intensities which shows up the nose are the curvature of the nasal ridge and the tip of the nose. In low resolution images these two characteristics completely disappear thereby making its detection impossible. Sometimes even that dark region disappears into the top of the mouth. The mouth Section poses quite a challenge due to the many possible variations that can be generated. By the time the Neural Network is trained, it generally accepts almost any region with a dark patch or line in the center. Due to this, a huge number of mouth Sections are identified in a given image. Sometimes each of the eye sockets is identified as a mouth Section, and then later rejected due to the detection of a very accurate eye region.

For all the three Section types (eyes/nose/mouth), the list of the regions selected under them are output to their respective files i.e., if the input file is *ImageFile.pgm*, then three list files of *ImageFile.pgm.eye*, *ImageFile.pgm.nos*, *ImageFile.pgm.mou* are generated. From here the image file and the three Section list files are sent to another program which joins these three sections to make up the final face boxes.

5.2.2 Sections Collection

Here face regions of eyes, nose and mouth which were detected are collected to detect the presence or absence of faces. Not all the listed section regions are ones belonging to real faces many of them belong to some dark patches in the image, shadows etc. randomly occurring in the image. A face is defined as a region where there are Eye, Nose and Mouth Sections in proximity in the mentioned order vertically down. Sometimes it is possible that a particular section is not identified - in those cases a face is selected based upon the spatial positioning of the other sections. The success of the face is highly dependent on the detection of the eyes. One of the observations of the process is that the Nose section is identified very infrequently, which leaves the entire identification process dependent on the position of the Eye and Nose sections. The permissible horizontal deviation for the sections is 1/6 of the window width. The permissible vertical deviation is exactly equal to or less than height of a Section.

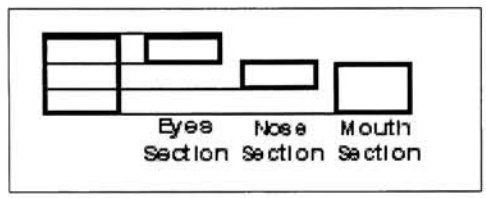

Figure 21: Permissible Vertical Ranges for Sections

Another problem to be dealt is the presence of multiple face boxes for the same face. For this situation, the union of the face boxes is taken as the final face box. The final face box is indicated by the presence of a distinguishable box draw around the area of the face, in the output image file.

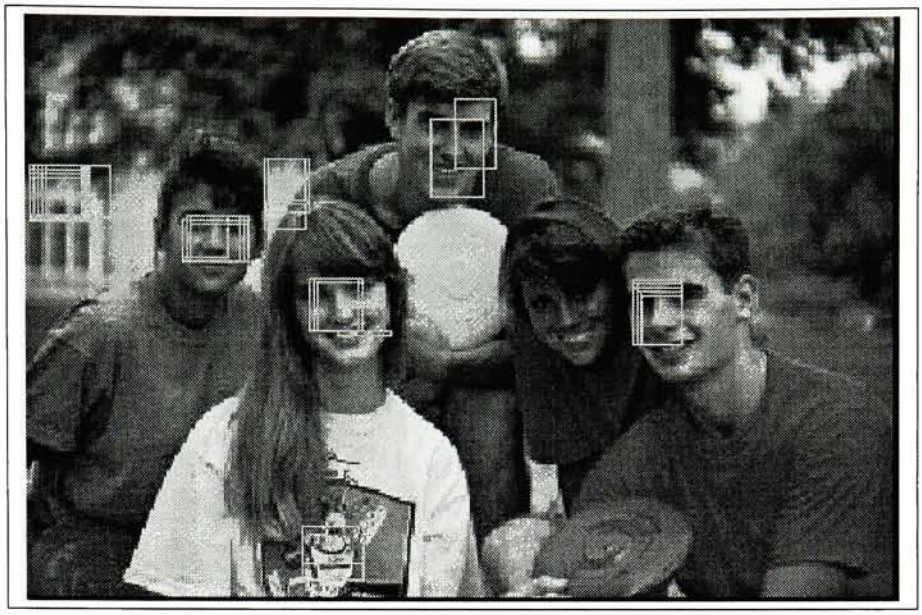

Figure 22: Multiple Face Boxes

6 Conclusion

6.1 Conclusive Remarks

As observed from the output images, the detection rate of the program is about 80% and above with a faulty acceptance rate less than 20%. Of the many weaknesses observed in the detection process, the one which had a direct impact on the results was the detection rate of the nose section. The low detection rate was not the result of poor training, but rather due to insufficiency of features and intensity variations - especially in low resolution or low contrast images where the nose was nothing but a dark patch at the bottom of the window. The effect of low identification rates resulted in the weakness of the chain of the three Sections - eyes, nose and mouth. Most of the times its absence would make the decision process depend solely on the correct positioning of the eyes and the mouth section.

One of the main goals of this work is the usage of minimal amount of data for training purposes. In the detection process the Section which had the most effective and accurate detection rate was the eyes Section. Depending on the contrast and quality of the image, the nose Section got identified, but generally the detection rates were very low. The mouth Section, due to the many possible variations of intensities, was infrequently identified.

The Clustering and Neural Network decisions go hand in hand in detecting faces. The Clustering decisions helped in localizing the decision volume while the Neural Network helped in correctly defining the decision surface within the volumes. Without the Clustering process the Neural Network would have a huge search area to classify, resulting in less accurate results. Without the Neural Network, the Clusters would have to make the decision surface based on extensive and effective training data and measurement metrics. Since this is not feasible errors appear in the Cluster decision process.

The influence of the Clustering process and Neural Network process can be shown with the help of two images, with results taken with Clustering Only, Neural Network Only, and with both of them.

Another entity which had an impact on the decision process was the position and size of the scanning window. The detection rate increased sharply when every possible window size is taken and scanned at every pixel of the entire image. This was because the Clustering and NN decision processes would accept a window at one position and completely reject it when analyzed with a very small shift. Part of the problem can be also be put on the

Figure 23: Image 1 - NN Vs Clustering

Table 5: Image 1 Results

Type Identified	NN Only	Cluster Only	Both
Person Faces	5	2	5
Non-Person Faces	1	1	1
Incorrect	9	7	1

Figure 24: Image 2 NN Vs Clustering

Table 6: Image 2 Results

Type Identified	NN Only	Cluster Only	Both
Person Faces	6	4	5
Non-Person Faces	–	–	–
Incorrect	8	9	2

high number of pixels in each window(900) where every possible shade and obstruction had a decent impact on the intensity contour and the decision process. In this implementation the window has been moved over images every alternate pixel on both axes and the window was resized by 3 pixels on every successive scan.

Possible improvements and corrective measure are presented in the next section.

6.2 Future Work and Improvements

Though the detection process works fine it is still very sensitive to many factors. Some of the things that can be improved upon are the speed and robustness. Speed can be easily improved due to the modular nature of the detection process - separate handling of the three sections by three different machines cuts the time required by a third. One of the many practical weaknesses is the speed of detection. Each image requires about 5 minutes for the results to come out. The bulk of the time is spent in moving the search window about the image rather than focusing on hot-spot regions. This could be helped with the use of color information to identify hot areas and quickly detect faces.

There is a definite trade-off between speed and accuracy here an eye section may not be identified if it is not correctly positioned in the window, and the processing could take unacceptable amount of time if every possible window position is searched. The robustness of the process can be improved in many ways, including better image processing of the input image. Things like gradient correction, shadow correction or elimination, better generalization techniques etc. can improve the generic quality of the input data. One of the changes which can have an impact is a smaller standard window size. Due to the huge size of the present window (30×30), much noise is being sent over to the decision process without getting generalized.

As discussed in the previous section, one of the important weaknesses was the ineffectiveness of the nose section. Due to this there arises a structural weakness to the three sections which *have* to be present in a particular order, to detect a face. A solution to this problem is to have only *two* sections -one for the eyes and one for both the nose and the mouth - this makes the process robust as the two sections are generally very easy and accurate to detect. Also an important plus, is that the processing power required is cut down by a third.

Basically generalization is a good thing in Face Detection as it is an attempt to detect the template, rather than distinguish between products of the template (Face Recognition). Since the ultimate use of work such as this is to have a real-time and foolproof working unit, a lot of work needs to be done in order to achieve this goal.

7 Appendix

7.1 Formats

7.1.1 PGM

P2 #Comments #Comments <Width> <space><Height> <Maximum Gray Value> <data1><space><data2><space>.... <dataX> <data...> <data...>
- P2: Ascii Based, P5: Binary Based - All numbers represented in Ascii Decimal for P2 format - No line should be more than 70 characters

7.1.2 CLS

<Number of dimensions>
<Width>
<Height>
{ <Cluster Number>
<Number of faces = nf>
<Cluster Centroid Data...>
<Face 1 Data>
<Face 2 Data>
...
<Face nf Data>
}
...
...
...
{ <Cluster Number>
<Number of faces = nf>
<Cluster Centroid Data...>
<Face 1 Data>
<Face 2 Data>
...
<Face nf Data>
}

- Has repetitive structures of Cluster Data
- All Face Data is in Binary format
- Clusters are read till no more data is available

7.1.3 BIN

<Number of dimensions> <Width> <Height> {Face Data...} {Face Data...} {Face Data...} ... ... {Face Data...}
- Face Data in binary format - Repetitive structures of Face Data - Size of Face Data determined by *Number of Dimensions*

7.1.4 NN

<No. of Nodes in Layer 1 = R1> <No. of Nodes in Layer 2 = R2> <No. of Nodes in Layer 3 = R3> ——————————— ——————————— {Neural Network Data ... $(R1 * R2) + (R2 * R3)$} {Anchor Cluster Centroid Data ... $(R1)$}
- Neural Network Data and Anchor Cluster Data in *float* representation

7.1.5 OUT

<Number of entries> <Input Pattern Number><Space><Desired NN Output> <Input Pattern Number><Space><Desired NN Output> <Input Pattern Number><Space><Desired NN Output> ... ... <Input Pattern Number><Space><Desired NN Output>
- All Data is in Ascii Format

References

[1] Henry A.Rowley, Shumeet Baluja, and Takeo Kanade. Rotation invariant neural network-based face detection. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.

[2] Henry A.Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *Pattern Analysis and Machine Intelligence*, January 1998.

[3] Shumeet Baluja. Probabilistic modeling for face orientation discrimination: Learning from labeled and unlabeled data. Technical report, Justsystem Pittsburgh Research Center, School of Computer Science, Carnegie Mellon University.

[4] David Beymer and Tomaso Poggio. Face recognition from one example view. *C.B.C.L.*, (Paper No.121), September 1995.

[5] B.S.Grewal. *Higher Engineering Mathematics.*

[6] Gilles Burel and Dominique Carel. Detection and localization of faces on digital images. *Pattern Recgnition Letters*, Vol. 15:pp. 963–967, 1994.

[7] Rafael C.Gonzalez and Richard E.Woods. *Digital Image Processing.* Addison Wesley.

[8] Brian Everitt. *Cluster Analysis.* H.E.B.

[9] S.E. Fahlman. An emprical study of learning speed in back-propagation networks. Tech. Report CMU-CS-88-162, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, September 1998.

[10] Laurence Fausett. *Fundamentals of Neural Networks Architectures, Algorithms and Applications.* Prentice-Hall, Inc., 1994.

[11] C. Fraley and A. E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. Technical Report Rep.No. 329, Department of Statistics, University of Washington, Box 354322, Seattle, WA 98195.

[12] John A. Hartigan. *Clustering Algorithms.* John Wiley and Sons.

[13] Antonio J.Colmenarez and Thomas S.Huang. Face detection with information-based maximum discrimination.

[14] Thomas K.Leung, Michael C.Burl, and Pietro Perona. Probabilistic affine invariants for recognition. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Santa Barbara, CA*, June 1998.

[15] J.F. Kolen and J.B.Pollack. Back propagation is sensitive to initial conditions. Technical Report Tech. Report TR 90-JK-RPSIC, 1990.

[16] Shang-Hung Lin and Sun-Yuan Kung. Face recognition/detection by probabilistic decision-based neural network. *IEEE Tansactions on Neural Networks*, Vol. 8(No. 1):pp. 114–131, Jan 1997.

[17] Jacek M.Zaruda. *Introduction to Artificial Neural Systems.*

[18] Edgar Osuna, Robert Freund, and Federico Girosi. Training support vector machines: an application to face detection. Technical report, Center for Biological and Computational Learning and Operations Research center, M.I.T., Cambridge, MA, 02139.

[19] R.Brunelli and T.Poggio. Face recognition: Features versus templates. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 15(No. 10):pp. 1042–1052, Oct 1993.

[20] Benjamin S.Duran and Patrick L.Odell. *Cluster Analysis A Survey.* Springer-Verlag.

[21] Gunturi Srimanth. Genetic algorithms using neural networks.

[22] Kah Kay Sung. *Learning And Example Selection For Object And Pattern Recognition.* PhD thesis, MIT, Artificial Intelligence Laboratory and Center For Biological And Computational Learning. Cambridge, 1995.

[23] C. Wren, A. Azarbayejani, T.Darrell, and A. Pentland. Pfinder: Real time tracking of the human body. *Proc. SPIE*, Vol. 2615:pp. 89–98, 1996.

[24] Guangzheng Yang and Thomas S.Huang. Human face detection in a complex background. *Pattern Recognition*, Vol. 27(No. 1):53–63, 1994.

[25] Jie Yang and Alex Waibel. A real-time face tracker. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.

Figure 25: Image 1

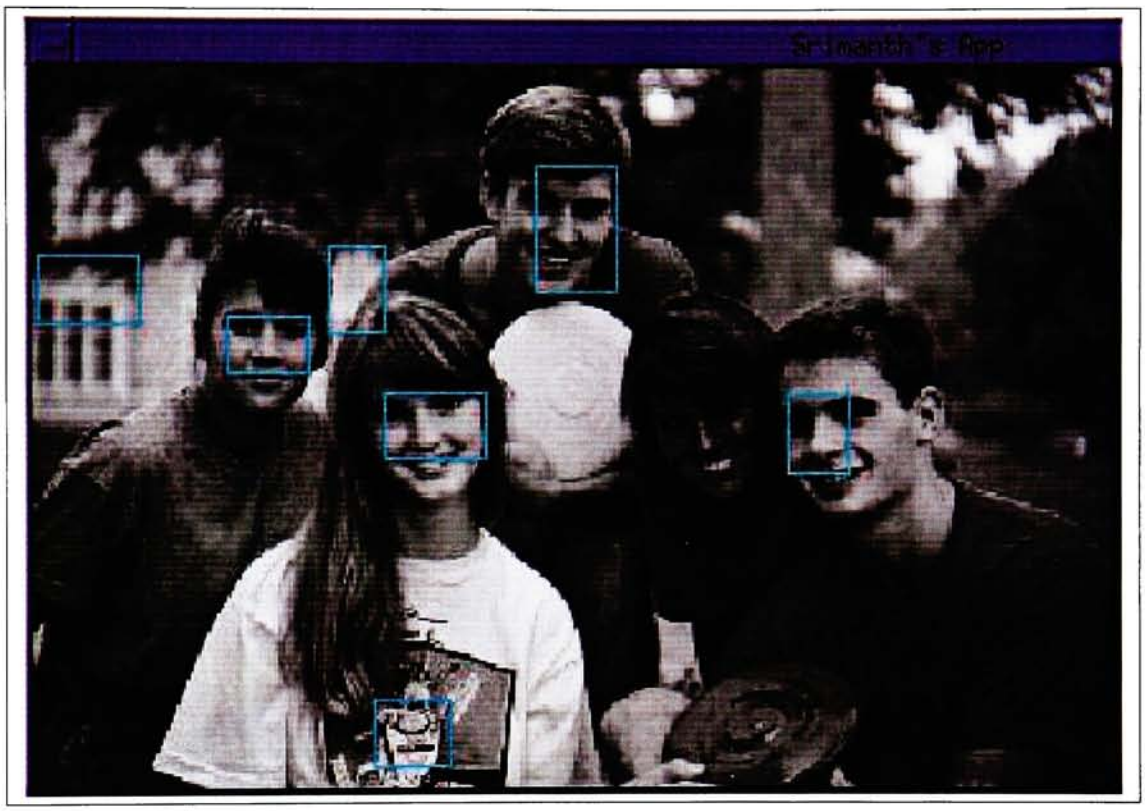

Figure 26: Image 2

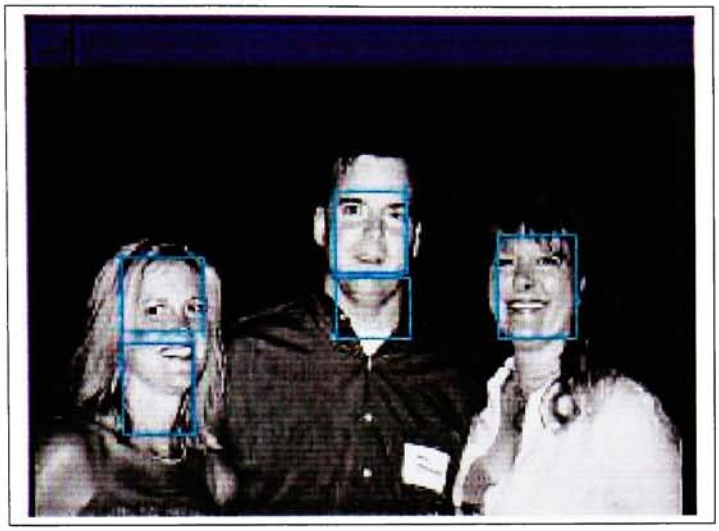

Figure 27: Image 3

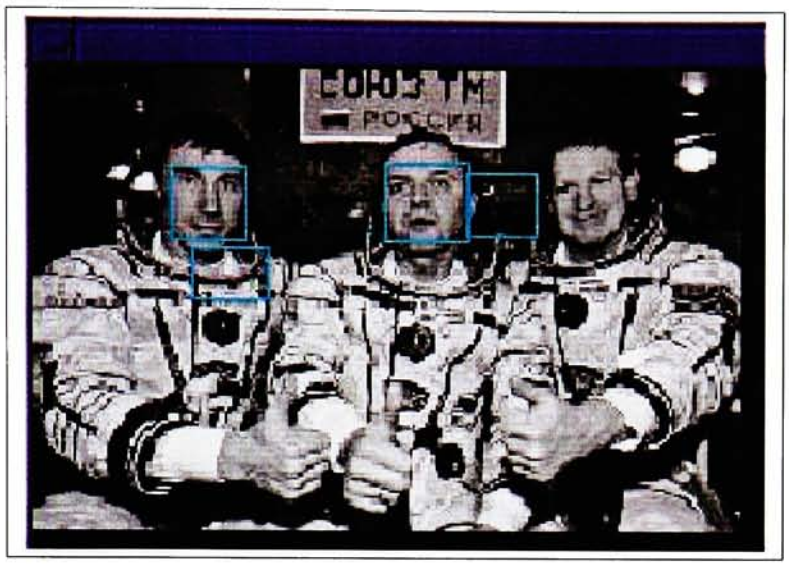

Figure 28: Image 5

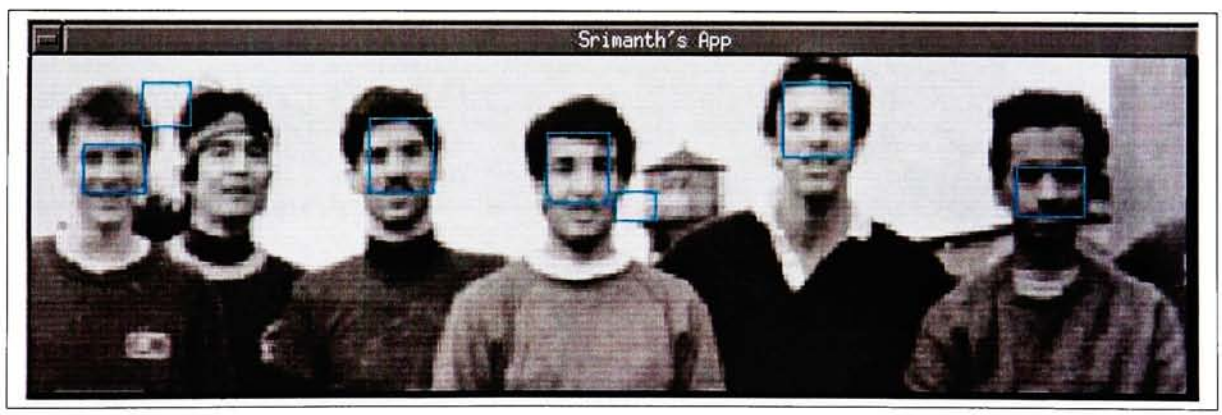

Figure 29: Image 6

Figure 30: Image 7

Figure 31: Image 8