FACE AND FRAME CLASSIFICATION USING GEOMETRIC FEATURES FOR A DATA-DRIVEN EYEGLASS RECOMMENDATION SYSTEM

Amir Zafar Asoodeh

A thesis
in
The Department
of
Computer Science

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science
Concordia University
Montréal, Québec, Canada

January 2015

Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Amir Zafar Asoodeh**

Entitled: **Face and Frame Classification using Geometric Features for a Data-driven Eyeglass Recommendation System**

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

Dr. E. Shihab ______ Chair
Dr. C. Y. Suen ______ Examiner
Dr. S. P. Mudur ______ Examiner
______ Examiner
Dr. T. Popa ______ Supervisor
______ Co-supervisor

Approved ______
Chair of Department or Graduate Program Director

______ 20 ______ ______

Abstract

Face and Frame Classification using Geometric Features for a Data-driven Eyeglass Recommendation System

Amir Zafar Asoodeh

Recommending glasses based on face and frame features is the main issue of this thesis. In this work we present an automatic shape extraction and classification method for face and eyeglass frame shapes. Our novel frame shape extraction algorithm can extract the polygonal shape of the frame accurately and reliably even for reflective sunglasses and thin metal frames. Additionally, we identify the key geometric features that can differentiate reliably the shape classes and we integrate them into a supervised learning technique for face and frame shape classification. Finally, we incorporate the shape extraction and classification algorithms into a practical data-driven frame recommendation system that we validate empirically with a user study. Using a supervised learning technique, we identified the geometric discriminatory features that can be used to classify both the face type and the eyeglass type form a single photograph. Our classification method reaches near 90% accuracy. We ran this classification on over 200 photographs and we surveyed 100 people on the compatibility between face and eyeglasses. Using this data we created an eyeglass recommendation system that we have validated experimentally.

Acknowledgments

My greatest appreciations to my parents and brother for their constant support and love in my endeavors throughout my lifetime.

I am grateful to my supervisor, Dr. Tiberiu Popa for initiating this research study as well as for his continued technical guidance during the completion of this thesis work.

I would also wish to acknowledge all my friends who have volunteered their help and their great corporation during the experimental stages of this work.

Contents

Chapter 1

Introduction

Many people require glasses for vision correction, protect against damage from the sun or other harmful causes. On the other side, glasses are worn by many people simply for aesthetic or fashion purposes.

There are many shapes, colors, and materials that can be used when designing frames and lenses that can be utilized in various combinations. Oftentimes, the selection of a frame is made based on how it will affect the appearance of the wearer. Some people with good natural eyesight like to wear eyeglasses as a style accessory.

According to the Vision Council of America [1] despite the gain in popularity of laser surgery and contact lenses, still, around 64% of the population wears corrective eyeglasses and nearly half requires them all the time. Additionally to the corrective eyeglasses, in this day and age, sunglasses are ubiquitous accessories recommended by physicians for comfort and protection against the sun in both summer and winter. While it is difficult to estimate globally, it is fair to say that eyeglasses, both corrective and for sun protection, are likely the most popular facial accessory if both genders are considered. Moreover, some older studies suggested that eyeglasses decreases self-esteem and the self-perception of beauty [2].

As a result, in our day and age eyeglasses and sunglasses have become a fashion items with iconic designs and an overwhelming supply of hundreds of frames available in both on-line and tradition retail stores[3, 4, 41], opening the path to many frame recommendation systems [3–6]. Most recommendation systems are generally procedural based on some a priori aesthetic rules and not necessarily based on real data. These rules are opaque to the user, occasionally contradicting each other and are based on rigid aesthetic principles that does not take into account current fashion, age or culture. A more flexible, data-driven recommendation framework is more desirable that can aggregate large databases of faces and frames.

In this work we present a scalable data-driven eyeglass recommendation system that

uses a survey on a large database of pictures to find the popular combinations of face and frame shapes. The recommendation system has two phases: a training phase and a testing phase. In the training phase, a database of pictures featuring subjects wearing glasses is created. Next, the face and frame shape of the subjects are automatically extracted and classified in a set of standard face and frame types.
A survey is then collected where a set of participants are asked to rate the aesthetic compatibility between the face and frame shapes.
In the testing phase, the input is an image of a person without wearing glasses. The face shape is automatically extracted and a recommendation of one or two frame shapes is made based on the most popular frame shapes associated with the subject's face shape obtained from the survey. We validated our recommendation results in a follow up survey where we used an on-line virtual eyeglass try-on system to recommend frames.
Overall, our contributions in this work are:

- ∈ An automatic face type classification system based on a single photograph.
- ∈ An automatic polygonal extraction of the frame shape from a single photograph.
- ∈ An automatic eyeglass type classification system based on a single photograph.
- ∈ A data driven eyeglass recommendation system.

In the next chapters, after an introduction of tools which used to develop the system, previous related research is explained. Afterwards, comprehensive descriptions of subsystems (face and frame extraction and classification systems) are addressed. Finally, the results of the research are reported and a conclusion presented.

Chapter 2

Background and Literature Review

It is said that the face is "the mirror of the soul" as indeed the face can give a lot of clues about a person such as emotions, body language and identity. Moreover, in social contexts, the perceptual aspect of facial attractiveness is very important and has been studied thoroughly by social psychologists [7–10] and has many application in medical fields such as plastic surgery [11] and in the fashion industry where it can be used to determine the most attractive hairstyle [12, 13], makeup [14, 15] and eyeglasses [3–6] for a specific person.

In tandem with social and medical disciplines, computational disciplines have tried to quantify and automate the detection of salient facial features. For instance, some work focuses on quantifying the attractiveness of a person automatically based on a single photograph [16–18]. Furthermore, applications in face detection, face feature extraction and authentication based on a single photograph has received a lot of attention [19–24]. In this work the main focus is on the aesthetic correlation between the shape of the face and the shape of eyeglasses. We employ a data-driven approach that has at its core a robust and fully automatic face and automatic frame shape extraction and classification methods.

While recent state of the art face extractors [22]are very reliable to automatically detect the important features such as face boundary and salient points such as the eyes from a single photograph, face shape classification still remains a difficult problem. Many general shape classification methods have been proposed [25–28]. Depending on the flavor, they work well if the shapes are significantly different to each other. However, In the case of faces, the shape variation from one category to another is often very subtle. Furthermore, some face shapes cannot be classified as one shape as they can be in fact a blend between two standard shapes. Therefore, a tailored approach is required in order to differentiate the subtle characteristic of each shape category and to account for the blending of the shapes.

The first practical eyeglass shape extraction method was introduced by [29]. The problem is formulated as an optimization problem on a set of active contours using the edges detected in the image. Their method is not very reliable with only about 50% of the frames are well reconstructed. A follow up technique [29] improves slightly the accuracy by computing a probability that an edge is part of the frame. This probability is learned from a set of examples. The reliability is still not very high and this probability has to be learned from a large class of glasses, which requires content updates whenever new glasses are designed. Several methods are targeted at eyeglass detection [2, 30, 31] or eyeglass removal from the image [32–35]. These methods typically do not extract an accurate outline of the glass shape and thus are not suitable for our application. A purely geometric method that requires no a priori knowledge was developed by [36], based on delaunay triangulation of the edges in the image. The results are not very clean and usually contain noise and outliers, which can further down the pipeline affect the shape classification. More recently, Borza et al. [37] presented a reconstruction method based on template matching of a database of preexisting shapes. This is inadequate for our application as any new glass designs will have to constantly be introduced to the database.

The frame classification problem suffers from the same challenges as the face classification: shape changes between different classes are subtle, making it difficult to differentiate using a generic shape classification algorithm. A more accurate and robust method tailored for this specific problem is desirable.

We chose a case based reasoning (CBR) framework approach that is suitable for both the face shape classification as well as the frame shape classification. The rest of the thesis is organized as follows: Section 3 presents an overview of the CBR framework. Section 4 presents the face classification method. Section 5 presents the frame shape extraction and classification. Section 6 presents our experiments and discussion and section 7 presents the conclusions and limitations.

We approach both our classification problems (face and frame shape classification) within a case-based reasoning framework (CBR) [24, 38] . A CBR framework consists of a training set and a query set defined over a feature space endowed with a distance metric. The classification of the elements of the training set are assumed to be known and the problem is to find the classification of the elements in the query set. The classification of a query element is based on a nearest search in the training set. This framework is very efficient and scales well to large dataset making it suitable for our classification problem.

There are some necessary tools that we use to implement this idea. The followings are the most essential components.

2.1 OpenCV

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. It focuses mainly on real-time image processing. OpenCV is written in C++ and some of its application areas are:

- ∈ 2D and 3D feature toolkits
- ∈ Facial recognition system
- ∈ Gesture recognition
- ∈ Motion tracking
- ∈ Human computer interaction (HCI)

We choose OpenCV by reason of its acceptable efficiency and speed in C++ environment. In this research, OpenCV roles proceed these targets:

- ∈ Image Edge Detection
- ∈ Image Contour Recognition
- ∈ Fitting geometric shapes to sets of 2D points
- ∈ Approximating the area of irregular convex shapes
- ∈ Calculating the convex hull of sets of points
- ∈ Measuring the symmetricity of closed polygons

Hence, OpenCV is the main C++ library that we used to implement the system.

2.2 Edge Detection Algorithms

Edge detection is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically

organized into a set of curved line segments termed edges.
Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.
The purpose of detecting sharp changes in image brightness is to capture important events and changes in properties of the world. It can be shown that under rather general assumptions for an image formation model, discontinuities in image brightness are likely to correspond to:

- ∈ discontinuities in depth.
- ∈ discontinuities in surface orientation.
- ∈ changes in material properties.
- ∈ variations in scene illumination.

In the ideal case, the result of applying an edge detector to an image may lead to a set of connected curves that indicate the boundaries of objects, the boundaries of surface markings as well as curves that correspond to discontinuities in surface orientation. Thus, applying an edge detection algorithm to an image may significantly reduce the amount of data to be processed and may therefore filter out information that may be regarded as less relevant, while preserving the important structural properties of an image. If the edge detection step is successful, the subsequent task of interpreting the information contents in the original image may therefore be substantially simplified.
Edges extracted from non-trivial images are often hampered by fragmentation, meaning that the edge curves are not connected, missing edge segments as well as false edges not corresponding to interesting phenomena in the image thus complicating the subsequent task of interpreting the image data.
There are many methods for edge detection, but most of them can be grouped into two categories, search-based and zero-crossing based. The search-based methods detect edges by first computing a measure of edge strength, usually a first-order derivative expression such as the gradient magnitude, and then searching for local directional maxima of the gradient magnitude using a computed estimate of the local orientation of the edge, usually the gradient direction. The zero-crossing based methods search for zero crossings in a second-order derivative expression computed from the image in order to find edges, usually the zero-crossings of the Laplacian or the zero-crossings of a non-linear differential expression. As a pre-processing step to edge detection, a smoothing stage, typically Gaussian smoothing, is almost always applied (see also noise reduction).
The well-known Edge Detection Algorithms are as follows:

- ∈ Canny method: detects a wide range of edges in images

- ∈ Sobel method: the output would be an image which emphasizes edges and transitions.
- ∈ Prewitt method: at each point in the image, the result of the Prewitt operator is either the corresponding gradient vector or the norm of this vector.
- ∈ Roberts method: approximate the gradient of an image through discrete differentiation which is achieved by computing the sum of the squares of the differences between diagonally adjacent pixels

In this research, the prominent Canny Edge Detection method is used as a part of frame extraction mechanism by reason of its reliability and accuracy.

2.3 Canny Edge Detection

Canny's aim is discovering the optimal edge detection algorithm. In this situation, an "optimal" edge detector means:

- ∈ Low error rate: Meaning a good detection of only existent edges.
- ∈ Good localization: The distance between edge pixels detected and real edge pixels have to be minimized.
- ∈ Minimal response: Only one detector response per edge.

To satisfy these requirements Canny used the calculus of variations a technique which finds the function which optimizes a given functional. The optimal function in Canny's detector is described by the sum of four exponential terms, but it can be approximated by the first derivative of a Gaussian.
Canny edge detection is a four step process:

1. A Gaussian blur is applied to clear any speckles and free the image of noise.
2. A gradient operator is applied for obtaining the gradients' intensity and direction.
3. Non-maximum suppression determines if the pixel is a better candidate for an edge than its neighbors.
4. Hysteresis thresholding finds where edges begin and end.

In Opencv, the Canny is developed and can be used by calling the method below:

$void\, Canny(InputArrayimage, OutputArrayedges, doublethreshold1, doublethreshold2$
$, intapertureSize = 3, boolL2gradient = false)$

The parameters are explained as follows:

- $\in$ $image$: single-channel 8-bit input image.
- $\in$ $edges$: output edge map; it has the same size and type as image .
- $\in$ $threshold1$: first threshold for the hysteresis procedure.
- $\in$ $threshold2$: second threshold for the hysteresis procedure.
- $\in$ $apertureSize$: aperture size for the Sobel() operator.
- $\in$ $L2gradient$: a flag, indicating whether a more accurate $L_2 norm = \sqrt{(dI/dx)^2 + (dI/dy)^2}$ should be used to calculate the image gradient magnitude ($L2gradient = true$), or whether the default $L_1 norm = \; dI/dx + dI/dy$ is enough ($L2gradient = false$).

The function finds edges in the input image image and marks them in the output map edges using the Canny algorithm. The smallest value between threshold1 and threshold2 is used for edge linking. The largest value is used to find initial segments of strong edges. The output edge is used as the input array for contour detection procedure.

2.4 Shape Contour Detection

Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition. In OpenCV, the following method calculates the contours of an input array:

$void\ findContours(InputOutputArrayimage, OutputArrayOfArrayscontours,$
$intmode, intmethod, Pointoffset = Point())$

The parameters are:

- $\in$ $image$: Source, an 8-bit single-channel image
- $\in$ $contours$: Detected contours. Each contour is stored as a vector of points.

∈ *hierarchy*: Optional output vector, containing information about the image topology. It has as many elements as the number of contours

∈ *mode*: Contour retrieval mode

1. CV_RETR_EXTERNAL: retrieves only the extreme outer contours.
2. CV_RETR_LIST: retrieves all of the contours without establishing any hierarchical relationships.
3. CV_RETR_CCOMP: retrieves all of the contours and organizes them into a two-level hierarchy.
4. CV_RETR_TREE: retrieves all of the contours and reconstructs a full hierarchy of nested contours.

∈ *method*: Contour approximation method

1. CV_CHAIN_APPROX_NONE: stores absolutely all the contour points.
2. CV_CHAIN_APPROX_SIMPLE: compresses horizontal, vertical, and diagonal segments and leaves only their end points.
3. CV_CHAIN_APPROX_TC89_L1,CV_CHAIN_APPROX_TC89_KCOS: applies one of the flavors of the Teh-Chin chain approximation algorithm.

∈ $offset$: Optional offset by which every contour point is shifted.

2.5 Case-Based Reasoning

"Case-Based Reasoning" (CBR) is used as a classification method to classify face and frame shapes into different categories.
"A case-based reasoner solves new problems by adapting solutions that were used to solve old problems."
"Case-based reasoning is a recent approach to problem solving and learning [1]." A case-based reasoning can be summarized as:

∈ To store previous cases in memory

∈ To solve new problems:

- Retrieve similar experience about similar situations from the memory
- Reuse the experience in the context of the new situation (complete or partial reuse, or adapt according to differences)

– Store new experience in memory (learning)

A case-based reasoning is a four-step process(Figure 2.1):

1. **Retrieve**: As per the problem definition, retrieve from memory cases that are relevant to solving it. A case consists of a problem, its solution and annotations about how the solution was derived.
2. **Reuse**: After retrieving the best possible solution from the previous case, we map the case to the problem definition. This may involve adapting the solution as needed to fit the new situation.
3. **Revise**: Having mapped the previous solution to the problem, we test the new solution in the real world and, if necessary, revise.
4. **Retain**: After the solution has been successfully adapted to the problem definition, we store the resulting experience as a new case in memory that can be used for future learning.

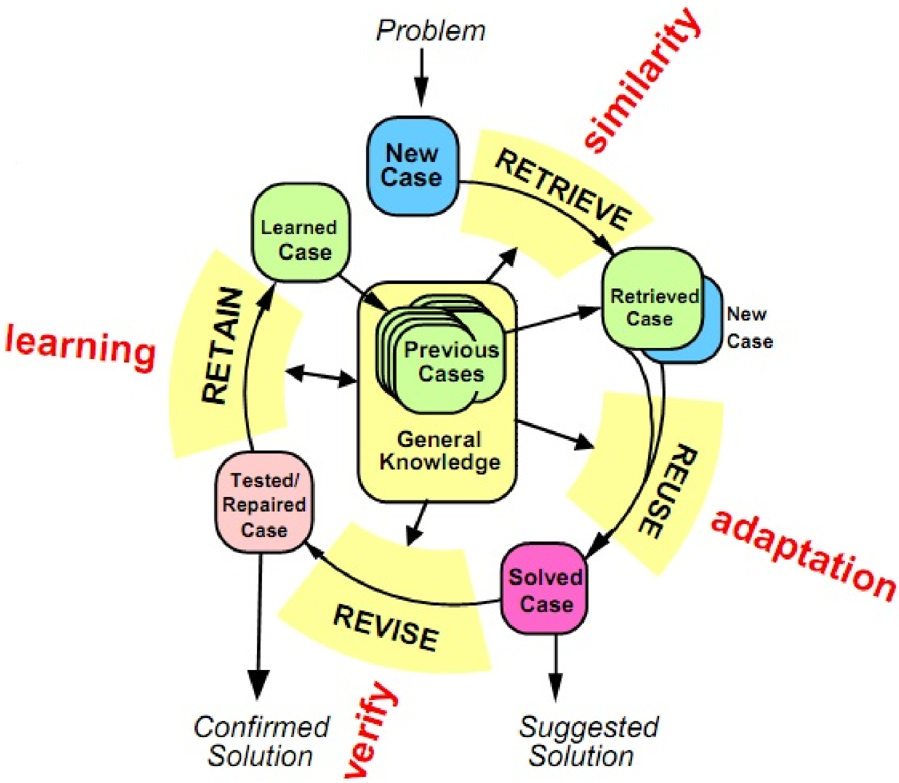

FIGURE 2.1: CBR Cycle (Aamodt & Plaza, 1994, AI Communications)

Besides, the advantages of CBR over other classification techniques are as follows:

1. High flexibility:

- $\in$ Knowledge can be distributed between the four containers according to application needs
- $\in$ Every container can hold the whole knowledge

2. Focus on knowledge in the case base.
3. Knowledge in the case base can be updated and maintained very easily.
4. Reduces the knowledge acquisition effort.
5. Requires less maintenance effort.
6. Improve problem solving performance through reuse.
7. Improve over time and adapt to changes in the environment.
8. High user acceptance.

2.6 State of the Art Face Tracker

Detection of the face edge is a part of this Project. There are many algorithms defined for the edge detection, such as Canny Edge detection, Prewitt Operator, Sobel Operator. However considering the mathematical model and calculations involved, the algorithm used is the Gradient Edge Detection (GED) Predictor Template [7].
Gradient Edge Detection (GED) Predictor Template Algorithm : GED algorithm uses 5 neighbors to determine local gradient and predict the current pixel value. These 5 adjacent pixels are: A and D in the same row, E and B in the same column and C which is a diagonal pixel.
GED algorithm uses fixed single direction (from left to right or from top to bottom) to analyze the image. However, it may not predict the proper pixel values. Therefore, a Multi - directional template of the GED is used. Furthermore it divides the image into 4 parts and each part is processed individually. This is because of the below reasons [7]:

1. Central part of the image covers the most of the information about the picture.
2. Regional characteristics and local gradient directions are mainly considered.

The algorithm Steps are as follows:

1. Calculate the local gradients (Horizontal and Vertical) as:

$$
\begin{aligned}
dh = \; D \quad A \, + \, C \quad B \; (Horizontal Gradient) \\
dv = \; C \quad A \, + \, E \quad B \; (Vertical Gradient)
\end{aligned} \tag{2.1}
$$

2. Prediction pixel value is calculated as:

if *(dv - dh > 80)* **then**
| I' = A
else
| **if** *(dv - dh < -80)* **then**
| | I' = B
| **else**
| | I'=[3(A+B)/2] + (C+D+E)/12
| **end**
end

3. Calculate the forecast error value as:

$$E(I, J) = \; I \quad I' \tag{2.2}$$

4. Determine edges of the image
To calculate the edges, between-cluster variance method is used. It follows the steps below:

∈ $E(I, J)$:gray value of the original image

∈ r: is the separated threshold value between foreground and background

∈ $G1$ / $E(I, J) <= T$: Foreground

∈ $G2$ / $E(I, J) > T$: Background

∈ $P_r(r_q) = (n_q/n)$: Rate of pixels number of gray scale rq for image pixels $(q = 0, 1, 2..L \quad 1)$

When σ^2 value is maximum for some value of T, then T is the best segmentation threshold and edges of the image are identified as below

if $E(i, j) > r$ **then**
| Mark the Pixel as edge point
else
| Mark the Pixel as non-edge point
end

Term/Equation	Definition
$w_0 = \sum_{q=0}^{T-1} p_q(r_q)$	Proportion of target pixels
$w_1 = \sum_{q=T}^{L-1} p_q(r_q) = 1 \quad w_0$	Proportion of background pixels
$\mu = \sum_{q=0}^{L-1} q p_q(r_q)$	Gray scale mean of entire image
$\mu_0 = \sum_{q=0}^{T-1} q p_q(r_q)/w_0$	Gray scale mean value of target pixel (μ_0)
$\mu_1 = \sum_{q=T}^{T-1} q p_q(r_q)/w_1$	Gray scale mean value of target pixel (μ_1)
$\sigma^2 = w_0(\mu_0 \quad \mu)^2 + w_1(\mu_1 \quad \mu)^2$	Need to take maximum of this value among all to determine maximum threshold.

TABLE 2.1: Terms used in Edge Detection Mechanism

5. Apply thinning algorithm
   Edge image pixel matrix (from previous step) contains multi-pixel wide edges. It has to be skeletonized to the single-pixel wide edge so that the contour points of the face will be determined.
   Following Thinning algorithm is applied to the edge image pixel matrix as:

   (a) Create two matrices (H_m for horizontal scan matrix and V_m for vertical scan matrix) with all default values set to 1.

   (b) Scan the edge image pixel matrix horizontally and check for any transition between pixels, such as 0 to 1 or 1 to 0. If found then keep the tag the value of 0 to 0 in new matrix. Rest of the values are left to default 1.

   (c) Repeat the same procedure from previous step for vertical scan (V_m).

   (d) At the end, by simple and logical operation, both H_m and V_m are combined together to give final face edge matrix.

In this research, a state of the art face tracker is used which tracks the face with all above rules and definitions [22]. We used the output of the face tracker as the input of our procedures.

Chapter 3

Glasses Recommendation System

After a detailed introduction in background and literature review, in this chapter, the implemented system will be discussed.
As mentioned, system is composed of different parts which makes the whole mechanism accurately executable. The main target of this chapter is to present various parts, throughly.

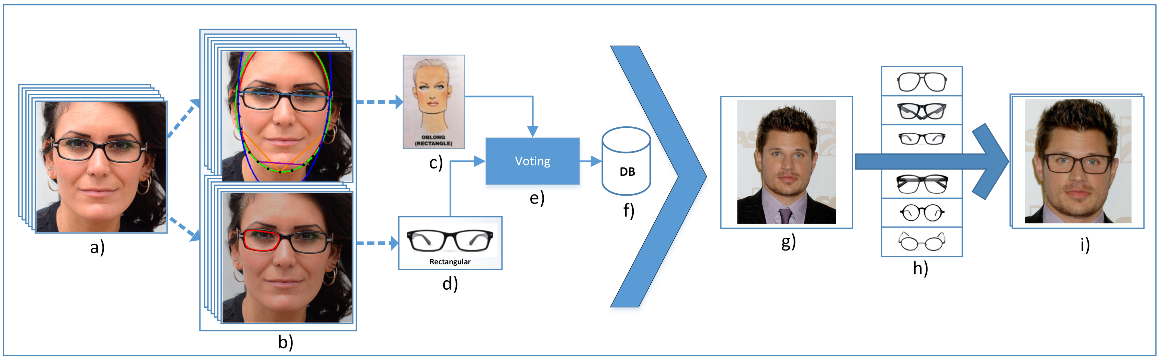

FIGURE 3.1: The overview of Glasses Recommendation System

3.1 Overview

The research is mainly divided into two parts, first is the classification procedures and second would be recommendation system. Figure 3.1 depicts an overview of the system.(a-f) parts show the data collection pipeline; where (g-i) present the recommendation pipeline. For each image, the system do the following:

1. Face Shape Recognition
2. Frame shape Recognition

Each of these algorithms contains preprocessing, feature extraction and classification phases. An image database of subjects wearing eyeglasses are presented to the system as input(a). Then, geometric features extracted from the face(b-Top) and Automatic frame shape extraction happened(b-Bottom). Based on these processes, Classification of face and frame shape are executed(c).
Having the mentioned mechanisms to recognize the shape of face and frame of images automatically, enables the system to produce data-driven surveys in order to rank the most fitted face/frame combinations based on users' conception.
Besides, rating of images from a survey are stored together with the face and eye glass classification. We extract the most popular combinations(e-f).
The right side of the Figure 3.1 depicts the Glasses recommendation system overview. The voting part is the main component of the system that is used to suggest eyeglasses to customers.
When an input image of a person without glasses is entered to the system(g), Face shape detection is performed on the image(f) subsequently. Afterwards, consequent and based on the results from (f) a recommendation is made from different types of the glasses available in the system(h). Finally, A frame of the recommended type overlapped the face(i).
In the next sections, each of above parts will be presented in detail.

3.2 Face Shape Extraction and Classification

One of the main parts of the thesis is Face Shape Recognition System (FSRS). It has received good attention in various areas such as fashion and psychology. Psychologists and stylists provide descriptions about each pattern of face shapes in order to provide people with suggestions about their face characteristics and style. Moreover, some on-line resources also recommend the different styles suitable for the particular face shape. The implications of such systems can also be found in haircut recommender systems [12, 13] as well as eye-glasses (frames) suggestion systems [3–6].
However, most of the on-line websites ask users to manually select their face shape according to their facial features. It becomes more difficult for the user to operate such

on-line resources, if user is not aware of his/her own face shape.
To automate this task, this project proposes a new approach to determine face shapes. FSRS overcomes this difficulty by accepting the users' image on-line and providing them their face shape.
In this part, various efficient and reliable algorithms are used to compute face shapes by comparing selected facial features and their respective inclination of face edges with variants of face shapes in a facial database.
FSRS works efficiently if the images with simple background are provided.
Before going into details of techniques and algorithms involved in recognizing various face shapes, a digression into the details of pattern recognition. The discipline, pattern recognition, includes all cases of recognition tasks such as speech recognition, object recognition, data analysis, and face recognition, etc. This section introduces the basic structure, general ideas and general concepts behind them.

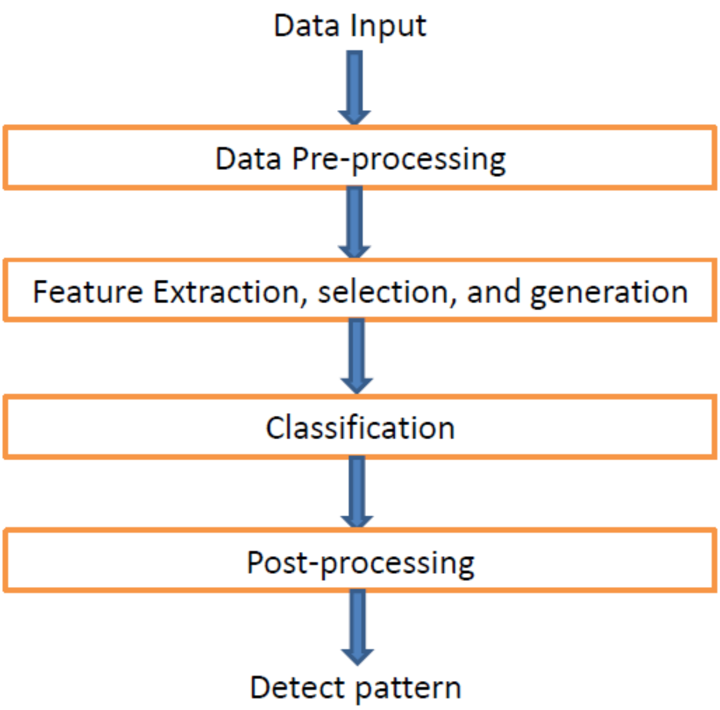

FIGURE 3.2: Phases in Pattern Recognition

The general structure of pattern recognition is shown in Figure 3.2. In order to generate a system for recognition, data sets are required for building categories and to compare similarities between the test data and each category. From Figure 3.2, Input data is passing through a pre-processing stage of the stored raw data. The sequences of data pre-processing operations are applied to images in order to put them into a suitable format ready for feature extraction. After this each raw data in the data sets are transformed into a set of features, and the classifier is mainly trained on these feature representations.

When a query comes in, similar data pre-processing is carried out followed by the same sequence of operations and then features are added into the trained classifier. The output of the classifier will be the optimal class (sometimes with the classification accuracy) label or a rejection note (return to manual classification).

Many techniques for the detection of facial feature points exists including approaches based on luminance, chrominance, facial geometry and symmetry. Quality control, mobile robot navigation, and detecting contours in images are some of the common applications of this technique.

Studies have shown that hair, face outline, eyes and mouth all have been determined to be important for recognizing and remembering human faces. Humans have a natural ability to recognize people by looking at their faces, but our minds recognize known people more easily than strangers. Human eyes and brain capability are limited to plethora of images; this is the reason why computerized methods have evolved particularly in the areas of security, surveillance and forensics.

The detection and localization of face and of its features are instrumental in the successful performance of subsequent tasks in related computer vision applications. Many high-level vision applications such as facial feature tracking, facial modeling and animation, facial expression analysis, and face recognition, require reliable feature extraction. The proposed method utilizes an analysis of different face shapes based on the boundary points across the edges of the face. The algorithm yields good results using a subset of feature points than many other typical applications.

The Database used for these experiments is an adequate number of images retrieved from on-line resources. A subset of these images was chosen to reduce problems associated with lighting, facial expressions and facial details. The image files used were in .PNG format.

For our experiment, the training data set had over 300 different face shapes that were recognized by domain experts i.e. Fashion Designers, Hair Stylists. In testing data we took 100 new face shapes and 50 randomly selected from training data.

Face shape recognition is a method to identify a person's face shape through a photograph of their face. The first module (see Figure 3.2) in face recognition processing is known as face detection. Implementing a face detection algorithm is beyond the scope of this project we have implemented a manual detection system. This step consists of providing rough estimates of the location and scale of specified boundary points on the face.

The second module in the face recognition processing is face point creation. In this module, an accurate localization is required. The database images selected for the project does not require us to implement this module. The reason would be using the state of the art face tracker [22] which Face shape recognition used it to have the face boundary points as system input.

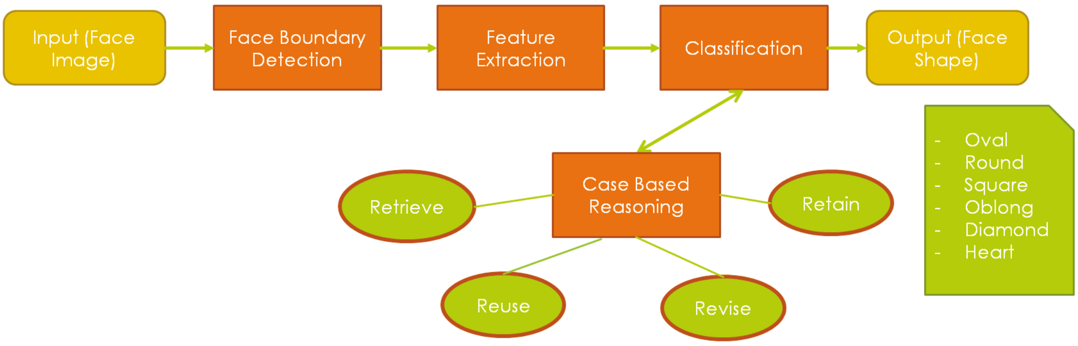

FIGURE 3.3: FSRS Process flow

The third module for processing is feature extraction, where we seek to extract effective information that will potentially distinguish faces of different persons. It is important in this module to obtain stable geometrical representations of the extracted features.
In the final module, feature matching, we compare the extracted feature vectors of a particular image to the face images we have stored in the database.
Figure 3.3 is the process flow of FSRS system. As explained before, the input is the clear image face and the output would be one of the six different available classes of faces which the face owner categorized in it by the FSRS. Next, the other parts of this process will be presented.

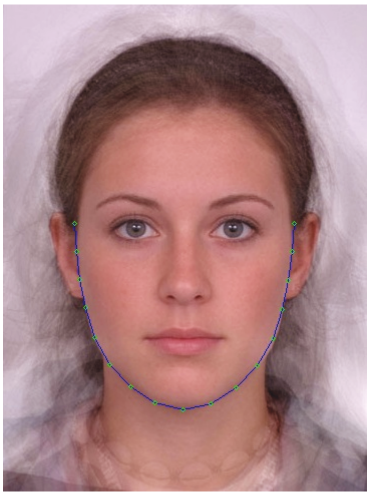

FIGURE 3.4: Sample output with detected face boundary

3.2.1 Feature Extraction

Based on algorithms and procedures in section 2.6 about Face Boundary detection, 17 points are extracted based on approximation and the use of natural symmetry of the face, which help to identify the features of the face. These points are shown in Figure 3.4 with green points, while the symmetric boundary is depicted with blue curved line. One of the 17 boundary points is located on the symmetry line of the face and there are 8 points on each side from lower jaw line to the eye-line. The results show that the quantity of these points is sufficient to have an accurate face shape recognition. Figure 3.5 illustrates the points, their way of numbering and position, exclusively.

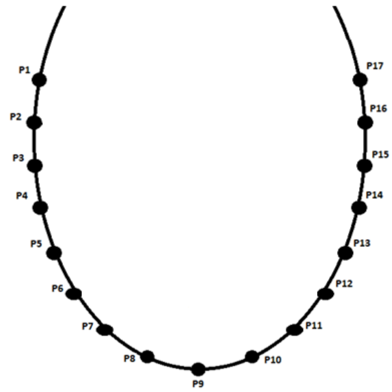

FIGURE 3.5: Face contour points

3.2.2 Face Shape Classification

In this section, classification procedure (third step in FSRS process flow (Figure 3.3)) will be presented.
Figure 3.6 is a presentation on how the geometric features vary with the all six face types.

Features: Considering the detected 17 points from the face, different comparable features are extracted. For comparison, 8 different face features are considered. Three of them are captured by drawing an ellipse from different points of the face and rest are determined by calculating distances.
There are 6 main elements of a face which are utilized to distinguish different face shapes. the elements are:

1. By drawing ellipses with face points, initially, three ellipses are drawn considering the different facial points:
   (a) Ellipse that accommodates most of the points
   This property helps to get the height of the face(Figure 3.7).
   (b) Ellipse that covers only chin points
   This property helps to identify if the chin is pointed (for heart shape) or round (for oblong shape) or other (for square shape)(Figure 3.8).

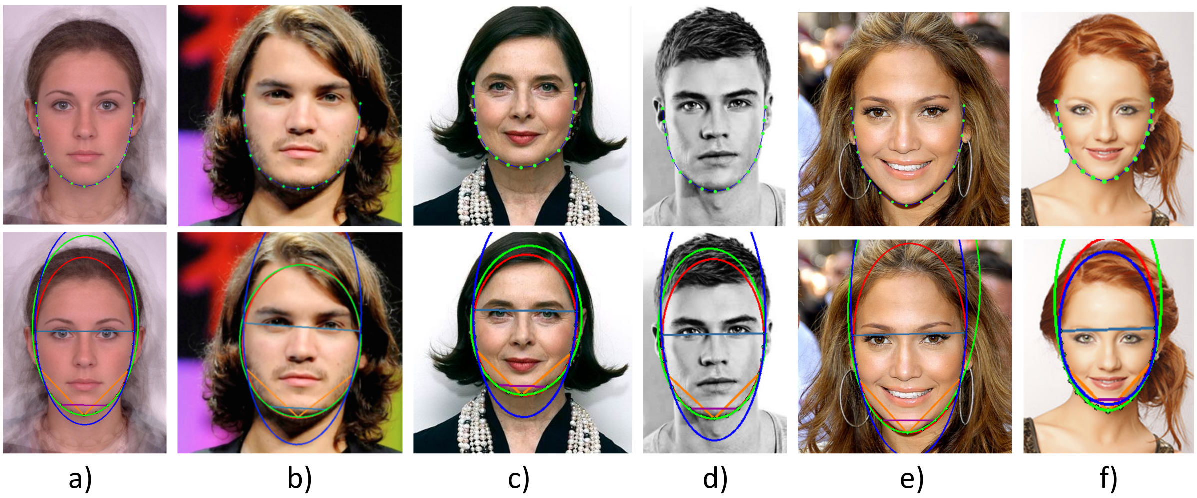

FIGURE 3.6: Illustration on how the geometric features vary with the face type: a) oval, b) round, c) square d) oblong, e) diamond, f) heart. Top: Pictures and the face boundary outlined. Bottom: Picture and the geometric feature extracted. Red: ellipse that best fits al boundary points. Green: ellipse that best fits the chin points. Blue: ellipse that best fits the cheek boundary points. Orange: diagonal line. Purple: jaw line. Lighter blue: eye line.

(c) Ellipse that considers the remaining points

This property helps to identify if the width of the face is same at eye line, cheekbone and jaw line. It helps to identify if the shape is oblong or square(Figure 3.9).

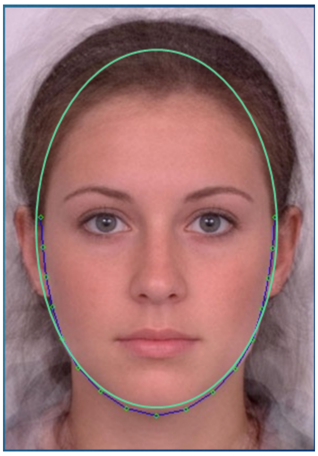

FIGURE 3.7: Ellipse from all points

2. Taking the distances of facial points from drawn ellipses in previous step.
   Distances are calculated of the points from the respective ellipse to recognize which type of ellipse covers a maximum number of points. (See Appendix B)
   Moreover, threshold value is set for the distances during training phase.
   Ex. Consider the ellipse that covers chin points. If the chin is pointed, then the distances of the chin points from ellipse will be more than threshold value. Such type of faces can be categorized in heart or diamond face shape.

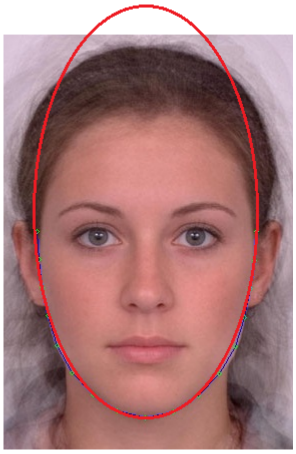

FIGURE 3.8: Ellipse from chin points

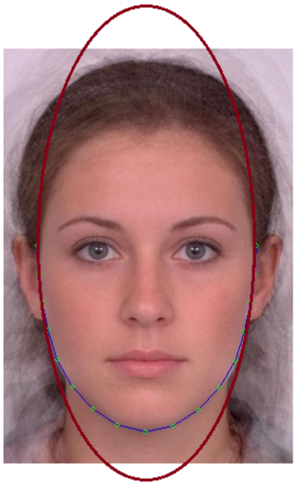

FIGURE 3.9: Ellipse from remaining points

However, if the distances are less than threshold value, then face can be categorized in round or oval face shape.

3. Eye Line length
   Eye Line Length is calculated by joining two points, P1 and P17.
   This property length is compared with cheek bone line and jaw line. It will help to determine if the these lengths are equal for oval or round faces, or differ a lot for heart and diamond face shape(Figure 3.10).

4. Jaw Line Length
   Calculation of Jaw line may not be accurate because few face shapes does not have explicitly defined jaw line such as round or Oval. Therefore, Jaw Line is defined by averaging 3 different lines that join face points P6 P12, P7 P11, P8 P10(Figure 3.11).

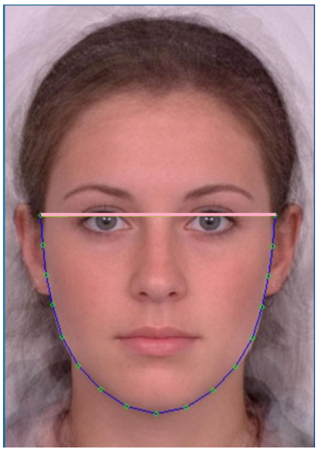

FIGURE 3.10: Eye-Line

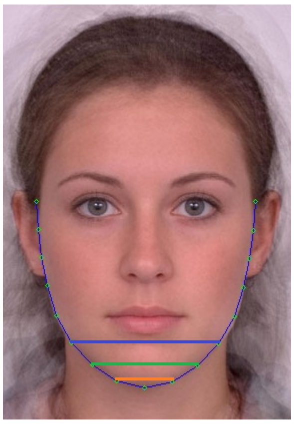

FIGURE 3.11: Jaw-Lines

5. A diagonal line length that connects jaw points with face symmetry point P9
   Four diagonal lines are drawn from points, P4, P5, P13, P14 to point P9.
   The lengths of these lines help to decide if the face is elongated. Moreover, these lines are used to calculate the distances of the remaining face points, explained in the next step.

6. Distances of points from diagonal lines, drawn in previous steps
   Though the distances from ellipses are compared to identify the face shapes, results will not be accurate as it may miss the angles present at the jaw line. To overcome this drawback, diagonal lines are drawn as shown in the picture (from previous step) and distances of the face points (near jaw) are calculated. These distances will accurately determine if the face shape is wider near face jaw(Figure 3.12).

Normalization: As the user may input image of any size, extracted image features may not be compared and results will not be accurate. Though the image is normalized to a specific fixed sized matrix, it cannot avoid the errors with the images having zooming

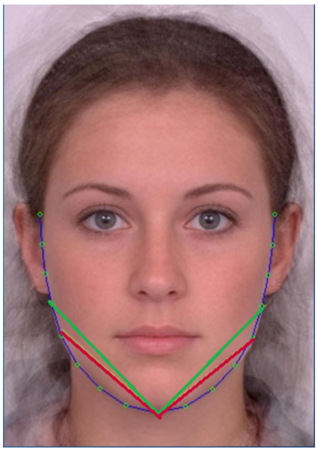

FIGURE 3.12: Diagonal Lines

	Feature	Explanation
1.	Height(all) / Width(all)	Height and Width of ellipse drawn from all facial points.
2.	Height(chin) / Width(all)	Height of ellipse drawn from chin points.
3.	Height(other) / Width(all)	Height of ellipse drawn from face side points.
4.	Distance(all) / Width(all)	Distance of the each facial point from respective ellipses drawn.
5.	Distance(chin) / Width(all)	
6.	Distance(other) / Width(all)	
7.	Eye-line / Jaw-line	Lines extracted from feature extraction steps 3 and 4.
8.	(Distance from bottom points to diagonal lines) / Width(all)	Lines extracted from feature extraction steps 6.

TABLE 3.1: FSRS Normalized Features

effects (either zoomed in or zoomed out).

To avoid such type of discrepancies, the natural symmetry of the face is utilized again to determine special features. Such features are described in table 3.1

Classification: In FSRS, six different types of face shapes are considered for identification. They are Heart, Oblong, Oval, Diamond, Square and Round. Facial properties for each shape are summarized in table 3.2 and 3.3. The variables are presented in table 3.4.

Type	Facial Features	Features in terms of equation
	Mostly balanced proportions. chin is slightly narrower than forehead and cheekbones are high.	1. Lh > Lcb 2. Lj ~= Lf 3. Lcb > Lj and Lcb > Lf
	Full cheeks, rounded chin with few angles. Width and length are in same proportions.	1. Lh = Lcb 2. Lf ~= Lcb ~= Lj
	Angular face with a strong jaw line, broad forehead and square chin. Proportional length and width.	1. Lh ~= Lcb 2. Lf ~= Lcb ~= Lj

TABLE 3.2: Oval, Round and Square Faces

Once all features are extracted, the next step is to perform classification based on the retrieved features. The main purpose of the classification is to assign each point in the space with a class label [4]. On the basis of a training set of data containing several observations, we identify to which of a set of categories a new observation belongs.
The term "classifier" is an algorithm that implements a classification. A variety of classification methods can be applied depends on the problem definition. Based on the feature extracted for Face Shape recognition, we have used "Case-Based Reasoning" (CBR)(section 2.5) as a classification method to classify face shape into different categories.

Figure 3.13 is an example scenario of case-based reasoning for FSRS. Therefore, we

	Narrow shape that's longer than it is wide. Angular features with high cheekbones, a longer nose and tall forehead.	1. Lh > Lcb 2. Lf ~= Lcb ~= Lj
	Narrow at the eye line and the jaw line with a small forehead and chin. Angular features with dramatic cheekbones.	1. Lh = Lcb 2. Lcb >> Lf and Lcb >> Lj
	Broad forehead and wide cheekbones that narrow to a small chin.	1. Lf >> Lj 2. Lcb > Lj

Table 3.3: Oblong, Diamond and Heart Faces

Symbols used for feature definition
Lh: Height Length
Lcb: Cheek Bone Length
Lj: Jaw Length
Lf: Forehead Length

Table 3.4: Variables uses in Faces' Comparison

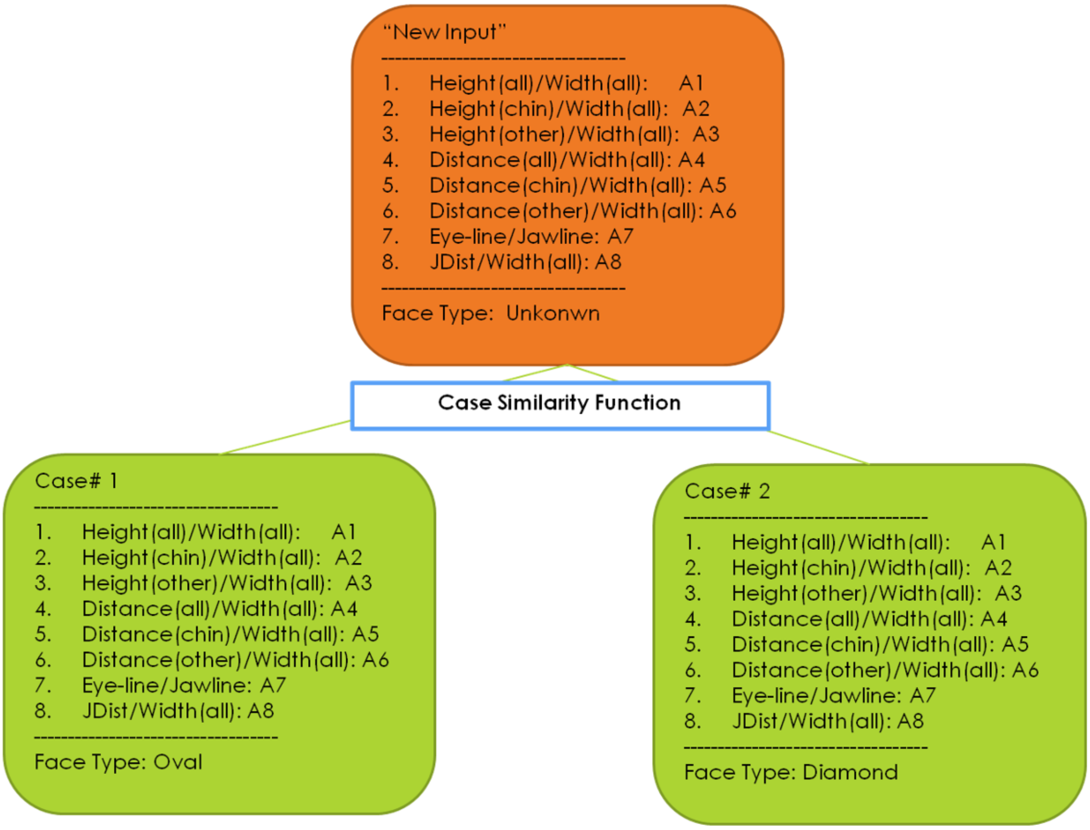

FIGURE 3.13: An example scenario of Case-based reasoning for Face Shape Recognition System

defined the distance as follows:

$$d(T, F^k) = \sum_{i=1}^{8} w_i (T_i \quad F_i^k) \tag{3.1}$$

where $T$is the feature vector extracted from the testing image, F^k is the feature vector of the k's element of the training images set, the subscript i denotes the components of the 8 dimensional feature vector and w_i is a weight associated to each of the feature to compensate for the relative magnitude and importance of the feature. The weights were determined experimentally and were set only once.

The CBR is defined in 2 separate phases:

1. Learning: Organizing the system in order to have maximum accuracy during executing the recognition. The Data used in this part is classified before by experts. This phase has four stages in CBR. The aim is to change the weights to have the best possible results.

2. Testing: Executing the system with real, unclassified data. This phase has 2 stages from CBR which are retrieve and reuse. The revise and retain parts are not running in order not to changing the weights.

With respect to CBR, the learning procedure of the method contains 4 steps:

1. Retrieve: Calculate the multidimensional distance (equation 3.1).
2. Reuse: Assign the nearest case label to the new case.
3. Revise: It is the weights' verification process. For each new case belongs to class m and m / $[1, n]$, the distance must be minimum, considering the weights. Thus, the new weights in learning process defined as:

$$w_i^* = \}w_i \sum w_i(T_i \quad F_i^m)^2 < \sum w_i(T_i \quad F_i^s)^2, \{s : s \ / \ [1, n], s \not\forall m| \tag{3.2}$$

   Where w_i^* stands for the new weights and w_i presents the current weights of the system.
4. Retain: Adding the case to the learning cases and substitute the new weights in the system.

Since CBR is designed for large amount of data and there are limited number of images in this step, we analyzed and experimented the weights changing procedure to assign the best weight to each parameter.

The final weights are: $(1.2, 1.0, 1.0, 1.2, 1.1, 1.2, 1.1, 1.1)^T$.
However, in the training only the dominant face shape is considered.
For training we used a set of 100 images from the Internet. We evaluated the accuracy of our classification on a different set of 300 pictures.
The results will be presented in Result section.

3.3 Frame Shape Extraction and Classification

Similarly to the previous part of thesis, our frame shape classification method consists of four steps. First, the polygonal shape of the frame is extracted from the image. Second, key geometric features tailored to differentiate between the various frame shapes are computed from the polygonal shape. Next these geometric features are converted

into a feature vector that can be used in the CBR framework. The CBR framework is trained on a set of known frame shapes and, finally, a query image is classified based on a nearest search on the elements of the training set.
As mentioned, The classification part of both face and frame was implemented identically(by using CBR). Whilst, Face shape and Frame shape Extraction have differences related to their structure, consequently.
In the next sections, the Various parts of the Frame Shape Extraction and Classification will be described.

3.3.1 Glasses Frame Extraction

To be able to classify glasses, the main parts of the frame should be extracted, undoubtedly. Since the Image of person wearing the glasses is provided to the system as input, there could be considerable quantity of noise in glasses frame extraction i.e. face properties, background, lights etc. Hence, a reliable algorithm should provided to ignore the noises. We used Edge detection and Symmetry concept to reach this goal.
The steps of Interior Frame Extraction are as follows:

1. Image Preprocessing
   (a) Convert the image to gray-scale format: using CV_BGR2GRAY of OpenCV.
   (b) Smooth the image: By using blur function of OpenCV, Each output pixel is the mean of its kernel neighbors(all of them contribute with equal weights). The kernel calculated as:

   $$\mathtt{K} = \frac{1}{\mathtt{ksize.width*ksize.height}} \begin{bmatrix} 1 & 1 & 1 & \bullet\bullet\bullet & 1 & 1 \\ 1 & 1 & 1 & \bullet\bullet\bullet & 1 & 1 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & 1 & 1 & \bullet\bullet\bullet & 1 & 1 \end{bmatrix}$$

   where $ksize$ is blurring kernel size, and width and height are calculated from image dimensions.
2. Augmented Polygon Extraction
   (a) Use canny algorithm to find the edges of the gray-scale image with a threshold.
   (b) Find contours on canny results.
   (c) Find symmetry line of the image.
   (d) Contour Reduction.
3. Symmetric contours selection(with a defined threshold) based on reflection of each one, using symmetry line.

4. Calculate convex hulls of symmetric contours.
5. Frame Polygon Selection Procedure
    (a) Check intersections.
    (b) Fit ellipse to each convex hull.
    (c) Calculate fitted oval area and convex hull area.
    (d) Rank the convex hulls based on fitted oval area, convex hull area and number of intersections.
    (e) Select the highest rank polygon
6. If there exists a convex hull which satisfies the condition, program is converged. Otherwise, go to step 2 with lower threshold.

Image preprocessing step prepares image for the next stages. By converting to gray-scale, noise will be decreased significantly. Smoothing the image leads to longer and more continuous contours.
In augmented polygon extraction step, the aim is to retrieve the smallest set of contours which contain the glasses polygon.
The glasses are located around the eyes. During contour detecting, as a consequence, nested contours would be selected as the output(eye, eyelashes and glasses frames). to prevent from this, CV_RETR_EXTERNAL is used in order to select only the outer contours. The contour method is CV_CHAIN_APPROX NONE since we can have all the contours for the next steps to have the actual frame extracted.
After finding the contours of the canny detection restult, the symmetry line of the image is calculated automatically based on the face size and angle in stage 2-c by using face tracker(section 2.6).
In step 2-d, contours around the glasses area would be kept in order to reduce the time and space complexity. the box area is depicted in Figure 3.15 c-f with light green lines. There is no doubt that symmetricity of the face and glasses are most distinguished properties existed. Accordingly, the symmetric contours would be retrieved in stage 3 of glasses extraction algorithm.
In stage 4 the convex hulls are calculated for all remaining contours to have closed polygons, by calling OpenCV convex hull method. The functions find the convex hull of a 2D point set using the Sklanskys algorithm [39] that has $O(NlogN)$ complexity in the current implementation.
Next, to filter and select the around eyes convex hulls, in step 4, the intersections of closed polygons and lines around the eyes will be calculated and the polygons with larger numbers of intersections would be stored for next processes.
Frame polygon selection procedure is a mechanism to determine the exact frame polygon

between remained ones.
Step 5-a calculates the number of intersections between the polygon and 6 lines which draws from eyebrows, noise and lips to the center of eyes; assuming that frame polygon is surrounded on that area.
In step 5-b, the fitted oval is calculated for all remaining convex hull areas. Afterwards, the extraction of exact frame polygon is happening based on oval area, convex hull area and number of intersections which are the thresholds based on possible size and geometry of frames. If the conditions are satisfied, the algorithm converged. Otherwise, the threshold of canny will be changed and all steps 2-6 are executed with lower threshold.

After image preprocessing, canny edge detection algorithm is executed on gray-scale image. The result is the input of contour detection mechanism and the output would be set of polygons that represent the edges of image. A number is defined as threshold which effects on canny stage, ranging from 140 to 10. In the case of lower threshold, the canny would be more sensitive and contains more polygons. Note that points which represent each contour available in higher thresholds will exist in lower ones. Thus, the output points of each threshold will be a subset of the points in lower ones. In other words, the points which selected in threshold equals to 140 is available in all lower thresholds including 10.
Since complexity is became higher by decreasing the threshold, algorithm is initialized with relatively high threshold and became lower when necessary (when the conditions are not satisfied in step 6). Figure 3.14 represents an image with different canny thresholds between 140 to 10.

The steps of frame extraction are visually presented in Figure 3.15. To extract the polygonal shape of the frame reliably we rely on three important observations: the frame shape is a closed polygon, it is nearly always convex and symmetric for the right and left eye [29]. First the contours are extracted from the image at different threshold levels using canny edge detector [40] (Figure 3.15b). For clarity only one level is shown in the figure. Next, the face boundary line is computed using a state of the art face tracker followed by the eye region (Figure 3.15c - green rectangles) and symmetry line (Figure 3.15c - dark blue line) are computed. Only he contours that intersect the eye region are kept (Figure 3.15c - purple lines). The contours are further refined by selecting only those that have a symmetric pair (Figure 3.15d - orange lines). Symmetry is achieved if more than 50% of the points have a corresponding symmetric pair on the paired curve (within a five by five pixel radius). Convex hulls are computed on the remaining polygons (Figure 3.15e - yellow polygons) and are candidates for the final polygon that approximates the frame shape. In order to avoid degenerate cases such as the triangular polygons in Figure 3.15e) a constraint is placed on both the size of the polygon (more than 20%

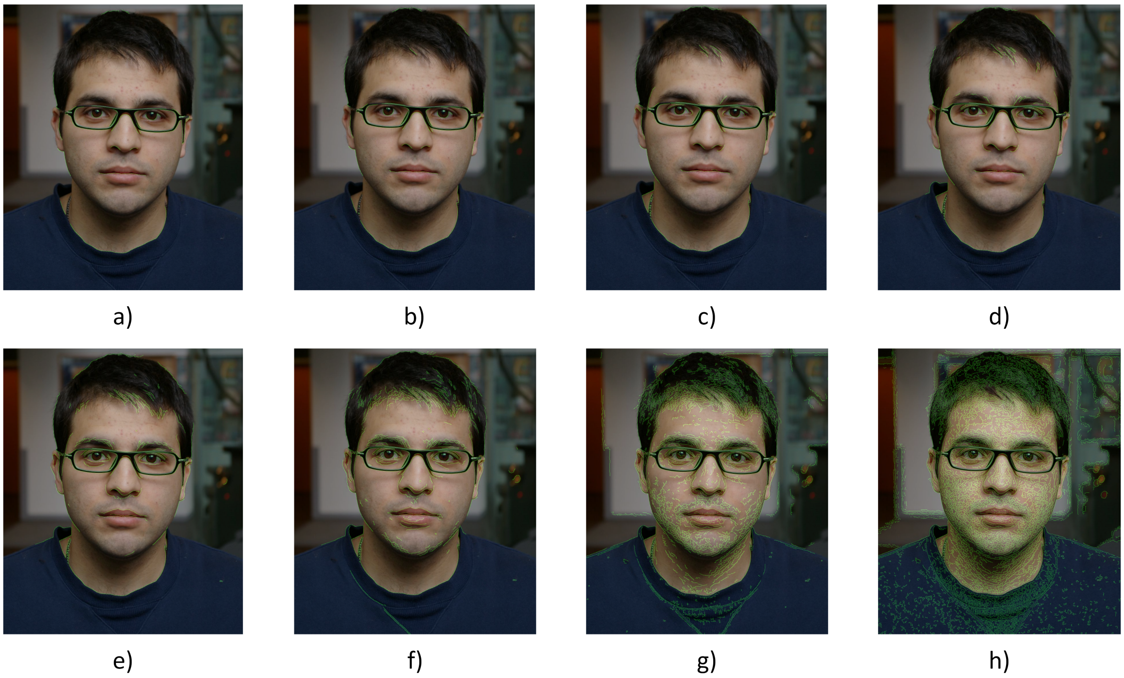

Figure 3.14: Image contour extraction with different thresholds: a)140, b)120, c)100, d)80, e)60, f)40, g)20, and h)10. By decreasing the threshold, the probability of having the frame edges and also quantity of noisy contours are increased .

of the eye region) as well as its roundness (the area of the polygon should be at least 80% of the area of the best fitting ellipse). If more than one candidate exist, the largest one is selected. If no candidates exist, the edge detection threshold is decreased and the process will be repeated.

The extraction mechanism works even on difficult input such as sunglasses with reflection (Figure 3.16a), partially occluded faces and thin metallic frames (Figure 3.16b) as well as subjects with significant pose rotation (Figure 3.16c).

3.3.2 Frame Classification

By considering the glasses stores, it is experimented that there are six most common and standard classes of glasses available. Figure 3.17 depicts these shapes.
The key distinctive features between the frame shape classes are the "roundness" of the shape, the orientation of the main axis, the size and the degree of symmetricity with respect to the principle axis. Therefore the only geometric feature that is extracted is the best fitting ellipse (Figure 3.18 - blue polygon)of the frame polygon (Figure 3.18 - red polygon, the principal axis that shows the degree of rotation in green).

Features: The features which are used in frame extraction are as follows:

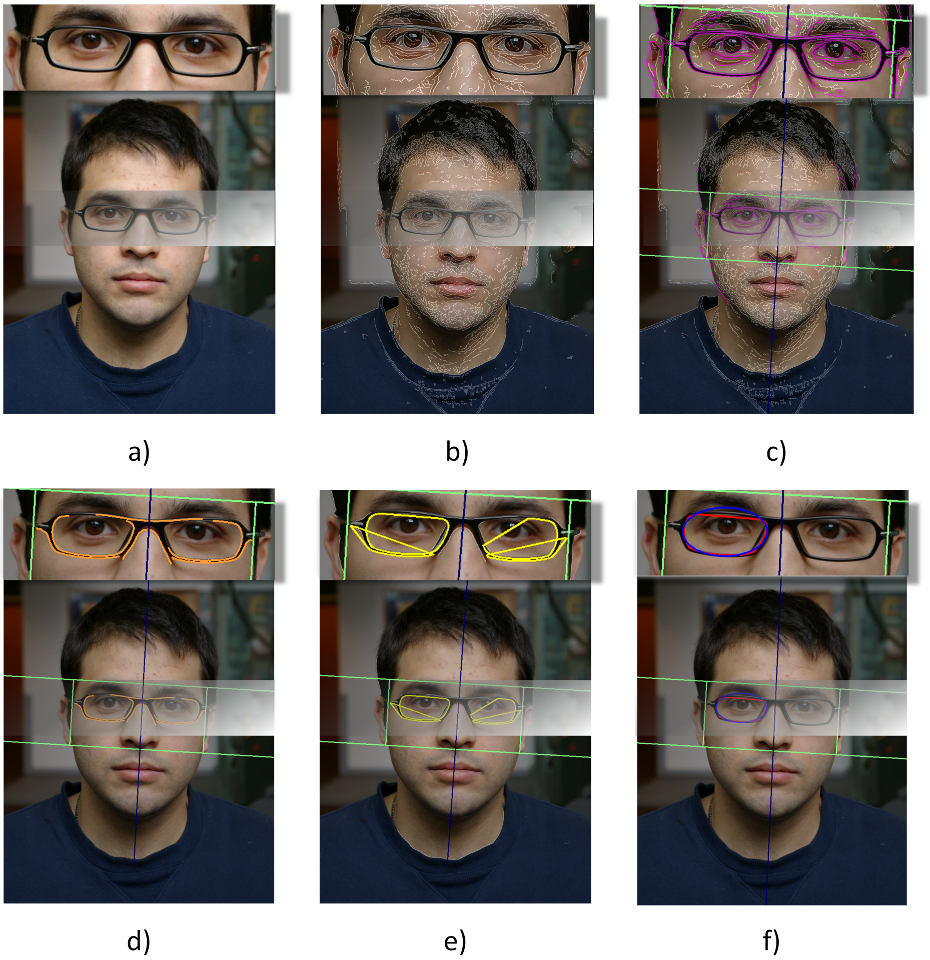

Figure 3.15: Frame shape extraction: a) Original image, b) Edge detected using Canny edge detector, c) Large edges that intersect the eye region, d) Contours that have a symmetric pair, e) Convex hull of these contours, and f) Select the optimal contour (red). A best fitting ellipse to this contour is rendered in blue.

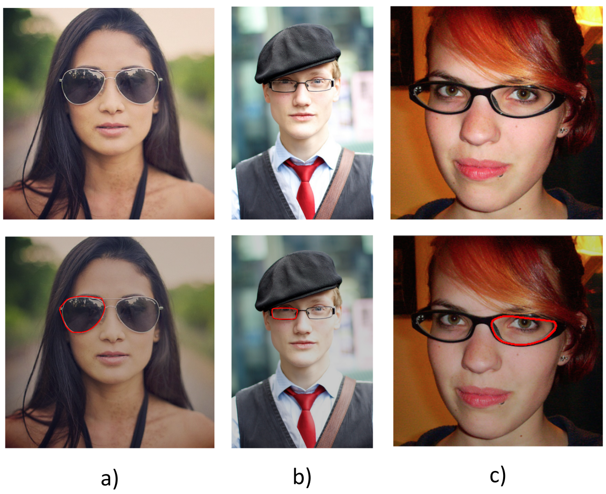

Figure 3.16: Difficult examples of the frame shape extractor: a) Sunglasses exhibiting complex reflection patterns, b) Thin small metallic frames, and c) Faces with significant rotations from the frontal position.

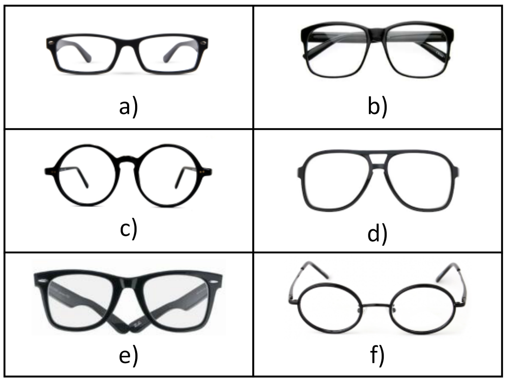

Figure 3.17: Example of the six standard frame shapes: a) rectangular, b) square, c) round, d) aviator, e) wayfarer, and f) oval.

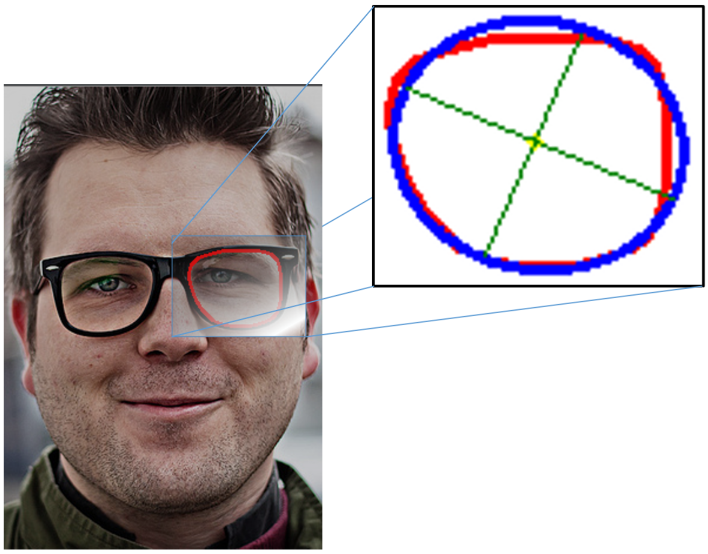

Figure 3.18: Geometric features that we used for the frame shape classification. Left: sample image with the frame shape polygon extracted (red). Right: Frame shape polygon (red), best fitting ellipse blue), ellipse axis of rotation (green).

1. Frame fitted oval occupancy.
2. Frame fitted oval height/width.
3. Frame fitted oval angle.
4. Distance from frame convex hull to frame fitted oval.
5. Symmetricity of the frame fitted oval.

Figure 3.19 illustrates the features in a wayfarer frame. As shown, the main component is the fitted oval of the convex hull polygons extracted. The occupancy of the ellipse is normalized by the face parameter to have the relevant size of the glasses.
Frames such as oval and rectangular do not have significant angle to the face, while wayfarer and aviator frames have remarkable angle. Consequently, fitted ellipse angle

could be used to approximate the degree between image and frame axis. In Figure 3.19, the angle would be between green and purple lines depicted.
To recognize round and angular frames, it is necessary to have the distance from the fitted ellipse to the extracted polygon. About the ellipse angle feature, because of probable rotation of the face, the correlative angle would be calculated. In Figure 3.19, this remarkable space is shown between the red and blue (frame and ellipse) polygons.
The symmetricity is important in order to differentiate symmetric frames like oblong and non-symmetric ones such az aviator. The symmetricity number of each polygon is calculated by realizing the distance from the center of mass and center of the window of the polygon. The results show that the mentioned features are sufficient to have an accurate frame recognition process.
Same as FSRS, the normalization of features is calculated by dividing the width of all boundary points of face on each feature.

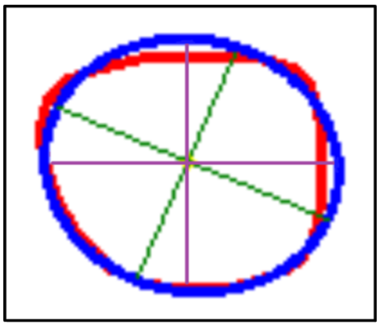

FIGURE 3.19: Demonstration of frame features. The Extracted frame polygon is illustrated in red, fitted oval in blue, ellipse width and height in green and x and y axes of the image in purple.

Classification: In Frame Classification, six different types of frame shapes are considered for identification. They are Aviator, Oblong, Oval, Wayfarer, Square and Round. Facial properties for each shape are summarized in table 3.2 and 3.3. The variables are presented in table 3.4.

As mentioned before, the classification aspect of both face and frame part of the thesis are following the CBR model with different number of elements and learning samples. Figure 3.20 depicted the actual frame and it's correspondence fitted oval for each 6 types of glasses.

Using the best fitting ellipse a 5 dimensional feature vector F is computed consisting of the following: the ratio between the height and the width of the best fitting ellipse, tilt angle of the best fitting ellipse, area of the shape polygon as a fraction of the eye region, average distance from the shape polygon to the ellipse and the distance between the center of mass of the polygon and the ellipse center. In this case the weights are

Type	Facial Features	Features in terms of equation
 Oval	Mostly balanced(Symmetric) proportions, with a bigger width than height. Ellipse angle about zero.	1. EHW<1 2. EA ~=0 3. ESym=1 4. ED ~= 0
 Round	Completely Symmetric proportions, with equal width and height.	1. EHW=1 2. ESym=1 3. ED ~=0
 Square	Completely Symmetric proportions, with equal width and height. The distance to ellipse is distinguishable.	1. EHW=1 2. ESym=1 3. ED>>

TABLE 3.5: Oval, Round and Square Frames

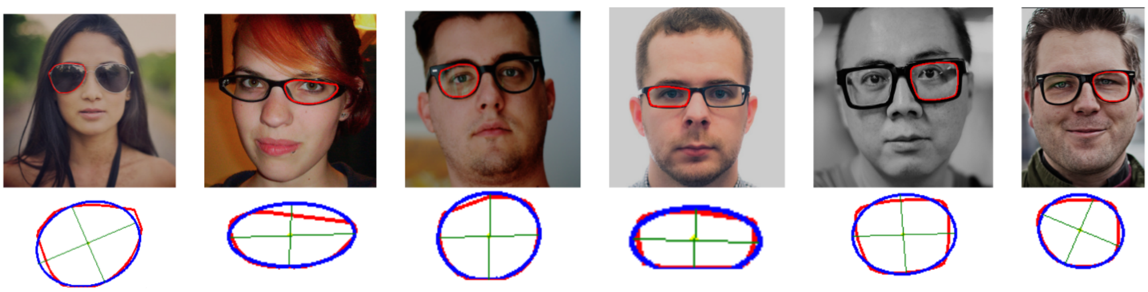

FIGURE 3.20: Example of the six standard frame shapes, their extracted convex hull and Fitted oval. From left: Aviator, Oval, Round, Rectangular, Square and Wayfarer

 Rectangular	Mostly balanced(Symmetric) proportions, with a bigger width than height. Ellipse angle about zero. The distance to ellipse is distinguishable.	1. EHW=1 2. ESym=1 3. ED>> 4. EA ~= 0
 Aviator	The size of the glasses and the ellipse angle are recognizable. The symmetricity is about zero.	1. ES>> 2. EA>> 3. ESym ~= 0
 Wayfarer	The size of the glasses and the ellipse angle are recognizable. The symmetricity is about zero. The distance to ellipse is more than Aviator frames.	1. ES>> 2. EA>> 3. ESym ~= 0 4. ED>>

Table 3.6: Rectangular, Aviator and Wayfarer Frames

Symbols used for feature definition
ES: Ellipse Size
EHW: Ellipse Height/Width
EA: Ellipse Angle
ED: Distance to Ellipse
ESym: Ellipse Symmetricity

Table 3.7: Variables uses in frames' Comparison

equal and equation 1 is used as the distance function.
Same as FSRS, the distance as follows:

$$d(T, F^k) = \sum_{i=1}^{5} w_i \dot{(} T_i \quad F_i^k) \tag{3.3}$$

where $T$is the feature vector extracted from the testing image, F^k is the feature vector of the k's element of the training images set, the subscript i denotes the components of the 5 dimensional feature vector and w_i is a weight associated to each of the feature to compensate for the relative magnitude and importance of the feature.

The CBR is defined in 2 separate phases which completely bear resemblance to face shape recognition.

1. Learning: To learn the weights of the system in order to have maximum recognition accuracy.
2. Testing: To classify the inputs by recognition system. Only retrieve and reuse stages of CBR is executable in this phase.

With respect to CBR, the learning procedure of the method contains 4 steps:

1. Retrieve: Calculate the multidimensional distance (equation 3.1).
2. Reuse: Assign the nearest case label to the new case.
3. Revise: It is the weights' verification process. For each new case belongs to class m and m / $[1, n]$, the distance must be minimum, considering the weights. Thus, the new weights in learning process defined as:

$$w_i^* = \} w_i \sum w_i \dot{(} T_i \quad F_i^m)^2 < \sum w_i \dot{(} T_i \quad F_i^s)^2, \{ s : s \ / \ [1, n], s \not\forall m | \tag{3.4}$$

   Where w_i^* stands for the new weights and w_i presents the current weights of the system.

4. Retain: Adding the case to the learning cases and substitute the new weights in the system.

Since CBR is designed for large amount of data and there are limited number of images in this step, we analyzed and experimented the weights changing procedure to assign

the best weight to each parameter.
The weights for frame shape classification are as follows:
$(1.0, 1.2, 1.1, 1.2, 1.0)^T$
One important note is that face and frame types do not always fall completely into one category, they can be a blend between two as shown in Figure 3.21. Therefore, we consider the closest two types and compute a blending score by simply dividing by their sum.
In frame recognition, we used 50 photographs for training and 250 for testing. If the scores are within 60%-40% range we label the face/frame type as a blend and during the recommendation type we allow recommendation from both face/frame types.

FIGURE 3.21: Example of mixed shape classes. Left: half oval and half rectangular face shape. Right: half oval and half rectangular frames.

3.4 The Survey

In previous sections of this chapter, the face and frame of the picture are automatically classified. In the next step, a learning method is required to gather information from the users. In other words, to complete the circle(Figure 3.1), preferred images should be chosen based on user's point of view.
In this stage, Php and mySQl are used to develop a survey mechanism to input the users idea based on images (which their face and frame classified, previously) and output the results. Figure 3.22 is a view of this website available in (http://users.encs.concordia.ca/am_zafa/survey/).

For each image, the user can choose a number from -3 to 3, excluding 0, means how much she/he likes the fitness of glasses on person's face. By target of having more reliable results, the pictures are repeated with a constant rate.
In our frame recommendation study we used a database of 240 pictures collected from the Internet and created a survey that asked the subjects to rate the compatibility between the face shape and the frame shape in each photograph on a scale from -3 to

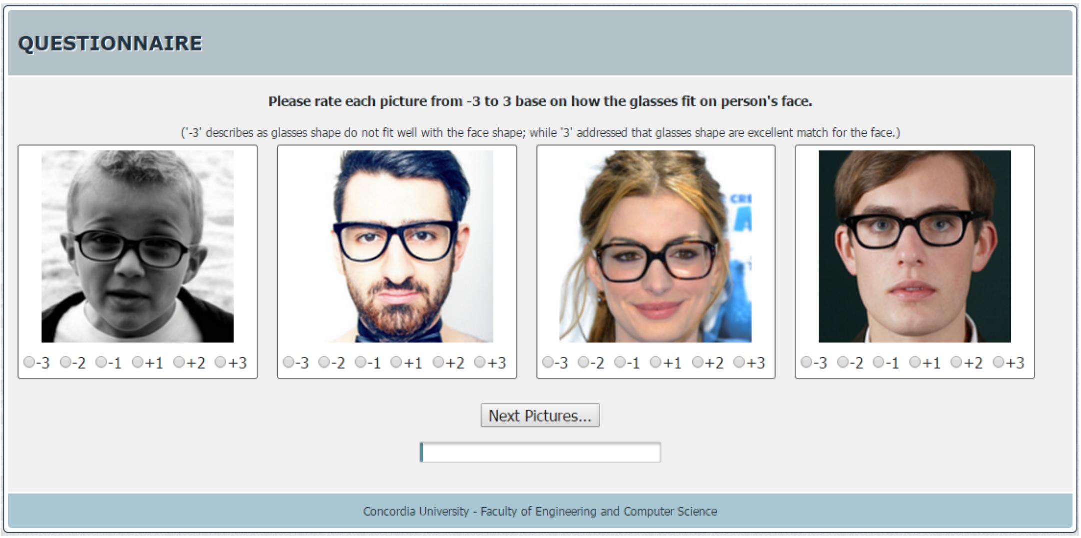

FIGURE 3.22: A view of survey website

3. To avoid too many neutral responses we did not allow a score of 0. Each subject voted on about 200 pictures picked at random.We had about 100 participants, obtaining overall more than 25,000 records.

Chapter 4

Results

After focusing on different parts of the system, in this chapter, the results of 3 main parts of the research will be considered.

4.1 Face Shape Recognition

Because of fuzzy nature of the problem, the degree of membership for a new pattern will be calculated on previous cases of CBR. Experiments clearly shows that system reuses the previously classified face patterns to predict the correct face shape type for any new face. we consider the closest two types and a blending score is computed by simply dividing by their sum. If the scores are within 60%-40% range we label the face type as a blend and during the recommendation type we allow recommendation from both face types.

Figure 4.1 is the confusion matrix calculated after testing phase, where the accuracy reached 80.33 %.

The lowest accuracy is for oblong class where the highest error is for misclassification of square samples. The main cause would be the distance to ellipse miscalculation because of face boundary detection error of the system. The highest mistakes rate is for misclassification of Heart and Diamond classes (13 and 11, respectively); because of the fact that they are exceedingly similar classes. The main discrepancy would be the relative eye-line of the face.

Note that the confusion matrix summation on only vertical lines is equal to zero (not horizontally), by the reason of 60%-40% rule that applied to the system which previously described. Increased accuracy clearly depicts that system trained itself after the first phase. This is because of the Case Based Reasoning, which allows system to recognize the face shape better after every iteration of testing with new samples.

	Oval	Diamond	Square	Heart	Oblong	Round
Oval	79	2	3	2	4	5
Diamond	3	82	2	13	3	2
Square	2	1	85	1	10	5
Heart	3	11	3	75	5	3
Oblong	8	3	1	2	77	1
Round	5	1	6	7	1	84

FIGURE 4.1: Confusion Matrix for FSRS system. the mean equals to 80.33%

We had over 100 different Face shapes that were recognized by domain experts like Fashion Designers, Hair Stylists.

300 new face shapes and 50 randomly selected from training data in 4 cycles (Cross validation). These 4 cycles include rotation of training, testing and cross validation. The mean recognition accuracy was calculated based on 4 fold, which come up with about 80 %.

Figure 4.2 illustrates the accuracy percentage of Face Classification during training procedure.

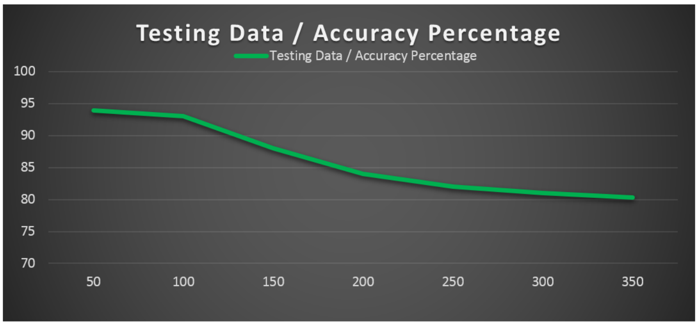

FIGURE 4.2: Accuracy Diagram during Testing Phase of FSRS

Note that during testing, the accuracy decreased to reach a steady state about 80%. Out of the 20% of misclassified faces, around half are misclassified due to partial failure of the face tracker. The classification on a dataset that has correct face extraction is about 90%.

4.2 Frame Shape Recognition

As mentioned earlier in classification sections, the face and frame parts are bear resemblance in using CBR as the classification method.

Since the frame extraction mechanism has a significant low error, the error in Frame classification is relatively small.

Figure 4.3 is the confusion matrix for frame recognition system. the mean value would be 88, while the highest accuracy is for rectangular class. In other words, system is able to classify the rectangular frame class with the highest precision, owing to the high accuracy of the frame extraction precision which it's result is the input of the frame classification.

The highest error is caused by misclassification of square frames (83%) in wayfarer frame class (11%). This occurred since the interior polygon of these two frame classes are almost identical. the only significant differences are distance to the fitted ellipse and symmetricity of the polygon.

	Oval	Wayfarer	Square	Aviator	Rectangular	Round
Oval	87	0	1	0	4	1
Wayfarer	0	90	11	9	0	3
Square	2	5	83	4	1	0
Aviator	0	4	3	84	1	5
Rectangular	6	0	2	1	93	0
Round	5	1	0	2	1	91

FIGURE 4.3: Confusion Matrix for Frame Recognition system. the mean equals to 88%

We used 60 (10 for each frame class) photographs for training and 240 for testing.

Our frame shape extractor reconstructs a correct polygon nearly 97% of the time and the shape classifier achieves an accuracy of 88%.

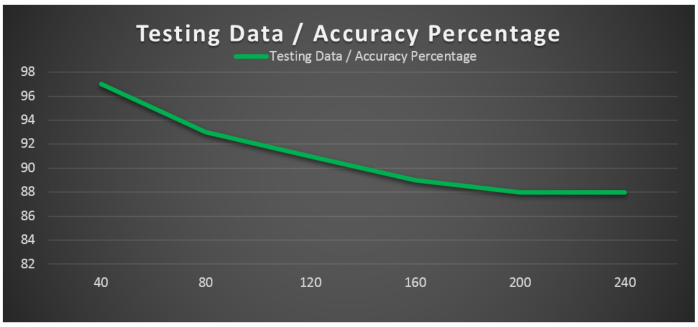

FIGURE 4.4: Accuracy Diagram during Testing Phase of FSRS

As indicated in overview of the system(Figure 3.1), after defining a group of pictures of people with glasses to the system as input, Face and Frame recognition procedures would be executed. The results clarify that classifications performed with over 80%

accuracy and ready for the survey to be filled by users. The next chapter indicates the glasses recommendation and validation phases results.

4.3 Glasses Recommendation System and Validation

For each face class, there are 6 frame classes available. the survey part of the research is responsible to weight these 36 different classes based on users' point of view.
In this part, 250 images of people with glasses collected from the web. Figure 4.5 depicts the population of each face/frame classes. in this diagram, x-axis is face/frame number. first number is face type (Oval, Square, Oblong, Round, Diamond, Heart) and second number is frame type (Oval, Round, Rectangular, Square, Aviator, Wayfarer). y-axis represents the number of pictures which related to this type of face/frame. As displayed, range of the figures in each category is from 5 to 25. The survey results show that the higher rated classes are from either high or low number of populations.

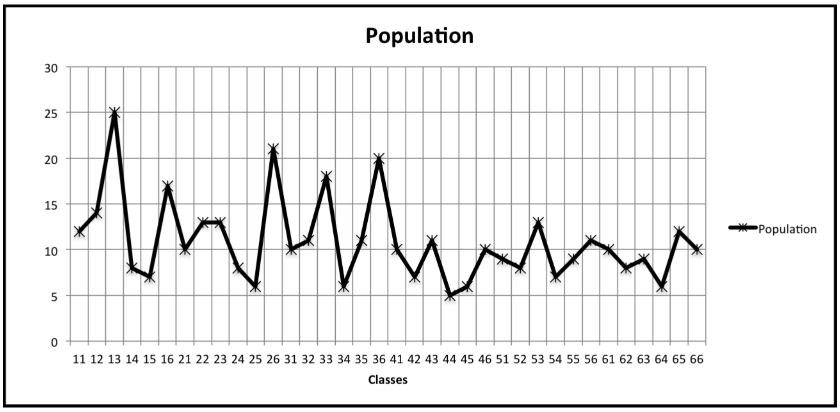

FIGURE 4.5: The population of each 36 face/frame classes (permutation of 6 face and 6 frame classes)

Figure 4.6 illustrates the number of votes for each image in each class from the user. In the survey, each of 250 images is repeated with a probability to results more reliable outputs. These diagrams depict the number of votes after polling process with about 100 participants. x-axis is face/frame number, while y-axis declares the number of votes. The range of votes is from 110 to 2470 for each single class. The classes with higher population received a higher number of votes, undoubtedly.

Table 4.1 reported the statistic results of the survey based on 100 participants ideas. In this table, Mean, Median, Mode and Standard Deviation for each class is calculated for different types of frames/faces. Therefore, each of the 36 possible combinations of face shape and frame type received between 500 and 2,500 records. The discrepancy between

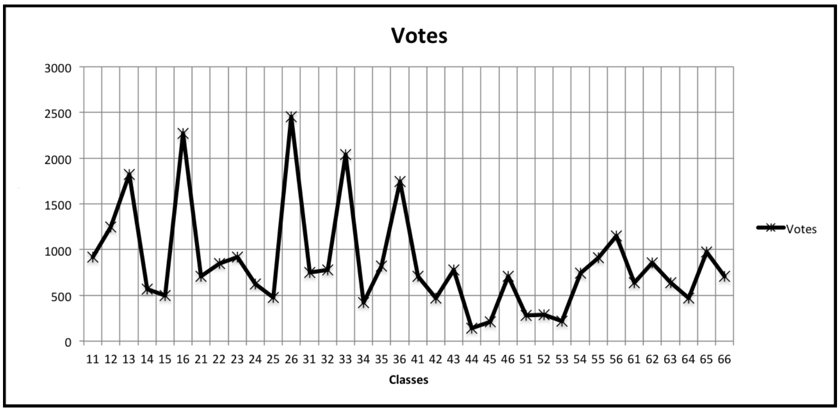

FIGURE 4.6: The number of votes for each 36 face/frame classes

the numbers comes from the fact that certain combination of face and eyeglass shapes are, unsurprisingly, more popular than others. Our recommendation system suggests the best two frame shapes for each input face shape. The results are summarized in Table 4.2.

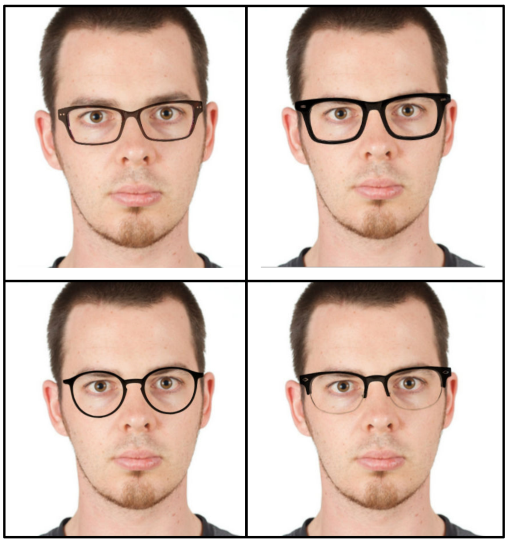

FIGURE 4.7: Example of one entry in the validation query: four frames were presented for the same person, two recommended by us and two other. Top row shows the system's top two recommendation.

To validate our recommendation system, we did a follow up study. A set of pictures of individuals without glasses and using one of the virtual try-on on-line systems [41] is collected. For each face type, four different types of frames are selected; two were the ones that the recommendation system selected and 2 selected randomly. Figure 4.7 shows one such example record. The 42 participants were ask to select their favorite frame. 82% of the times the selection made coincided with one of our recommendation. Table 4.3 depicts the results of validation based on survey selected cases for each face shape.

	Faces/Frames	Mean	Median	Mode	Standard Deviation
1	Oval/Oval	0.67215	1	2	1.86738
2	Oval/Round	0.501486	1	1	1.89754
3	Oval/Rectangular	1.1245	2	2	1.65835
4	Oval/Square	0.78836	1	2	1.95929
5	Oval/Aviator	0.199597	1	1	1.94959
6	Oval/Wayfarer	0.358466	1	2	1.94859
7	Square/Oval	0.476728	1	2	1.86908
8	Square/Round	0.44	1	1	1.88225
9	Square/Rectangular	1.24078	2	2	1.58919
10	Square/Square	1.49718	2	2	1.44043
11	Square/Aviator	0.868545	2	2	1.8466
12	Square/Wayfarer	0.544048	1	2	1.91986
13	Oblong/Oval	0.714085	1	1	1.85721
14	Oblong/Round	0.193342	1	1	1.97546
15	Oblong/Rectangular	1.14759	2	2	1.65368
16	Oblong/Square	0.55477	1	1	1.46563
17	Oblong/Aviator	0.569961	1	2	1.88805
18	Oblong/Wayfarer	0.860629	1	2	1.80551
19	Round/Oval	0.562764	1	2	1.83311
20	Round/Round	0.344758	1	1	1.86005
21	Round/Rectangular	0.260563	1	2	1.79861
22	Round/Square	0.658974	1	1	1.94168
23	Round/Aviator	0.943662	2	2	1.91648
24	Round/Wayfarer	0.207921	1	1	1.89111
25	Diamond/Oval	0.24735	1	1	1.85964
26	Diamond/Round	0.71831	1	1	1.79148
27	Diamond/Rectangular	1.13681	2	2	1.6843
28	Diamond/Square	0.4321	1	1	1.65321
29	Diamond/Aviator	1.21127	2	2	1.89822
30	Diamond/Wayfarer	0.739744	1	2	1.92958
31	Heart/Oval	0.54717	1	2	1.88155
32	Heart/Round	1.07042	1	1	1.70603
33	Heart/Rectangular	1.22727	2	2	1.62274
34	Heart/Square	1.08511	1	2	1.55052
35	Heart/Aviator	0.577465	1	2	2.094
36	Heart/Wayfarer	0.59887	1	2	1.93616

Table 4.1: Results of the survey based on about 100 participants

Face	1st Choice	2nd Choice
Oval	Rectangular	Square
Diamond	Aviator	Rectangular
Square	Square	Rectangular
Heart	Rectangular	Square
Oblong	Rectangular	Wayfarer
Round	Aviator	Square

Table 4.2: Summary of our recommendations based on collected data.

Face	1st Recommended	2nd Recommended
Oval	Rectangular (64%)	Square (9%)
Diamond	Aviator (58%)	Rectangular (30%)
Square	Square (54%)	Rectangular (33%)
Heart	Rectangular (68%)	Square (20%)
Oblong	Rectangular (58%)	Wayfarer (23%)
Round	Aviator (35%)	Square (46%)

Table 4.3: Validation Results. 82% of the times the selection made coincided with one of our recommendations.

Chapter 5

Conclusion

In this work we introduced a data-driven eyeglass frame recommendation system based on geometric feature extraction and analysis.
At the core of our system are a novel shape extraction method for eyeglass frame shape and classification methods for both face and frame shapes. The frame extraction method is efficient and more accurate and robust than previous methods. It works even on difficult subjects such as sunglasses with reflection, partially occluded face and thin metallic frames as well as subjects with significant pose rotation.
The classification method uses geometric features tailored for this particular problem in a case based reasoning framework. It provides good results despite the little shape variance in the shape of the face and frame shape boundary.
Our recommendation system is scalable and efficient and additional validation survey suggests that our systems recommendation is consistent to the reality.
In this chapter, after summarization of different parts of the system, limitations of the research will be described. Finally, based on these restrictions, possible future works of this research will be presented.

5.1 A Summary

Face Shape Recognition System: The main challenge of this part of the research was to extract special facial features that vary from regular facial features. (Regular facial features include distances between eye centers or nose points etc.) As this project deals with only face boundary points, regular facial features are of no use. Additionally, face shape patterns have many intersections and it could be impossible for a system to classify an instance of a single set (which equals to any face shape pattern). Hence, a fuzzy system is required to calculate a degree of membership to each pattern for any

testing data (input faces) of the system and classify them accordingly.
Moreover, Another difficulty faced was the calculation of angles present at the face edge points near the jaw.
As this project deals with the face points below the eye line, the important feature of the face, overall face height cannot be determined directly. Therefore to predict the height of the face, the ellipse drawn; considering all face points, is used. The height of this ellipse can be termed as the overall height of the face. Through experiments this idea was validated, as face shapes were getting detected properly.
This project accepts input image of any size. With regular normalization methods, it may not avoid the zooming effects in the image, such as an image which has face only 30 % of overall size (zoomed out effect).
To overcome this problem, instead of using direct extracted features of the face, the ratios between them are used such as instead of just considering Height of the face, (Height(all) / Width(all)) is considered.
With this solution of taking ratios, zooming effects do not affect the classification of images.
Our goal towards the implementation of a face shape recognition system is successfully accomplished . FSRS determines various face shapes using boundary points across the edges of faces. By choosing effective normalized accurate features our system is capable of detecting six different face shapes i.e. Oval, Round, Oblong, Square, Diamond and Heart with reliability.
In this part, various efficient and reliable algorithms were used to determine face shapes by extracting selected facial features and computing their inclination of face edges with variants of face shapes in a facial database. We have achieved a recognition accuracy of 80.33 % for a testing data of 150 images which included 100 new faces and 50 randomly selected from training data.

Glasses Frame Shape Recognition System: In this part, three main sub challenges have been addressed. First, extraction of glasses frame of the person in a reliable way. Second, clarifying the main features of the extracted frame in order to recognize different types of glasses and third, Frame shape classification procedure.
To have the frame polygon, Canny edge detection and Contour finding algorithms were used initially. Next, by utilizing the symmetric line of the face, the contours are pruned. Convex hull, Ellipse fitting, polygon intersection and area approximation algorithms are used to extract the exact polygon of the frame with over 97% precision.
After experimenting various types of glasses, There are 5 main features are defined to identify six different frame types. most of them are related to the fitted ellipse of the frame closed polygon. A normalization mechanism is applied to features. this process

ables system to have different size of pictures/faces as input.
In order to classify based on defined features of the frame, CBR was employed by the reason of its accuracy and swiftness. CBR ables the designer to weight different features due to their preference in the system. Weighting procedure is performed experimentally by investigating different images/frames.
88% precision is achieved in testing of 240 images with a set of 60 photograph training set.
Glasses Recommendation System: Face and Frame classifications are designed to establish the data-driven glasses recommendation system. A survey is ranked face and frame classified images base on participants ideas. Each image is rated from -3 to 3, excluding 0, base on how the glasses fit on person's face. '-3' describes as glasses shape do not fit well with the face shape; while '3' addressed that glasses shape are excellent match for the face.
Because of the automatic characteristic of face and frame classifications, It is possible to have various types of surveys with different quantity of images.
A validation process is designed to measure the accuracy of the recommendation process. For each face type, a photo without glasses is provided. 4 different frames are put on each face type. 2 of them are system's most recommended and the rest are chosen randomly. It showed that the glasses selection is more than 82% matched to the glasses recommendation choices.

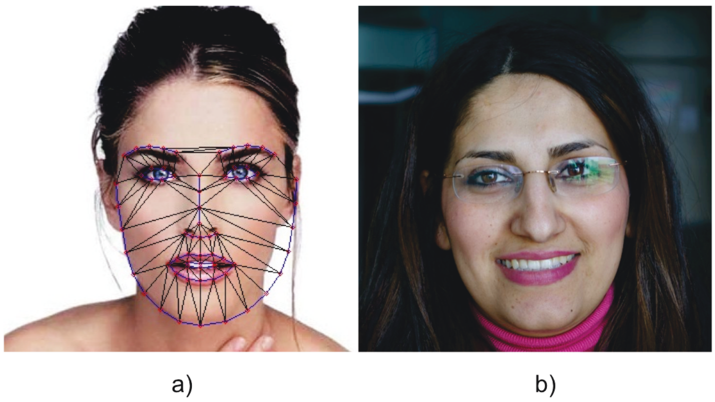

FIGURE 5.1: Failure cases. a) Face tracker failure. b) Rimless glasses.

5.2 Limitations

Although the classification mechanism is very robust, it can occasionally fail either due to failure of the face tracker 5.1a) or failure to correctly detect the polygon when the edges are not too strong as it is the case particularly with rimless frames 5.1b).

Another limitation of our experiment is that the database of pictures that we used is relatively small. However, this is not a fundamental limitation of the method. The CBR mechanism scales easily to thousand of pictures. and in fact we are planning in the future to make the survey and the recommendation system publicly available.

5.3 Future Work

The future work of the research could be designing a more precise face tracker to have better results, researching on mechanism to have more rotated images as input, algorithm to have video stream as input of classification process and a procedure to extract asymmetric glasses from images.

Bibliography

[1] What percentage of the population wears glasses? http://glassescrafter.com/information/percentage-population-wears-glasses.html, 2013.

[2] Roger L Terry and Carol S Brady. Effects of framed spectacles and contact lenses on self-ratings of facial attractiveness. *Perceptual and motor skills*, 42(3):789–790, 1976.

[3] URL http://www.framesdirect.com/landing/a/face-shape-guide.html.

[4] URL http://www.lenscrafters.com/lc-us/face-shape.

[5] URL http://iris.ca/choosing-the-right-frames-for-your-face-shape/.

[6] . URL http://www.youbeauty.com/face/sunglasses-for-square-face-shapes.

[7] Karl Grammer and Randy Thornhill. Human facial attractiveness and sexual selection: The role of symmetry and averageness. *Journal of comparative psychology*, 108(3):233, 1994.

[8] Joanna E Scheib, Steven W Gangestad, and Randy Thornhill. Facial attractiveness, symmetry and cues of good genes. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 266(1431):1913–1917, 1999.

[9] Randy Thornhill and Steven W Gangestad. Facial attractiveness. *Trends in cognitive sciences*, 3(12):452–460, 1999.

[10] Bernhard Fink and Ian Penton-Voak. Evolutionary psychology of facial attractiveness. *Current Directions in Psychological Science*, 11(5):154–158, 2002.

[11] Ira D Papel. Computer imaging for facial plastic surgery. *Facial plastic and reconstructive surgery (2nd edition), Thieme Medical Publishers, Inc, New York*, pages 110–115, 2002.

[12] Find the best haircut for your face shape. http://www.allure.com/hair-ideas/2011/find-the-best-haircut-for-your-face-shape, 2014.

[13] How to find the best hairstyle for face shape. http://www.glamour.com/lipstick/2009/03/how-to-find-the-best-hairstyle-for-your-face-shape/1, 2014.

[14] URL http://www.lancome.ca/.

[15] . URL http://www.youbeauty.com/face/makeup-for-your-face-shape.

[16] Tommer Leyvand, Daniel Cohen-Or, Gideon Dror, and Dani Lischinski. Data-driven enhancement of facial attractiveness. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2008)*, 27(3), August 2008.

[17] Andrea Bottino and Aldo Laurentini. The analysis of facial beauty: an emerging area of research in pattern analysis. In *Image Analysis and Recognition*, pages 425–435. Springer, 2010.

[18] Fangmei Chen and David Zhang. A benchmark for geometric facial beauty study. In David Zhang and Milan Sonka, editors, *Medical Biometrics*, volume 6165 of *Lecture Notes in Computer Science*, pages 21–32. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-13922-2. doi: 10.1007/978-3-642-13923-9_3. URL http://dx.doi.org/10.1007/978-3-642-13923-9_3.

[19] Ashok Samal and Prasana A Iyengar. Automatic recognition and analysis of human faces and facial expressions: A survey. *Pattern recognition*, 25(1):65–77, 1992.

[20] Jianguo Wang and Tieniu Tan. A new face detection method based on shape information. *Pattern Recognition Letters*, 21(6):463–471, 2000.

[21] Erik Hjelmås and Boon Kee Low. Face detection: A survey. *Computer vision and image understanding*, 83(3):236–274, 2001.

[22] Jason M Saragih, Simon Lucey, and Jeffrey F Cohn. Face alignment through subspace constrained mean-shifts. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1034–1041. IEEE, 2009.

[23] Sandro Schönborn, Andreas Forster, Bernhard Egger, and Thomas Vetter. A monte carlo strategy to integrate detection and model-based face analysis. In *Pattern Recognition*, pages 101–110. Springer, 2013.

[24] Christopher K Riesbeck and Roger C Schank. *Inside case-based reasoning*. Psychology Press, 2013.

[25] Kang B Sun and Boaz J Super. Classification of contour shapes using class segment sets. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 727–733. IEEE, 2005.

[26] Lena Gorelick, Meirav Galun, Eitan Sharon, Ronen Basri, and Achi Brandt. Shape representation and classification using the poisson equation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):1991–2005, 2006.

[27] Jamie Shotton, Andrew Blake, and Roberto Cipolla. Multiscale categorical object recognition using contour fragments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(7):1270–1281, 2008.

[28] Xinggang Wang, Bin Feng, Xiang Bai, Wenyu Liu, and Longin Jan Latecki. Bag of contour fragments for robust shape classification. *Pattern Recognition*, 47(6): 2116–2125, 2014.

[29] Zhong Jing and Robert Mariani. Glasses detection and extraction by deformable contour. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 2, pages 933–936. IEEE, 2000.

[30] Haiyuan Wu, Genki Yoshikawa, Tadayoshi Shioyama, Shihong Lao, and Masato Kawade. Glasses frame detection with 3d hough transform. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 2, pages 346–349. IEEE, 2002.

[31] Michelle Lai, Ipek Oruç, and Jason JS Barton. The role of skin texture and facial shape in representations of age and identity. *Cortex*, 49(1):252–265, 2013.

[32] Jeong-Seon Park, You Hwa Oh, Sang Chul Ahn, and Seong-Whan Lee. Glasses removal from facial image using recursive pca reconstruction. In *Audio-and Video-Based Biometric Person Authentication*, pages 369–376. Springer, 2003.

[33] Jeong-Seon Park, You Hwa Oh, Sang Chul Ahn, and Seong-Whan Lee. Glasses removal from facial image using recursive error compensation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(5):805–811, 2005.

[34] Chenyu Wu, Ce Liu, Heung-Yueng Shum, Ying-Qing Xy, and Zhengyou Zhang. Automatic eyeglasses removal from face images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(3):322–336, 2004.

[35] Xiaodong Jia and Jiangling Guo. Eyeglasses removal from facial image based on phase congruency. In *Image and Signal Processing (CISP), 2010 3rd International Congress on*, volume 4, pages 1859–1862. IEEE, 2010.

[36] Yi Xiao and Hong Yan. Extraction of glasses in human face images. In *Biometric Authentication*, pages 214–220. Springer, 2004.

[37] Diana Borza, Adrian Sergiu Darabant, and Radu Danescu. Eyeglasses lens contour extraction from facial images using an efficient shape description. *Sensors*, 13(10): 13638–13658, 2013.

[38] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59, 1994.

[39] Jack Sklansky. Finding the convex hull of a simple polygon. *Pattern Recognition Letters*, 1(2):79–83, 1982.

[40] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.

[41] URL http://www.glassesusa.com.