ÇANKAYA UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING

MASTER THESIS

FINDING THE ETHNICAL IDENTITY OF HUMAN FACE

MERVE YENİCE

SEPTEMBER 2012

Title of the Thesis: **Finding the Ethnical Identity of Human Face**

Submitted by **Merve YENİCE**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University

Prof. Dr. Taner ALTUNOK
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Murat SARAN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

(Title and Name)
Co-Supervisor

Asst. Prof. Dr. Reza HASSANPOUR
Supervisor

Examination Date : 17.09.2012

Examining Committee Members (first name belongs to the chairperson of the jury and the second name belongs to supervisor)

Prof. Dr. Mehmet R. TOLUN (TED Univ.) ....

Asst. Prof. Dr. Reza HASSANPOUR (Çankaya Univ.) ....

Asst. Prof. Dr. Abdul Kadir GÖRÜR (Çankaya Univ.) ....

STATEMENT OF NON-PLAGIARISM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : MERVE YENİCE

Signature :

Date : 17.09.2012

ABSTRACT

FINDING THE ETHNICAL IDENTITY OF HUMAN FACE

Yenice, Merve

M.S.c., Department of Computer Engineering

Supervisor : Asst. Prof. Dr. Reza HASSANPOUR

September 2012, 62 pages

In this thesis, how to find a human being's ethnical identity from his/her face is analyzed. Parts of the face like eyes, nose, mouth, skin colour are used for defining the face. In addition to this, some programs like C# and Luxand are also used in correctly defining and calculating the facial parts. This calculation is very important and necessary in fractionating the human face and finding the dimensions of members of the face, because it gives the main idea about the shape, length and colour of face. The most important issues in determining and finding the ethnical identity of human face are shape, length and skin colour of the face. After finding these items the ethnical identity of a human can easily be found. The evidence found after working in this thesis is shown that the thesis has reached its aim willingly.

Keywords: Ethnical Identity, Ethnic ID

ÖZ

İNSAN YÜZÜNDEN ETNİK KİMLİK BULMAK

Yenice, Merve
Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı
Tez Yöneticisi : Asst. Prof. Dr. Reza HASSANPOUR

Eylül 2012, 62 sayfa

Bu çalışmada, bir insanın yüzünden etnik kimliğini bulabilme işlemi incelenmiştir. Gözler, burun, ağız, ten rengi gibi yüz parçaları yüzün tanımlanabilmesi için kullanılmıştır. Buna ek olarak, C# ve Luxand gibi yazılımlar da yine, yüzün parçalarının doğru tayin edilmesi ve hesaplanması için kullanılmışlardır. Ortaya çıkan bu hesaplama, yüz parçalarının bölümlere ayrılması ve boyutlarının hesaplanması için çok önemli ve gereklidir çünkü, ancak yapılan bu hesaplama yüzün şekli, uzunluğu ve rengi hakkında temel fikri verir. Bir insan yüzünden onun etnik kimliğini tanımlamak ve bulmak için en önemli gereksinimler, yüzün şekli, uzunluğu ve cilt rengidir. Bunları bulduktan sonra, bir insanın etnik kimliği de kolayca bulunabilir. Bu çalışma sonucunda elde edilen bulgular tezin istenilen biçimde amacına ulaştığını gösterir.

Keywords: Etnik Kimlik, Etnik Benzerlik

ACKNOWLEDGMENTS

I want to thank to my teachers who helped during my master education and preparation of this thesis. Moreover, I want to introduce my great thanks to Prof. Dr. Mehmet R. TOLUN, Asst. Prof. Dr. Reza HASSANPOUR. Finally, thanks to member of jury Asst. Prof. Dr. Abdul Kadir GÖRÜR.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

Ethnical identity can roughly be defined as one of the main facts that differs folks and nations from each other according to their physical appearances, language and religion. Ethnic identity is an important part of understanding who you are and where you came from. There are billions of people living all over the world who came from different ethnical roots for many many years. In addition to this, most people become cosmopolitan with the globalization of the world. For these reasons, ethnic cultures are interwining nowadays. This situation shows the richness of ethnic cultural heritage of the world. However it leads to nearly impossible to talk about a person's pure ethnical identity. Lately, only DNA tests give some nearly correct results about people's root. However it is not certain.

Because of the reasons mentioned above three types of people from different ethnical roots are studied in this thesis. African people, Asian people and others are accepted as European people. More deeply root identification is not the subject of this thesis. These three types ethnical identities are investigated in this thesis because of giving the certain results about them. Some physical characteristics of these people give hints about their ethnical identities to us. One of the definition of ethnic root is "groups differentiated by color." So an Asian (or European) and an African obviously can be distinguished in the first look from their physical view. In spite of that, a person can easily realizes the differences between them and separates each other, it is not as easy as differing them for a computer. Because a computer is a device that gives the correct result what you enter. However it can not make any logical comparisons, if no entry is defined into the computer.

The main subject of this thesis is studied in finding the ethnical identity of a person from his/her face. It is no way of finding the ethnical identity directly for a computer. Some data must be loaded, some calculations must be done and according to these results a comparison can be made and reached to the wanted solution. There are two main steps for a computer to determine the ethnical identity of a human face. Calculating the size of face and then determining ethnical root.

First of all calculating the face and face parts. According to this thesis work, face is the only and only data that is used in finding the ethnical identity of a person by using a computer. It seems and sounds easy. However a computer doesn't work as a human brain, so it isn't as easy as thinking it. At this stage the computer has to be made think as a human brain. For satisfying this condition in this thesis, calculating the size of human faces the is the first main method which is studied. A computer can not understand or be defined a human face as a whole. For that reason, a human face can be introduced to the computer by dividing into smaller parts and then combining them. Just after that, a computer can understand the vision. The dimensions of facial parts are very beneficial in understanding the face and determining the ethnical identity.

The second step is making the comparisons between facial parts and type which the clues have been taken from the previous work. Then determining the root of a person.

CHAPTER I

1.1. DETERMINING FACE AND MOTIVATION APPLICATION

Face is one of the most important organs of a human. People know and look each other face first. Most of the sense organs are found at face. Fore head, eyes, mouth, nose, chin and eyebrows are found at face. So we can learn the environment with the help of our face. The feelings can be easily seen from the face. Also the definition of the beauty starts at face.

This thesis is studied in face types and face structure because it is a new area in computer engineering. It is very popular as it attracts the attention of many people. In this thesis, detailed information about face types, face structure, skin color and measuring the facial components are given.

The first, most important and difficult work is determining the face to the computer as a face. It is very hard in the beginning because computers have no logical comparison ability for doing that without a program. All photos are same for them which made up of zeros and ones. In addition it does not make any sense if it is a photo or text file. Image processing is a technique that is used to solve identification problem. It uses some techniques to analyze the photo and identify shades, colors and relationships like perceiving by the human eye. After that step, pattern recognition is needed to complete the identification. It is aimed to identify the face without active human participation in the decision process of computer. To overcome this problem, a software called “Luxand” is used in the thesis. Luxand is a Software Development Kit (SDK) that is used to develop image processing and pattern recognition. Luxand is very useful because it is compatible with

.NET. Face Software Development Kit Detecting Face (FSDK_DetectFace) function in Luxand is used to find the front view face in a photo or image. The function turns to the locations of the face in the photo. The efficiency and trustfulness of face detection is checked by parameters and Threshold functions. FaceSDK also satisfies the function called FSDK_DetectFacialFeatures. It helps to find the parts and specialities of the face in the photo. The centres of eyes are found by the function of FSDK_DetectEyes. These functions are basic ones for finding the front view of the face and assigning centre of eyes or facial parts. The same functions that have in region speciality don't have any face detection, facial part detection or centre of eyes properties. The characteristics of facial parts are saved in the data structure of FSDK_Features which is a kind of data type array. Moreover it contains another function called FSDK_FACIAL_FEATURE_COUNT that gives the number of facial parts' points.

Luxand FaceSDK detects 66 facial feature points as shown in the picture below. There are 66 critical points at the face which allows the computer to understand the photo if it is a face or not. Each point has a unique meaning that represents and introduces the parts of the face.

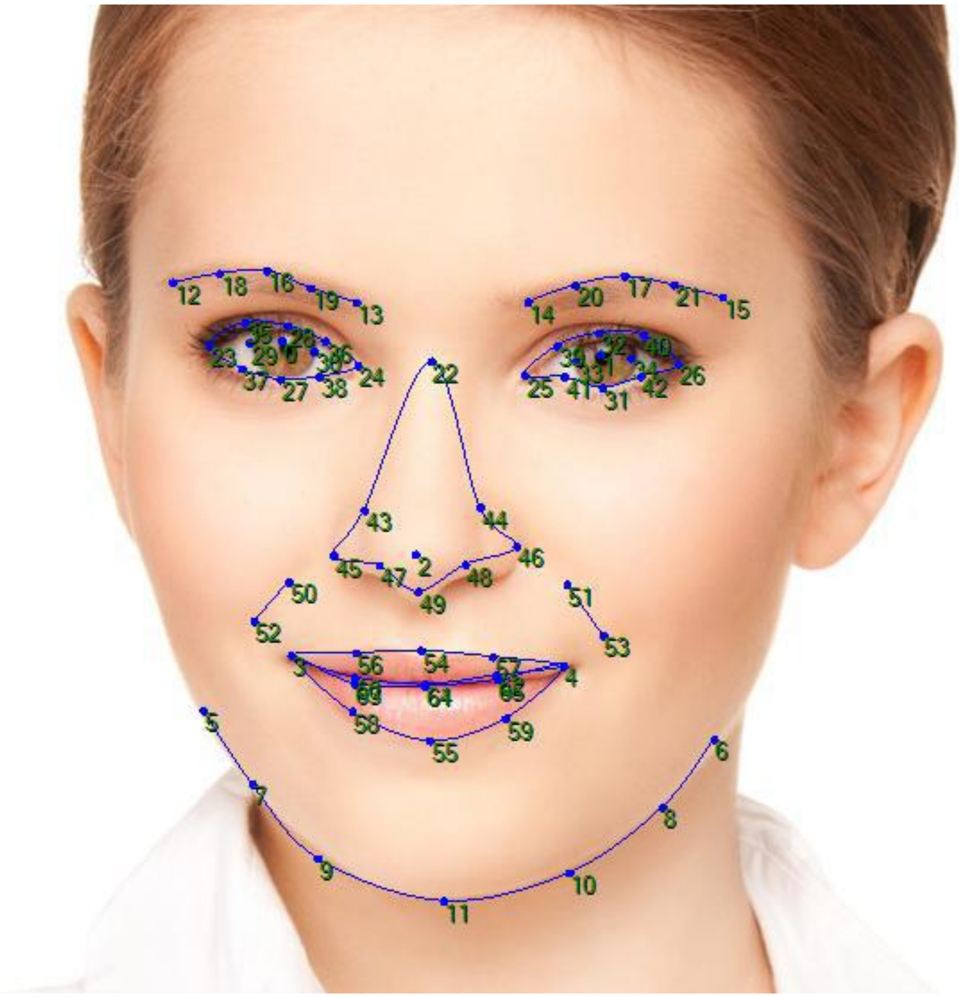


Figure 1 - Face Features for Mask

After defining these points, they are put in an array. These facial feature points can be reached by their names in the array found FSDK_Features function. There is a list of 66 points that is defined as an array with their names and places where to define. In the end of this work, the face is defined to the computer which is composed of 66 points. The names and array values defined in the program are listed below.

Facial Feature Name	Value
FSDKP_LEFT_EYE	0
FSDKP_RIGHT_EYE	1
FSDKP_NOSE_TIP	2
FSDKP_MOUTH_RIGHT_CORNER	3
FSDKP_LEFT_EYE_INNER_CORNER	24
FSDKP_LEFT_EYE_OUTER_CORNER	23
FSDKP_LEFT_EYE_LOWER_LINE1	38
FSDKP_LEFT_EYE_LOWER_LINE2	27
FSDKP_LEFT_EYE_LOWER_LINE3	37
FSDKP_LEFT_EYE_UPPER_LINE1	35
FSDKP_LEFT_EYE_UPPER_LINE2	28
FSDKP_LEFT_EYE_UPPER_LINE3	36
FSDKP_LEFT_EYE_LEFT_IRIS_CORNER	29
FSDKP_LEFT_EYE_RIGHT_IRIS_CORNER	30
FSDKP_RIGHT_EYE_INNER_CORNER	25
FSDKP_RIGHT_EYE_OUTER_CORNER	26
FSDKP_RIGHT_EYE_LOWER_LINE1	41
FSDKP_RIGHT_EYE_LOWER_LINE2	31
FSDKP_RIGHT_EYE_LOWER_LINE3	42
FSDKP_RIGHT_EYE_UPPER_LINE1	40
FSDKP_RIGHT_EYE_UPPER_LINE2	32
FSDKP_RIGHT_EYE_UPPER_LINE3	39
FSDKP_RIGHT_EYE_LEFT_IRIS_CORNER	33
FSDKP_RIGHT_EYE_RIGHT_IRIS_CORNER	34
FSDKP_LEFT_EYEBROW_INNER_CORNER	13
FSDKP_LEFT_EYEBROW_MIDDLE	16
FSDKP_LEFT_EYEBROW_MIDDLE_LEFT	18
FSDKP_LEFT_EYEBROW_MIDDLE_RIGHT	19
FSDKP_LEFT_EYEBROW_OUTER_CORNER	12
FSDKP_RIGHT_EYEBROW_INNER_CORNER	14
FSDKP_RIGHT_EYEBROW_MIDDLE	17
FSDKP_ RIGHT_EYEBROW_MIDDLE_LEFT	20
FSDKP_ RIGHT_EYEBROW_MIDDLE_RIGHT	21
FSDKP_RIGHT_EYEBROW_OUTER_CORNER	15
FSDKP_NOSE_BOTTOM	49
FSDKP_NOSE_BRIDGE	22
FSDKP_NOSE_LEFT_WING	43
FSDKP_NOSE_LEFT_WING_OUTER	45
FSDKP_NOSE_LEFT_WING_LOWER	47
FSDKP_NOSE_RIGHT_WING	44
FSDKP_NOSE_RIGHT_WING_OUTER	46
FSDKP_NOSE_RIGHT_WING_LOWER	48
FSDKP_MOUTH_LEFT_CORNER	4
FSDKP_MOUTH_TOP	54
FSDKP_MOUTH_TOP_INNER	61
FSDKP_MOUTH_BOTTOM	55
FSDKP_MOUTH_BOTTOM_INNER	64
FSDKP_MOUTH_LEFT_TOP	56
FSDKP_MOUTH_LEFT_TOP_INNER	60
FSDKP_MOUTH_RIGHT_TOP	57
FSDKP_MOUTH_RIGHT_TOP_INNER	62
FSDKP_MOUTH_LEFT_BOTTOM	58

FSDKP_MOUTH_LEFT_BOTTOM_INNER	63
FSDKP_MOUTH_RIGHT_BOTTOM	59
FSDKP_MOUTH_RIGHT_BOTTOM_INNER	65
FSDKP_NASOLABIAL_FOLD_LEFT_UPPER	50
FSDKP_NASOLABIAL_FOLD_LEFT_LOWER	52
FSDKP_NASOLABIAL_FOLD_RIGHT_UPPER	51
FSDKP_NASOLABIAL_FOLD_RIGHT_LOWER	53
FSDKP_CHIN_BOTTOM	11
FSDKP_CHIN_LEFT	9
FSDKP_CHIN_RIGHT	10
FSDKP_FACE_CONTOUR1	7
FSDKP_FACE_CONTOUR2	5
FSDKP_FACE_CONTOUR12	6
FSDKP_FACE_CONTOUR13	8

After defining this 66 points at the face, the computer has an idea about the face and its features. The meanings of the arrays are very important, because each point represents a specific part on the face and once it is defined in the array, it cannot be changed. At the end of using this arrays, some specifications of facial parts are obtained. They can be summarized like below.

The width and height of the eyes are obtained, so vertical measure is generally less than the horizontal distance. Also the length between two eyes is generally an eye width. The shape or length of eyebrows and the distance between eyebrows and eyes. The eyebrows give a special shape to the human face and give a meaning to a human’s eyes and forehead. They also have another main task. Eyebrows make communication easier. In addition they affiliate meaning to feelings and expressions, such as pleasure or anger. The length and width of a nose and nose wings are taken. The shape of the nose is formed by the bone of the skull between the eyes and at the roof of the nose and the septum in the nose that seperates the left and right airways in the nose. The dimensions of a perfect nose look like a cone. Generally a male nose is bigger than a nose of a female. The length and width of lips, upper lips and lower lips. The distance between nares and lips. The shape of chin. In addition the face contour. In the end the 66 points that described in the arrays give the dimensions of the face and face feature, so size of human faces can be calculated.

From now on, image processing and pattern recognition tools that are found in Luxand helps to the computer to understand a photo if it is a face or not. In addition to this with the help of the arrays the computer uses patterns and detects the parts of the face in detail.

1.2. METHODS

After defining the arrays in Luxand, it gives the important information about face and its dimensions. However there must be some restrictions to fix the pixel an centimeter measurement. First all the length and width of facial parts are measured one by one with the help of a ruler in centimeters. Then a photo is taken and measurement of facial parts are realized by the help of computer. But unfortunately, it didn’t give the wanted solution. Because while the measurement is done in centimeters in real life, it is done in pixels in computer. This is the most important problem that I have introduced while working on face calculating. To overcome this problem, what is the equivalent of 1 pixel in terms of centimeters must be evaluated. After measuring parts of the face, the size is calculated for both in real and picture. So, how many pixel equals to how many centimeters is found. In addition to this some user restrictions are needed to define the real size nearly same to the size in picture. The first restriction is the specification of the camera. It must be HD 4.0. The resolution is 16.2 Megapixel. The dimensions of the photo must be 1024X768. Also 72 dpi and 24 bit depth. The zoom property must be 4.9. These are the limitations of the device to catch the right solution. Another restriction is related to the distance. It must be 1.80 cm between the camera and the person whose photo is taken. In the end, the real size of the face is same with the size of photo and the calculations will give the correct result.

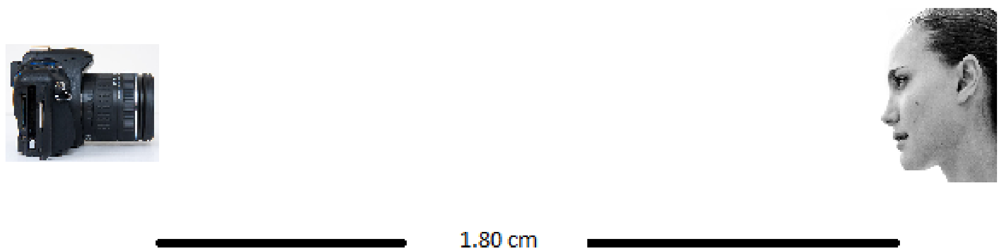


Figure 2 - Taking Photo Method

1.3. RESULTS OF DETERMINING FACE

In this part, determining the face and calculating the size of facial parts of human face is given. First the photo of human face is taken. Then the computer finds the points of facial parts by using x-y coordinate system. While doing that, it uses pixel and then changes into centimeter (cm). After that the computer determines the sizes of facial parts by using the program of Luxand. Finally, the computer detects both the places, sizes of facial parts and finds the face itself. Also calculating the size of human faces is realized in this step with the restrictions to be followed that are explained above.

1.4. ETHNICAL IDENTITIES AND DIFFERENCES

Figure 3 - Ethnic Differences in a Face

Ethnic identity refers to the ethnicity that someone identifies with, often through family, community, or nationality. So it is an important part of understanding who you are and where you came from. Ethnical identification can be described as a relation between a human and the folk that his / her elder parents belong to. They all share common properties like physical or cultural. Thefore ethnical group is is defined as people who have shared same language and cultural heritage.

Ethnicity isn't exactly too important nowadays. It needs considerability for people just for learning their past and history. So they can have information about the origin of their elder parents and ancestors, also their physical appearances and way of living. It is still brought up in pop culture because people are still racist, so there needs to be ways to combat this. In this thesis, ethnical root is an important reason just for the current academic interest.

The traditional definition of different ethnical roots is identified with biological and sociological reasons orderly. Ethnicity expresses a human's physical appearance like eye shape, skin color, bone structure etc. Looking at it the other way, ethnical identity interests in cultural factors like national status, culture, ancestries, languages and beliefs.

In this thesis work, human face is used to understand the ethnical root of a person. Because types of faces show differences between the races. There are many different race types all over the world, but three types of race is studied here. These are African, Asian and European.

The main characteristics and differences of African face type can be defined as

- Thick lower lip.
- Thick upper lip.
- Wider nose and nares such that the lateral aspect of the nare is on the naso-labial fold.
- Eye brows in the brow zone is upper part of the face.
- Lateral border of the face slightly narrower.

Figure 4 - African Face Type

The main characteristics and differences of Asian face type can be defined as

- Medial and lateral epicanthic fold.
- Lateral border of the face significantly wider.
- Eye brows slightly superior to that of the face with shorter tails.
- Slightly wider nose and nares extend laterally.
- Superiorly positioned nasal columella creating a longer upper lip.

Figure 5 - Asian Face Type

The main characteristics and differences of European face type can be defined as

- Slightly vertically thin upper and lower lips
- Flat eyebrow (very little arch)
- Slightly wider nose
- Lateral border of the face slightly wider than the Mask
- Possible: Narrow eyes, longer vertical chin, longer nose

Figure 6 - European Face Type

So the main differences between three types of ethnic roots are determined and accepted like defined above. Then they must be introduced to the computer for determining the ethnical roots of face types from the photos. A limitation is defined to the computer for the width of face. Also the width of mouth and nose are looking together, at the same time. For example, for the Africans the skin color is the most distinguishing factor. So, the facial parts are looked together by the computer and after making the comparisons some are defined above, shapes of the facial parts define the specific ethnic root.

CHAPTER II

2.1. GENERAL INFORMATION

The work that is studied on this thesis is used in many areas all over the world. The working areas of this subject is limitless, however it is a new technology and for that reason it must be developed. The subject of the thesis is both related to image processing and pattern recognition. Image processing is used in determining the face and calculating the size of human faces. Pattern recognition is used in finding the ethnical identity of the human face. Three types of race is used in this work, because ethnic guess is very complex and the ethnic identity of the world is on very friendly terms with each other. So the dominant three races are chosen at this work.

2.2. PREVIOUS WORK

Ethnicity Identification from Face Images : This work is done by Xiaoguang Lu and Anil K. Jain in the USA. In this work, the researchers accept the
facial images of human for different population and folks like ethnicity or gender. The Linear Discriminant Analysis (LDA) is used in this work. LDA is a popular statistical technique that reflects the given multidimensional information to the lower dimension. Here, Linear Discriminant Analysis is a method which determines the ethnical classification. However in this work it represents two different ethnicities whether Asian or non Asian. In addition to this, three different racial groups are studied on this work. There are 25 points from face and head used for determining if a person is from North America, Africa-America or China. These 25 points measurements give the main racial differences between these three ethnicities. For example, the width of face and nose is greatest at Chinese people.

LDA accepts the facial image which has two dimension as one dimensional vector. While doing that, it joins each row or column of the facial image. So, as described above this analysis provides a high dimensional fave feature vector space from a low dimensional representation. The transformation matrix is aimed the face vector and its coefficients are satisfied the representaion of facial image. In this work, for each class, only a single Gaussian models the density estimation in the sample space. LDA is successful at different, multiple scale images for providing information for distinct leels of visual signs. In this implementation, three distinct scales is used for resizing each facial image. A Linear Discriminant Analysis based classifier is used at each scale. So, it satisfies the equation between the number of classifiers and number of scales. Only the final ensemble consists of these scales. In the ensemble, different scales stisfy the confidence values of class for each image. Each facial images are divided into each parts of the face like the mouth, nose and eyes. Because of being appearance based analysis, the LDA is very sensitive to the quality of alignment. In this work all of the divided facial images are aligned with the two eyes' center. With the help of different types of light the intensity contrast is extended for normalizing. Each vector of the facial image is aimed normalizing for being a unit length.

Briefly, the work studied on this paper aimed finding ethnicity identification by using facial images. The Linear Discriminant Analysis (LDA) is used for determining the difference between two ethnicities if it is Asian or non Asian. An ensemble content uses the LDA for the input face images at different and distinct scales improve the classification performance for the ethnical identification.

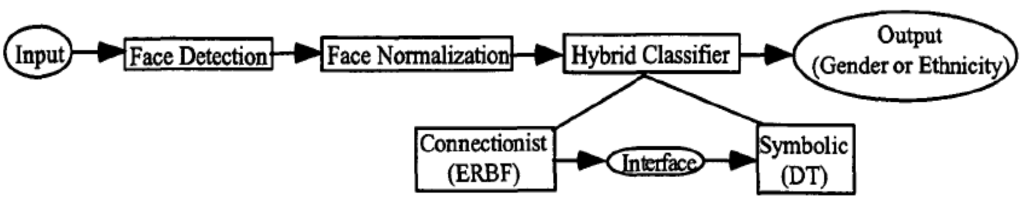
Principal Component Analysis of Gender, Ethnicity, Age, and Identity of Face Images: This is a common work done by Samarasena Buchala, Neil Davey, Tim M. Gale, Ray J. Frank Department of Computer Science and Department of Psychiatry in UK. Principal Component Analysis (PCA) is a most common used analysis for representing data of facial images in a subspace with low dimension. PCA is a very effective method for that reason, this analysis is chosen for this study. It is very successful at analyzing different parts of the faces and different specialities like age, gender or ethnical identity. In addition to PCA, Linear Discriminant Analysis (LDA) is also used in this work. The duty of LDA is controlling the working principles of PCA whether it works how efficient. As a result it proves that PCA encodes different and multiple specialities.

Briefly, this paper analyzes different features of faces, age, gender or ethnical identity using PCA on facial images. While doing that LDA helps controlling if it is work true or false. As a result the work can be summarized in three substances.

1. PCA codes facial image components like age, gender or ethnical identity effectively. The performances of the classification for all specialities are very high.
2. Different properties of PCA code distinct specialities of a face. Many different specialities are used for defining age and gender. However a very wide components are required for defining ethnical identity.
3. There can be some factors that which encode different and distinct properties. For example let take age, gender or ethnical identity are the three components. Each have equal importance and if one is missing, the others have no meaning.

Gender and Ethnic Classification of Human Faces Using Hybrid Classifiers : Done by Srinivas Gutta and Harry Wechsler in the USA. This paper includes paragenetic classification techniques for gender and ethnical properties from the human faces. In this work 3006 facial images are used for correlating 1009 different specialities from the database of FERET. The paragenetic approach includes a group of RBF networks and Decision Trees (DT) which are inductive. Experimental Cross Validation reaches an avarage conclusion that satisfies a rate of 96% correctness on the classification of genders, also gives a rate of 94% correction on the classification of ethnical identities. There are some beneficial results of paragenetic classification like satisfying strongness at querying by a group of RBF network unity. Decision Trees provide an adaptive and flexible threshhold values in contrast to ad hoc and hard values. There is an important issue to be underlied. The paragenetic approach is related to reduction. The complex problems are solved analyzing slowly. Moreover, this approach is based on logical difference between cognitive operations. Automatic and reflexive, low level and controlled (like perception), high level and deliberative (like decision making). The paragenetic systems assign reflexive processes to the related sub system and deliberative processes to the symbolic sub system. Intelligent paragenetic (heterogeneous) systems include characteristic hierarchical steps of knowledge. These are defined particularly and corresponding interface. Especially, related, fuzzy and symbolic steps are included in this hierarchy. Each level consists of an even tempered architecture in itself. The paragenetic classification techniques used in this paper is determining for genders and ethnical identities. In addition to this, these techniques include symbolic and related modules. Paragenetic approach gets the comprehensive stencil matching with distinct methods together. While doing that, numerical and symbolic values are used in order. The relating step is also defined by using a group of terms from Radial Basis Function (RBF) networks. Moreover, the symbolic step includes Decision Trees (DT). The relating Radial Basis Function networks are also studied on distinct datasets correlative to different types of the first taken data. This gives a birth to increasing incertainity at distinct topologies of the RBF networks.

The architecture of paragenetic or hybrid for defining genders and ethnical identities can be summarized as shown in the figure below.



Learning Appearance Primitives Of Iris Images For Ethnic Classification : This research is done by Xianchao Qiu, Zhenan Sun, and Tieniu Tan in China. The shape and form of the iris is a very important feature which shows different environmental issues distinct from the genetic specialities. In this study, the images of iris are worked. The main aim of this paper is proving that there is a relation between iris and ethnical identities. Also learning the appearance of rude people from the given iris image is purposed. Therefore, it approves both an iris shows ethnical identity and finding from the iris images how the rude people of Asian and non Asian look distinct from each other. The human eye's iris has a round shape form. There is an extremely big difference at texture between the dark part and white part of the human eye. This texture has a casual and specifically single structure for each person. Because of that reason, iris recognition is used at finding biometric solutions in personal identification. The work done in this paper, a new and original method is used for detecting the ethnical identity automatically. This method is exploit from the distinguishing visual images of the iris.

Briefly, the additions of this work to the science world can be summarized like below:

1. An iris image can be thought as a visual dictionary. The words are learnt from the images of iris. Each word can be accepted as different and distinct iris images.
2. Asian and non-Asian iris images are roughly classified in this work. Therefore the relation between genotypical pattern of the iris image is proven scientifically.

Integrating Faces, Fingerprints, and Soft Biometric Traits for User Recognition: is a work done by Anil K. Jain, Karthik Nandakumar, Xiaoguang Lu, and Unsang Park in the USA. Eye colour, weight, height, age, gender and ethnic identity can be said as soft biometric properties. However they have no distinguishing and genetic specialities. In contrast, some other soft biometric properties like DNA, iris, fingerprint or microbioms are unique at each person and can be used as distinctive specialities. In this work, the researchers studied on biometric systems which use face and fingerprint. Also the soft characteristics used in this paper are height, gender and ethnical identity. Moreover, the effects of biometric properties from facial image on identification and fingerprint identification that uses soft biometric propertis are studied.

While realizing this study, a database which is made up of 263 people is used. As a result of experiments show that using biometric data upgrades the performance of identification in facial and fingerprint detection. The developing steps of experiments can be summarized like this. Each of 263 people has been taken 10 facial images. In addition, the fingerprints of 160 users have been taken. The fingerprints have been taken by using a Veridicom sensor. Four image from four fingers are obtained

from every user in the database. These four fingers are left and right index fingers and left and right middle fingers. So, there are 640 fingers used in total for this experiment. However, because of using 263 facial images, 263 of these fingers are selected and signed for each faces in the database. Linear Discriminant Analysis (LDA) is an effective analysis that is used in matching. For that reason LDA is used in facial comparisons for this work. The facial images are used at two steps during the experiments. Eight images of ten for each experimental are used at training step and two images are used at test step. The comparision score vector of face has a length of 263. This number is equal to the number of total facial images in the database. The computation of these test images for each one leads to computing similarity. This similarity is computed as taking eight largest scores. 2104 (263 X 8) represents facial images are used at training step. The eight largest score of a unique experimental is chosen as the comparison score of that experimental.
Another work studied on this paper is identifying ethnicity. The classifier used here is for identifying ethnical identities as if the experimentals are Asian or non Asian. The experiment gives a result of 96.3% correctness. There can be put a reject option to the system. If this option is added, the classification gives a mistake of less than 1%. The gender classification works with the same logic as ethnical classifier. The classifier used for identifying gender gives a result of 89.6% correctness. A reject option can also be added to this system. In that case it gives a result less than 2% fault. Sometimes gender or ethnical identity classifier can not decide and gives a result which is not dependable. In such a situation the biometric system can not take the advantage of given information.

Multimodal Facial Gender and Ethnicity Identification : done by Xiaoguang Lu, Hong Chen, and Anil K. Jain in the USA. There are some distinct procedures like range, intensity on human faces which satisfy some hints for a person's gender and ethnical identity classification. The researchers who studied in this paper have used Support Vector Machine (SVM) for ethnical identification from human faces. The range and intensity of facial images are the properties that registered to the system. These properties are combined and integrated for gender and ethnical identification in this work. The database used in this paper consists of 1240 facial images that have 376 subjects. Therefore the experiments are managed by using this database. The experiments prove that the range method has overwhelming seperative property on identifying gender and ethnicity according to the intensity method. The integration pattern has an hegemony for each method in identifying gender and ethnical identity.

Sensing and recognizing in facial images compromise gender and ethnical identification. The researchers who work on this paper have divided the classisfication of ethnicity into two groups. The study works on if the experimentals are Asian or non Asian.

The scientists who relate the science of measuring the human body have obtained some statistical results about facial differences of people's ethnicity. At the end of these working it is proved that there is a strong relationship between the human face which has 3D shape and his/her ethnical identity. The three dimensional facial image database used for this paper has been taken from two universities. One of them is taken from University of Notre Dame (UND). The database taken from here has included 944 facial images that have 276 subjects. The other one is taken from Michigan State University (MSU). The database taken from here has included 296 facial images that have 100 subjects. The facial images are scanned and gathered up with Minolta Vivid series 3D scanner. Then the three dimensional frontal facial scanned images are collected together for the evaluation of this paper's aimed job.

Ten facial images for each experimental are selected. Nine of them are used in training step, the remaining one is used at test step. This situation satisfies cross validation for ethnical identification. The scanned images which have same subject are taken into the same place. This is a compulsatory for satisfying that the similarity is not affected from the data used in both training and testing steps in identifying ethnicity. There are some mean and standard deviation values obtained while comparing error rates are taken here. These values are taken down for ten experiments. The same application is also used for identification of gender. After that, both gender and ethnical identification are satisfied. At the end of the experiment, the range of facial images are accepted as the information of three dimensional images and the intensity of facial images are accepted as the information gathered from two dimensional images. Therefore, it is proven that both the union of range and intensity is the best for determining gender and ethnic identity.

As a summary, this paper has mentioned on finding gender and ethnical identities. While doing this work, researchers used range and the intensity of human face images. The information of range has been gathered from three dimensional facial images and used at determining ethnicity. Then combining both range and intensity of images of scanned faces, gender and ethnical identification are completed. As a result of experiments the outcomes have given satisfied conclusions.

A Unified Learning Framework for Real Time Face Detection and Classification : is a research done by Gregory Shakhnarovich, Paul A. Viola and Baback Moghaddam in the USA. In this paper, the researchers worked on a system that analyzes and decides combined and strong real time face detection which can be used at further works that is related to inhabitants. The researchers Viola and Jones have developed an algorithm for face detection in another research. This algorithm is also used in this paper too, because it is fast enough for this kind of work. In addition to this there is a classifier which tries to separate people according to their population and works as the same logic as face detection algorithm. This classifier is also fast and chosen because of giving a less error rate than the most known classifiers.

In this paper, it is mentioned on detecting faces automatically across time and categorizing them as their gender and ethnicity. This ethnical classifier give a result as if they are Asian or non Asian. Because of working with fast algorithms, the researchers have aimed to work the system both online and real time. Face detection algorithm has been processing for 15 images per second. After that classification starts immediately. There is no need for any calibration or correction adjustment, because the classifier used in this paper works very effectively. The purposed augmentation of this study is classifying the population correctly and working fast in real time which can be used in the real world.

The main processes that have been worked on this paper are classified as; face detection, gender and ethnical identification. They can be summarized as below.

First of all face detection has been a challenging issue for many years. In the algorithms that have been developed use neural network techniques in the logic of detection. Because of that reason, this situation makes the system work slower. On the other hand, the Bayesian method works slowest but has the highest correct detection. However the algorithm that has been developed by Viola and Jones 10 times faster than the other algorithms and gives more correct results.

Secondly, classifying a human as a male or a female is an issue that has been working on since 1990s. Also neural network systems are used in this classification too. However the image should be taken from frontal view. A new algorithm has been inspired from Quinlan’s algorithm and developed by the researchers

Gutta, Phillips and Wechsler. In this algorithm, the method uses neural network techniques (RBF) and decision trees. The database of the images used for this experiment has been obtained from the FERET. Their original size were 384 x 256 pixels, but normalized to 64 x 72 pixels for this work and algorithm. After operating the algorithm, the results gave the gender classification with an error rate of approximately 4%. Another similar experiment done by Moghaddam and Yang was studied on images that have low resolution and 21 x 12 pixels from the same database. They have also used cross validation technique and support vector machines. Then in the end, they have reached a solution with an error rate at 3.4% by guessing 1044 males and 711 females of 1755 people.

Finally, ethnical classisfication is mentioned on this paper. While the researches have studied on this paper there is only a few number of works done that are related to finding ethnicity. Also the studies must have been worked on facial images that have the same scale. An example can be given for such a previous work. It was done by again Gutta. He has used RBFs and decision trees. In this paper, an easier and better working classifier which depends on binary classification has been used. It divides the people as they are Asian or non Asian.

An extra classification technique studied on this paper is both gender and ethnicity classifiers. The working principle can be summarized like this. Both gender and ethnical identity detection have two classes; male-female and Asian-non Asian. So, the facial image is taken as an input **x.** Then its output will be $f(\mathbf{x})$ and must have a scalar value. The class is detected by the polarity which is the sign of $f(\mathbf{x})$. The exact decision is taken by taking the magnitude $||f(\mathbf{x})||$. The working principles of binary classifiers are generally the same as described above. On the other hand, Fisher classifier, Quadratic classifier or Linear classifier are density based classifiers and work with a different way from binary classifiers. In these kinds of classifiers the output function gives a logarithmic ratio. RBF, Nearest-Neighbor and SVM are kernel based classifiers. The output of these ones can be defined as the distance from boundary. The researchers have accepted the output classifier as a union of simple and single discriminants. Finally binary classifier is used especially for more complicated algorithms like video images.

As a result, the work done in this paper can be encapsulated like below. The researchers have studied on ethnical identification. To reach this aim, the system should work in order which starts with face detection first. Then it goes on identifying gender and ethnicity. The classification gives a result as if a human is Asian or non

Asian. Also the comparisons of classifiers with respect to SVM are inspected in the paper and the results are given in a comparative, detailed way.

Multimodal 2D and 3D Facial Ethnicity Classification : Guangpeng Zhang, Yunhong Wang made a study in China. Ethnic identity is a very important property that gives information about the history of folks. Because of getting a developing subject this issue has an increasing importance. Especially detecting ethnicity automatically from face detection is needed for using in different kind of areas. For that reason, the researchers have studied on both two dimensional and three dimensional face detection and ethnical identification on this paper. A multi scale and multi ratio local binary pattern method which is called MM LBP is applied for ethnical identification. The working style of this technique is based on histograms. The multi scale and multi ratio rectangular regions are detected first, then the histograms are outdrawn to texture and range of the images. Moreover, one of the machine learning techniques Adaptive Boosting also called as Adaboost is used here. This technique uses LBP histograms and is chosen because of classifying powerfully. In the last step is decision takes place. The database used for this paper is applied on FRGC. This database is chosen because both the two dimensional and three dimensional facial images are combined correctly and works suitable with the MM BBP algorithm with an optimum performance for ethnical identification. At the end of the experiments, the obtained result at determining ethnicity has a success ratio of more than 99.5%.

Face detection and ethnical identification from this detection have very different application areas. Especially making this work automatically is used at wide ares like airports, supermarkets etc. It is very useful and gives hints about finding wanted people. Three dimensional shapes of human faces consist of information about their racial background. For example Asian people have slanting eyes or some races have characteristic and sharp face lines. Nowadays, with the help of developing technologies, three dimensional facial images are captured by sensors and converted into two dimensional images. These are textured. The researchers have mentioned about an ethnical identification technique on this paper such that ethnicity can be appointed either it is two diemensional or three dimensional. This is named as multimodal identification. MM LBP is technique that gives an opportunity of classifying ethnical identity correctly. The angle of an image or lighting can change the image. The view can have a different look. But this MM LBP method helps to prevent this and satisfies giving correct decision.

Multi scale, multi ratio local binary pattern is a very important algorithm. The working of this algorithm takes place as mentioned below. LBP is used at identifying texture. The gray scale is constant so that can be used at texture. In LBP, the first step is taking a pixel that has been accepted as centre. Then 3x3 neighbour pixels are selected and used for making binary numbers. These binary numbers are converted to their equivalent decimal numbers. The acquired value after these operations represents a label the central pixel. Then the histograms of these labels are obtained and represent the texture defined above. This process allows LBP to form homogenous. Also any center pixel can be chosen with any wanted number of neighbor pixels or radius. Because of being homogenous and binary, it let having transition values between zero and one. Here, the main aim of using LBP is determining face, but in the future it can be developed and used for defining facial expressions or sentimental expressions. The texture images can be transformed into range images, because LBP is very strong and powerful. Therefore, finding disjoint and separated areas for depicting texture and figural discrepancies can be possible with the use of multi scale, multi ratio LBP.

In the experimens done for this paper, the success of ethnical classifiers is affected by the union of texture and range properties. As the result of the experiments, there are some integrity results obtained. If there is a comparison in consonance with the results, the range properties are more successful than texture properties in ethnical identification. The last step, decision level fusion has increased the success too. The ethnical identification is tested on 100 experimental according to these subjects and the result gives an error ratio 1.56% for texture, 1.19% for range and 0.42% for decision level.

Shortly, the researchers have studied on finding ethnical identification from both two dimensional and three dimensional facial images in this paper. An algorithm named MM LBP is also used in the works done. This is realized by extracting the histograms rectangularly using multi scale and ratio. Moreover Adaboost is utilized to increase the correctness of the system. Successful results are obtained.

Ethnicity is very important property because of giving information about folks' background. These ethnical specialities will stay with us all our lives and go through to our children. In addition to this, the ethnical identities can be guessed from people's facial appearances. However this is a little bit more complicated to realize this with the computer. Face detection, normalization, classification and identification are different problems to be solved. After researching and studying on different related works done by different people from all over the world, it gave me different ideas about my thesis work. In the studies made before had used two dimensional or texture method generally, because it is known that distinct races have distinct texture images. In addition to this, there are also some methods trying to explain the problem by using 3D images. All of the works done in these papers are defined in detailed above. My work is related to these works as being used 2D photos by explaining the problem.

Moreover, with the formation of the World there has been appeared many different races and ethnicities. Most of the different races have very distinct characteristics that can be realized at first look, but some can not be understood. After defining the faces it is also difficult to make computers decide the accurate differences of different ethnicities. For that reason, there is very little number of works done before in the area of facial ethnical classification. Also most of these work make a decision between the races of Asian and non Asian. My thesis work has an improvement that differs from these previous works. It has developed a face detection method and decided the ethnical identities of faces as if they are Asian, African or European.

Finally, I want to mention another technique for deciding ethnic identity. The new DNA tests are developed and sold at markets especially in the USA. They give the most correct answer about a person's ethnic identity. This system works with the help of the DNA which is found in the blood. A drop of blood from the person is taken and put into the tester kit. Then it gives the answer of which the ethnicity is that person belongs to.

2.3. METHODS USED IN THE THESIS

In this thesis, the most important hardware used is a good working camera. For making the right calculations and finding the result which is very close to the real life, some restrictions must be defined. One of them is related to the camera. It must have HD 4.0, 16.2 Megapixel and 4.9 zoom properties. Also the distance between the person and camera is a restriction for the user, but needed to find the exact solution.

The software part of the thesis is more complex than the hardware part. It has two steps. In the first step, the photo must be perceived if it is face or not. To overcome this problem, Luxand is used. After defining the program the 66 points of face, it firstly finds the points of facial parts and then determines if it is face or not. Then the computer calculates the size of human face. In the second step, C# is used for programming. It uses patterns for the ethnic roots defined and be proved in scientific works. It compares the skin color from the photo’s pixel darkness. And compares the facial parts one by one and combined together.

CHAPTER III

3.1. IMPLEMENTATION

The first implementation is done in Luxand SDK, by defining the 66 points of face. The SDK satisfies the places and situations of 66 facial points that contains eyebrows, eyes, nose, mouth and face contours. It has strong, powerful frontal face detection and fixation of multi faces in an image. It is easy to configure and use. Luxand works very fast in calculation, detection and comparison. This program is chosen fort his thesis, because it is very reliable and compatible with .NET which will be used later. FaceSDK provides detecting facial features in an image by just detecting the eyes first. First, these functions find a front view of face in a photo, then detect its facial properties by using (x,y) coordinates of face center. Then the efficiency and trustfulness of face identification is controlled by some detection parameters and Threshold functions used in Luxand SDK. So it detects a front view of face in a photo and saves information about the face and enables easy manipulation of images.

After detecting the face and calculating the sizes of facial parts, it is time for finding the ethnical identities of human faces. According to the race specifications, some patterns are defined for African, Asian and European people. By making comparisons, the program finds and decides about the photo, which ethnical root he or she belongs to.

Some parts of the program are written in Turkish, however their English meanings are also defined in Chapter 1, Introduction part. This C# code is also combined the codes in Luxand SDK. After running the program it gives correct results for the entered photos which are taken under the restricted conditions that are defined before.

3.2. EXAMPLES AND RESULT

For an African face type, the face is like shown below.

Figure 7 - African Face Type Match with Mask

After working the program it detects the face and ethnical identity by taking the mask below.

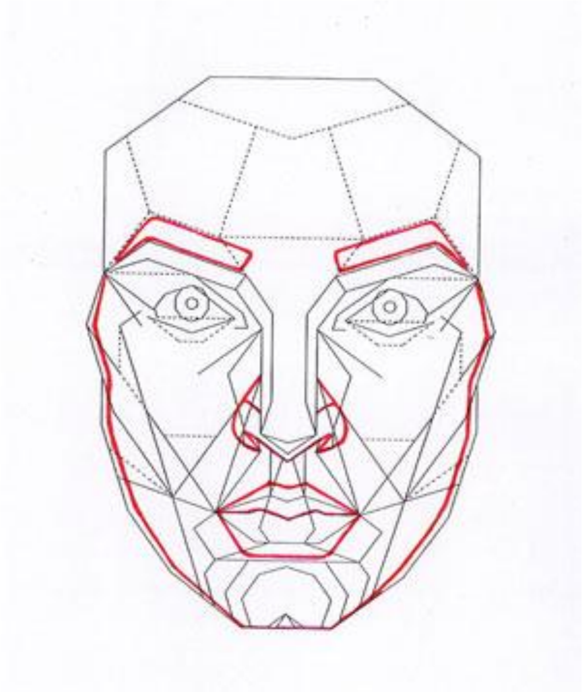


Figure 8 - African Face on Mask

For an Asian face type, the face is like shown below.

Figure 9 - Asian Face Type Match with Mask

After working the program it detects the face and ethnical identity by taking the mask below.

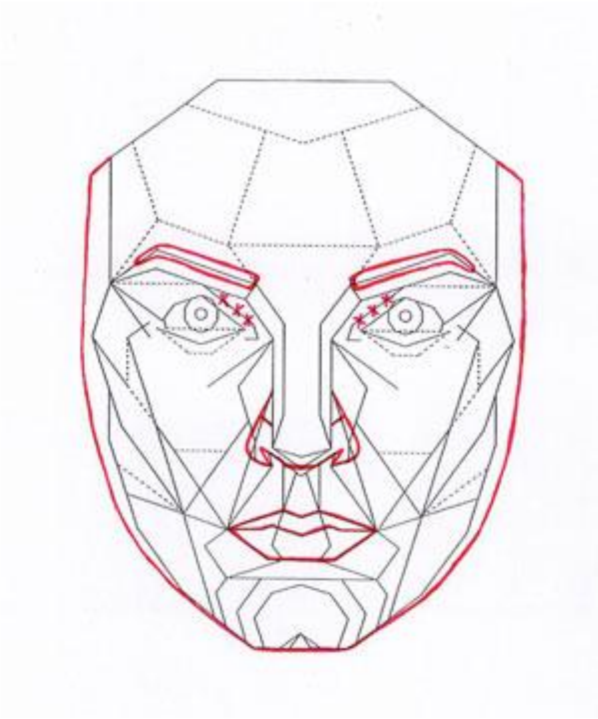


Figure 10 - Asian Face on Mask

For a European face type, the face is like shown below.

Figure 11 - European Face Type Match with Mask

After working the program it detects the face and ethnical identity by taking the mask below.

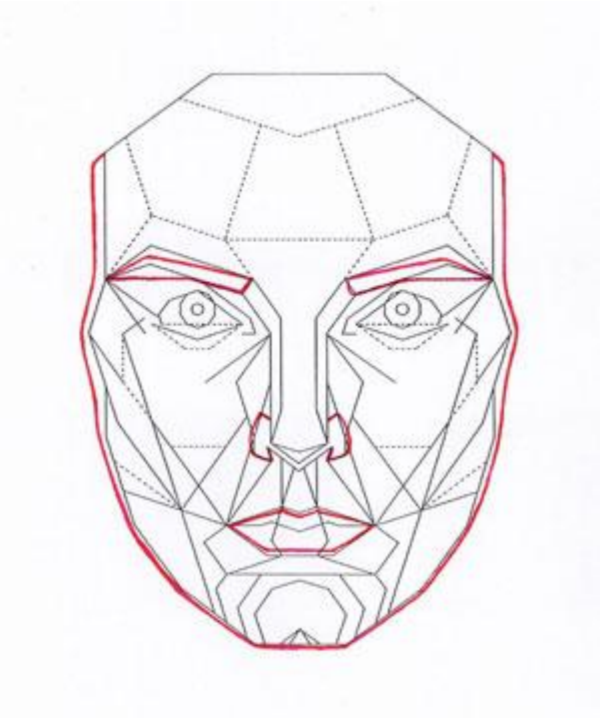


Figure 12 - European Face on Mask

The most reliable age interval is between 20 and 50 years old. Determining the ethnical identity is given in type of percentage. These are defined as %0, %25, %50, %75 and %100. For example, the ethnical identity is accepted as the highest percentage ratio. When the ratio is %50, the program fails. In my thesis work, I have tried 24 photos from different ethnical identities. Some of them gave %100, some gave %75. However only one sample gave %50 Asian, %50 European. So the program failed only in this sample. As a result the program works with a accuracy more than %95.

CONCLUSION

First of all, I have chosen this subject for my master thesis, because it is related to my field of interest. The study is very useful for me. It consists of both image processing and pattern recognition, so I have learned lots of things about these subjects. Of course, some difficulties are occurred while working. The most difficult part was changing the pixel value to centimeters (cm) for taking the correct result. To overcome this problem, some restrictions are applied like the specifications of the camera and the distance between the camera and the person who is taken photo. After making some trials, the correct result is obtained as explained in the chapters above.

There are some disadvantages or points that have to be cared while working on this thesis. The program can give error or wrong result for children, old people whose faces are degenerated. Also it can give wrong result for ill people like down syndrome because they have a specific face type that ruins the rule defined in thesis. In addition to this, people who have plastic surgery aren't suitable for this project, because some characteristics of facial parts lose their properties. Another disadvantage can be the restriction of ethnical identities with three. However it is nearly impossible under the technology that is used today. As described before only with the help of DNA testing, the ethnical identities can be detected.

The advantages of this thesis take away the disadvantages. "Thee ethnical identities" is an acceptable value for such a work all over the world. It also can be adapted into many different areas for daily use. For example it can be used at airports or undergrounds for security. It is an alternative at police departments. Finally, I want to mention games and mobile applications. The facial and ethnical determination can be interesting in these areas, too. In the further step, this subject can be developed and used in much more areas like anthropology or plastic surgery.

All of the researches and applications that are worked in this thesis are applied under different conditions to find the right solution. After many trials, the optimum conditions are fixed. Taking photos under defined restrictions and applying the program for different ethnical faces always give the correct result for this thesis.

It can be said that the program gives the correct result and the thesis study has reached it aim.

REFERENCES

[1] Luxand - Download, Luxand Face SDK 4.0, http://www.luxand.com/download/, (Released: 22/07/2011)

[2] Repose Frontal View, Beauty Analysis, http://www.beautyanalysis.com/mba_reposefrontalviewethnicvariations_page.htm

[3] The facial propositions of beautiful people, Majority Rights, http://majorityrights.com/weblog/comments/the_facial_proportions_of_beautiful_people

[4] Photo Instruction, Fas Diagnostic & Prevention Network, http://depts.washington.edu/fasdpn/htmls/photo-face.htm

[5] A Model's Secrets: The Perfect Face - Golden Ratio Beauty Calculator, A Model's Secrets by Shelley GoodStein, http://facethis.blogspot.com/2012/01/perfect-face-golden-ratio-beauty.html

[6] Natalie Portman Face Left Profile, Mega WallPaper, http://www.mega-wallpaper.com/31032_natalie_portman_face_left_profile.html

[7] Racial Differences in Parkinson Disease, Viartis, http://www.viartis.net/parkinsons.disease/news/090506.htm

[8] Xiaoguang Lu and Anil K. Jain, Ethnical Identification from Face Images, Department of Computer Science & Engineering, Michigan State University, East Lansing, MI 48824, http://www.cse.msu.edu/biometrics/Publications/Face/LuJain_EthnicityFace_SPIE04.pdf

[9] Samarasena Buchala, Neil Davey, Tim M. Gale, Ray J. Frank, Principal Component Analysis of Gender, Ethnicity,
Age, and Identity of Face Images, Department of Computer Science, University of Hertfordshire, College Lane, Hatfield, AL10 9AB, UK, Department of Psychiatry, QEII Hospital, Welwyn Garden City, AL7 4HQ, UK.
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.59.2853&rep=rep1&type=pdf

[10] Srinivas Gutta, Harry Wechsler , Gender and Ethnic Classification of Human Faces Using Hybrid Classifiers, NY 10510, Department of Computer Science George Mason University Fairfax, VA 22030, http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00830815

[11] Xianchao Qiu, Zhenan Sun, and Tieniu Tan, Learning Appearance Primitives of Iris Images for Ethnic Classification, NLPR, Institute of Automation, Chinese Academy of Sciences, China, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4379178

[12] Xiaoguang Lu, Hong Chen, and Anil K. Jain, Multimodal Facial Gender and Ethnicity Identification, Michigan State University, East Lansing, MI 48824, http://pdf.aminer.org/000/274/145/multimodal_facial_gender_and_ethnicity_identification.pdf

[13] Gregory Shakhnarovich, Paul Viola, Baback Moghaddam, A Unified Learning Framework for Real Time Face Detection & Classification, Mitsubishi Electric Research Laboratories, May 2002, http://www.merl.com/publications/docs/TR2002-23.pdf

[14] Integrating Faces Fingerprints and Soft Biometric Traits for User Recognition, http://pdf.aminer.org/000/168/694/integrating_faces_fingerprints_and_soft_biometric_traits_for_user_recognition.pdf

APPENDIX I - SOME PARTS OF CODES

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Luxand;
using System.Collections;

namespace Find_Ethnical_Identity
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }



        public string imageLocation = null;
        ArrayList pointsArray = new ArrayList();
        ArrayList pointPlacesArray = new ArrayList();
        FaceFeaturesSize faceFeaturesSize = new FaceFeaturesSize();
        public double onepixel = 24.23;
        FSDK.TFacePosition facePosition;
        private void Form1_Load(object sender, EventArgs e)
        {
            if (FSDK.FSDKE_OK !=
FSDK.ActivateLibrary("NEhE0W8svetqEJTww/8OqRIZYPYOjYHdRZ8tfNg/R+/9hg6CRGxN0DZV
Xdofuqdw9IIqkzNFRsZ8ZGL5KaFAMDX0mTgKJniR5OTISntYI2eCh0ArdvLMPUUxYLh4v96rJMSogd
zxiwrFNtTfqMDaSDxYLvx74asbWhFmMpjYZNE="))
            {
                MessageBox.Show("Please run the License Key Wizard (Start -
Luxand - FaceSDK - License Key Wizard)", "Error activating FaceSDK",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                System.Windows.Forms.Application.Exit();
            }


            FSDK.InitializeLibrary();
            FSDK.SetFaceDetectionParameters(false, true, 384);
        }

        private void btnLoadImage_Click(object sender, EventArgs e)
        {
            OpenFileDialog openFile = new OpenFileDialog();
            if (openFile.ShowDialog() == DialogResult.OK)
```

```
{
                FSDK.CImage image = new FSDK.CImage(openFile.FileName);
Image frameImage = image.ToCLRImage();
                inputpictureBox.Image = frameImage;
                inputpictureBox.Refresh();
                imageLocation = openFile.FileName;
            }

            PictureBox layer2 = new PictureBox();
            layer2.Width = 512;
            layer2.Height = 384;
            layer2.ImageLocation = "d:\\belgelerim\\visual studio 
2010\\Projects\\Find_Ethnical_Identity\\Find_Ethnical_Identity\\europ.png";
            ShowFaceFeatures();
            calculateFaceFeatures(facePosition);
            outputpictureBox.Controls.Add(layer2);

        }

        private void ShowFaceFeatures()
        {

            pointsArray.Clear();
            pointPlacesArray.Clear();

            FSDK.CImage image = new FSDK.CImage(imageLocation);
            Image frameImage = image.ToCLRImage();
            Graphics gr = Graphics.FromImage(frameImage);
            facePosition = image.DetectFace();
            if (0 == facePosition.w)
                MessageBox.Show("No faces detected", "Face Detection");
            else
            {
                int left = facePosition.xc - facePosition.w / 2;
                int top = facePosition.yc - facePosition.w / 2;
                gr.DrawRectangle(Pens.LightGreen, left, top, facePosition.w + 
10, facePosition.w + 10);
                gr.DrawRectangle(Pens.Red, left, top, 1, 1);
                gr.DrawRectangle(Pens.Yellow, facePosition.xc, 
facePosition.yc, 1, 1);
                gr.DrawRectangle(Pens.Yellow, facePosition.xc - 
(facePosition.xc - left) * 81 / 100, facePosition.yc, 1, 1);
                gr.DrawRectangle(Pens.Yellow, facePosition.xc + 
(facePosition.xc - left) * 81 / 100, facePosition.yc, 1, 1);



                FSDK.TPoint[] facialFeatures = 
image.DetectFacialFeaturesInRegion(ref facePosition);

                int i = 0;
                foreach (FSDK.TPoint point in facialFeatures)
                    gr.DrawEllipse((++i > 2) ? Pens.Red : Pens.Blue, point.x, 
point.y, 1, 1);

                findPlaces(facialFeatures);
                gr.DrawRectangle(Pens.Yellow, 
((PointPlaces)pointPlacesArray[22]).point.X, 
(int)(((PointPlaces)pointPlacesArray[22]).point.Y * 0.95), 1, 1);
                gr = circleMaskPoints(gr);
```

```
                gr.Flush();

            }

            // display image
            outputpictureBox.Image = frameImage;
            outputpictureBox.Refresh();

        }

        private Graphics circleMaskPoints(Graphics gr)
        {
            MaskPoints maskPoints = new MaskPoints();
            gr.DrawEllipse(Pens.Blue, maskPoints.solKasBaslangiciSagdan.X,
maskPoints.solKasBaslangiciSagdan.Y, 10, 10);

            return gr;
        }

        public void calculateFaceFeatures(FSDK.TFacePosition facePosition)
        {
            faceFeaturesSize.ikiGozArasi =
Math.Sqrt((((PointPlaces)pointPlacesArray[24]).point.X -
((PointPlaces)pointPlacesArray[25]).point.X) *
(((PointPlaces)pointPlacesArray[24]).point.X -
((PointPlaces)pointPlacesArray[25]).point.X) +

(((PointPlaces)pointPlacesArray[24]).point.Y -
((PointPlaces)pointPlacesArray[25]).point.Y) *
(((PointPlaces)pointPlacesArray[24]).point.Y -
((PointPlaces)pointPlacesArray[25]).point.Y)) / onepixel;

            faceFeaturesSize.sagGozGenisligi =
Math.Sqrt((((PointPlaces)pointPlacesArray[23]).point.X -
((PointPlaces)pointPlacesArray[24]).point.X) *
(((PointPlaces)pointPlacesArray[23]).point.X -
((PointPlaces)pointPlacesArray[24]).point.X) +

(((PointPlaces)pointPlacesArray[23]).point.Y -
((PointPlaces)pointPlacesArray[24]).point.Y) *
(((PointPlaces)pointPlacesArray[23]).point.Y -
((PointPlaces)pointPlacesArray[24]).point.Y)) / onepixel;

            faceFeaturesSize.solGozGenisligi =
Math.Sqrt((((PointPlaces)pointPlacesArray[25]).point.X -
((PointPlaces)pointPlacesArray[26]).point.X) *
(((PointPlaces)pointPlacesArray[25]).point.X -
((PointPlaces)pointPlacesArray[26]).point.X) +

(((PointPlaces)pointPlacesArray[25]).point.Y -
((PointPlaces)pointPlacesArray[26]).point.Y) *
(((PointPlaces)pointPlacesArray[25]).point.Y -
((PointPlaces)pointPlacesArray[26]).point.Y)) / onepixel;

            faceFeaturesSize.agizGenisligi =
Math.Sqrt((((PointPlaces)pointPlacesArray[3]).point.X -
((PointPlaces)pointPlacesArray[4]).point.X) *
(((PointPlaces)pointPlacesArray[3]).point.X -
((PointPlaces)pointPlacesArray[4]).point.X) +
```

```
(((PointPlaces)pointPlacesArray[3]).point.Y -
((PointPlaces)pointPlacesArray[4]).point.Y) *
(((PointPlaces)pointPlacesArray[3]).point.Y -
((PointPlaces)pointPlacesArray[4]).point.Y)) / onepixel;

            faceFeaturesSize.solGozBebegiGenisligi =
Math.Sqrt((((PointPlaces)pointPlacesArray[33]).point.X -
((PointPlaces)pointPlacesArray[34]).point.X) *
(((PointPlaces)pointPlacesArray[33]).point.X -
((PointPlaces)pointPlacesArray[34]).point.X) +

(((PointPlaces)pointPlacesArray[33]).point.Y -
((PointPlaces)pointPlacesArray[34]).point.Y) *
(((PointPlaces)pointPlacesArray[33]).point.Y -
((PointPlaces)pointPlacesArray[34]).point.Y)) / onepixel;

            faceFeaturesSize.sagGozBebegiGenisligi =
Math.Sqrt((((PointPlaces)pointPlacesArray[29]).point.X -
((PointPlaces)pointPlacesArray[30]).point.X) *
(((PointPlaces)pointPlacesArray[29]).point.X -
((PointPlaces)pointPlacesArray[30]).point.X) +

(((PointPlaces)pointPlacesArray[29]).point.Y -
((PointPlaces)pointPlacesArray[30]).point.Y) *
(((PointPlaces)pointPlacesArray[29]).point.Y -
((PointPlaces)pointPlacesArray[30]).point.Y)) / onepixel;

            faceFeaturesSize.noseHeight =
Math.Sqrt((((PointPlaces)pointPlacesArray[22]).point.X -
((PointPlaces)pointPlacesArray[49]).point.X) *
(((PointPlaces)pointPlacesArray[22]).point.X -
((PointPlaces)pointPlacesArray[49]).point.X) +

(((PointPlaces)pointPlacesArray[22]).point.Y -
((PointPlaces)pointPlacesArray[49]).point.Y) *
(((PointPlaces)pointPlacesArray[22]).point.Y -
((PointPlaces)pointPlacesArray[49]).point.Y)) / onepixel;

            faceFeaturesSize.noseWidth =
Math.Sqrt((((PointPlaces)pointPlacesArray[45]).point.X -
((PointPlaces)pointPlacesArray[46]).point.X) *
(((PointPlaces)pointPlacesArray[45]).point.X -
((PointPlaces)pointPlacesArray[46]).point.X) +

(((PointPlaces)pointPlacesArray[45]).point.Y -
((PointPlaces)pointPlacesArray[46]).point.Y) *
(((PointPlaces)pointPlacesArray[45]).point.Y -
((PointPlaces)pointPlacesArray[46]).point.Y)) / onepixel;

            faceFeaturesSize.faceHeight = Math.Sqrt((facePosition.xc -
((PointPlaces)pointPlacesArray[11]).point.X) * (facePosition.xc -
((PointPlaces)pointPlacesArray[11]).point.X) +
                                     (facePosition.yc -
facePosition.w / 2 - ((PointPlaces)pointPlacesArray[11]).point.Y) *
(facePosition.yc - facePosition.w / 2 -
((PointPlaces)pointPlacesArray[11]).point.Y)) / onepixel;

            faceFeaturesSize.betweenPupils =
Math.Sqrt((((PointPlaces)pointPlacesArray[0]).point.X -
```

```
((PointPlaces)pointPlacesArray[1]).point.X) *
(((PointPlaces)pointPlacesArray[0]).point.X -
((PointPlaces)pointPlacesArray[1]).point.X) +

(((PointPlaces)pointPlacesArray[0]).point.Y -
((PointPlaces)pointPlacesArray[1]).point.Y) *
(((PointPlaces)pointPlacesArray[0]).point.Y -
((PointPlaces)pointPlacesArray[1]).point.Y)) / onepixel;

            faceFeaturesSize.betweenNares =
Math.Sqrt((((PointPlaces)pointPlacesArray[47]).point.X -
((PointPlaces)pointPlacesArray[48]).point.X) *
(((PointPlaces)pointPlacesArray[47]).point.X -
((PointPlaces)pointPlacesArray[48]).point.X) +

(((PointPlaces)pointPlacesArray[47]).point.Y -
((PointPlaces)pointPlacesArray[48]).point.Y) *
(((PointPlaces)pointPlacesArray[47]).point.Y -
((PointPlaces)pointPlacesArray[48]).point.Y)) / onepixel;

            faceFeaturesSize.betweenEyeBrowsChin =
Math.Sqrt((((PointPlaces)pointPlacesArray[22]).point.X -
((PointPlaces)pointPlacesArray[11]).point.X) *
(((PointPlaces)pointPlacesArray[22]).point.X -
((PointPlaces)pointPlacesArray[11]).point.X) +

(((PointPlaces)pointPlacesArray[22]).point.Y * 0.95 -
((PointPlaces)pointPlacesArray[11]).point.Y) *
(((PointPlaces)pointPlacesArray[22]).point.Y * 0.95 -
((PointPlaces)pointPlacesArray[11]).point.Y)) / onepixel;

            faceFeaturesSize.betweenEyeBrows =
Math.Sqrt((((PointPlaces)pointPlacesArray[13]).point.X -
((PointPlaces)pointPlacesArray[14]).point.X) *
(((PointPlaces)pointPlacesArray[13]).point.X -
((PointPlaces)pointPlacesArray[14]).point.X) +

(((PointPlaces)pointPlacesArray[13]).point.Y -
((PointPlaces)pointPlacesArray[14]).point.Y) *
(((PointPlaces)pointPlacesArray[13]).point.Y -
((PointPlaces)pointPlacesArray[14]).point.Y)) / onepixel;

            faceFeaturesSize.betweenEyeBrowsLips =
Math.Sqrt((((PointPlaces)pointPlacesArray[22]).point.X -
((PointPlaces)pointPlacesArray[61]).point.X) *
(((PointPlaces)pointPlacesArray[22]).point.X -
((PointPlaces)pointPlacesArray[61]).point.X) +

(((PointPlaces)pointPlacesArray[22]).point.Y * 0.95 -
((PointPlaces)pointPlacesArray[61]).point.Y) *
(((PointPlaces)pointPlacesArray[22]).point.Y * 0.95 -
((PointPlaces)pointPlacesArray[61]).point.Y)) / onepixel;
            faceFeaturesSize.faceWidth = (2 * (facePosition.w / 2) * 81 / 100)
/ onepixel;

            lblIkiGozArasi.Text =
faceFeaturesSize.ikiGozArasi.ToString("0.##");
            lblRightEyeWidth.Text =
faceFeaturesSize.sagGozGenisligi.ToString("0.##");
```

```
            lblLeftEyeWidth.Text = 
faceFeaturesSize.solGozGenisligi.ToString("0.##");
            lblMouthWidth.Text = 
faceFeaturesSize.agizGenisligi.ToString("0.##");
            lblLeftEyePupilWidth.Text = 
faceFeaturesSize.solGozBebegiGenisligi.ToString("0.##");
            lblRightEyePupilWidth.Text = 
faceFeaturesSize.sagGozBebegiGenisligi.ToString("0.##");
            lblNoseHeight.Text = faceFeaturesSize.noseHeight.ToString("0.##");
            lblNoseWidth.Text = faceFeaturesSize.noseWidth.ToString("0.##");
            lblFaceHeight.Text = faceFeaturesSize.faceHeight.ToString("0.##");
            lblBetweenPupils.Text = 
faceFeaturesSize.betweenPupils.ToString("0.##");
            lblBetweenNares.Text = 
faceFeaturesSize.betweenNares.ToString("0.##");
            lblBetweenEyeBrowsChin.Text = 
faceFeaturesSize.betweenEyeBrowsChin.ToString("0.##");
            lblBetweenEyeBrows.Text = 
faceFeaturesSize.betweenEyeBrows.ToString("0.##");
            lblbetEyeBrows_Lips.Text = 
faceFeaturesSize.betweenEyeBrowsLips.ToString("0.##");
            lblFaceWidth.Text = faceFeaturesSize.faceWidth.ToString("0.##");
        }

        public class FaceFeaturesSize
        {
            public double ikiGozArasi;
            public double sagGozGenisligi;
            public double solGozGenisligi;
            public double agizGenisligi;
            public double solGozBebegiGenisligi;
            public double sagGozBebegiGenisligi;
            public double noseHeight;
            public double noseWidth;
            public double faceHeight;
            public double betweenPupils;
            public double betweenNares;
            public double betweenEyeBrowsChin;
            public double betweenEyeBrows;
            public double betweenEyeBrowsLips;
            public double faceWidth;

        }

        public class MaskPoints
        {
            public Point solKasBaslangiciSagdan;
            public MaskPoints(){
                solKasBaslangiciSagdan.X = 150;
                solKasBaslangiciSagdan.Y = 50;

            }

        }

        public class PointPlaces
        {
            public string place;
            public Point point = new Point();
            public PointPlaces(string place, Point point)
```

```
            {
                this.place = place;
                this.point = point;
            }

        }

        public void findPlaces(FSDK.TPoint[] facialFeatures)
        {
            for (int j = 0; j < facialFeatures.Length; j++)
            {
                pointsArray.Add(new Point(facialFeatures[j].x,
facialFeatures[j].y));
            }


            for (int k = 0; k < pointsArray.Count; k++)
            {
                Point point = (Point)pointsArray[k];

                if (k == 0)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ göz bebeği",
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz
bebeği\r\n";
                }
                else if (k == 1)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz bebeği",
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz
bebeği\r\n";
                }
                else if (k == 2)
                {
                    pointPlacesArray.Add(new PointPlaces("Burun üst ucu",
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Burun
üst ucu\r\n";
                }
                else if (k == 3)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağdan ağız
başlangıcı 1", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağdan
ağız başlangıcı 1\r\n";
                }
                else if (k == 4)
                {
                    pointPlacesArray.Add(new PointPlaces("Soldan ağız
başlangıcı 1", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Soldan
ağız başlangıcı 1\r\n";
                }

                else if (k == 5)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağdan çene
başlangıcı 1", point));
```

```
                //textBox1.Text += point.X + " " + point.Y + "  - Sağdan
çene başlangıcı 1\r\n";
            }
            else if (k == 6)
            {
                pointPlacesArray.Add(new PointPlaces("Soldan çene
başlangıcı 1", point));
                //textBox1.Text += point.X + " " + point.Y + "  - Soldan
çene başlangıcı 1\r\n";
            }
            else if (k == 7)
            {
                pointPlacesArray.Add(new PointPlaces("Sağdan çene
başlangıcı 2", point));
                //textBox1.Text += point.X + " " + point.Y + "  - Sağdan
çene başlangıcı 2\r\n";
            }
            else if (k == 8)
            {
                pointPlacesArray.Add(new PointPlaces("Soldan çene
başlangıcı 2", point));
                //textBox1.Text += point.X + " " + point.Y + "  - Soldan
çene başlangıcı 2\r\n";
            }
            else if (k == 9)
            {
                pointPlacesArray.Add(new PointPlaces("Sağdan çene
başlangıcı 3", point));
                //textBox1.Text += point.X + " " + point.Y + "  - Sağdan
çene başlangıcı 3\r\n";
            }
            else if (k == 10)
            {
                pointPlacesArray.Add(new PointPlaces("Soldan çene
başlangıcı 3", point));
                //textBox1.Text += point.X + " " + point.Y + "  - Soldan
çene başlangıcı 3\r\n";
            }
            else if (k == 11)
            {
                pointPlacesArray.Add(new PointPlaces("Çene Ucu", point));
                //textBox1.Text += point.X + " " + point.Y + "  - Çene
Ucu\r\n";
            }
            else if (k == 12)
            {
                pointPlacesArray.Add(new PointPlaces("Sağ kaş başlangıcı
sağdan 1", point));
                //textBox1.Text += point.X + " " + point.Y + "  - Sağ kaş
başlangıcı sağdan 1\r\n";
            }
            else if (k == 13)
            {
                pointPlacesArray.Add(new PointPlaces("Sağ kaş başlangıcı
sağdan 5", point));
                //textBox1.Text += point.X + " " + point.Y + "  - Sağ kaş
başlangıcı sağdan 5\r\n";
            }
            else if (k == 14)
            {
```

```
                    pointPlacesArray.Add(new PointPlaces("Sol kaş başlangıcı
soldan 5", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol kaş
başlangıcı soldan 5\r\n";
                }
                else if (k == 15)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol kaş başlangıcı
soldan 1", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol kaş
başlangıcı soldan 1\r\n";
                }

                else if (k == 16)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ kaş başlangıcı
sağdan 3", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ kaş
başlangıcı sağdan 3\r\n";
                }

                else if (k == 17)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol kaş başlangıcı
soldan 3", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol kaş
başlangıcı soldan 3\r\n";
                }

                else if (k == 18)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ kaş başlangıcı
sağdan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ kaş
başlangıcı sağdan 2\r\n";
                }

                else if (k == 19)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ kaş başlangıcı
sağdan 4", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ kaş
başlangıcı sağdan 4\r\n";
                }

                else if (k == 20)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol kaş başlangıcı
soldan 4", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol kaş
başlangıcı soldan 4\r\n";
                }

                else if (k == 21)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol kaş başlangıcı
soldan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol kaş
başlangıcı soldan 2\r\n";
                }
```

```
                else if (k == 22)
                {
                    pointPlacesArray.Add(new PointPlaces("Burun Başlangıcı",
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Burun
Başlangıcı\r\n";
                }


                else if (k == 23)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı
sağdan 1", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz
bebeği\r\n";
                }

                else if (k == 24)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı
soldan 1", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz
başlangıcı soldan 1\r\n";
                }

                else if (k == 25)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı
sağdan 1", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz
başlangıcı sağdan 1\r\n";
                }

                else if (k == 26)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı
soldan 1", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz
başlangıcı soldan 1\r\n";
                }

                else if (k == 27)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı
alttan orta", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz
başlangıcı alttan orta\r\n";
                }

                else if (k == 28)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı
üstten orta", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz
başlangıcı üstten orta\r\n";
                }

                else if (k == 29)
                {
```

```
                    pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı
ortadan sağdan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz
başlangıcı ortadan sağdan 2\r\n";
                }

                else if (k == 30)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı
ortadan soldan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz
başlangıcı ortadan soldan 2\r\n";
                }

                else if (k == 31)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı
alttan orta", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz
başlangıcı alttan orta\r\n";
                }

                else if (k == 32)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı
üstten orta", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz
başlangıcı üstten orta\r\n";
                }

                else if (k == 33)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı
ortadan sağdan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz
başlangıcı ortadan sağdan 2\r\n";
                }

                else if (k == 34)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı
ortadan soldan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz
başlangıcı ortadan soldan 2\r\n";
                }

                else if (k == 35)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı
üstten sağdan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz
başlangıcı üstten sağdan 2\r\n";
                }

                else if (k == 36)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı
üstten soldan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz
başlangıcı üstten soldan 2\r\n";
```

```
                }

                else if (k == 37)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı 
alttan sağdan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz 
başlangıcı alttan sağdan 2\r\n";
                }

                else if (k == 38)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı 
alttan soldan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz 
başlangıcı alttan soldan 2\r\n";
                }

                else if (k == 39)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı 
üstten sağdan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz 
başlangıcı üstten sağdan 2\r\n";
                }

                else if (k == 40)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı 
üstten soldan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz 
başlangıcı üstten soldan 2\r\n";
                }

                else if (k == 41)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı 
alttan sağdan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz 
başlangıcı alttan sağdan 2\r\n";
                }

                else if (k == 42)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı 
alttan sağdan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz 
başlangıcı alttan sağdan 2\r\n";
                }

                else if (k == 43)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ burun üstü", 
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ 
burun üstü\r\n";
                }

                else if (k == 44)
                {
```

```
                    pointPlacesArray.Add(new PointPlaces("Sol burun üstü",
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol
burun üstü\r\n";
                }

                else if (k == 45)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ burun
başlangıcı", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ
burun başlangıcı\r\n";
                }

                else if (k == 46)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol burun
başlangıcı", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol
burun başlangıcı\r\n";
                }

                else if (k == 47)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ burun deliği",
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ
burun deliği\r\n";
                }

                else if (k == 48)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol burun deliği",
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol
burun deliği\r\n";
                }

                else if (k == 49)
                {
                    pointPlacesArray.Add(new PointPlaces("Burun alt ucu",
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Burun
alt ucu\r\n";
                }

                else if (k == 50)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ burun altı
üstten 1", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ
burun altı üstten 1\r\n";
                }

                else if (k == 51)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol burun altı
üstten 1", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol
burun altı üstten 1\r\n";
```

```
                }

                else if (k == 52)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ burun altı 
üstten 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ 
burun altı üstten 2\r\n";
                }

                else if (k == 53)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol burun altı 
üssten 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol 
burun altı üssten 2\r\n";
                }

                else if (k == 54)
                {
                    pointPlacesArray.Add(new PointPlaces("Üst dudak üst orta", 
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Üst 
dudak üst orta\r\n";
                }

                else if (k == 55)
                {
                    pointPlacesArray.Add(new PointPlaces("Alt Dudak alt orta", 
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Alt 
Dudak alt orta\r\n";
                }

                else if (k == 56)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağdan ağız 
başlangıcı üst dudak üstten 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağdan 
ağız başlangıcı üst dudak üstten 2\r\n";
                }

                else if (k == 57)
                {
                    pointPlacesArray.Add(new PointPlaces("Soldan ağız 
başlangıcı üst dudak üstten 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Soldan 
ağız başlangıcı üst dudak üstten 2\r\n";
                }

                else if (k == 58)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağdan ağız 
başlangıcı alt dudak alttan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağdan 
ağız başlangıcı alt dudak alttan 2\r\n";
                }

                else if (k == 59)
                {
```

```
                    pointPlacesArray.Add(new PointPlaces("Soldan ağız
başlangıcı alt dudak alttan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Soldan
ağız başlangıcı alt dudak alttan 2\r\n";
                }

                else if (k == 60)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağdan ağız
başlangıcı üst dudak alttan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağdan
ağız başlangıcı üst dudak alttan 2\r\n";
                }

                else if (k == 61)
                {
                    pointPlacesArray.Add(new PointPlaces("Üst Dudak alt orta",
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Üst
Dudak alt orta\r\n";
                }

                else if (k == 62)
                {
                    pointPlacesArray.Add(new PointPlaces("Soldan ağız
başlangıcı üst dudak alttan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Soldan
ağız başlangıcı üst dudak alttan 2\r\n";
                }

                else if (k == 63)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağdan ağız
başlangıcı üst dudak üstten 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağdan
ağız başlangıcı üst dudak üstten 2\r\n";
                }

                else if (k == 64)
                {
                    pointPlacesArray.Add(new PointPlaces("Alt dudak üst orta",
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Alt
dudak üst orta\r\n";
                }

                else if (k == 65)
                {
                    //gr.DrawEllipse(Pens.Black, point.X, point.Y, 3, 3);
                    pointPlacesArray.Add(new PointPlaces("Soldan ağız
başlangıcı alt dudak alttan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Soldan
ağız başlangıcı alt dudak alttan 2\r\n";
                }

            }
        } }}
```

APPENDIX II - CV

PERSONAL INFORMATION

Surname, Name : **YENİCE, MERVE**

Nationality : **T.C.**

Birth Date and Place : **1984 ANKARA**

Marital Status : **SINGLE**

Phone : **0536 459 4373**

E-Mail : **merveyenice@hotmail.com**

EDUCATION

DEGREE	INSTITUTION	GRADUATION DATE
UNDERGRADUATE	Atılım University	2009
HIGH SCHOOL	Arı College	2002

FOREIGN LANGUAGE

English – Advanced Level