2012

Development Of Emotional Health Monitoring And Feedback System Based On Facial Expression Recognition,Traditional Chinese Medicine Theory, And Regional Agriculture

Franshaun Latrice Hardy
North Carolina Agricultural and Technical State University

DEVELOPMENT OF EMOTIONAL HEALTH MONITORING AND FEEDBACK SYSTEM BASED ON FACIAL EXPRESSION RECOGNITION, TRADITIONAL CHINESE MEDICINE THEORY, AND REGIONAL AGRICULTURE

by

Franshaun Latrice Hardy

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Department: Electrical and Computer Engineering
Major: Electrical Engineering
Major Professor: Dr. Corey A. Graves

North Carolina A&T State University
Greensboro, North Carolina
2012

School of Graduate Studies

North Carolina Agricultural and Technical State University

This is to certify that the Master Thesis of

Franshaun Latrice Hardy

has met the thesis requirements of

North Carolina Agricultural and Technical State University

Greensboro, North Carolina

2012

Approved by:

Dr. Corey A. Graves Major Professor	Dr. Gary Lebby Committee Member
Dr. Christopher Doss Committee Member	Dr. John Kelly Department Chairperson

Dr. Sanjiv Sarin
Associate Vice Chancellor of
Research and Graduate Dean

DEDICATION

This thesis is dedicated to my loving mother. She has instilled in me the values of hard work diligence, perseverance, self–discipline, courage and faith; a true woman of virtue. I would not have achieved this great accomplishment without her love and support. I truly love and appreciate her with all of my heart.

BIOGRAPHICAL SKETCH

Franshaun Latrice Hardy was born April 25, 1988 in Richmond, Virginia. She is the second child of four biological siblings and the third of five sisters. She received her high school diploma from Huguenot High School of Richmond, Virginia in 2006. She received her Bachelor of Science in Computer Engineering at North Carolina Agricultural and Technical State University in 2010. She is a candidate for a Master's of Science in Electrical Engineering with a concentration in Computer Engineering.

ACKNOWLEGEMENTS

I would first like to honor my Lord and Savior Jesus Christ for his favor and many blessings that He has bestowed upon me. Without the direction and leadership of God this degree would have been impossible to achieve.

I greatly appreciate the support of my graduate advisor, Dr. Corey A. Graves and my committee members, Dr. Gary Lebby and Dr. Christopher Doss.

I would like to thank my parents Jerry and LaDonna Hardy for their love and support throughout my collegiate years. They instilled in me the true values of working hard to achieve every goal that I have set forth.

I would like to thank my siblings Jasmin, Jerry Jr., Steven, Vakaya , Alexis and La'Nita for all of their support, prayers, and words of encouragements. I am truly grateful to have a host of sisters, who are there to support my every effort. They have truly made this journey easier to tackle. I appreciate the love of my brothers, with the random phone calls and text messages just to check up on me. I would like to thank my niece Joy for her prayers and just for being the simple "bundle of joy".

I am appreciative of my significant other, Taye Esho, who has been that great support system throughout my graduate journey. He is the true definition of a man after God's heart. His prayers, words of encouragement and presence are what pushed me to continue this great task. I am truly honored to be in love and in a relationship with such a great man of God. I look forward to what the future holds for us.

I am grateful for my loving church family at Mt. Zion Baptist Church, who gave me the word to push a little further. My wonderful young adult pastor, DeJuan Harris and his wife Sannyu Harris were that direct covering and support for my spiritual and educational growth.

To my wonderful alma mater North Carolina Agricultural and Technical State University, thank you for this awesome journey of undergraduate to graduate studies. I started this college career as an immature girl with some direction and am leaving as a mature young woman, who is confident, know what she wants and know how to handle and conduct herself in a professional setting. Thank you for the many opportunities. I will always remember and have AGGIE PRIDE!

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES **PAGE**

LIST OF TABLES

TABLES **PAGE**

ABSTRACT

Hardy, Franshaun Latrice. DEVELOPMENT OF EMOTIONAL HEALTH MONITORING SYSTEM BASED ON FACIAL EXPRESSION RECOGNITION, TRADITIONAL CHINESE MEDICINE THEORY, AND REGIONAL AGRICULTURE **(Major Advisor: Dr. Corey A. Graves)**, North Carolina Agricultural and Technical State University.

An individual's emotional, as well as physical state plays a valuable role in their overall productivity. If one has a balance of physical and emotional well being, the likelihood of their performance will increase towards their daily tasks and assignments. One's nutritional diet plays a vital role in creating and producing balance into their lives. Studies have shown that the foods you eat have a significant effect on your physical, mental and emotional state.

The objective of this research was to develop a system that will enable more research for emotion monitoring and feedback with the long term goal of developing a system that improve the emotional health and overall nutritional well being of an individual. The system estimates the emotion of an individual through the EigenFace Expression Recognition approach developed by Md. Iftekhar Tanveer, and then suggests a list of healthy snacks based on their tastes to an individual to improve their mood through the influence of Traditional Chinese Medicine. This research studied five facial expressions: neutral, anger, happy, disgust and sad, which in return expressed the emotional state of the person. The neutral expression represents the reference of the system. The intensity of a particular expression can be identified by its level of dissimilarity from the neutral expression.

The Traditional Chinese Medicine Theory uses the Five Element System to treat and diagnose their patients. This system includes earth, wood, fire, water, and metal to create balance in the body and its corresponding taste and emotion. Based on the particular emotion, which is the diagnosis portion, the taste that corresponds to the element represents the treatment for that emotion. A series of trials were run to test the accuracy of the Emotional Health Monitoring System. The highest performing emotion was the anger at a rate of 80% and the lowest performing emotion was sad at 50 %. This caused the system to perform at an overall accuracy rate of 64%.

CHAPTER 1

INTRODUCTION

In today's society, the work force is largely confined to an office/cubicle environment. The components of a typical work space include the desktop/ laptop computer, telephone, cabinets, drawers and a chair. A thoughtful employer has the office space setup to accommodate the individual in such a space to promote optimal productivity [20]. Ideally, every part of the work area should be measured according to the height and width of the person. For example, the computer should be set up at the eye level of the user and the desk should be positioned such that the user should be comfortable with his/her legs cleared underneath it [20].

Due to improper setup, injury in the work place has become more prevalent. The injuries consist of neck, back, hand and arm problems [21]. These injuries are caused from six or more hours of computer usage without proper breaks and stretches. Furthermore inappropriate work station setup is another cause of work related injuries. Due to these injuries, many companies have taken action to prevent as many injuries as possible by focusing on Ergonomics [21]. Ergonomics consist of the proper fit of an office space with an individual. Companies have begun to use various software applications that log and keep track of the activity on the individual's computer. The applications suggest 5, 10, or 15 minute breaks and sample stretches [21].

As mentioned previously, companies are concerned with the physical state and well being of their employees. However, companies are not as concerned with the emotional state of their employees. A person's emotional state plays a role in their productivity in the work place, as much as physical health. If one has a balance of physical and emotional well being, chances are that they will perform better at their daily tasks and assignments. Studies have shown that the foods you eat have a significant effect on both your mental and emotional state [2]. For example, if you have ever reached for a cup of coffee or soda to keep you awake or alert, you have experienced a food mood change. The foods you consume not only affect your body physically, but affect the functionality of your mind.

Using the Five Element System of earth, fire, wood, water and metal to create balance to the body, the Traditional Chinese Medicine theory method is used to treat and diagnose their patients. Each element has a corresponding taste and emotion. Based on the particular emotion, which is the diagnosis portion, the taste that corresponds to the element is then controlled by the controlling cycle and the taste of the stronger element represents the treatment for that emotion. For example, the wood element corresponds to the anger emotion and a sour taste, the fire element corresponds to the joy emotion and a bitter taste. The earth element corresponds to the worry emotion and a sweet taste, as the metal element corresponds to the sadness emotion and a pungent taste and the water element corresponds to the fear emotion and a salty taste [1].

This research focuses on the emotional well-being of an individual in the work place or one whom spends six or more hours on a computer daily. This research produces an application that monitors the emotional states of the individual throughout the day.

The system randomly snaps a photo of the user, estimates their facial expression, as well as an indication of their mood/emotion. From a collection of snap shots, the system suggest healthy snacks specific to the southeastern region of the United States and the correct season of the user that is intended to improve their mood.

CHAPTER 2

BIOMETRICS

Biometrics is a system that extracts physical features and behavioral characteristics from a person. This system is used for verification and identification purposes. Verification confirms or denies a person's claimed identity based on a 1:1 matching ratio, while identification recognizes who the person is from a list of users in a dataset of images. Facial Recognition, Iris Recognition, Finger Print Authentication, and Facial Expression Recognition are all examples of biometric applications. Only two of the four applications mentioned will be discussed in this paper, (Facial Recognition and Facial Expression Detection).

2.1 Facial Recognition

Facial Recognition is an automated technique which identifies/ verifies a digital image of an individual based on their facial features and characteristics. This is done by comparing selected facial features of an image with a database of facial images [3]. One of the major advantages of facial recognition is that it can be done at a distance without the user being aware of the process [3]. Facial recognition is used in various applications such as surveillance in airports, stores, banks, security access to computers, and authorization of important documents/files.

2.2 Facial Expression Recognition

Facial expressions play a vital role in nonverbal communication. In human-to human interaction it is easy to determine ones emotion based on common knowledge and experience, however this process is very difficult in human to computer interaction. Facial Expression Recognition is an automated technique that identifies and verifies a digital image of an individual based on their facial features and characteristics that resembles a particular expression which indicates their emotion.

2.3 Methods for Facial Expression and Facial Recognition

Various methods of facial expression recognition have been proposed and implemented based on the performance of the face image variations of particular expressions. According to [8] facial expression recognition techniques are categorized into two methods, appearance based and geometric base. He states that appearance based approaches have proven to be better than geometric features, because geometric features are very sensitive to noise. The approach stated in his research was the Fusion of Gabor and Local Binary Pattern features [8].

Tanveer implemented the EigenFace facial expression approach using Principal Component Analysis adopted from the EigenFace Approach developed by Turk and Pentand [5]. Li implemented a real time user independent facial expression recognition system that combined Linear Discriminat Analysis and Principal Component Analysis to reduce dimensionality [12].

The methods considered for this research includes all of which are in previous statements and are thoroughly: Local Binary Pattern Method Combined with various neural networks and filters and EigenFace Approach for Facial Recognition and Facial Expression Recognition and the Linear Discriminant Analysis for real time implementation and FisherFace analysis [8], [11], [3], [5], [13], [12].

2.3.1 Local Binary Pattern

Local Binary Patterns (LBP) is a very popular method that describes the texture and shape of an image which is prevalent for feature extraction in facial expression recognition [6]. Ojala et al first introduced LBP operator and showed its high discriminative power for texture classification [7]. Local Binary Patterns is an ordered set of binary comparisons of pixel intensities between the center pixels and the surrounding neighboring pixels [6]. LBP quantifies intensity patterns or textures of local pixel neighborhood patches, which are pre divided sections of a given image. The Local Binary Pattern code equation is expressed below:

$$LBP(x, y) = \sum_{n=0}^{7} s(\, i_n - i_c) 2^n \qquad (1)$$

Where i_c corresponds to the gray level center pixel of (x, y), i_n represents the 8 neighboring pixels of the image. The s$(i_n - i_c)$ function is defined as follows:

$$s(i_n - i_c) = \begin{cases} 0, & i_n - i_c < 0 \\ 1, & i_n - i_c > 0 \end{cases} \qquad (2)$$

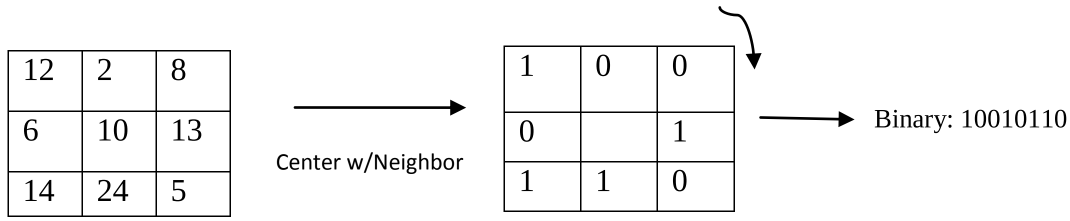

Figure 2.1. LBP Binary Comparison Example

Each bit of a LBP code has the same significance level and two successful bit values can have a totally different meaning [6]. An extension to local binary patterns is classifying the patterns as uniformed or non-uniformed. A uniform pattern is when there are 2 or less intensity changes. An intensity change is when a bit value changes from 1 to 0 or 0 to 1. If there are more than 2 changes then the pattern is said to be non-uniform. For example: 11111111 (0 transitions) uniformed pattern, 01011000 (4 transitions) non-uniformed pattern, and 10011111 (2 transitions) uniformed pattern.

All feature values from the uniformed patterns are quantified into 59 bins based on uniformity. The histogram contains information about the distribution of the local micro-patterns, such as edges, and flat areas across the whole image [8]. Face images are composed of the micro patterns which are described by LPB features. The LBP histogram of the entire face only encodes occurrences of the micro patterns and does not indicate pattern locations. Next, LBP features from each patch are concatenated into a single feature histogram. The histogram provides information of the features at the pixel and regional levels which are very much useful to Facial Recognition and Facial Expression Recognition [8].

2.3.2 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a statistical approach for classifying samples of unidentified classes based on training samples with identified classes [9]. This method aspires to maximize between-class (across users) variance and minimize within class (within user) variance [9]. LDA is a very common technique used in Facial Recognition in which it looks for linear combinations of variables which best explain the data. The Functionality of LDA is as follows, according to Etemad and Chellappa [9]:

1. Obtain a training set composed of a relatively large group of subjects with diverse facial characteristics.
2. For each image and sub image, construct the lexicographic vector expansion. This vector corresponds to the initial representation of the face.
3. Establish a framework for performing cluster separation analysis in the feature space for the faces of different subjects.
4. After the subjects are separated into their classes, compute the within and between– class scatter matrices.

$$S_w^{(V)} \sum_{i=1}^{L} \Pr(C_i)\Sigma_i \quad (3)$$

$$S_b^{(V)} \sum_{i=1}^{L} \Pr(C_i)(\mu - u_i)(\mu - \mu_i)^T \quad (4)$$

S_w is the within-class scatter matrix showing the average scatter of the sample vectors (V) of different classes C_i around their respective mean, vectors μ_i. S_b is the between-class scatter matrix, representing the scatter of the conditional mean vector (μ_i) around the overall mean vector μ. PrC_i is the probability of the *ith* class [9].

5. The discriminatory power of a representation is quantified by using the separation matrix which shows the combination of the within-and between-class scatters of feature points in the representation space. The class separation and measure of seperability is computed as follows:

$$S^{(V)} = S_{w^{-1}} S_b \tag{5}$$

$$J_{(V)} = sep(V) = trace(S^{(V)}) \tag{6}$$

J_v is the measure of the discrimination power (DP) of a given representation V [9].

One of the most familiar algorithms that use Linear Discriminant Analysis (LDA) for Facial Recognition is FisherFace. FisherFace was introduced and implemented by (Joao, Hespanha and Kriegman) [13]. They classified FisherFace as a class specific and reliable method that is used to reduce the dimensionality and simplify classifiers in the reduced feature space by shaping the scatter in classification [13]. This method avoids the problem of singularity in the scatter within class by projecting the image to a lower dimensional space which causes the within scatter class matrix to be non-singular. This method was implemented by using LDA to reduce the dimensionality of the feature space and then applied the standard Fisher Linear Discriminant [13].

2.3.3 Principal Component Analysis

Principal Component Analysis (PCA) is a common statistical technique used for finding patterns of data with high dimension, which conveys the similarities and differences between data [4]. This method does not experience much information loss, once patterns are found in data and then compressed by the reduction of the number of dimensions. PCA is an immensely useful tool for data analysis in image processing techniques that are used in biometric applications, i.e. Facial Recognition and Facial Expression Recognition. The functionality of Principal Component Analysis is as follows:

1. The First step in PCA analysis is to retrieve or create a dataset of 2-D values.
2. The next step is to subtract the mean of the dataset values $(x - \bar{x})$. Subtracting the mean calculates the average across each dimension [4]. For every x value there is an $\bar{x}$ (the average of all x values in dataset) subtracted from it. For every y value there is $\bar{y}$ subtracted from it. This produces a data set which equates to zero.
3. Thirdly, calculate Covariance Matrix.

$$\frac{\sum_{j=0}^{n}(x - \bar{x})(y - \bar{y})}{n-1} \quad (7)$$

The covariance matrix is a matrix that represents all of the covariance values calculated from equation 2 for all the dimensions in the dataset [4]. The covariance matrix must be a square matrix such that the eigenvectors/eigenvalues can be calculated.

4. The fourth step is to calculate the eigenvectors and eigenvalues of the covariance matrix. This process characterizes the data. All eigenvectors of a matrix are perpendicular to each other. The eigenvector with the highest eigenvalue is the principal component of the dataset. The next step is to choose components and form a feature vector, which is a matrix of eigenvectors selected by the order of significance [4].
5. The last step in PCA is to take the transpose of the matrix and multiply it by the original dataset transposed. This transformation will provide the original dataset in terms of the eigenvectors selected [4]. This process simply transforms the data and expresses it in patterns that describe the relationships between the data [4].

One of the most familiar algorithms that implement the concepts of PCA is the EigenFace approach [3]. EigenFace is used for Facial Recognition and Facial Expression Recognition applications.

EigenFace Approach

According to Turk and Pentland, the concept of EigenFace is to find the principal components of the distribution of faces, or the eigenvectors of the covariance matrix of the set of face images, treating an image as a point (or vector) in a high dimensional space [3]. Eigenvectors, often looked upon as feature vectors are ordered, accounting for a different amount of the variation among the face images [3]. An eigenface is a ghost like face which is the result of the image location with respect to the eigenvector. Each eigenface digresses from the uniform gray appearance when facial features differ among the set of training faces that indicates the map of variations between faces [3].

An individual face can be represented by the linear combination of eigenfaces or by also using the eigenfaces with the largest eigenvalues that have the most variance within a set of face images [3]. EigenFace was motivated by a technique created by Sirovich & Kirby (1987) and Kirby & Sirovich (1990) for effectively representing pictures of faces using principal component analysis [3]. Kirby and Sirovich stated that any collection of face images can approximately be reconstructed by storing a small collection of weights for each face and a small set of standard pictures [3]. What follows is a description of the EigenFace Facial Recognition approach according to Turk and Pentland.

EigenFace Face Recognition Algorithm

Initialization operations:

1. Obtain an initial set of images (training set)
2. Calculate the eigenfaces from the training set. Keep the N- images, which define the face space that corresponds to the highest eigenvalues
3. Calculate the corresponding distribution in N- dimensional weight space for each known individual by projecting their face images onto the face space.

Having initialized the system; the following steps are used to recognize new face images:

1. Calculate a set of weights based on the input image and the N- eigenfaces by projecting the input image onto each of the eigenfaces
2. Determine whether the image is a face, by checking to see if the image is close to “face space”
3. If it is a face, classify the weight pattern as either a known person or unknown person
4. Update the eigenfaces and/or weight patterns
5. If the same unknown face is seen several times, calculate its characteristics weight pattern and incorporate into the known faces

EigenFace Facial Expression Recognition Approach

The functionality of the EigenFace Expression approach is similar to EigenFace for facial recognition, except it identifies an expression as opposed to a face of the individual. This algorithm was developed by Md. Iftekhar Tanveer and motivated by the Turk and Pentland Facial Recognition approach [5]. The purpose of this algorithm is to project facial expression images onto a feature space that spans the significant distinctions of known facial expression images as opposed to only corresponding to specific facial features such as eyes, nose, and mouth [3]. This approach characterizes an individual face by the weighted sum of eigenvectors or principal components of the particular face. In order to recognize a particular facial expression it is necessary to compare the weights to a known facial expression [3]. Below is Md. Iftekhar Tanveer’s EigenFace Expression Recognition Algorithm [5].

EigenFace Facial Expression Recognition Algorithm

1. Acquire an initial set of facial expression images (the training set)
2. The train images are utilized to create a low dimensional face space. This face space along with projected versions of all training images is created by performing Principal Component Analysis to produce principal components.
3. The test set of images are projected onto a face space. All test images are represented in terms of the selected principal components
4. The Euclidean distance of a projected test image from all the projected training images are calculated and the minimum value is chosen in order to find out the training image which is most similar to the test image. The test image is assumed to fall in the same class that the closest train image belongs to.
5. In order to determine the intensity of a particular expression the Euclidean distance from the mean projected neutral images are calculated. The farther the distance is from the neutral expression, the stronger the expression.

CHAPTER 3

TRADITIONAL CHINESE MEDICINE

Traditional Chinese Medicine (TCM) is an ancient system of health and healing that embodies the laws and patterns of nature (harmony and balance) and relates these concepts to the human body to employ moderation and prevention [16], [14]. TCM considers every aspect of a person: mind, body, spirit and emotions as a complete circle in terms of treatment rather than taking care of each component individually.

TCM practitioners use herbs, acupuncture (the insertion of thin needles into the skin to remove blockages in the flow of *chi* and maintain health), moxibustion (the use of heat from burning the herb moxa at the point acupuncture to stimulate the flow of *chi* and restore health), mind-body therapy, massage, exercise and dietary therapy [14] to treat patients. It focuses on stimulating the body's natural curative abilities.

3.1 Five Element Theory

The five element theory (five phase theory) embarks upon the notion of harmony and balance. This Theory states that everything in the universe including ones health is governed by five elements: earth, wood, fire, water and metal. These elements describe the five essentials of *chi* (life force energy), that are used to improve, manage or balance a space depending on the particular needs [17]. Each of the five elements has the ability to shift ones energy or the energy of one's space with corresponding components such as taste and emotion. Based on the particular emotion, which is the diagnosis portion, the taste that corresponds to the element is then controlled by the controlling cycle and the taste of the stronger element represents the treatment for that emotion in Table 3.1.

The earth element focuses on the act of settling down and being receptive and is excellent when things are disordered and need more stability, particularly when one is dealing with issues of assurance [17]. The taste and emotional component of the earth element is sweetness and thought, often referred to as worry [18]. The wood element focuses on the process of easy growth as well as progress that is connected with new beginnings and assists in level improvement [17]. The emotional and taste attributes of this element are anger (rage) and sourness [18].The focal point of fire element is excitement, expansion and quickness. This element increases heat and is useful when one needs to increase passion or intensity in a particular area in life [17]. The tastes and emotional state of the fire element are happiness and bitterness [18].The water element is centered on flow and connection. This element is very vital in terms of getting out of a "sticky" situation and improving communication [17]. Fear and salty are the corresponding emotion and taste with the water element [18]. The metal element is associated with focus, reduction and sharpness. Metal helps in the area of concentration and clearing one's head to handle detail [17]. Sadness and pungent are the corresponding emotion and taste for the metal element [18].

Table 3.1 Five Element Theory Elements with Corresponding Emotion and Taste

Element	Emotion	Taste
Earth	Worry	Sweet
Wood	Anger	Sour
Fire	Happiness	Bitter
Water	Fear	Salty
Metal	Sadness	Pungent

3.2 The Controlling Cycle

When an element is extremely strong, a stronger element is needed to bring the initial element back into balance, which is the purpose of the controlling cycle shown in Figure 3.1. The controlling cycle is implemented as follows: metal breaks down wood, wood breaks up the earth, earth blocks water, fire softens metal and water puts out fire [17]. This cycle is used to determine which emotion corresponding to each element is needed to change the emotion of an individual through the treatment of taste to bring the individual into balance. Based on the controlling cycle: the Pungent taste decreases the anger emotion, as the Sour taste decrease the worry (disgust) emotion. The Sweet taste decreases the fear (disgust) emotion, as the Salty taste decreases the happy emotion, and the Bitter taste decrease the sad emotion [17].

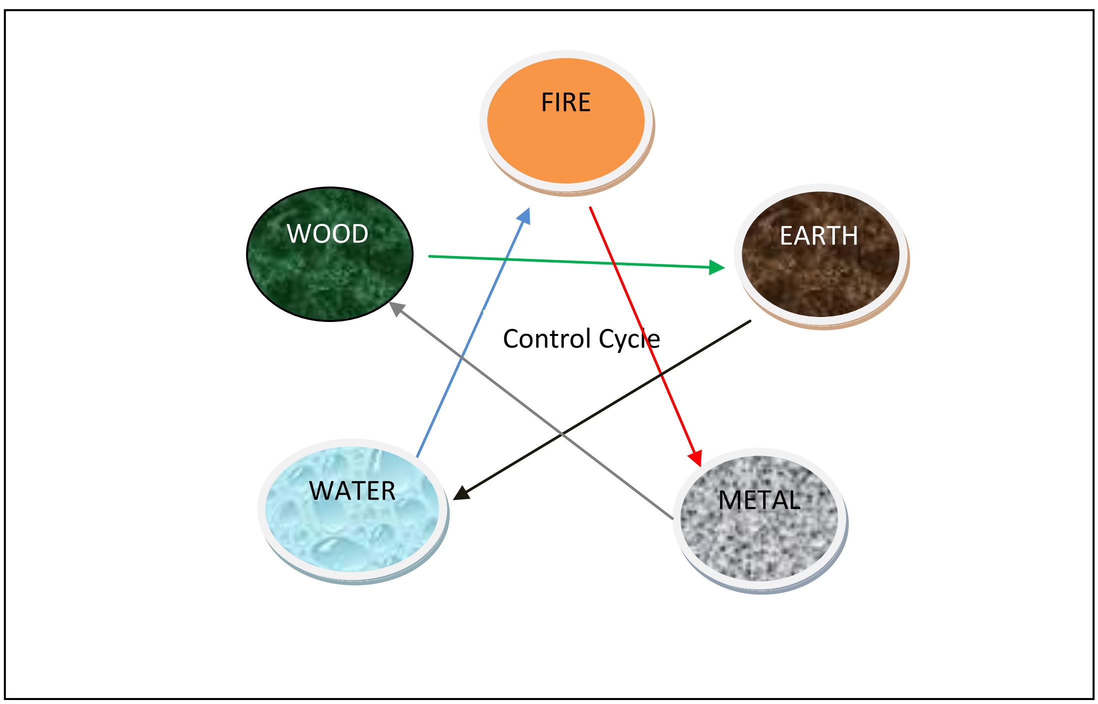

Figure 3.1.The Controlling Cycle

Each arrow that is pointing in the direction of the element represents the element that the element without the arrow controls. For example, the fire element controls the metal element; the earth element controls the water element, the metal element controls the wood element and the water element controls the fire element.

CHAPTER 4

METHODOLOGY

The Emotional Health Monitoring system consists of three major components that contributed to the overall methodology of this research. The first component is the Expression Recognition portion which used the EigenFace Expression recognition algorithm. The next phase is the Five Element Theory of Traditional Chinese Medicine, which creates a controlling cycle that controls and improves the emotion that corresponds to one of the five elements through the taste or strength of the stronger element that creates balance. The last component of this system is the southeastern regional food database. This database provides the user of the system with a list of healthy seasonal snacks geared to the suggested taste that can improve their mood based on the controlling cycle of TCM. All three components combine into a system shown in Figure 4.1 that retrieves the emotion of the user through facial expression recognition and determines which element is best to control or decrease that emotion through the controlling cycle of TCM. This system then suggests a healthy, in- season, snack to the user based on the southeastern regional seasonal database that will change the mood of the individual.

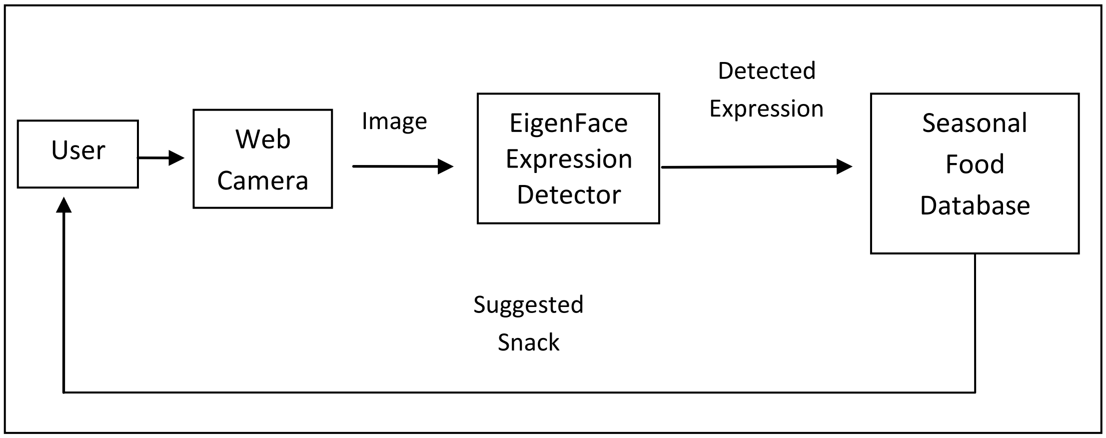

Figure 4.1: Block Diagram of Emotional Health Monitoring System

4.1 EigenFace Expression Recognition Algorithm

The EigenFace Expression Recognition algorithm developed by Md. Iftekhar Tanveer [5] was the emotion detection method used in the Emotional Health Monitoring System to classify five emotions: happiness, anger, sadness, neutral and disgust. This system was implemented using Matlab software. The absence of an expression represents the neutral expression and the other expressions were determined based on its intensity, which implies the Euclidean distance between the emotion and the neutral image. Tanveer stated that the greater the intensity value from the neutral expression classifies that particular expression as different emotion [5]. This algorithm used the EigenFace approach introduced by Turk and Pentland [3], which simply uses Principal Component Analysis (PCA) to extract the features of the face to classify every emotion.

The system is setup to have two databases of Images, Training and Testing Images. Training Images are used to train each emotion that is considered for detection. Tanveer recommends a training set of 50 images that represents10 images of each expression compared with a Testing Set of 31 images. In order for the system to work accurately the testing images should not be a subset of the training images, but rather a set of its own. Also it is imperative that a neutral image is represented in the training set, if not the algorithm would stop working. The testing images are used as inputs to the system to verify each expression compared with the train images.

Then a Label File is created or presented which simply states the emotion that corresponds to every image in the training set database. This text file gives words to the training images, which allows the system to know which expression each image represent. Once the training set, testing set and Label file are created, the user is then able to execute the software. As mentioned previously, this algorithm uses the PCA method to extract the principal components of the eigenvectors of the important images. It then reconstructs the training and test images onto a face space and creates eigenface images, which are ghost like images as shown in Figure 4.2 and Figure 4.3.

Figure 4.2 EigenFace Female Images

Figure 4.3. EigenFace Male Images

Next, the system calculates the Euclidean distance of the training images and the testing images from the mean of neutral images, which provides the intensity of the image tested from the mean of the neutral image. After the calculation of the intensity, the Euclidean distance is calculated between the other images in the training set that are not neutral. The algorithm then take note of the minimum distance, obtain the position of the minimum distance and then return the expression detected for the image based on its position.

This system was chosen to be the expression recognition method because of the availability of the developer to allow users to use and modify code [5]. It also provided the system to have a recognition accuracy rate of 70.9% before implementing it into the research [5]. The next component used and implemented into the overall setup was the Five Element Theory of the Chinese Traditional Medicine [17]. The picture depiction of this algorithm is located in Figure 4.4.

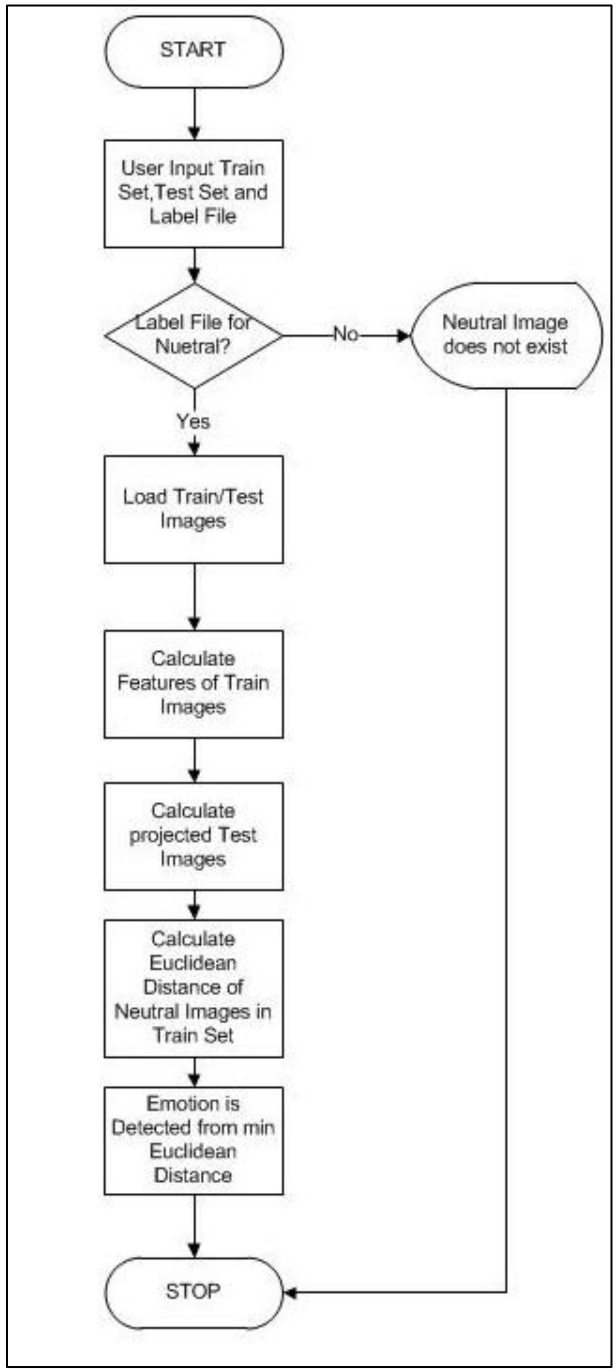

Figure 4.4 Flowchart of EigenFace Expression Recognition

4.2 Five Element Theory of Traditional Chinese Medicine

The Five element theory of the Traditional Chinese Medicine (TCM) simply states that the universe and overall health are based on five elements: earth, fire, wood, water, and metal. Each of these elements corresponds to a food taste and emotion [17]. According to TCM theory, in order to totally change and create balance for a particular element the controlling cycle steps in to do such [17]. The controlling cycle simply uses a stronger element to change the condition of the strong element that needs balance [17].

This cycle is relevant to this research, such that it provides balance to an individual's emotion by using the taste of the controlling element that would create balance for the individual and change their emotion [17]. For example, TCM theory states that the Pungent (spicy) taste decreases the anger emotion; the Sour taste decrease the worry (disgust) emotion; the Sweet taste decrease the fear (disgust) emotion; the Salty taste decrease the happy emotion; and the Bitter taste decrease the sad emotion [16].

4.3 Southeastern Regional Food Database

The Southeastern regional food database is a database composed of food according to the different seasons for the south east region of the United States. This reason was chose because programmatically the seasons can be determined according to the current date of the user. Matlab has a function that extracts the current month and day from the time stamp of the user's computer. Each season has five files, each which corresponds to tastes, pungent, sour, sweet, salty and bitter. This database is implemented in directories that are accessed from the EHMS system created in Matlab.

4.4 Hardware Setup for Emotional Health Monitoring System (EHMS)

The hardware setup of the emotional health monitoring system consists of a desktop computer, with a flat panel display and an 8 mega pixel c905 Logitech portable web camera Figure 4.5. The functionality of the system is through a graphical user interface (GUI) created in Matlab software presented in Appendix A. The face detection algorithm code is provided in Appendix B. This GUI allows the system to be user friendly, which heightens the overall experience of the emotional health monitoring system Figure 4.6 (a-c). Figure 4.6a displays the GUI of the system before the EHMS system is executed. Figure 4.6b displays the EHMS system after execution. It shows the picture taken by the user, the expression detected, and the button used to start the program. Figure 4.6c shows the message box with the list of suggested healthy snacks.

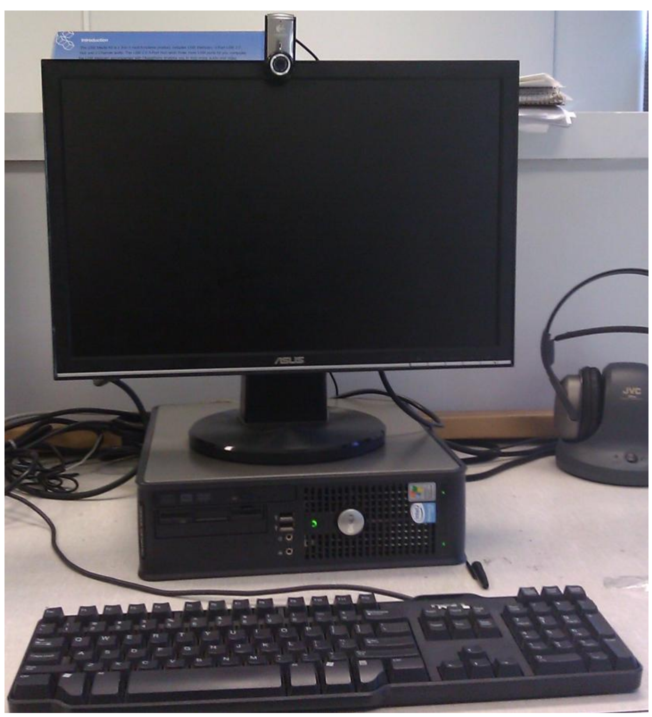

Figure 4.5 Experimental Setup of Emotional Health Monitoring System

4.4.1 User Interface to EHMS

Training Procedure

The EHMS GUI requires interaction and input from the user. The user must first have a testing and training set of images labeled “TrainImages” and “TestImages”. Next the user must create a Label File, which is a text document that consists of a list of the image filenames in the training set along with the corresponding word describing the emotion. This provides the system with the five emotions: happy, sad, anger, disgust and neutral as well as the images that depict these emotions. There is a textbox named “Label File”, which needs the user to key in the name of the Label File text document, with the “.txt” extension. For example: “LabelFile.txt” would need to be entered into the textbox shown in Figure 4.6. Failure to include the Label File name or a Label File name without the extension will cause an error and the GUI will not work.

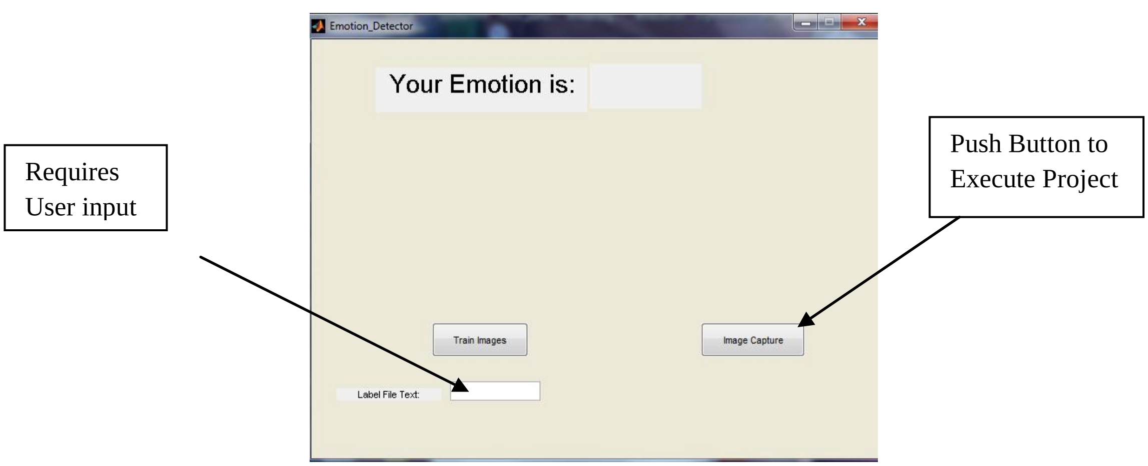

Figure 4.6 Graphical User Interface of Emotional Health Monitoring System

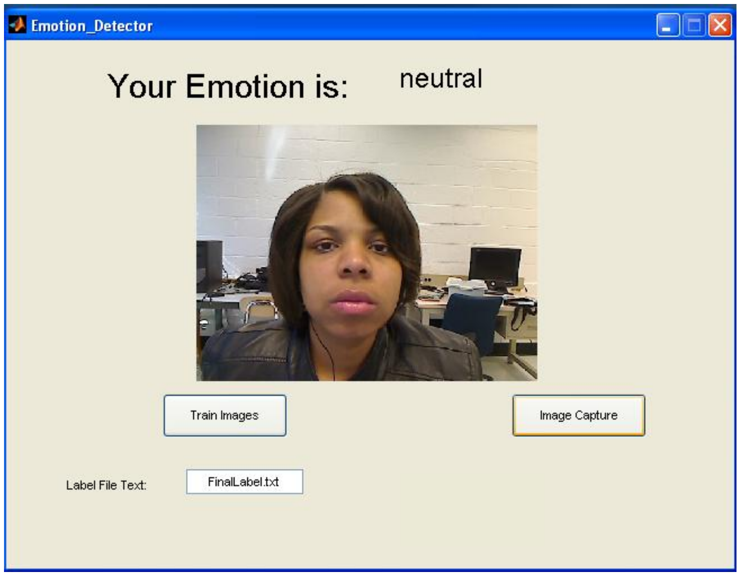

Figure 4.7 Graphical User Interface of Emotional Health Monitoring System

Figure 4.8 Healthy Suggestions Message Box of Emotional Health Monitoring System

Normal User procedure

The next interaction required by the user is to press the Image Capture button which will snap a photo of the user and determine the emotion. The user has the option to use the button to snap the photo or allow the program to automatically snap photos of the user throughout the day to give an average of the mood of the user over a specified time period. The picture taken by the user is then displayed in the GUI with the emotion detected. A pop- up box of suggested healthy snacks are displayed to create balance in user by improving his/her mood Figure 4.7, 4.8. Figure 4.9 displays the flowchart of the EHMS system.

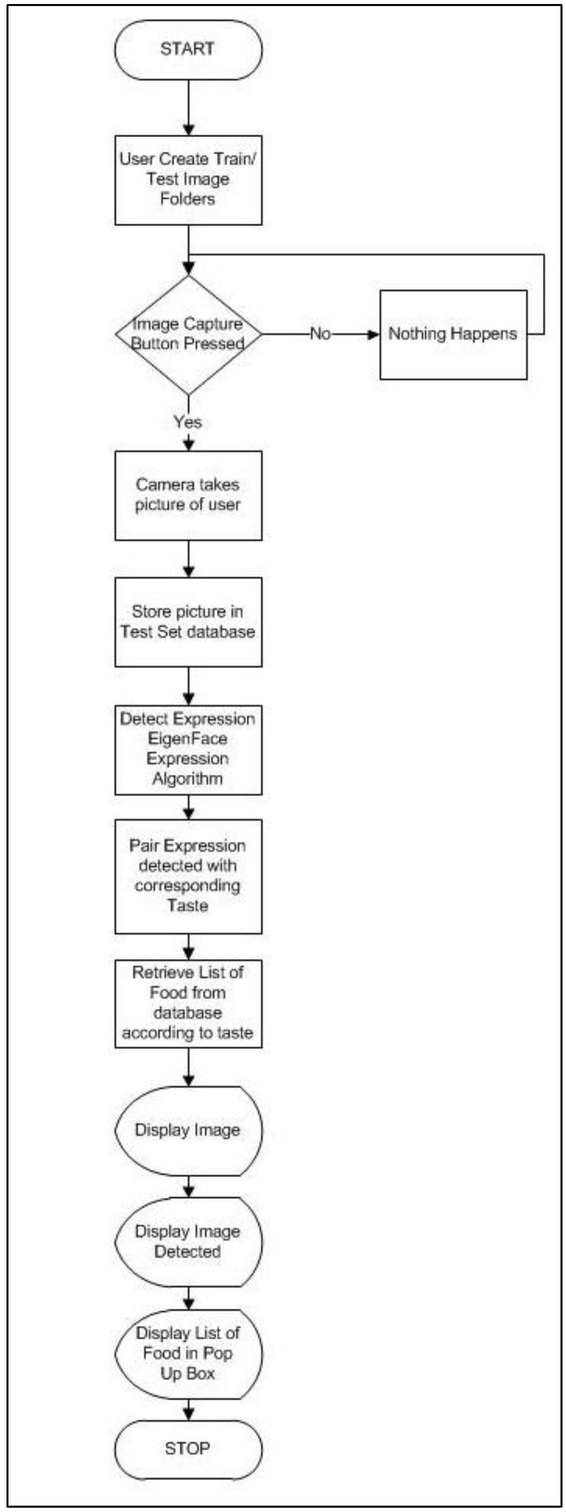

Figure 4.9 Flow Chart of Emotional Health Monitoring System

4.4.2 Experimental Procedure

Variation in Training Set

The initial experiments conducted for this research were to test the EigenFace Expression Recognition algorithm. This software provided sample images of 50 Training and 31 Testing, which allowed the user to understand the functionality of the algorithm. The initial tests consisted of varying the number of training images to retrieve accurate results in the functionality of the algorithm. It was understood that the more images used in training; the more accurate the result of the emotion is determined. The objective of this set of experiments was to find the minimal of training images needed to still achieve the highest accuracy.

Variation in Testing

The second set of experiments investigated the number of test images required to accurately detect the correct emotion. The goal of this set of experiments was to see at what least amount of test images will provide accurate results. The decreasing ranges of test images were 5-1. The number 5 was selected based on the number of emotions tested: happy, anger, sad, disgust, and anger. The results of these experiments are located in the following section of this paper.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Variation in Training Set

This variation in training images consisted of these sets: 50, 40, 30, 20, 15 and 10. The results for the training sets of 50 down to 10 are located in Table 5.1. Figure 5.1, displays the graphical representation of the data in the Table 5.1. The table and graph display that as the numbers of training images decrease the accuracy of the system decrease. This data was tested on 31 test images. The dataset used in this experiment were provided with the software for the EigenFace Expression Recognition, by Md. Iftekhar Tanveer [5]. The overall accuracy of this system is 70.9% with 20 or more training images and 35 % below 20 train images. The training image set selected for use in the overall system was the contained 20 images because it was the least number of training images that had the overall percent accuracy of 70.9.

Table 5.1 Train Set and Percent Accuracy of System with 31 Test Images

Train Set Images	Percent Accuracy
10	35.5
15	35.5
20	70.9
30	70.9
40	70.9
50	70.9

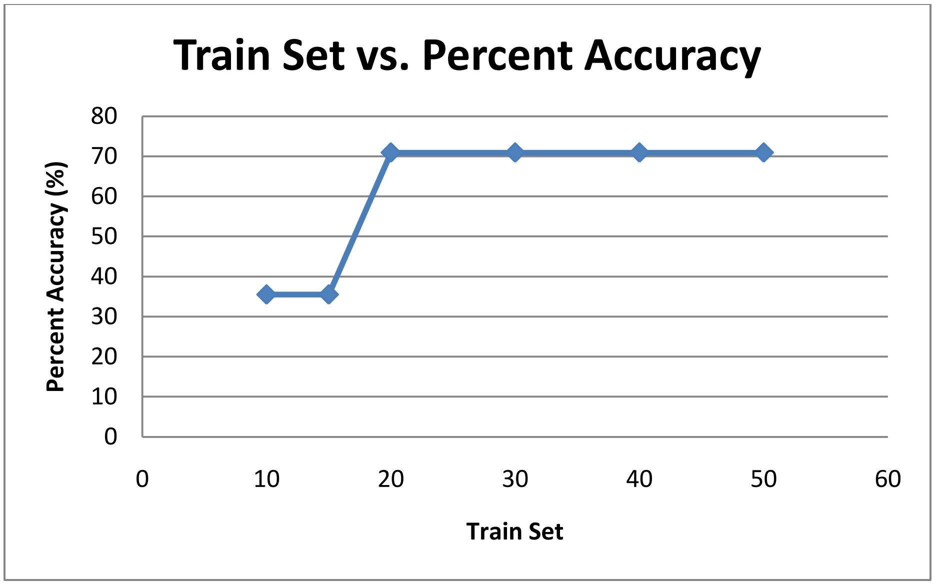

Figure 5.1 Train Set vs. Percent Accuracy

5.2 Variation in Testing Set

The variation in testing set of experiments tested the accuracy of training images 30, 20, 15 and 10 against tests sets of 1-5 images. The image database used in these experiments was of the user using the LG web camera. The overall objective of these experiments were to determine what number of training images are needed to have a good accuracy rate with a small amount of test images. The test images range from 1 to 5 as the overall goal is to determine the emotion for 1 image; as the user takes a picture and there was not a need to use multiple images for testing to have great accuracy. The results for these experiments are shown in Figures 5.2 – 5.5.

Figure 5.2 displays the accuracy of 1-5 test images with a training set of 30. The 5 images in the testing set represented each emotion: happy, sad, anger, disgust and neutral. A picture was then subtracted from the testing set one at a time to determine how accurate the system would classify the emotions in the testing set as the correct emotion. The results show that as the number of test image decreases, the accuracy percentage also decreases. With a collection of 5 and 4 test images (i.e. emotions) the system classified each emotion correctly. At 3 test images, the system's accuracy dropped to 75 % and then to 0% for 1 and 2 test images.

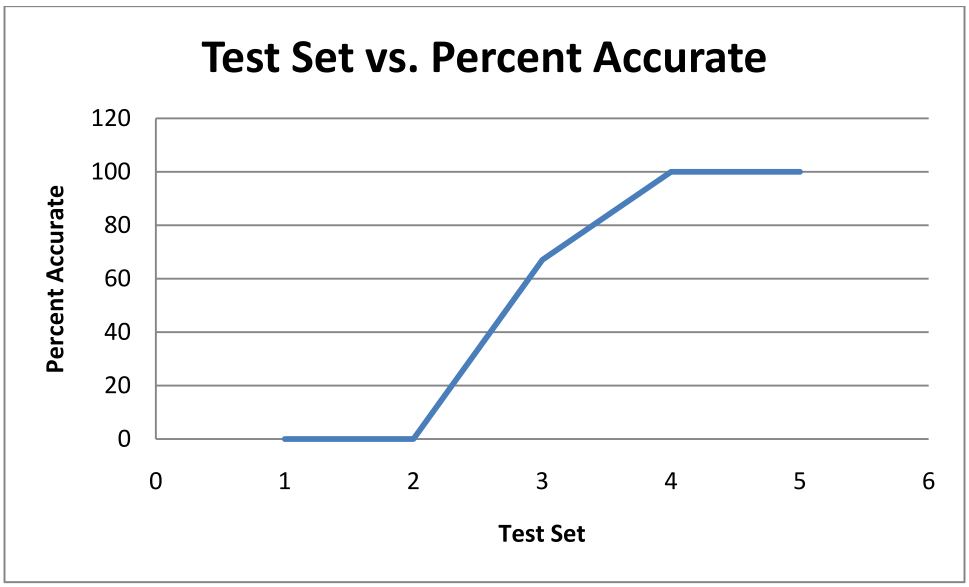

Figure 5.2 30 Train Images, Test Set vs. Percent Accurate

Figure 5.3 displays the results for a training set of 20 images. The results slightly differ from those of the 30 image training set. The difference between the two are that the results for 4 images are at an accuracy of 75 % and for 3 images are at 0%. The results for 5, 2, and 1 test images remain consistent. Figure 5.4 displays the results for the training set of 15 images. These results are similar to those of the 20 image training set.

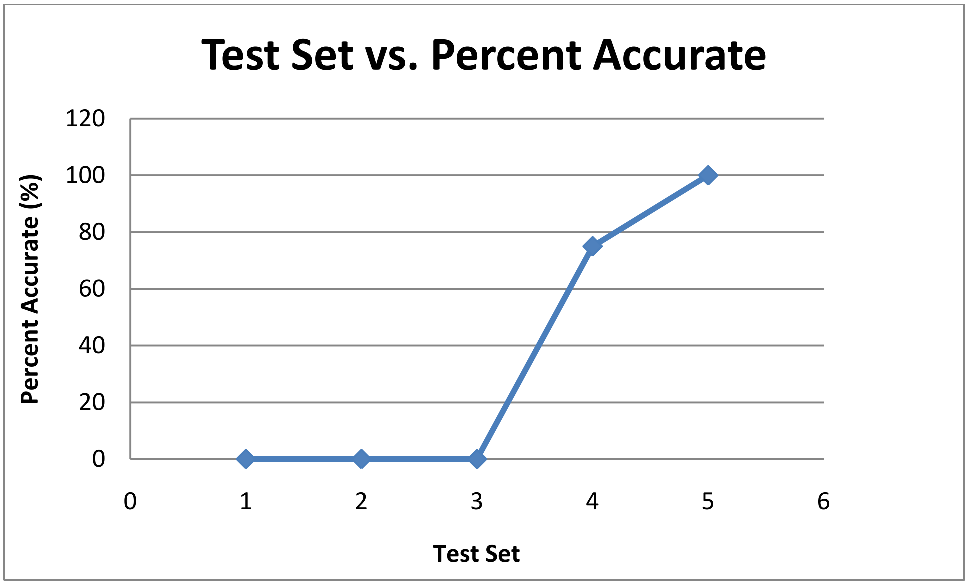

Figure 5.3 20 Train Images, Test Set vs. Percent Accurate

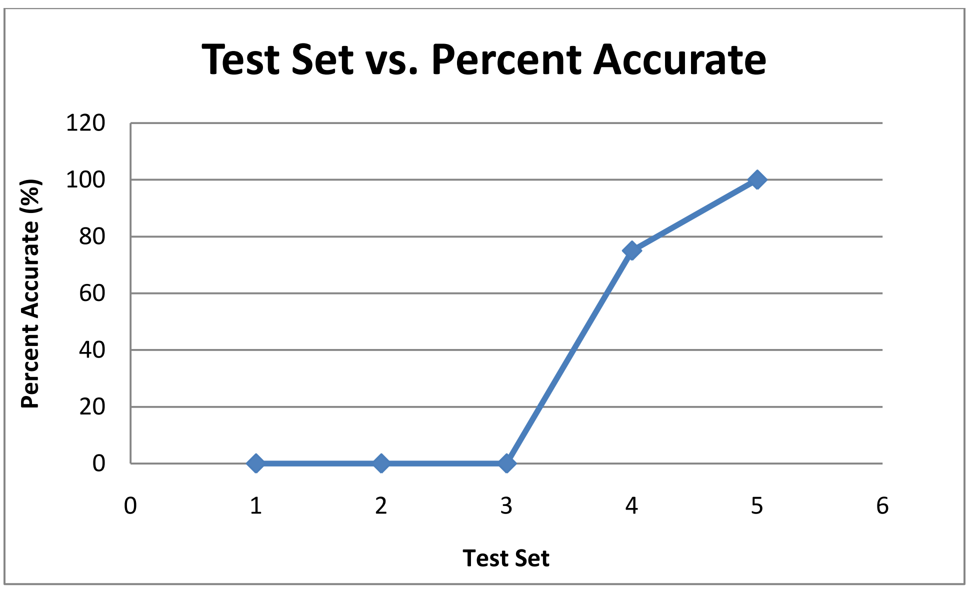

Figure 5.4 15 Train Images, Test Set vs. Percent Accurate

Figure 5.5 displays the results of 10 train images. These results differ from the others such that the accuracy for the 5 test images was 60 % as opposed to the others who had an accuracy of 100%. The accuracy of 4 test images had an accuracy of 100%. Test images of 3, 2 and 1 had 0% accuracy.

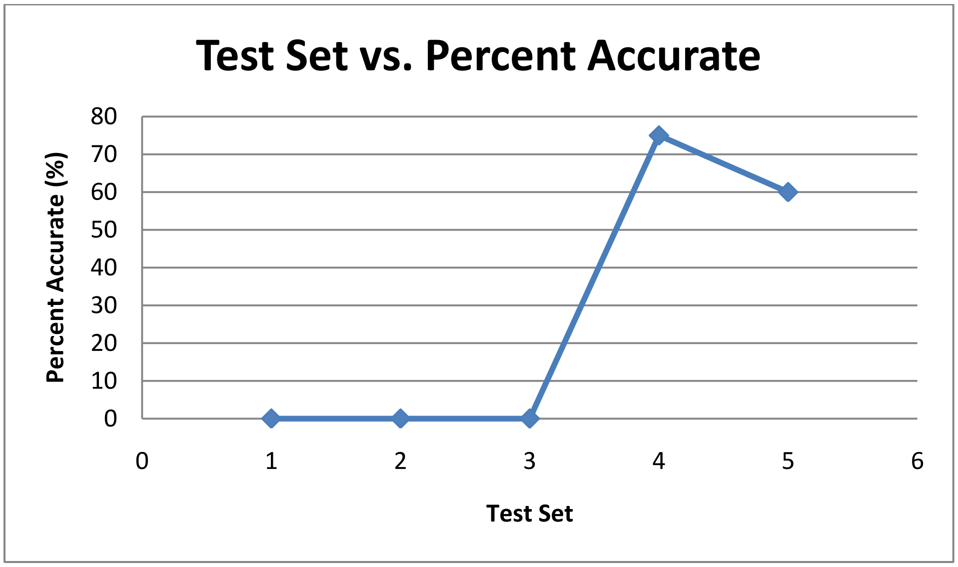

Figure 5.5 10 Train Images, Test Set vs. Percent Accurate

These results show that in order to properly classify the expression correctly, there needs to be a minimum of 5 images in the testing set, one for each emotion. The initial decision to fix this problem was to copy the same image five times to classify the images properly. Unfortunately this solution did not provide accuracy to the system; it in return classified each picture as the same incorrect expression. The next solution to this problem was to use 4 bias images as an offset, in the testing set, that each represented a different expression besides neutral. It was understood that different expressions needed to be present and "feed off" each other to accurately classify the expressions in the testing set. After testing this theory, it was proven that this was the best solution to the problem.

5.3 Accuracy of Each Expression

The accuracy of each expression was determined based on a series of 10 tests. A picture of each expression was randomly taken using the Emotional Health Monitoring system setup. The results of each expression were determined by10 trials of each expression. The order in which expressions were taken was random and not in a sequential order for each emotion. Table 5.2 displays the accuracy of each emotion. The anger emotion had the highest accuracy rate of 80 %, as the neutral expression was at 70%, followed by happy and disgust at 60 % and ranking last was sad at 50 % accuracy. According to Tanveer [5] this is due to the intensity of the emotion. The more intense an emotion or expression is, the easier it is to classify the expression as its own set. The sad emotion was easily confused with the neutral and disgust expression. Next, the overall accuracy of the system was calculated based on the average of the accuracies of the emotions combined. Based on the results of the accuracy of the five emotions; one could conclude that the overall accuracy of the system is approximately 64%. Figures 5.6 -5.10 display every expression using the emotional health monitoring system. The GUI displays the determined emotion, the image of the user, and the message box of the list of seasonal foods to improve the mood of the user.

Table 5.2 Accuracy of Emotion

Emotion	Percent Accuracy (%)
Anger	80
Neutral	70
Happy	60
Disgust	60
Sad	50

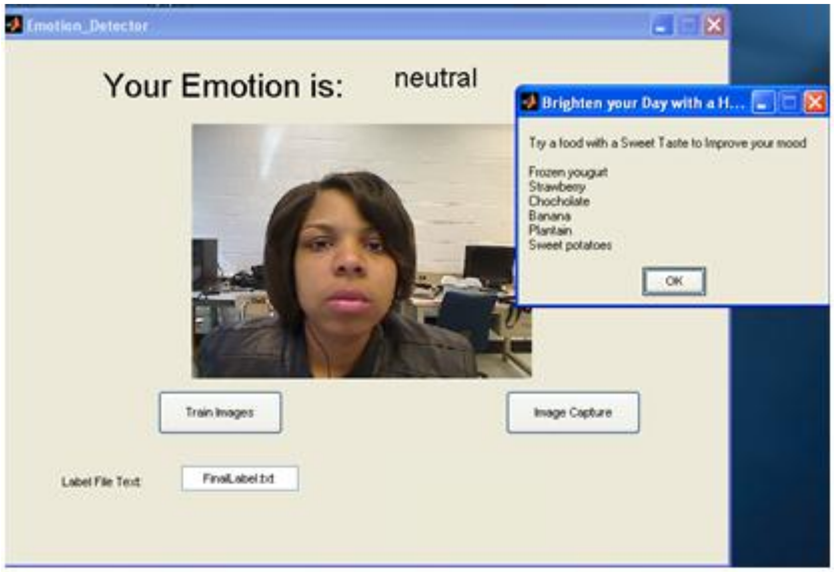

Figure 5.6 The Neutral Expression Identified

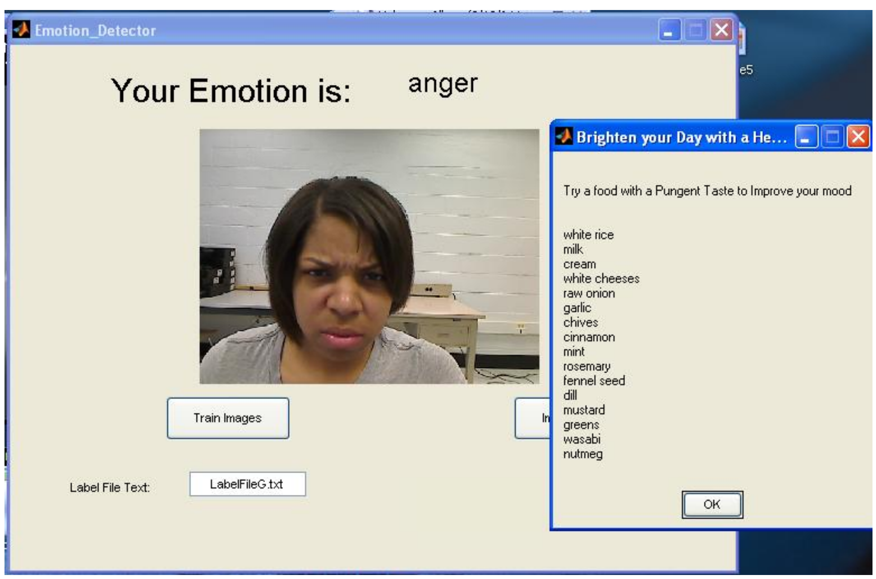

Figure 5.7 The Anger Expression Identified

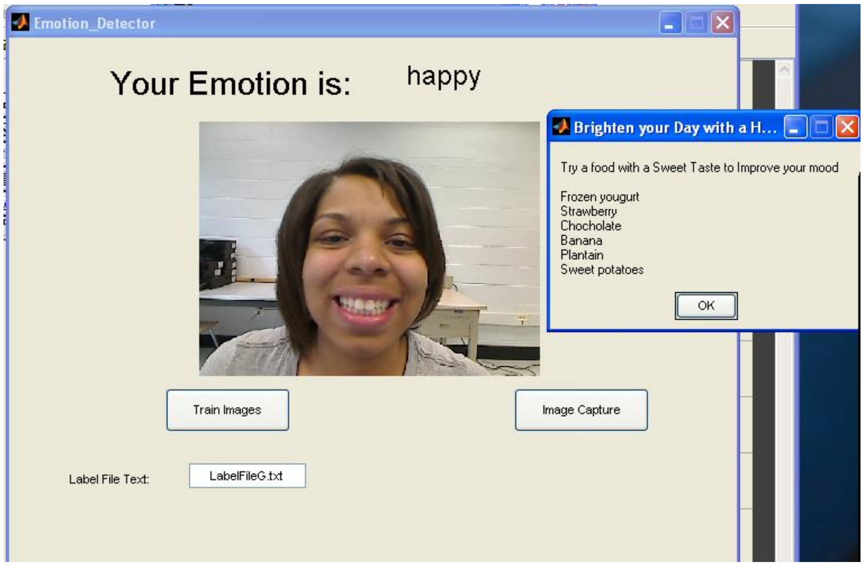

Figure 5.8 The Happy Expression Identified

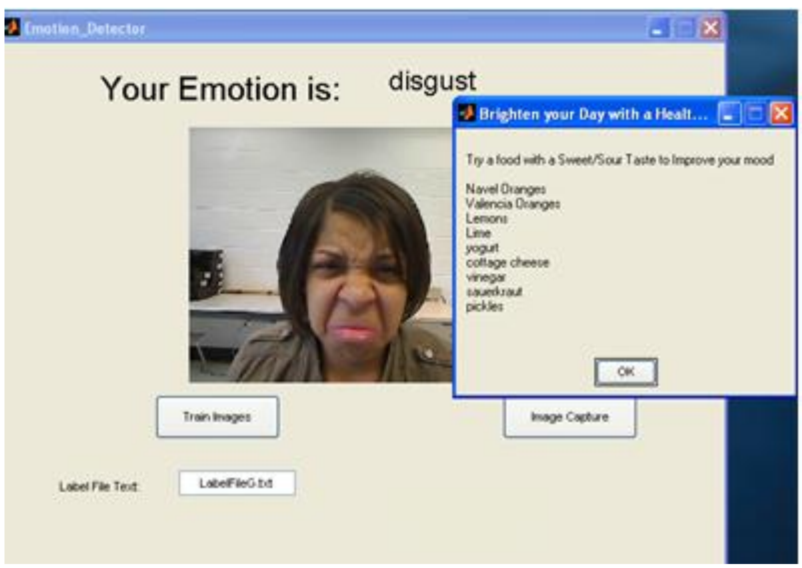

Figure 5.9 The Disgust Expression Identified

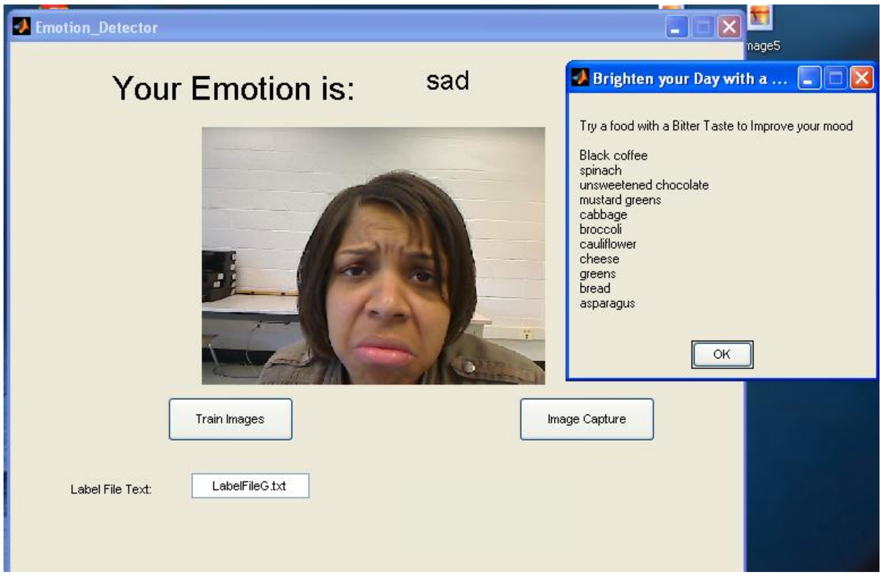

Figure 5.10 The Sad Expression Identified

As mentioned above Figures 5.6 – 5.10 display each of the five emotions tested within this system. The figures also display the message box which suggests a list of healthy snacks based on taste to help improve the mood of the user.

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

In conclusion, the objective of this research was to create a system that estimates the emotion of an individual to improve the emotional state and overall nutritional well being of an individual with the development of a near-real time emotional health monitoring system. The system estimates the emotion of an individual through the EigenFace Expression Recognition approach developed by Md. Iftekhar Tanveer [5], and then suggests a list of healthy snacks based on TCM Theory, and local in season food. This research studied five facial expressions: happiness, anger, sadness, neutral, and disgust, which in return expressed the emotional state of the person.

This system used 20 images to train, 4 of each expression that classified the expressions into its five categories: happiness, anger, sadness, neutral and disgust. The system is then tested with the image taken by the user to classify the expression correctly. The testing set includes 4 bias images of the user, which are of different, non neutral expressions. This method allows the expression of the image taken through the system to be higher. The accuracy of the overall system is 64% which is based on the accuracies of each emotion. The anger emotion had accuracy 0f 80 % and the sad emotion had an accuracy of 50 %. This system functions at 100% accuracy in linking the expression detected with the proper list of healthy snacks in season, which prompts users to improve their mood with a healthy seasonal snack. Numerous factors can play into the accuracy rate, such as noise due to the background in which the image is taken against; also the resolution of the camera can affect the accuracy of the classifications of expressions. Another factor is the facial expression intensity of images in the training set. It is imperative that each emotion expressed in the training set has a different intensity range compared with the other four expressions. This is an important concept because the EigenFace expression recognition approach [5] classifies each emotion based on its intensity level.

6.2 Recommendations

The experimental recommendations for this research are to change the algorithm into user independency, which means that the expressions are not directed to one particular person, but has the ability to determine any person's emotion in the database. Another recommendation is to test the difference when the user take pictures while looking directly into the camera and when they are simply looking at their computer. Also the database of healthy seasonal foods could link to a website of seasonal foods and pull data from the site or simply create a database for each region dependent on the user. The last recommendation for this research is to combine the PCA method with other methods to increase the accuracy of the Facial Expression Recognition Algorithm.

REFERENCES

[1] Balensi, Joseph (2008). Five Elements. *chinesemedicinesampler.com*. Retrieved November21, 2011 from http://www.chinesemedicinesampler.com/theory5elements.html

[2] Sarfati, Debbie (2002, September). The Food-Mood Connection: How Your Diet Effects Your Physical And Emotional State. *Women's Magazine*. Retrieved November 10, 2011 from http://www.wholenourishment.com/article5.htm

[3] Turk,Mattew, Pentland, Alex. (1991). Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*. Volume 3, No. 1

[4] Smith, I. Lindsay (February 2002). *A tutorial on Principal Component Analysis.* Retrieved February 28, 2012 from http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

[5] Tanveer, Md. Iftekhar . (2008) *Eigenfaces for Expression Detection.* Eigenface based Facial Expression Recognition System. Retrieved from http://www.mathworks.com/matlabcentral/fileexchange/33325-eigenface-based-facial-expression-classification/content/EigenFace.m

[6] Huang , Ming-Wei, Wang, Zhe-wei， Ying, Zi-Lu. *A New Method for Facial Expression Recognition Based On Sparse Representation Plus LBP*. Unpublished paper presented at the 2010 3rd International Congress on Image and Signal Processing

[7] Ojala, T., Pietikainen, M. and Harwood, D. *A comparative study of texture measures with classification based on feature distributions*, J. Pattern Recognition vol. 29, No.1 pp. 51-59, 1996.

[8] Zhao, Quan-You, Bao-Chang, Pan, Jian-Jia, Tang, Yuan-Yan. (2008). *Facial Expression Recognition Based On Fusion of Gabor and LBP Features*. Unpublished paper presented at the 2008 International Conference on Wavelet Analysis and Pattern Recognition, Hong Kong

[9] *Face Recognition*. (n.d.). Retrieved March 10, 2012 from National Science and Technology Council Committee on Technology Committee on Homeland and National Security Subcommittee on Biometrics Web site: http://www.biometrics.gov/Documents/facerec.pdf

[10] Etemad, Kamran and Chellappa, Rama (1997). *Discriminant analysis for recognition of human face images*. Optical Society of America. vol 14, No 8

[11] Cao, Jianqiang and Tong, Can. *Facial expression recognition based on LBP-EHMM*. Unpublished paper presented at 2008 Congress on Image and Signal Processing

[12] Li, He, Buenaposad, José M., Baume, Luis. (2008).IEEE. *Real-Time Facial Expression Recognition with Illumination-Corrected Image Sequences*

[13] Belhumeur, Peter N., Hespanha, Joao P., and Kriegma David J. (1997). IEEE Transaction on Pattern Analysis and Machine Intelligence. *Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection*. volume, No. 7

[14] *What is Traditional Chinese Medicine?* (2005-2012).Traditional Chinese Medicine World Foundation. Retrieved March 10, 2012 from http://www.tcmworld.org/what_is_tcm/

[15] *Traditional Chinese Medicine: An Introduction*. National Institutes of Health. Retrieved March 5, 2012 from http://nccam.nih.gov/health/whatiscam/chinesemed.htm

[16] *Traditional Chinese Medicine.* Retrieved March 1, 2012 from http://medical-dictionary.thefreedictionary.com/traditional+Chinese+medicine

[17] Roberts, Stephanie (2002). *A Quick Guide to the 5 Elements*. Kahului, HI. Lotus Pond Press

[18] Parkinson, Rhonda. *The Five Elements Theory of Chinese Cooking: Learn about the Role of the Five Elements in Chinese Cooking.* Retrieved January 17, 2012 from http://chinesefood.about.com/library/weekly/aa041900a.htm

[19] Watson, Molly. *Southern Seasonal Fruits & Vegetables. What's in Season for the South?* Retrieved March 12, 2012 from http://localfoods.about.com/od/searchbyregion/a/SouthFruitVeg.htm

[20] Adams, Chris. *Ergo 101 - What is Ergonomics. What is Ergonomics?* Retrieved March 25, 2012 from http://ergonomics.about.com/od/ergonomicbasics/a/ergo101.htm

[21] WISHA Services Division. *Office Ergonomics. Practical solutions for safer workplace.* Retrieved from http://www.lni.wa.gov/IPUB/417-133-000.pdf

APPENDIX A

EMOTIONAL HEALTH MONITORING SYSTEM CODE GUI

```
function varargout = Emotion_Detector(varargin)
% EMOTION_DETECTOR M-file for Emotion_Detector.fig
%      EMOTION_DETECTOR, by itself, creates a new EMOTION_DETECTOR or raises the existing
%      singleton*.
%
%      H = EMOTION_DETECTOR returns the handle to a new EMOTION_DETECTOR or the handle to
%      the existing singleton*.
%
%      EMOTION_DETECTOR('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in EMOTION_DETECTOR.M with the given input arguments.
%
%      EMOTION_DETECTOR('Property','Value',...) creates a new EMOTION_DETECTOR or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before Emotion_Detector_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to Emotion_Detector_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Emotion_Detector

% Last Modified by GUIDE v2.5 06-Mar-2012 13:23:29

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Emotion_Detector_OpeningFcn, ...
                   'gui_OutputFcn',  @Emotion_Detector_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT



% --- Executes just before Emotion_Detector is made visible.
```

```
function Emotion_Detector_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Emotion_Detector (see VARARGIN)


% Choose default command line output for Emotion_Detector
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Emotion_Detector wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Emotion_Detector_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)

vid = videoinput('winvideo',1);
%Saves image in current directory
cd TestI
for counter = 1:1
    imageSize=[600,800];
     img = getsnapshot(vid);% take a snapshot of the video
     img1=imresize(img,imageSize);
     axes(handles.axesImage)
     imshow(img1,[]);
      imshow(img1);
     fname = ['Image5' ,'.jpg'];%
     imwrite(img1,fname,'jpg');%saves image to f_Test
     disp('test Picture Taken')

end
cd ..


disp('This Program is written by Md. Iftekhar Tanveer (go2chayan@gmail.com)');
disp('Copyleft 2009');

isSucceed = 0;
```

```
if (exist('strTrainPath')==0)
   strTrainPath = 'TrainI'
   %input('Enter Train Folder Name:','s');
   %'Train18'
   %TrainImages


end
if (exist('strLabelFile')==0)
   strLabelFile = get(handles.edit3,'string')
   %get(handles.edit3,'string')

end
if (exist('strTestPath')==0)
   strTestPath = 'TestI'
   %input('Enter Test Folder Name:','s');
   %'Test18'
   %TestImages

end

fid=fopen(strLabelFile);
imageLabel=textscan(fid,'%s %s','whitespace',',');
fclose(fid);

NeutralImages=[];
for i=1:length(imageLabel{1,1})
   if (strcmp(lower(imageLabel{1,2}{i,1}),'neutral'))
      NeutralImages=[NeutralImages,i];
   end
end
if (length(NeutralImages)==0)
   disp('ERROR: Neutral Expression is not available in training');
   return;
end

structTestImages = dir(strTestPath);
numImage = length(imageLabel{1,1});  % Total Observations: Number of Images in training set
lenTest = length(structTestImages);

if (lenTest==0)
   disp('Error:Invalid Test Folder');
   return;
end

TrainImages='';
for i = 1:numImage
   TrainImages{i,1} = strcat(strTrainPath,'\',imageLabel{1,1}(i));
end

j=0;
for i = 3:lenTest
    if ((~structTestImages(i).isdir))
```

```
        if  (structTestImages(i).name(end-3:end)=='.jpg')
            j=j+1;
            TestImages{j,1} = [strTestPath,'\',structTestImages(i).name];
        end
    end
end
numTestImage = j; % Number of Test Images
clear ('structTestImages','fid','i','j');pack

imageSize = [280,180];         % All Images are resized into a common size

%% ################# Load Train Data & Preprocess  ########################
%% Loading training images & preparing for PCA by subtracting mean

%%Reads in image from training set
%%Creates a vector of zeros based on the size of the training images
img = zeros(imageSize(1)*imageSize(2),numImage);
for i = 1:numImage %number of images in train set
   %%resize the images of the training set
   aa = imresize(detect_face(imresize(imread(cell2mat(TrainImages{i,1})),[375,300])),imageSize);
   img(:,i) = aa(:);
   %%load the training images
   disp(sprintf('Loading Train Image # %d',i));
end

%Calculates the mean of the images in the training set by rows
meanImage = mean(img,2);

%Calculate the difference between the image and the meanImages of the
%training data
img = (img - meanImage*ones(1,numImage))';     % img is the input to PCA
%% #########################################################################

%% ################# Low Dimension Face Space Construction ################
%%C is the Coefficient
%%S is the score
%%L is the latent

%%Returns the latent that are not zero which are the eigenvalues in the
%%covariance matrix
[C,S,L]=princomp(img,'econ');                 % Performing PCA Here
EigenRange = [1:11];   % Defines which Eigenvalues will be selected based on Test Images atleast 2
images
C = C(:,EigenRange); %%?? All Eigen Vectors in C
%% #########################################################################


%% ############# Load Test Data and project on Face Space #################
%%create space for the Test images
img = zeros(imageSize(1)*imageSize(2),numTestImage);
for i = 1:numTestImage
   aa = imresize(detect_face(imresize(imread(TestImages{i,1}),[375,300])),imageSize);
   img(:,i) = aa(:);
```

```
    disp(sprintf('Loading Test Image # %d',i));
end
%Calculate the mean of the test images accorging to the rows in the vector
meanImage = mean(img,2);
%Calculates the difference between the face space vector and the test
%images
img = (img - meanImage*ones(1,numTestImage))';

%Projected test image
%Test image created
Projected_Test = img*C;
%% ###########################################################################
%% ################# Calculation of Distance from Neutral ##################
meanNutral = mean(S(NeutralImages,EigenRange)',2)
for Dat2Project = 1:numTestImage
    TestImage = Projected_Test(Dat2Project,:);
    % Picking the image #Dat2Project

    Eucl_Dist(Dat2Project) = sqrt((TestImage'-meanNutral)'*(TestImage' ...
        -meanNutral))
        % Here, the distance between the expression under test and
        % the mean neutral expressions is being calculated
end
%Eucl_Dist = Eucl_Dist/max(Eucl_Dist
%mds= Eucl_Dist - Other_Dist(:)
%% ###########################################################################

%% ################# Calculation of other Distances #######################
Other_Dist = zeros(numTestImage,numImage);
for Dat2Project = 1:numTestImage
    TestImage = Projected_Test(Dat2Project,:);
    % Picking the image #Dat2Project
    for i = 1:numImage
        Other_Dist(Dat2Project,i) = sqrt((TestImage'-S(i,EigenRange)')' ...
            *(TestImage'-S(i,EigenRange)'));

    end
end
[Min_Dist,Min_Dist_pos] = min(Other_Dist,[],2)

%% ###########################################################################



%% ########################## Display Result ###############################
fid = fopen('Results.txt','w');
fid1=fopen('EmotionResult.txt','w');

%fprintf(fid,'//Test Image,Distance From Neutral, Expression,Best Match\r\n');

for i = 1:numTestImage
    b = find(TestImages{i,1}=='\');
    Test_Image = TestImages{i,1}(b(end)+1:end);
    Dist_frm_Neutral = Eucl_Dist(i);
```

```
    Best_Match = cell2mat(imageLabel{1,1}(Min_Dist_pos(i)));
    Expr = cell2mat(imageLabel{1,2}(Min_Dist_pos(i)));
    fprintf(fid,'%s,%0.0f,%s,%s\r\n',Test_Image,Dist_frm_Neutral,Expr,Best_Match);

end

for i = numTestImage
    b = find(TestImages{i,1}=='\');
    Test_Image = TestImages{i,1}(b(end)+1:end);
    Dist_frm_Neutral = Eucl_Dist(i);
    Best_Match = cell2mat(imageLabel{1,1}(Min_Dist_pos(i)));
    Expr = cell2mat(imageLabel{1,2}(Min_Dist_pos(i)));
    fprintf(fid1,'%s\r\n',Expr);

end
fclose(fid);
fclose(fid1);
%% ###########################################################################
% isSucceed = 1;
% disp('Done')
% disp('Output File = .\Results.txt');
% disp('Output File = .\EmotionResults.txt');
% Willexit = input('Press Enter to Quit ...','s');
%Willexit = set(handles.edit4,'String','Recogntion Done');
% %Restrieve the month from the current date

date= datestr(now,'mmm');
Winter1 = strcmp(date,'Dec');
Winter2 = strcmp(date,'Jan');
Winter3 = strcmp(date,'Feb');

Spring1 = strcmp(date,'Mar');
Spring2 = strcmp(date,'Apr');
Spring3 = strcmp(date,'May');

Summer1 = strcmp(date,'Jun');
Summer2 = strcmp(date,'Jul');
Summer3 = strcmp(date,'Aug');

Fall1 = strcmp(date,'Sep');
Fall2 = strcmp(date,'Oct');
Fall3 = strcmp(date,'Nov');

if (Winter1 |Winter2| Winter3 == 1)
    Season = 'Winter';
end

if (Spring1 |Spring2| Spring3 == 1)

        Season = 'Spring';
end

if (Summer1 |Summer2| Summer3 == 1)
```

```
        Season='Summer';
end

if (Fall1 |Fall2| Fall3 == 1)
           Season='Fall';
end

if (exist('Result')==0)
    Emotion = textread('EmotionResult.txt' ,'%s')
end

 cd(Season)

 happy_Compare =strcmp(Emotion,'happy');
 sad_Compare =strcmp(Emotion,'sad');
 anger_Compare =strcmp(Emotion,'anger');
 disgust_Compare =strcmp(Emotion,'disgust');
 neutral_Compare =strcmp(Emotion,'neutral');

  set(handles.text5,'String',Emotion);

    if (happy_Compare == 1)
        fid1= textread('Sweet.txt' ,'%s', 'delimiter', '\n')
        Message= msgbox(fid1,'Brighten your Day with a Healthy Snack');
        disp(Message)
    end

    if (sad_Compare == 1)
        fid1= textread('Bitter.txt', '%s', 'delimiter', '\n')
        Message= msgbox(fid1,'Brighten your Day with a Healthy Snack');
        disp(Message)
    end

    if (anger_Compare == 1)
        fid1= textread('Pungent.txt', '%s', 'delimiter', '\n')
        Message= msgbox(fid1,'Brighten your Day with a Healthy Snack');
        disp(Message)
   end

   if (disgust_Compare == 1)
         fid1= textread('SweetSour.txt', '%s', 'delimiter', '\n')
         Message= msgbox(fid1,'Brighten your Day with a Healthy Snack');
         disp(Message)
   end

    if (neutral_Compare == 1)
         fid1= textread('Sweet.txt','%s', 'delimiter', '\n')
         Message= msgbox(fid1,'Brighten your Day with a Healthy Snack');
         disp(Message)
    end
cd ..
% % hObject    handle to pushbutton1 (see GCBO)
% % eventdata  reserved - to be defined in a future version of MATLAB
```

```
% % handles    structure with handles and user data (see GUIDATA)
% %Calls the EigenFace Function
% EigenFace
%


% --- Executes on button press in pushbutton2.
function pushbutton7_Callback(hObject, eventdata, handles)
%Creates Training Folder

% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
vid = videoinput('winvideo',1);
%Saves image in current directory
cd TrainImages

for counter = 1:24
   imageSize=[600,800];
    iTime = clock; % Retrieves the live clock
    img = getsnapshot(vid);% take a snapshot of the video
    img1=imresize(img,imageSize);
    elapsed = etime(clock, iTime);
    fname = ['Image' num2str(counter),'.jpg'];%names the image
    if (counter <= 20)
    figure, imshow(img1);
    disp('Train Image Picture Taken')
    imwrite(img1,fname,'jpg');%saves image to f_Test
    pause(10 - elapsed);%pause the amount of seconds between each image according to the counter
    else
        cd ..
        cd TestImages
    figure, imshow(img1);
    disp('Test Picture Taken')
    imwrite(img1,fname,'jpg');%saves image to f_Test
     end

end
cd ..

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function t_input_Callback(hObject, eventdata, handles)
% hObject    handle to t_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of t_input as text
%        str2double(get(hObject,'String')) returns contents of t_input as a double


% --- Executes during object creation, after setting all properties.
function t_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to t_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton6.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)




function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double


% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)



% --- Executes on button press in pushbutton10.
function pushbutton11_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%cd RandImage
vid = videoinput('winvideo',1);
for counter = 1:5
    imageSize=[600,800];
     iTime = clock; % Retrieves the live clock
     img = getsnapshot(vid);% take a snapshot of the video
     img1=imresize(img,imageSize);
     elapsed = etime(clock, iTime);
     fname = ['Image' num2str(counter),'.jpg'];%names the image
     figure, imshow(img1);
     disp('Train Image Picture Taken')
     imwrite(img1,fname,'jpg');%saves image to f_Test
     pause(15 - elapsed);%pause the amount of seconds between each image according to the counter

      %fname = ['Image5' ,'.jpg']
     copyfile(fname,'RandImage')
     movefile(fname,'c:/Documents and Settings/flhardy/Desktop/Eigen/TestG/Image5.jpg')

    disp('This Program is written by Md. Iftekhar Tanveer (go2chayan@gmail.com)');
disp('Copyleft 2009');

sSucceed = 0;
if (exist('strTrainPath')==0)
    strTrainPath = 'TrainG'
    %input('Enter Train Folder Name:','s');
    %'Train18'
    %TrainImages


end
if (exist('strLabelFile')==0)
    strLabelFile = get(handles.edit3,'string')
    %get(handles.edit3,'string')
```

```
end
if (exist('strTestPath')==0)
    strTestPath = 'TestG'
    %input('Enter Test Folder Name:','s');
    %'Test18'
    %TestImages

end

fid=fopen(strLabelFile);
imageLabel=textscan(fid,'%s %s','whitespace',',');
fclose(fid);

NeutralImages=[];
for i=1:length(imageLabel{1,1})
    if (strcmp(lower(imageLabel{1,2}{i,1}),'neutral'))
        NeutralImages=[NeutralImages,i];
    end
end
if (length(NeutralImages)==0)
    disp('ERROR: Neutral Expression is not available in training');
    return;
end

structTestImages = dir(strTestPath);
numImage = length(imageLabel{1,1});  % Total Observations: Number of Images in training set
lenTest = length(structTestImages);

if (lenTest==0)
    disp('Error:Invalid Test Folder');
    return;
end

TrainImages='';
for i = 1:numImage
    TrainImages{i,1} = strcat(strTrainPath,'\',imageLabel{1,1}(i));
end

j=0;
for i = 3:lenTest
     if ((~structTestImages(i).isdir))
         if  (structTestImages(i).name(end-3:end)=='.jpg')
             j=j+1;
             TestImages{j,1} = [strTestPath,'\',structTestImages(i).name];
         end
     end
end
numTestImage = j; % Number of Test Images
clear ('structTestImages','fid','i','j');pack

imageSize = [280,180];          % All Images are resized into a common size
```

```
%% ################# Load Train Data & Preprocess  ########################
%% Loading training images & preparing for PCA by subtracting mean

%%Reads in image from training set
%%Creates a vector of zeros based on the size of the training images
img = zeros(imageSize(1)*imageSize(2),numImage);
for i = 1:numImage %number of images in train set
   %%resize the images of the training set
   aa = imresize(detect_face(imresize(imread(cell2mat(TrainImages{i,1})),[375,300])),imageSize);
   img(:,i) = aa(:);
   %%load the training images
   disp(sprintf('Loading Train Image # %d',i));
end

%Calculates the mean of the images in the training set by rows
meanImage = mean(img,2);

%Calculate the difference between the image and the meanImages of the
%training data
img = (img - meanImage*ones(1,numImage))';     % img is the input to PCA
%% #########################################################################

%% ################# Low Dimension Face Space Construction ################
%%C is the Coefficient
%%S is the score
%%L is the latent

%%Returns the latent that are not zero which are the eigenvalues in the
%%covariance matrix
[C,S,L]=princomp(img,'econ');                % Performing PCA Here
EigenRange = [1:4];   % Defines which Eigenvalues will be selected based on Test Images atleast 2 images
C = C(:,EigenRange); %%?? All Eigen Vectors in C
%% #########################################################################


%% ############# Load Test Data and project on Face Space #################
%%create space for the Test images
img = zeros(imageSize(1)*imageSize(2),numTestImage);
for i = 1:numTestImage
   aa = imresize(detect_face(imresize(imread(TestImages{i,1}),[375,300])),imageSize);
   img(:,i) = aa(:);
   disp(sprintf('Loading Test Image # %d',i));
end
%Calculate the mean of the test images accorging to the rows in the vector
meanImage = mean(img,2);
%Calculates the difference between the face space vector and the test
%images
img = (img - meanImage*ones(1,numTestImage))';

%Projected test image
%Test image created
Projected_Test = img*C;
%% #########################################################################
```

```
%% ################ Calculation of Distance from Neutral #################
meanNutral = mean(S(NeutralImages,EigenRange)',2)
for Dat2Project = 1:numTestImage
   TestImage = Projected_Test(Dat2Project,:);
   % Picking the image #Dat2Project

   Eucl_Dist(Dat2Project) = sqrt((TestImage'-meanNutral)'*(TestImage' ...
      -meanNutral))
      % Here, the distance between the expression under test and
      % the mean neutral expressions is being calculated
end
%Eucl_Dist = Eucl_Dist/max(Eucl_Dist
%mds= Eucl_Dist - Other_Dist(:)
%% ########################################################################

%% ################ Calculation of other Distances ######################
Other_Dist = zeros(numTestImage,numImage);
for Dat2Project = 1:numTestImage
   TestImage = Projected_Test(Dat2Project,:);
   % Picking the image #Dat2Project
   for i = 1:numImage
      Other_Dist(Dat2Project,i) = sqrt((TestImage'-S(i,EigenRange)')' ...
         *(TestImage'-S(i,EigenRange)'));

   end
end
[Min_Dist,Min_Dist_pos] = min(Other_Dist,[],2)

%% ########################################################################


%% ########################## Display Result #############################
fid = fopen('Results.txt','w');
fid1=fopen('EmotionResult1.txt','w');
fid2=fopen('EmotionResult2.txt','w');
fid3=fopen('EmotionResult3.txt','w');
fid4=fopen('EmotionResult4.txt','w');
fid5=fopen('EmotionResult5.txt','w');

%fprintf(fid,'//Test Image,Distance From Neutral, Expression,Best Match\r\n');

for i = 1:numTestImage
   b = find(TestImages{i,1}=='\');
   Test_Image = TestImages{i,1}(b(end)+1:end);
   Dist_frm_Neutral = Eucl_Dist(i);
   Best_Match = cell2mat(imageLabel{1,1}(Min_Dist_pos(i)));
   Expr = cell2mat(imageLabel{1,2}(Min_Dist_pos(i)));
   fprintf(fid,'%s,%0.0f,%s,%s\r\n',Test_Image,Dist_frm_Neutral,Expr,Best_Match);

end

if (counter == 1)
for i = numTestImage
```

```
    b = find(TestImages{i,1}=='\');
    Test_Image = TestImages{i,1}(b(end)+1:end);
    Dist_frm_Neutral = Eucl_Dist(i);
    Best_Match = cell2mat(imageLabel{1,1}(Min_Dist_pos(i)));
    Expr = cell2mat(imageLabel{1,2}(Min_Dist_pos(i)));
    fprintf(fid1,'%s\r\n',Expr);

end
end

if (counter == 2)
for i = numTestImage
    b = find(TestImages{i,1}=='\');
    Test_Image = TestImages{i,1}(b(end)+1:end);
    Dist_frm_Neutral = Eucl_Dist(i);
    Best_Match = cell2mat(imageLabel{1,1}(Min_Dist_pos(i)));
    Expr = cell2mat(imageLabel{1,2}(Min_Dist_pos(i)));
    fprintf(fid2,'%s\r\n',Expr);

end
end

if (counter == 3)
for i = numTestImage
    b = find(TestImages{i,1}=='\');
    Test_Image = TestImages{i,1}(b(end)+1:end);
    Dist_frm_Neutral = Eucl_Dist(i);
    Best_Match = cell2mat(imageLabel{1,1}(Min_Dist_pos(i)));
    Expr = cell2mat(imageLabel{1,2}(Min_Dist_pos(i)));
    fprintf(fid3,'%s\r\n',Expr);

end
end

if (counter == 4)
for i = numTestImage
    b = find(TestImages{i,1}=='\');
    Test_Image = TestImages{i,1}(b(end)+1:end);
    Dist_frm_Neutral = Eucl_Dist(i);
    Best_Match = cell2mat(imageLabel{1,1}(Min_Dist_pos(i)));
    Expr = cell2mat(imageLabel{1,2}(Min_Dist_pos(i)));
    fprintf(fid4,'%s\r\n',Expr);

end
end

if (counter == 5)
for i = numTestImage
    b = find(TestImages{i,1}=='\');
    Test_Image = TestImages{i,1}(b(end)+1:end);
    Dist_frm_Neutral = Eucl_Dist(i);
    Best_Match = cell2mat(imageLabel{1,1}(Min_Dist_pos(i)));
    Expr = cell2mat(imageLabel{1,2}(Min_Dist_pos(i)));
```

```
    fprintf(fid5,'%s\r\n',Expr);

end
end
fclose(fid);
fclose(fid1);
fclose(fid2);
fclose(fid3);
fclose(fid4);
fclose(fid5);
end
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton11.
```

APPENDIX B

FACE DETECTION ALGORITHM CODE

```
%###########################################################################
%%##################### Face Detection Algorithm

% Tolga Birdal's original code is heavily modified by Md. Iftekhar Tanveer
% (go2chayan@gmail.com)
% Modifications by Md. Iftekhar Tanveer:
% code optimized for the following assumptions:
% 1. Only one face in scene and it is the primary object
% 2. Faster noise reduction and face detection

% Originally by Tolga Birdal
% Implementation of the paper:
% "A simple and accurate face detection algorithm in complex background"
% by Yu-Tang Pai, Shanq-Jang Ruan, Mon-Chau Shie, Yi-Chi Liu

% Additions by Tolga Birdal:
%  Minimum face size constraint
%  Adaptive theta thresholding (Theta is thresholded by mean2(theata)/4
%  Parameters are modified by to detect better. Please check the paper for
%  parameters they propose.
% Check the paper for more details.

% usage:
%  I=double(imread('c:\Data\girl1.jpg'));
%  detect_face(I);
% The function will display the bounding box if a face is found.


function [aa,SN_fill,FaceDat]=detect_face(I)
% No faces at the beginning
Faces=[];
numFaceFound=0;

I=double(I);

H=size(I,1);
W=size(I,2);

%%%%%%%%%%%%%%%%%% LIGHTING COMPENSATION %%%%%%%%%%%%%%%

C=255*imadjust(I/255,[0.3;1],[0;1]);

% figure,imshow(C/255);
% title('Lighting compensation');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%% EXTRACT SKIN
%%%%%%%%%%%%%%%%%%%%%%
```

```
YCbCr=rgb2ycbcr(C);
Cr=YCbCr(:,:,3);

S=zeros(H,W);
[SkinIndexRow,SkinIndexCol] =find(10<Cr & Cr<255);
for i=1:length(SkinIndexRow)
    S(SkinIndexRow(i),SkinIndexCol(i))=1;
end

m_S = size(S);
S(m_S(1)-7:m_S(1),:) = 0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%

%%%%%%%%%%%%%%%% REMOVE NOISE %%%%
% figure imshow(S);

SN=zeros(H,W);
for i=1:H-5
    for j=1:W-5
        localSum=sum(sum(S(i:i+4, j:j+4)));
        SN(i:i+5, j:j+5)=(localSum>20);
    end
end

%     figure;imshow(SN);

 Iedge=edge(uint8(SN));

%  figure;imshow(Iedge);

SE = strel('square',9);
SN_edge = (imdilate(Iedge,SE));


%
% SN_edge =  SN_edge1.*SN;

%  figure;imshow(SN_edge);

 SN_fill = imfill(SN_edge,'holes');
%  figure;imshow(SN_fill);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%

%%%%%%%%%%%%%%%% FIND SKIN COLOR BLOCKS %%%%

[L,lenRegions] = bwlabel(SN_fill,4);
AllDat  = regionprops(L,'BoundingBox','FilledArea');
AreaDat = cat(1, AllDat.FilledArea);
[maxArea, maxAreaInd] = max(AreaDat);
```

```
FaceDat = AllDat(maxAreaInd);
FaceBB = [FaceDat.BoundingBox(1),FaceDat.BoundingBox(2),...
    FaceDat.BoundingBox(3)-1,FaceDat.BoundingBox(4)-1];

aa=imcrop(rgb2gray(uint8(I)).*uint8(SN_fill),FaceBB);

 %figure,imshow(aa);
 %title('Identified Face');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

end
```