1. **INTRODUCTION**

**1.1 Image Processing**

In electrical and computer engineering, image processing is a subset of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or, a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a 2-D signal and applying standard signal-processing techniques to it. Image processing usually refers to as digital image processing and the *acquisition* of images (producing an input image in the first place) is referred to as imaging.

Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modelled in the form of Multidimensional Systems.

The task of face analysis with whatever scope is characterized by highly complicated conditions. In the first place, the appearance of faces varies between different persons in their outer shape, the relative spatial position of facial features (eyes, brows, nose, mouth, cheeks), the shape of facial features, and the skin colour. Due to the high complexity and variance of face recognition and interpretation, the evolution even dedicated an own area of the human visual brain cortex to this task (Journal of Neuroscience, 1997, Wikipedia). Therefore, human face analysis constitutes one of the greatest challenges in the modern computer vision area.

***1.1.1 Image Processing Tasks***

**Image Acquisition**: A digital image is produced by one or several image sensors, which, besides various types of light-sensitive cameras, include range sensors, tomography devices, radar, ultra-sonic cameras, etc. Depending on the type of sensor, the resulting image data is an ordinary 2D image, a 3D volume, or an image sequence. The pixel values typically correspond to light intensity in one or several spectral bands (gray images or colour images), but can also be related to various physical measures, such as depth, absorption or reflectance of sonic or electromagnetic waves, or nuclear magnetic resonance.

**Pre-Processing**: Before a computer vision method can be applied to image data in order to extract some specific piece of information, it is usually necessary to process the data in order to assure that it satisfies certain assumptions implied by the method. Typical examples of such pre-processing techniques are

* Re-sampling in order to assure that the image coordinate system is correct.
* Noise reduction in order to assure that sensor noise does not introduce false information.
* Contrast enhancement to assure that relevant information can be detected.
* Scale-space representation to enhance image structures at locally appropriate scales.

**Feature Extraction**: Image features at various levels of complexity are extracted from the image data. Typical examples of such features are

* Lines, edges and ridges.
* Localized interest points such as corners, blobs or points.
* More complex features may be related to texture, shape or motion.

**Detection/Segmentation**: At some point in the processing a decision is made about which image points or regions of the image are relevant for further processing. Examples are

* Selection of a specific set of interest points
* Segmentation of one or multiple regions which contain a specific object of interest.

**High-Level Processing**: At this step the input is typically a small set of data, for example a set of points or an image region which is assumed to contain a specific object. The remaining processing deals with, for example:

* Verification that the data satisfy model-based and application specific assumptions.
* Estimation of application specific parameters, such as object pose or object size.
* Image recognition: classifying a detected object into different categories.
* Image registration: comparing and combining two different views of the same object.

***1.1.2 Challenges for Face Detection and Feature Extraction***

Some challenges for performing successful face detection and feature extraction using image processing are described below.

**Pose:** The images of a face vary due to the relative camera-face pose (frontal, 45 degree, profile, upside down), and some facial features such as an eye or the nose may become partially or wholly occluded.

**Presence or Absence of Structural Components:** Facial features such as beards, mustaches, and glasses may or may not be present and there is a great deal of variability among these components including shape, color, and size.

**Facial Expression:** The appearance of faces is directly affected by a person’s facial expression making it difficult for face detection and by extrapolation background structures may appear as faces.

**Occlusion:** Faces may be partially occluded by other objects. In an image with a group of people, some faces may partially occlude other faces.

**Image Orientation:** Face images directly vary for different rotations about the camera’s optical axis.

**Imaging Conditions:** When the image is formed, factors such as lighting (spectra, source distribution and intensity) and camera characteristics (sensor response, lenses) affect the appearance of a face.

**1.2 Neural Network**

***1.2.1 Brief History of Neural Networks***

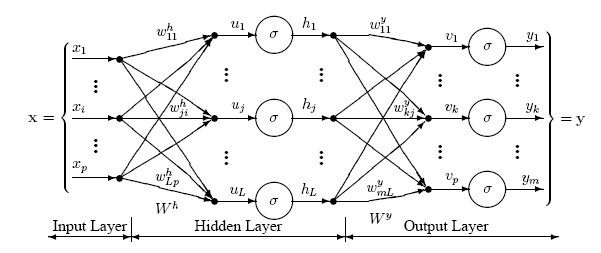
***Neural networks*** are predictive models loosely based on the action of biological neurons.

The original “Perceptron” model was developed by Frank Rosenblatt in 1958. Rosenblatt’s model consisted of three layers, (1) a “retina” that distributed inputs to the second layer, (2) “association units” that combine the inputs with weights and trigger a threshold step function which feeds to the output layer, (3) the output layer which combines the values. Unfortunately, the use of a step function in the neurons made the perceptions difficult or impossible to train. A critical analysis of perceptrons published in 1969 by Marvin Minsky and Seymore Papert pointed out a number of critical weaknesses of perceptrons, and, for a period of time, interest in perceptrons waned.

Interest in neural networks was revived in 1986 when David Rumelhart, Geoffrey Hinton and Ronald Williams published “Learning Internal Representations by Error Propagation”. They proposed a multilayer neural network with nonlinear but differentiable transfer functions that avoided the pitfalls of the original perceptron’s step functions. They also provided a reasonably effective training algorithm for neural networks.

***1.2.2 The Multilayer Perceptron Model***

Multilayer perceptrons (MLPs) are layered feed forward networks typically trained with static back propagation by specifying the number of hidden layers. These networks have found their way into countless applications requiring static pattern classification. Their main advantages are that they are easy to use, and that they can approximate any input/output map. The key disadvantages are that they train slowly, and require lots of training data (typically three times more training samples than network weights).



“Figure1. Perceptron network with three layers”

The network architecture shown in “Figure 1” has an **input layer** with three neurons, one **hidden layer** (in the middle) with three neurons and an **output layer** with three neurons. There is one neuron in the input layer for each predictor variable. In the case of categorical variables, *N*-1 neurons are used to represent the *N* categories of the variable.

**Input Layer** — A vector of predictor variable values (*x1...xp*) is presented to the input layer. The input layer (or processing before the input layer) standardizes these values so that the range of each variable is -1 to 1. The input layer distributes the values to each of the neurons in the hidden layer. In addition to the predictor variables, there is a constant input of 1.0, called the *bias* that is fed to each of the hidden layers; the bias is multiplied by a weight and added to the sum going into the neuron.

**Hidden Layer** — arriving at a neuron in the hidden layer, the value from each input neuron is multiplied by a weight (*wji*), and the resulting weighted values are added together producing a combined value *uj*. The weighted sum (*uj*) is fed into a transfer function, σ, which outputs a value *hj*. The outputs from the hidden layer are distributed to the output layer.

**Output Layer** — Arriving at a neuron in the output layer, the value from each hidden layer neuron is multiplied by a weight (*wkj*), and the resulting weighted values are added together producing a combined value *vj*. The weighted sum (*vj*) is fed into a transfer function, σ, which outputs a value *yk*. The *y* values are the outputs of the network.

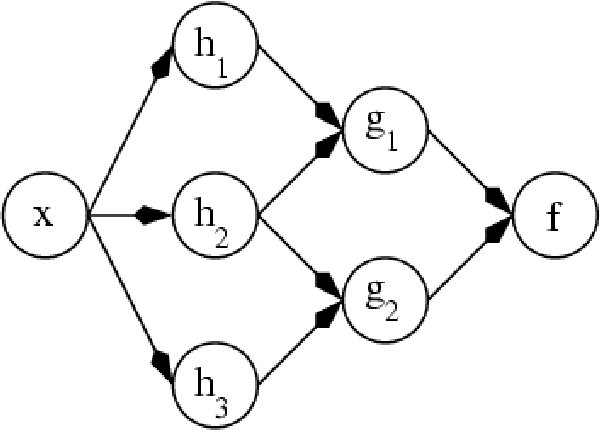
If a regression analysis is being performed with a continuous target variable, then there is a single neuron in the output layer, and it generates a single y value. For classification problems with categorical target variables, there are *N* neurons in the output layer producing *N* values, one for each of the *N* categories of the target variable.

The network diagram shown in “Figure 1” is a full-connected, three layer, feed-forward, perceptron neural network. “Fully connected” means that the output from each input and hidden neuron is distributed to all of the neurons in the following layer. “Feed forward” means that the values only move from input to hidden to output layers; no values are fed back to earlier layers (a Recurrent Network allows values to be fed backward).

***1.2.3 Multilayer Perceptron Architecture***

The network diagram shown in “Figure 2” is a four layers, feed-forward, Perceptron neural network with two hidden layers. “Feed forward” means that the values only move from input to hidden to output layers; no values are fed back to earlier layers.

All neural networks have an input layer and an output layer, but the number of hidden layers may vary. When there is more than one hidden layer, the output from one hidden layer is fed into the next hidden layer and separate weights are applied to the sum going into each layer.

****

“Figure2. Perceptron network with two hidden layers and four total layers”

***1.2.4 Neural Network Training***

The goal of the training process is to find the set of weight values that will cause the output from the neural network to match the actual target values as closely as possible. There are several issues involved in designing and training a multilayer perceptron network:

* Selecting how many hidden layers to use in the network.
* Deciding how many neurons to use in each hidden layer.
* Finding a globally optimal solution that avoids local minima.
* Converging to an optimal solution in a reasonable period of time.
* Validating the neural network to test for over fitting.

**1.3 Human Computer Interaction (HCI)**

Human–computer interaction (HCI) is the study, planning and design of the interaction between people (users) and computers. It is often regarded as the intersection of computer science, behavioural sciences, design and several other fields of study. Interaction between users and computers occurs at the user interface (or simply *interface*), which includes both software and hardware; for example, characters or objects displayed by software on a personal computer's monitor, input received from users via hardware peripherals such as keyboards and mice, and other user interactions with large-scale computerized systems such as aircraft and power plants. The Association for Computing Machinery defines human-computer interaction as "a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them." An important facet of HCI is the securing of user satisfaction.

Because human-computer interaction studies a human and a machine in conjunction, it draws from supporting knowledge on both the machine and the human side. On the machine side, techniques in computer graphics, operating systems, programming languages, and development environments are relevant. On the human side, communication theory, graphic and industrial design disciplines, linguistics, social sciences, cognitive psychology, and human factors are relevant. Engineering and design methods are also relevant. Due to the multidisciplinary nature of HCI, people with different backgrounds contribute to its success. HCI is referred to as man–machine interaction (MMI) or computer–human interaction (CHI).

Attention to human-machine interaction is important, because poorly designed human-machine interfaces can lead to many unexpected problems. A classic example of this is the Three Mile Island accident where investigations concluded that the design of the human-machine interface was at least partially responsible for the disaster. Similarly, accidents in aviation have resulted from manufacturers' decisions to use non-standard flight instrument and/or throttle quadrant layouts: even though the new designs were proposed to be superior in regards to basic human-machine interaction, pilots had already ingrained the "standard" layout and thus the conceptually good idea actually had undesirable results.

**1.4 Problem Statement**

***1.4.1 Background***

The task of finding a person’s expression either real life or in a picture seems to be effortless for a human to perform. However it is far from simple for a machine of current technology to do the same. In fact, development of such a machine or system has been widely and actively studied in the fields of image processing, artificial intelligence and machine learning and can loosely be termed as image understanding. For the past few decades there have been applications constructed having machine vision and face recognition in mind. Moreover in recent years, the research activities in this area have intensified as a result of its applications being extended towards video representation and coding purposes.

The main objective of this project is to design a system that can find a person’s facial expression from given image data, extract features from the image and identify expression shown in the expressive image. This problem is commonly referred to as expression recognition or emotion detection. Regardless of the terminology, they all share the same objectives. However, note that the problem usually deals with finding the position and contour of a person’s face preliminarily. Followed by extraction of features from the face this stage represents a conversion of image data to a mathematical format that can be used by the system and finally training a neural network to identify the expression being demonstrated.

Vision and facial gestures are some of the oldest tools humans use for interaction among each other. It is therefore one of the most natural ways to interact with the computers as well. Although image processing is now good enough to allow decomposition of an image to multiple formats, emotion recognition can increase the overall efficiency of interaction and may provide everyone a more comfortable user interface. It is often trivial for humans to get the emotion of an individual and adjust their behavior accordingly. Emotion recognition will give the programmer a chance to develop an artificial intelligence that can meet the expresser’s feelings that can be used in many scenarios from computer games to virtual sales-programs. Three base emotions, smile, surprise and frown are taken into account. Various image sets that belong to these emotion groups are extracted from different people and used for training and testing. The neural network is capable of distinguishing these test samples. Neural networks are chosen for the solution because a basic formula cannot be devised for the problem. The neural networks are also quick to respond which is a requirement as the emotion should be determined almost instantly. The training takes a long time but is irrelevant as the training is mostly done off-line.

* + 1. ***Definition***

Given two frontal face images of an individual under adequate lighting conditions, the first being a non-expressive (base image) and the second showing an expression (expressive image) recognize the expression being displayed as either a “frown” “smile” or “surprise”.

* + 1. ***Aims and Objectives***
* Take user input in the form of images (“.jpg”) or video (“.avi”) using an intuitive and user friendly GUI.
* Perform face detection and background noise elimination.
* Identify Region’s of Interest (ROI) namely eyes, cheeks, nose, mouth and eyebrows.
* Extract features from regions of interest i.e. get specific points in the regions of interest.
* Convert image processing data into a useable mathematical form.
* Train neural network to recognize expressions based on data extracted from image.
  + 1. ***Scope of the Project***
* Uses “.jpg” images and “.avi” videos.
* Minimum of two user input images for expression detection.
* Requires well functioning neural network.

**1.5 Organisation of the Report**

The report is divided into six parts (1) Introduction – provides generic background information regarding the areas of interest in this project. Aims to gives a new user insight into these research areas and provide a better understanding of this project. (2) Literature Survey – An account of articles referred during the course of the project (3) Methodology and Timeline – step by step account of how the project has progressed from inception to completion. (4) Workflow and Implementation – gives details of mechanisms and algorithms used during various stages of the project and their corresponding results. (5) Conclusions and Future work. (6) References.

**1.6 Technologies Used**

* Open CV – for image processing.
* Python – for User Interface and Scripting.
* C/C++ - v.4.5.0 gcc compiler – Coding language.
* Ubuntu and Fedora Linux – Operating System.

# LITERATURE SURVEY

### 2.1 Background

This section describes the research papers from leading journals and conferences that were studied prior to and during the project. These articles have had a profound impact during various stages of development. This is to gain familiarity with the subject, give an idea of what problem statements are part of this field and basic challenges faced, current research trends and technologies commonly used along with standard procedures followed, challenges faced while working on problem statements and to develop an idea or intuition to solve problems and develop techniques in the chosen area.

Some important papers and the information they have provided are.

1. Analysis of Emotion Recognition using Facial Expressions, Speech and Multimodal Information [1]:

This paper analyzes the strengths and the limitations of systems based only on facial expressions or acoustic information. It also discusses two approaches used to fuse these two modalities: decision level and feature level integration. Using a database recorded from an actress, four emotions were classified: sadness, anger, happiness, and neutral state. By the use of markers on her face, detailed facial motions were captured with motion capture, in conjunction with simultaneous speech recordings. The results reveal that the system based on facial expression gave better performance than the system based on just acoustic information for the emotions considered. Results also show the complementarily of the two modalities and that when these two modalities are fused, the performance and the robustness of the emotion recognition system improve measurably.

1. Emotion Detection of Infants from Facial Expressions and Cries [2]:

The paper describes a system that is designed to analyze the facial image and sound of the crying infant to derive the reason why the infant is crying. The image and the sound represent the same cry event. The image processing module determines the state of certain facial features, certain combinations of which determine the reason for crying. The sound processing module analyzes the data for the fundamental frequency and the first two formants and uses k-means clustering to determine the reason of the cry. The decisions from the image and sound processing modules are then fused using a decision level fusion system. The accuracy of the fused decision is seen to be greater than that of each method individually.

1. Biometric Recognition Performing in a Bio-inspired System [3]:

In this paper, a set of biometric recognition experiments are proposed in conditions similar to real operating systems. This implies a change from the usual laboratory conditions to a more real situation where the amount of variability between training and testing samples is large. Here experiments are presented with face, hand-geometry, and signature recognition training a ‘‘universal classifier’’ able to decide if two input samples belong to the same person or not.

**2.2 Critical Information**

This is information which finds direct application in the project. It includes online tutorials and directly relevant research papers. Some of these sources provided insight to methods that were replaced by more apt and powerful (with respect to our goals) techniques. However at some point of time during the research and development cycle they have provided critical knowledge and hence can be found in this section.

##### 2.2.1 Online Tutorials.

1. **Computing constrained Delaunay triangulation in the plane by** [Samuel Peterson](http://www.geom.uiuc.edu/%7Esamuelp/welcome.html) - University of Minnesota Undergraduate:

This tutorial explains the algorithms used to generate a Delaunay Triangulation for a given set of points as well as an implementation and examples of applications of this method.

1. Algorithms for Constructing a Delaunay Triangulation – By Peter Fleischmann:

This tutorial briefly overviews Delaunay Triangulation algorithms for a given point set P without constraining boundaries. It names six different algorithms to compute Delaunay Triangulation the most commonly used being Naïve Edge Swapping and [Divide-and-Conquer](http://www.iue.tuwien.ac.at/phd/fleischmann/node45.html)

##### 2.2.2 Critical Research Papers

1. Facial Feature Detection Using Haar Classifiers: By Phillip Ian Wilson, University of Texas A&M [3].

In this paper it is proposed that the area of the image being analyzed for a facial feature needs to be regionalized to the location with the highest probability of containing the feature. By regionalizing the detection area, false positives are eliminated and the speed of detection is increased due to the reduction of the area examined. This method has been used extensively in our project to identify different parts of the face.

1. Automatic facial emotion recognition: By Micheal Baratheon, Department of Computer Science, University of Michigan [4].

This paper presents a system for recognizing emotions through facial expressions displayed in live video streams and video sequences. The system is based on the Piecewise B´ezier Volume Deformation tracker and has been extended with a Haar face detector to initially locate the human face automatically. This paper is regarded as critical because it provided us the initial impetus to pursue the path we are on. The overall framework of action described in this referred paper is similar to what we have adapted to achieve our goals.

# **Facial feature location with Delaunay triangulation/Voronoi diagram calculation [5]:**

This paper advocates that an automatic extracting algorithm is developed to locate "key points" of facial features. The Delaunay Triangulation/Voronoi Diagram technique well known in computational geometric is applied on the edge enhanced facial image. Facial features are classified and extracted in terms of various types of Delaunay triangles and the dual of a subset of the Delaunay triangles; Voronoi edges form the skeleton of facial skin. That is, facial feature's shape is described by Delaunay Triangulation/Voronoi Diagram. Furthermore, the facial features can be identified. The method succeeds in locating facial features in the facial region exactly and is insensitive to face deformation. The method is executable in a reasonably short time.

4. Face Segmentation using skin-colour map in videophone application [6]: This referred paper addresses a proposed method to automatically segment out a person’s face from a given image that consists of a head-and-shoulders view of the person and a complex background scene. The method involves a fast, reliable, and effective algorithm that exploits the spatial distribution characteristics of human skin color. A universal skin-color map is derived and used on the chrominance component of the input image to detect pixels with skin-color appearance. Then, based on the spatial distribution of the detected skin-color pixels and their corresponding luminance values, the algorithm employs a set of novel regularization processes to reinforce regions of skin-color pixels that are more likely to belong to the facial regions and eliminate those that are not. The performance of the face segmentation algorithm is illustrated by some simulation results carried out on various head-and-shoulders test images.

5. Detecting Faces in Images: A Survey [7]:

Images containing faces are essential to intelligent vision-based human computer interaction, and research efforts in face processing include face recognition, face tracking, pose estimation, and expression recognition. However, many reported methods assume that the faces in an image or an image sequence have been identified and localized. To build fully automated systems that analyze the information contained in face images, robust and efficient face detection algorithms are required. Given a single image, the goal of face detection is to identify all image regions which contain a face regardless of its three-dimensional position, orientation, and lighting conditions. Such a problem is challenging because faces are non-rigid and have a high degree of variability in size, shape, colour, and texture. Numerous techniques have been developed to detect faces in a single image, and the purpose of this paper is to categorize and evaluate these algorithms.

6. Voice Activity Detection by Lip Shape Tracking Using EBGM [8]:

This paper gave us come very important information related to lip detection which is one of the most challenging aspects of image processing encountered in this project. The system proposed in this paper extracts the lip movement of the target speaker by measuring the lip aspect ratio. An infrared camera is used to cope with the change of lighting environment. In order to extract the lip from gray scale images, Elastic Bunch Graph Matching is employed. Since our project assumes ideal frontal lighting conditions we can continue without an infrared camera. However we found that this algorithm fails for our form of image processing and provides no useable data because it is a template based algorithm and our system does not support template matching. The other significant reason we did not proceed with this method is because we were inspired by another lip detection research paper which was more suitable to our needs.

7. Detecting Eyelash and Reflection for Accurate Iris Segmentation [9]:

Accurate iris segmentation is presented in this paper, which is composed of two parts, reflection detection and eyelash detection. Eyelashes are classified into two categories, separable and multiple. An edge detector is applied to detect separable eyelashes and intensity variances are used to recognize multiple eyelashes. Reflection is also divided into two types, strong and weak. A threshold and statistical test are proposed to recognize the strong and weak reflection, respectively. We make use of a simplified version of this algorithm to locate the pupils.

8. Emotion Recognition Using Neural Networks [10]:

This is one of the most important papers we studied. It introduced us to the power of the neural network and its application specifically for recognition of emotional features from voice signals and gave us the confidence to proceed with neural network training for expression recognition on an image after it has been subject to appropriate image processing. Speech and emotion recognition improve the quality of human computer interaction and allow more easy to use interfaces for every level of user in software application. In this study, they have developed an emotion recognition neural network (ERNN) to classify the voice signals for emotion recognition.

9. Comparative Analysis of Lip Features for Person Identification [11]:

This paper provides us with the primary motivation for lip detection. It is from this paper we get the basic idea for lip detection upon which we perform our own algorithm to detect lips accurately. In this paper a comparative evaluation of different visual features from lips are presented. Geometric and appearance based features were extracted and their relevance to identifying people was studied by feature selection methods.

* 1. **Outcome of Literature Survey**

After the literature survey we have a sound knowledge of the background in image processing, artificial intelligence and human computer interaction. We came across a gamut of methods, techniques and algorithms used for image processing, feature extraction and noise cancellation. After analyzing these algorithms and identifying their constraints, accuracy, performance and limitations, we arrived at several promising possibilities for tackling the problem statement. Armed with this knowledge we were able to identify various stages of the projects and possible methods for implementation. We also understood the concepts behind different neural networks and how to select, design and parameterise one best suited to achieve our targets using Open CV along with other technologies.

.

1. **METHODOLOGY ADOPTED**

**3.1 Project Outline**

“Figure3. Timeline of the system”

***3.1.1 Work done in VII Semester.***

This section describes in detail the steps taken and decisions made in the project from its inception in July 2010 to completion in April 2011.

1. Selected Facial Expression Recognition / Facial Gesture Recognition as Major Project topic in collaboration with India Innovation, Bangalore. Read research papers in ancillary areas provided by project guide to gain some familiarity with the topic. The most important have been mentioned in the “Survey of Other Works Section”.
2. Understood that OpenCV would play a key role in the implementation of the project.

While going through background information it was observed that most researchers make use of Open CV, which is an open source collection of computer vision libraries, providing functions like pixel data, frame capturing etc. This along with encouragement by IIL coordinators and the Project Guide influenced our decision to use OpenCV as a means of implementing our CV algorithms.

1. Learnt basics of Open CV. This step marks the concrete beginning of our major project. It involved writing simple apps using OpenCV such as a limited function movie player.
2. Decided to use Delauny Triangulation as a means of mapping the image to a workable mathematical model, based on advice given by IIL and seconded by Project Guide.

Now after attaining operational capacity we began to crack on the problem statement. The first step was naturally to convert an image which a collection of pixel data into a mathematical representation whose properties we can analyze. Expertise from people with more experience in the field led us to choose the Delaunay Triangulation as a means of doing so.

1. Needed to decide how to find track able points that can be used to create the triangulated mesh. Now, we know that Delaunay triangulation for a set P of points in the plane is a triangulation DT(P) such that no point in P is inside the circumcircle of any triangle in DT(P). Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation; they tend to avoid skinny triangles.

BUT to create this triangulation or mesh we require a point set P. The next task we had was to decide how to get this set of points on the face that can be used to draw a mesh. The points should not be arbitrary but represent certain features that can be tracked if the face is distorted or trans-located. (I.e. pupil good point, but arbitrary point on cheek will not have this track ability). This problem boiled down to being able to identify various parts of the face such as eyes, nose, mouth etc. (after that fixing points becomes trivial).

1. Uncovered Haar Classifiers while searching for suitable point set generator.

Haar Classifiers form an integral part of most object recognition algorithms and it while searching for a method to identify the facial features we came across.

1. Discovered the usefulness of Haar Classifiers.

Although many different algorithms exist to perform face detection, each has its own weaknesses and strengths. Some use flesh tones, some use contours, and other are even more complex involving templates, neural networks, or filters. These algorithms suffer from the same problem; they are computationally expensive. Haar Cascading Classifiers act by approximating features on the face, for example once face is detected the eyes will generally be in the top third of the face, in this region we perform a contrast, colour or edge detection algorithm which can be used to pinpoint the position of the eyes. After obtaining the position of the eyes we can then estimate position of nose being between the 2 eyes so on and so forth.

1. Decided to incorporate Haar Classifiers to identify features of the face. In this way we have been able to identify mouth, nose and eyes so far.
2. Generated Point set using features identified using the Haar Cascading Classifiers method. This step is trivial after locating position of other features like top, bottom, left most and rightmost point in the rectangle surrounding the mouth were chosen.
3. Constructed Delaunay Triangular mesh by piping information from Haar classified image to Delaunay triangulation constructor program.
4. Found almost impossible to quantize distortions in mesh for different frames accurately- began searching for a new method to identify emotions.

We tried many approaches to quantize the distortion between meshes of different images, but were unable to come up with any suitable subtraction function. We then realized that such an approach might not be feasible or even possible and an alternative method of making sense (recognizing expression) from the triangulated meshes would have to be found.

1. Decided to try using Machine Learning techniques for emotion recognition based on inputs from several referred papers.
2. Identified three probable machine learning techniques that can be used and compared for correctness and efficiency, they are;

a) K-Means Clustering. Use centroids of the triangles formed after triangulation and perform k-means clustering and compare against a standard image. The number of points for consideration in different images will be the same because the number of triangles remains a constant for a given face, however distortion in the shapes of the triangles between different images will result in shifted k-means; hence a shift in the k-means may be able to identify a particular expression.

b) Linear Transformations on an n-dimensional vector. The vector will most probably consist of centroids and areas of the triangles (this will be a mathematical representation of the image) and then making use of a neural network to teach the computer what these vectors represent. So when a new vector is input the neural network should be able to identify the expression.

c) Fuzzy logic : Marking angles as “low”, ”high” or “medium” and measuring angles in some order to form a vector and apply fuzzy logic to identify a smile. (This is the least understood of the three alternatives).

* + 1. ***Work done in VIII Semester.***

1. Realised that Haar classifier and other information is good for getting regions of interest, but need more powerful feature extraction features so decided to concentrate more on feature extraction and in general image processing before proceeding with machine learning techniques to identify an expression. This logic is well founded because if input data is not accurate enough it is very difficult for machine learning techniques (neural network etc.) to function as intended to. Also machine learning codes are well researched and are readily available and do not form a focus for our project per se this implies that the core of our project lies in image processing. Therefore image processing became and remains to be our primary focus. Underlying principle is that the machine learning techniques are as powerful as the image processing techniques that feed them data.
2. By the end of the even semester we were able to detect the regions of interest viz. Eyes, nose and mouth. Now we needed to extract features from these regions of interest. First we detected eyes using a simplified version of accurate iris segmentation which is an important step for automatic iris recognition and iridology. In previous iris segmentation approaches, the inner and outer boundaries of an iris can be taken as two non-centric circles and the upper and lower eyelids are modelled by two parabolas. Hough transform and other curves fitting techniques are capable of effectively determining the parameters of the circles and parabolas. By detecting a region of black at the centre of the iris we are able to detect pupils.
3. Next step was nasal feature extraction. Using Haar nose classifiers we were able to identify the nasal region on the face, coupled with some heuristics (nose found between and below the eyes). As the nose on its own does not undergo mush distortion during different expressions we need nasal points mainly as place holders to maintain consistency over the image. To meet our requirements it was observed that simple extreme points would suffice (i.e top of nose, bottom of nose, nostrils or left and right extreme points of nose). Therefore we simply took the heuristically accurate points from the Haar classified “Nose” region of interest and avoided complex image processing algorithms. In this way we have improved performance by removing redundant step excess image processing in the form of edge detection for nose etc.
4. After obtaining pupils and nasal features another very important feature to be tracked are cheeks. This is because during expressions such as “Smile” and “Surprise” there is significant shift in the cheek region of the face. Therefore to accurately recognize these expressions identifying the movement or shift in cheeks necessary. The initial focus was on using Delaunay triangulation as means to track the cheek moving however this failed for two reasons. (1) Mesh formation on the face is done by first creating a mesh and then trying to fit it on the face, but in our method we were trying to generate a mesh from points we get on the face, this is in essence the opposite of how the method is designed to work and hence fails. (2) We reached a paradoxical situation where we needed track able points on the cheek (which is inherently untrackable) to create the mesh, but the mesh was designed to solve the problem of cheek being untrackable. So at this stage we abandoned Delaunay Triangulation as a method of quantizing the extracted features.
5. In order to track cheeks we came up a marker-less, marker-based hybrid model where only the cheeks need to be marked using coloured markers. We tried markers of various colours to identify which can be most easily identified against skin tone and experimental verification gave us “Blue” (other competitors were green, yellow and orange) as the optimal colour for marking. We then applied a dominant colour detection algorithm to identify the marker and in this way we are able to track the cheek points and calculate shift in cheeks during an expression.
6. At this stage we had found that in spite of majority background noise elimination after masking of area NOT identified as a “face” using Haar Classifiers. Some residual background noise is found in the parts of the image very close to the actual face. This noise was interfering with the more sophisticated algorithms and had to be removed. To facilitate this we performed a skin detection algorithm that effectively “cuts-out” the skin part of the image. It is only in this cut out skin regions that more advance algorithms (like lip detection) are run. This intuitively improves accuracy.
7. For expression recognition even hair on the head and facial hair like beards constitute “noise” as they have no use for expression recognition and they interfere with certain other algorithms such as the eyebrow detection algorithm. In order to remove this excess hair we developed our own algorithm that identifying regions with maximum number of connected pixels of a particular threshold value and then eliminate it (masking acts as noise removal). The threshold value (range) is an RGB colour range that is commonly associated with hair. In conformance with the most commonly occurring colour for hair in test data sets program implementation is done by taking an RGB range of 0, 0, 0 to 50, 50, 50 (commonly called “black”).
8. After removing the excess hair (mainly hair on the head) we were able to perform eyebrow detection using a method similar to the hair removal algorithm. In order to track the hair of the eyebrows we modified our hair removal algorithm that tracks regions with maximum number of connected pixels of the RGB range of 0,0,0 to 50,50,50(“Black”) threshold value range. This algorithm is run for the region above the eyes and since hair on the head has been eliminated whatever regions are detected by the algorithm must be the eyebrows. Again this is in conformance with the most commonly occurring color for hair in test data sets. After identifying the eyebrow region we take the center of the eyebrow and the left most and right most points as the eyebrow points. i.e we have three points that we can identify per eyebrow.
9. The penultimate stage we performed in the feature extraction is the lip detection. For this we developed our own algorithm based on “Canny’s edge detection algorithm”. The first algorithm is based on a well accepted edge detection (Canny’s) method, it consists of two steps, the first one is a lip enhancing colour transform and the second one is edge detection based on active contours. Several colour transforms have already been proposed for either enhancing the lip region independently or with respect to the skin. Here, after evaluating several transforms; a colour transform that involves converting the image to grey scale is considered and this method is able to trace the outline of the lips.
10. Another lip detection method used corner detection algorithm and enabled us to arrive at the corner points of the lips (top, bottom, left and right) points.
11. We evaluated the usefulness and accuracy of the lip detection algorithms developed in step 23 and 24. It was found that the edge detection algorithm used in step 23 was superior to that in step 24 for the following reasons (1) Naive analysis indicated that the accuracy for step 23 is superior to that of step 24. The algorithm of step 24 was rarely able to mark more than two out of four points with any accuracy whereas the algorithm in step 23 was able to trace the outline of the lip with great clarity. (2) Due to the inaccuracy of step 24 it was very difficult to arrive at a technique to determine whether the mouth is open or closed. It is this parameter (mouth open or closed) that plays an important role in determining the expression “Surprise” and hence we comprehensively chose the algorithm used in step 23 for lip detection.
12. The final stage in feature extraction was to determine whether the mouth is open or close. Here we made use of data we obtained from the lip detection algorithm stated in step 23. The condition for mouth open was if there are a maximum number of connected pixels of a particular threshold value whose length was greater than 60% the length of the lip. As this would indicate the black/grey/pink/white regions of the mouth that are visible when it (the mouth) is wide open.
13. The next step in the work flow was to obtain a set of parameters (points) from the feature extraction. The points we were able to generate are
    1. Pupils x 2
    2. Eyebrows x 3(per eyebrow) x 2(number of eyebrows)
    3. Nose x 4 (extreme points)
    4. Lips x 4 (extreme points)
    5. Mouth (closed, partially open, closed)
    6. Cheeks x 2

That is a total of 17 parameters that represent the expressive regions of the face.

1. The next step is to normalize the data over the input images. This involved transformation of an absolute co-ordinate system to a more suitable system, where origin is shifted from top left corner of image to midpoint of line joining pupils. The distances are then unitized by taking distance between pupils to be 50 units.
2. After Normalizing the data sets obtained for the images we needed to take the difference in parameters. The difference being a simple geometric subtraction of vectors (taking each point to be a vector in a 2-D Euclidian space). This subtraction represents the shit or change in facial features between base and expressive image.
3. After completing the image processing and normalization; an appropriate machine learning technique is used to “teach” the system what these changes in the images “mean” is to recognize the expression being displayed in the expressive image.
4. We used a multilayer perceptron neural network which are layered feed forward networks typically trained with static back propagation. We chose this because they are easy to use, and that they can approximate any input/output map. We developed a MLP and then tested by training it to understand sine and cosine values.
5. After training with 1000 and then 100,000 sine and cosine values we tested the MLP to check if it had “learnt” the values of sine and cosine theta. The results were encouraging indicating almost 100% accuracy with small delta error after the network was trained for 100,000 values.
6. We then began training neural network on our images (pictures of project members) for two expressions (“Smile” and “Surprise”) and have found that it is able to recognize these two emotions with good accuracy. Due to time constraints we were not able to train the neural network for all three expressions for a large number of individuals.
7. **IMPLEMENTATION AND RESULTS**
   1. **Pipelined Work Flow**

This section briefly shows the stages which the system undergoes from input to output. It is a pipelined approach that combines many algorithms within itself to provide us with concrete output. The rest of this section will detail the inner workings of the macro pipeline stages. The program passes through various phases of the Image Processing and Data Processing in a step-wise manner as shown in “Figure 4”.

“Figure4. Pipelined process flow of the system”

* 1. **Input Mechanism and Technologies Used.**

The whole project has been bundled in a GUI for ease of Usage. The GUI or front-end has been developed using Python as the base language whereas the back-end is developed using “C” as the base language and OpenCv.

Input: Two Pictures of an individual

* + Picture 1: Base Image – Image where no expression is shown.
  + Picture 2: Expressive Image – Image where the individual in the base image make one of the three expressions being tested for (“smile”, ”frown” or “surprise”).

The UI supports parametric input in the form of Static, Video feed or Cam feed. Static takes two images that are already there on the users comp. Video feed takes in a video of a person changing his expression from a base starting point, the system samples every 20 frames. Cam feed is similar to Video feed, only it takes a live video feed from a web camera.

The system currently supports “.jpg” image type and “.avi” video. We plan to extend for more file types in the future.

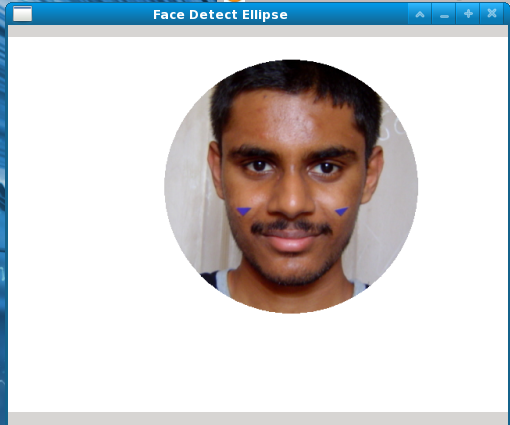
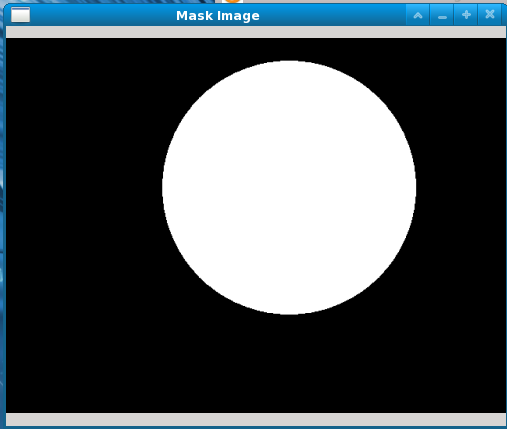
Verbose feature is also implemented where the execution stops after every step. This mechanism is good for debugging and observing the inner mechanisms of action.

**4.3 Face, Eyes, Nose, Mouth Detection using Haar Classifiers**

***4.3.1 Face Detection:***

This is the first phase of Image Processing makes use of something called as Haar Classifiers.Haar Classifiers make use of wavelets and contours to detect features. Haar Classifiers are a well established means of breaking up an image to pieces that contain specific features. It makes use of a visual neural network recognizer.

The Haar Classifiers give a rectangular box to mark the possible presence of a Human Face. But there is a chance of minimal data loss. Hence, we improvised it a little bit by expanding the rectangular region into an ellipsoidal region with a slight expansion to recover for the lost data as shown in “Figure 5” and these are trained using Haar training.Once the Ellipsoidal region is determined, a mask is generated and everything except in the Region of Interest if masked out as shown in “Figure 6”. Hence, improving the performance of application for further Image Processing. This partially masked image is then used for the next set of Image Processing operations.

“Figure6 . Mask Background”

“Figure5. Face Detection”

***4.3.2 Eye Detection:***

TheEyes are the only very "Trackable" features on Human Face. Due to the presence of both hard edges and presence of high contrast and their stable and constant location on the face they are the only feature of the face that can be identified immediately and with maximum accuracy within the face as shown in “Figure 7”.

This second phase of Image processing again makes use of Haar Classifiers which are trained to recognize the eye region. Since we have limited of region of interest to the face the detection is almost 100% accurate.

Here the outputs are two points marking the two pupils. The performance of Haar Classifiers for the eye-detection is very high because of the inherent distinctness of the pupil. Hence, we get a very high performance statistics for eye-detection.

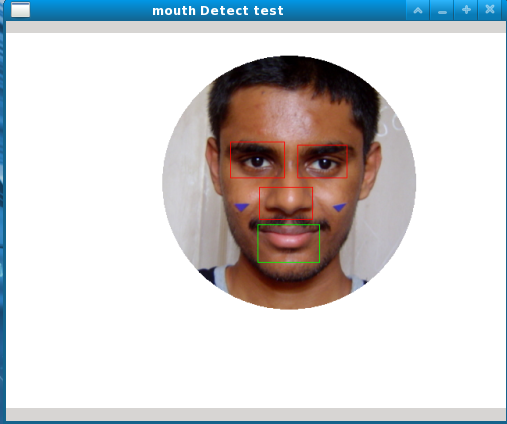
***4.3.3 Nose Detection:***

This is the third phase in image filtering and re-employs Haar Classifiers as the technique to identify the nose as shown in “Figure 7”. This technique of using Haar Classifiers within a Haar classified image is called cascading Haar classification. But the performance of the algorithm degrades to an extent as compared to eye-detection,due to the inherently less “trackable” nature of the nose.

Accuracy is improved by taking the nose to be in the region between and below the eyes, this heuristic greatly improves hit rate for finding the nasal region in the face.

***4.3.4 Mouth Detection:***

This phase again makes use of cascading Haar Classifiers trained on the mouth and lips as shown in “Figure 7”. The performance of native mouth-detection using Haar Classifiers is again lower when compared to eye-detection but is on par with nose-detection. Again heuristics are employed to improve accuracy.Here, we improvised on the native Haar Classifiers Detection. Since for all Human Faces, we can generalize that (1) Mouth will always be present in between the eyes, and(2) Mouth will always be below the Nose and hence since we have a grasp on the location of eye and nose in the face, we can reduce our Region Of Interest by manifolds. Hence, greatly improving time complexity for later algorithms that act on specific regions as well as their accuracy. Accuracy of the Haar Classification is illustrated in “Table 1”.



“Figure 7 . Eyes,Nose,Mouth Detect”

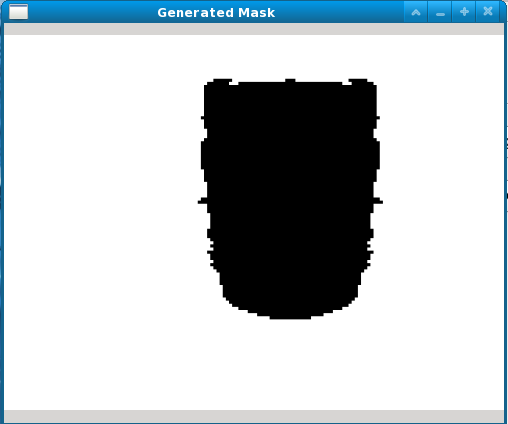
|  |  |  |  |
| --- | --- | --- | --- |
| Person | Region of Interest | Successful Identification | Accuracy |
|  | Face | 18 | 90% |
| Veeresh | Nose | 17 | 85% |
|  | Eyes | 19 | 95% |
|  | Mouth | 19 | 95% |
|  |  |  |  |
|  | Face | 20 | 100% |
| Srinivas | Nose | 14 | 70% |
|  | Eyes | 19 | 95% |
|  | Mouth | 19 | 95% |
|  |  |  |  |
|  | Face | 19 | 95% |
| Sushant | Nose | 17 | 85% |
|  | Eyes | 20 | 100% |
|  | Mouth | 16 | 80% |

“Table 1. Basic Feature Detection using Haar Classifiers”

* 1. **Skin Segmentation:**

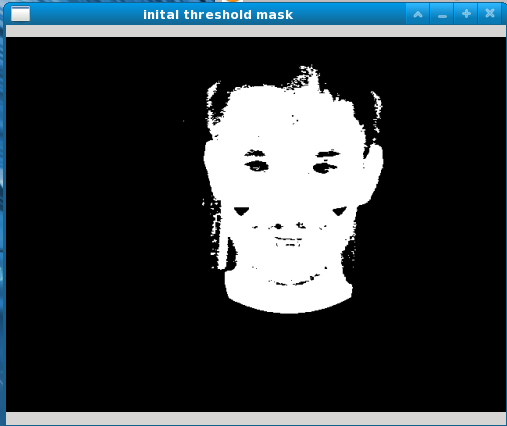
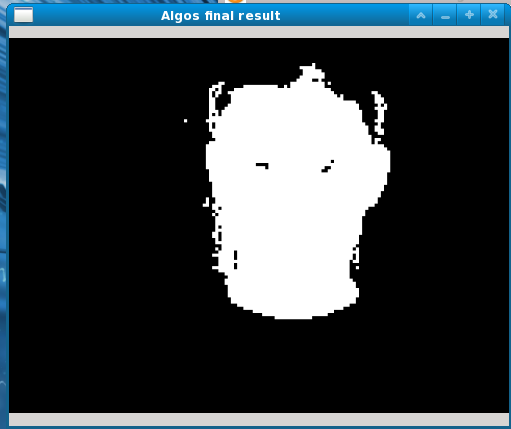
This algorithm addresses our proposed method to automatically segment out a person’s face from a given image that consists of a head-and-shoulders view of the person and a complex background scene as shown in “Figures 8 and11”. The method involves a fast, reliable, and effective algorithm that exploits the spatial distribution characteristics of human skin color.

A universal skin-color mapis derived and used on the chrominance component of the input image to detect pixels with skin-color appearance. Then, based on the spatial distribution of the detected skin-color pixels and their corresponding luminance values, the algorithm employs a set of novel regularization processes to reinforce regions of skin color pixels that are more likely to belong to the facial regions and eliminate those that are not (elimination shown in “Figures 9 and 10”). The performance of the face segmentation algorithm is illustrated by simulation results carried out on various head-and-shoulders test images.

“Figure 9 .Generate Mask””

“Figure 8 . Begin Skin Segmentation””

*** ***

“Figure 11 .Final Skin Region””

“Figure 10 .Initial Threshold Mask””

**4.5 Hair Removal on Head:**

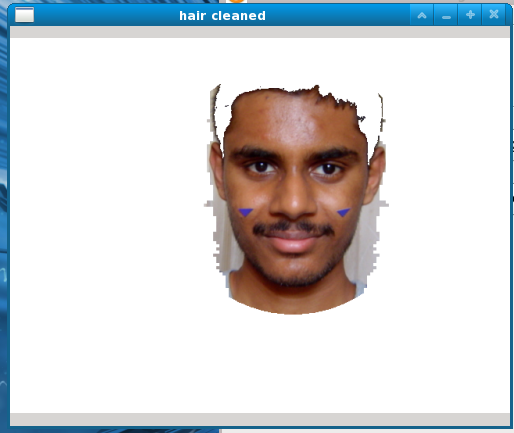
This is one of the indigenously developed algorithms that we developed and the need for this algorithm arose when the facial hair especially that on the head began interfering with the detection of eyebrows. It was then that we decided that a method for removing hair had to be devised. The algorithm works by shading the area covered by hair on the head and hence gives us a very good hold on the positioning of fore-head and eyebrows.***4.5.1******Functioning of the Algorithm:***

The algorithm makes use of two major concepts from Graph Theory:

* Connected Components in Graphs
* Depth-First Traversal of Graph

We start from the top of the Region of Interest of the full-face i.e. taking hold of the top-most pixel with hair in it and then propagating through connected components in a FS fashion till we encounter a non-black pixel, which is where we terminate the graph. Removal of hair using this algorithm has the major advantage of getting a better hold on the eye-brows.

The results are shown in “Figure 12”.



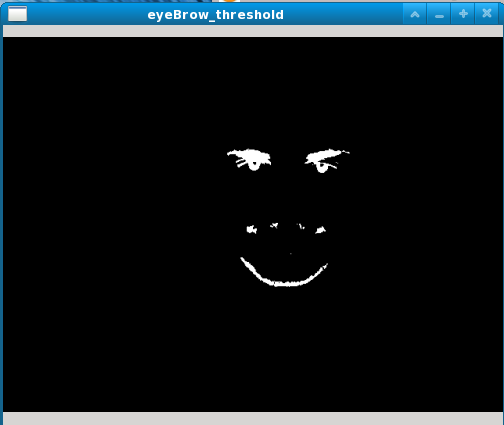
“Figure 12 .Hair Removal””

**4.6 Eye-Brow Detection:**

This phase of the image-processing is based on the assumptions that:

* The eye-brow hairs are all black – This assumption stems from the fact that in all our test data hair is “black”. However this can be changed if different coloured hair individuals use this image.
* The eyebrows are confined to the region above eyes.

Using this heuristic and black as threshold color, we detect the position of eye-brows use the same algorithm as the head hair removal technique as shown in “figure 13”.



“Figure 13. Eyebrow Threshold Generation””

**4.7 Cheek Detection:**

This is one the critical and most challenging features we came across during various feature recognitions of the face. Due to inherent un-trackability of the cheek-bones, we had to choose one of the following options:

1) Ask for user-input of cheek bones. (Not done because, it can't be ensured that same points will be marked again in the second run).

2) Extrapolate the points that we get from eye, nose and mouth detection. Highly inaccurate and prone to failure

3) Use marker based detection to get cheek-bones. This is easy to implement and reliable.

We chose the third option and ran the algorithm with various colour thresholds whose detection rates illustrated in “Table 2”

|  |  |  |
| --- | --- | --- |
| Colour | Markers Detected | Accuracy |
| Blue | 46 | 92% |
| Green | 42 | 84% |
| Yellow | 29 | 58% |
| Orange | 34 | 68% |

Table 2. Marker Detection Algorithm Performance

Finally, we chose Blue as the marker color, one of the explanations as to why blue is a good colour to use as a marker is: Skin color has some proportions of yellow and red in it and hence interferes with the marker detection.

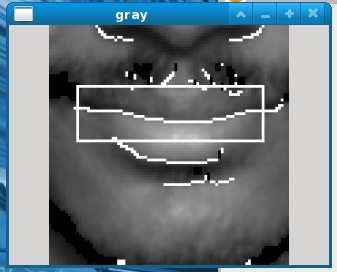
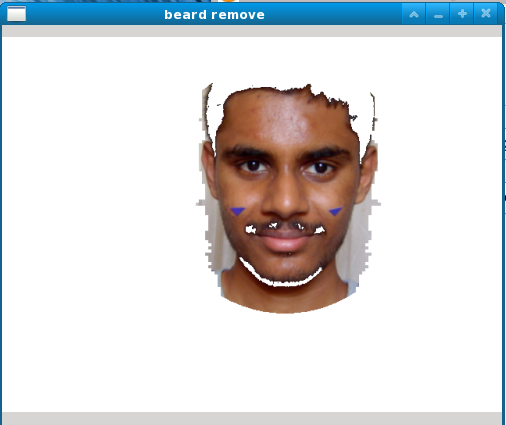
**4.8 Lip Contour Detection:**

Here we first do the beard subtraction (shown in “Figure 15”) using the same "Hair Removal Algorithm", used earlier. Once the hair is removed we run an edge detection algorithm (in our case Canny), which gives us a gray-scale image with all the edges. We then found the longest of the edges and set a rectangular region enclosing the edge with a height (Base Image) as shown in “Figure 14”.

For an image with emotion, we again run the same edge-detection algorithm and again find out the edges in the same region. The results are shown in “Table 3”.1) No edge means the mouth is opened.2) More than one longer edge implies that the mouth is open.3) Else it is closed.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Algorithm | 4 extreme points found | 3 extreme points found | 2 extreme points found | 1 extreme point found | Accuracy |
| Corner Detection | 2 | 8 | 22 | 8 | 52.5% |
| Edge Detection | 16 | 21 | 3 | 0 | 83.1% |

Table 3. Algorithm Performance

“Figure 14 .Grey Scale Lip Contour Detection””

“Figure 15 .Beard Removal””

**4.9 Neural Network Training**

Multilayer Perceptron (MLPs) are feed forward neural networks trained with the standard back propagation algorithm based on supervised learning and require a desired response to be trained. They learn how to transform input data into a desired response, so they are widely used for pattern classification. With one or two hidden layers, they can approximate virtually any input-output map. They have been shown to approximate the performance of optimal statistical classifiers in difficult problems. Most neural network applications involve MLPs.

***4.9.1 Why MLP?***

* Popularity - the most used type of NN; Universal Approximators - general-purpose models, with a huge number of applications;
* Nonlinearity - capable of modelling complex functions;
* Robustness - good at ignoring irrelevant inputs and noise;
* Adaptability - can adapt its weights and/or topology in response to environment changes;
* Ease of Use - black-box point of view, can be used with few knowledge about the relationship of the function to be modelled.

***4.9.2 Neural Network Design***

Key features while designing a Multi Layer Perceptron:

* No of input nodes
* No of output nodes
* No of hidden or intermediate layers
* No of neurons at each layer
* Learning rate
* Activation Functions for each layer
* Weights of the edges

The number of input and output units is defined by the problem (there may be some uncertainty about precisely which inputs to use, a point to which we will return later. However, for the moment we will assume that the input variables are intuitively selected and are all meaningful). The number of hidden units to use is far from clear. As good a starting point as any is to use one hidden layer, with the number of units equal to half the sum of the number of input and output units.

The key issues can be enlisted as:

• Selecting how many hidden layers to use in the network.

• Deciding how many neurons to use in each hidden layer.

• Finding a globally optimal solution that avoids local minima.

• Converging to an optimal solution in a reasonable period of time.

• Validating the neural network to test for over fitting.

There are a set of rules to decide on quantization of above stated parameters.

* For nearly all problems, one hidden layer is sufficient. Two hidden layers are required for modelling data with discontinuities such as a saw tooth wave pattern. Using two hidden layers rarely improves the model, and it may introduce a greater risk of converging to a local minima. There is no theoretical reason for using more than two hidden layers.
* Number of neurons at each layer is given by using following set of **thumb** rules:
  + The number of neurons at any layer is generally chosen to be a value in between the input and output nodes.
  + The number of neurons at a layer is determined as 2/3 of the input nodes to the layer, plus the number of output nodes.
  + The number of hidden neurons should be less than twice the size of the input.
* Total numbers of weights to be determined are
  + *Weights* = (1+ *n*)\**m*
* There is a simple method to find out Neural Network hidden Neurons. Assume a Back Propagation Neural Network Configuration is l-m-n. Here l is input neurons, m is hidden neurons and n is output layers. If we have two input and two output in our problem than we can take same number of hidden neurons. So our configuration becomes 2-2-2. (l=2 input neurons=2 hidden neurons and n=2 output neurons.)

**Activation Functions:**

In computational networks, the activation function of a node defines the output of that node given an input or set of inputs. A standard computer chip circuit can be seen as a digital network of activation functions that can be "ON" (1) or "OFF" (0), depending on input. This is similar to the behaviour of the linear perceptron in neural networks. However, it is the *nonlinear* activation function that allows such networks to compute nontrivial problems using only a small number of nodes.

Types of Activation Functions are:

* Linear: y = x ;
* Tanh: y = tanh(x) ;
* Logistic (or sigmoid): y = 1/1+e−x

**Node Activation:**

Each node outputs an activation function applied over the weighted sum of its inputs: si = f (wi ,0 + Pj2I wi ,j × sj ).

#### Choosing a Cost Function:

While it is possible to define some arbitrary, ad hoc cost function, frequently a particular cost will be used, either because it has desirable properties (such as convexity) or because it arises naturally from a particular formulation of the problem (e.g., in a probabilistic formulation the posterior probability of the model can be used as an inverse cost). Ultimately, the cost function will depend on the desired task.

***4.9.3 Neural Network Implementation:***

MLP, a feed-forward neural network shown in “Figure 16” has the following functions:

* Takes in the batch files
* Reads the set of inputs and outputs from the formatted feed
* Build the network on the first entry
* Update the network with every new set of input/output pairs we give.
* Save the network at the end of the training for further decision making

Neurons

Neurons

Neurons

17 input nodes

2 output nodes

17 x 12 connecting lines

2

12

17

“Figure 16 . Design of implemented MLP”

**Salient Features of Implemented Neural Network:**

* No of input nodes : 17
* No of output nodes : 2
* No of hidden or intermediate layers : 1
* No of neurons at each layer : {17, 12, 2}
* Learning rate : {0.8}
* Activation Functions for each layer : {Sigmoid, Sigmoid, Tanh}

***4.9.4 Neural network performance analysis***

“Table 4” illustrates accuracy of neural network trained on values of SinƟ and CosƟ i.e given a Ɵ(angle) value the neural network predicts a Sine and Cosine value.

Basically this table indicates the accuracy and reliability of a neural network based on the size of the training set and the learning rate that we supply to the network.

For testing we have taken 50 arbitrary Ɵ(angle) values and fed them as input to the network and have tolerated an accuracy of ±0.01.

|  |  |  |
| --- | --- | --- |
| Number of training sets | Learning rate | Accuracy |
|  |  |  |
|  | 0.5 | 9% |
| 100 | 0.6 | 17% |
|  | 0.7 | 33% |
|  | 0.8 | 23% |
|  |  |  |
|  | 0.5 | 22% |
| 1000 | 0.6 | 46% |
|  | 0.7 | 56% |
|  | 0.8 | 41% |
|  |  |  |
|  | 0.5 | 52% |
| 100000 | 0.6 | 63% |
|  | 0.7 | 80% |
|  | 0.8 | 73% |
|  |  |  |
|  | 0.5 | 52% |
| 100000 | 0.6 | 63% |
|  | 0.7 | 80% |
|  | 0.8 | 73% |

Table 4. Neural Network Performance Analysis

1. **CONCLUSIONS AND FUTURE WORK**

**5.1 Conclusions**

* **The system is able to detect expressions for given image sets with high accuracy provided the neural network has been trained on an individual’s pictures.**
* Final output depends directly on the accuracy of intermediate steps AND on how well they (intermediate stages) integrate with each other. For example one algorithm A may not maintain the assumptions of algorithm B, hence we must avoid a situation where algorithm A is run before algorithm B.
* Neural networks show high accuracy for trained images. This is again possible only because of powerful image processing, feature extraction and normalization.
* Selection of algorithms is dependent on nature of test data sets and heuristics. In other words some algorithms work well with certain data sets and we can determine optimum algorithm for a particular goal heuristically.
* As image processing algorithms are not guaranteed and are tested under specific conditions and subject to heuristics improvements must be made to satisfy your system’s demand. In other words one algorithm will NOT suit all and a deep understanding of how it works and how to customize it to your needs must be made.
* Variations in testing conditions such as lighting and image format have a bearing on the outcome. So the correct testing conditions and image formatting should be maintained for high accuracy output.
* Due to the differential nature of test data sets with respect to testing conditions, image quality, differences between individuals (for some people certain features are detected with almost 100% accuracy whereas others have lower hit rates) for a concrete outcome data presented must be averaged over a large data set.
* There are many other methods and image processing algorithms present and we have used only those that seemed most appropriate, but this is by no means the last word in image processing many other exciting and challenging prospects still exist in this area for the same problem statement.

**5.2 Future Work**

* Test neural network for more image sets that can be found locally
* Test system for benchmarked data sets.
* If results continue to be as encouraging as we have seen so far we plan to continue our research in the following areas
  + Pipelined expression recognition system using trained MLP.
  + Lips and Mouth detection to identify whether mouth is open or closed
  + Optimization of Hair removal algorithm

**References**

[1] Emotion Detection of Infants from Facial Expressions and Cries: By Pritam Pal,

Electrical & Computer Engineering Department, Temple University, May 2006.

[2] Biometric Recognition Performing in a Bioinspired System: By Joan Fabregas, University of Barcelona, June 2010.

[3] Facial Feature Detection Using Haar Classifiers: By Phillip Ian Wilson, University of Texas A&M, April 2006.

[4] Automatic facial emotion recognition: By Aitor Azcarate et al, Department of Computer Science, University of Amsterdam, June 2005.

# **[5] Facial feature location with Delaunay triangulation/Voronoi diagram calculation: By Yi Xiao, Hong Yan, University of Sydney.** ISBN:0-909-92589-5.

[6] Face Segmentation Using Skin-Colour Map In Videophone Applications: By Douglas Chai, Student member IEEE. INSPEC Accession Number: 6280597.

[7] Detecting Faces in Images: A Survey: By Ming-Hsuan Yang, Member, IEEE et al. INSPEC Accession Number: 7165080.

[8] Voice Activity Detection by Lip Shape Tracking Using EBGM: By Masaki Aoki*,* Graduate School of Engineering, Kobe University. ISBN: 978-1-59593-702-5.

[9] Detecting Eyelash and Reflection for Accurate Iris Segmentation: By Wai-Kin Kong and David Zhang, Biometrics Research Centre, Department of Computing, The Hong Kong Polytechnic University. Current Version Date 07 August 2002.

[10] Emotion Recognition Using Neural Networks: by MEHMET S. UNLUTURK et al. Department of Software Engineering, Izmir University of Economics, 10th WSEAS International Conference on NEURAL NETWORKS.

[11] Comparative Analysis of Lip Features for Person Identification: By Usman Saeed,

COMSATS Institute of Information Technology, ISBN: 978-1-4503-0342-2.

[12] Neural Network Classification of Diesel Spray Images: By S. D. Walters, Engineering Research Centre, School of Engineering, The University of Brighton, April 2004.