THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY

AUTOMATIC EXTRACTION OF AFFECTIVE

MULTIMODAL FACE VIDEOS

Master Thesis

CAN KANSIN

ISTANBUL, 2012

THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY

THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES ELECTRICAL AND ELECTRONICS ENGINEERING

AUTOMATIC EXTRACTION OF AFFECTIVE MULTIMODAL FACE VIDEOS

Master Thesis

Can KANSIN

Supervisor: Assoc. Prof. Çiğdem EROĞLU ERDEM

ISTANBUL, 2012

THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
ELECTRICAL AND ELECTRONICS ENGINEERING

Title of the Master’s Thesis: Automatic Extraction of Affective Multimodal Face Videos
Name/Last Name of the Student: Can KANSIN
Date of Thesis Defense: 10.09.2012

The thesis has been approved by the Graduate School of Natural and Applied Sciences.

Assoc. Prof. Tunç BOZBURA
Director

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

Assoc. Prof. Ufuk TÜRELİ
Program Coordinator

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

Examining Committee Members ______________________ Signatures ______________________

Assoc. Prof. Çiğdem EROĞLU ERDEM (supervisor) ______________________

Asst. Prof. Devrim ÜNAY ______________________

Asst. Prof. Olcay KURŞUN ______________________

ACKNOWLEDGEMENTS

First of all I would like to thank my thesis supervisor Assoc. Prof. Çiğdem Eroğlu Erdem for her guidance and patience. During my thesis work I was able to receive from her a true educational experience on image and video processing and scientific research methods. I would like to thank her for personally investing so much time in me, and bringing out my inner potential.

I would like to express my deepest gratitude to the members of the thesis committee, Asst. Prof. Olcay Kurşun and Asst. Prof. Devrim Ünay for their scientific advice and knowledge and many insightful discussions and suggestions.

This thesis has been supported by TÜBİTAK (Technical and Scientific Research Council of Turkey) under project EEAG-110E056.

07.09.2012 Can KANSIN

ABSTRACT

AUTOMATIC EXTRACTION OF AFFECTIVE MULTIMODAL FACE VIDEOS

Can Kansın

Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Çiğdem EROĞLU ERDEM

September 2012, 100 pages

Detection of human faces and estimation of affective information from facial images and videos is a research field, which has been very active in the last decade. Designing a system for estimation of the emotional (affective) and mental state of a person requires large annotated databases for the training and test phases. The available affective databases today are mostly recorded in controlled laboratory environments and contain acted expressions. Therefore, large databases that contain close to spontaneous expressions, with varying illumination conditions, subject ethnicities and subject ages are needed. However, such databases are very difficult and laborious to collect. In order to fulfill this need, we present an automatic system that can extract audio-visual facial clips from readily available movies and TV series, which are shot under close to real life conditions. The proposed system first automatically detects, and tracks all faces in a given video. The landmarks on the face are tracked using a Constrained Local Model based method. When the face tracking is no longer possible due to occlusions or a scene cut, the facial audio-visual video clip is extracted and written to a file, together with subtitles if available. The extracted video clips are manually evaluated in terms of their affective content and they are added to the database after quality check and annotation stages. The system has been successfully used to create an affective audio-visual database containing video clips in English and Turkish. The database (BAUM-2: BAhçeşehir University Multimodal affective database) is open to researchers and can easily be extended to include audio-visual clips in other languages.

Keywords: Facial Expression recognition, Emotion recognition, Affect Recognition, Spontaneous Expressions, Constrained Local Model, Audio-Visual Database

ÖZET

DUYGU İÇERİKLİ ÇOK BİÇİMLİ YÜZ VİDEOLARININ ELDE EDİLMESİ İÇİN OTOMATİK BİR YÖNTEM

Can Kansın

Elektrik Elektronik Mühendisliği Yüksek Lisans Programı

Tez Danışmanı: Doç. Dr. Çiğdem Eroğlu Erdem

Eylül 2012, 100 sayfa

Resim ve videolarda yer alan insan yüzlerini bulunması, takip edilmesi ve duygusal/zihinsel durum bilgisinin kestirilmesi uzun süredir araştırma yapılan bir alandır. Duygusal ve zihinsel durum kestirimi amacıyla bir yöntem geliştirmek için duygu içeriği olan resim ve video veri tabanlarına ihtiyaç duyulmaktadır. Hâlihazırda araştırmacıların erişimine açılmış olan veri tabanları, genellikle yapay yüz ifadeleri içerirler ve kontrollü koşullar altında laboratuvar ortamında kaydedilmişlerdir. Doğal duygusal ifadeler ve farklı aydınlatma koşulları içeren, farklı yaş ve etnik gruplardan insanlardan oluşan veri tabanlarına ihtiyaç vardır. Böyle veri tabanlarının derlenmesi ise oldukça zaman alan ve zahmetli bir süreçtir. Bu tezde, doğala daha yakın duygu ifadeleri içeren ses ve yüz veri tabanı elde edebilmek için otomatik bir yöntem geliştirilmiştir. Hâlihazırda var olan sinema filmleri ve televizyon dizileri çokça duygu içerikli yüz videoları içermektedir. Bu filmlerde yer alan insan yüzleri otomatik olarak tespit edilip, sahne değişimi ya da örtüşme nedeniyle takip edilmez oluncaya kadar kısıtlı yerel modeller kullanılarak otomatik olarak izlenmektedir. Yüzün başarıyla takip edildiği video klibi, ona ait ses ve varsa altyazı bilgileri ile beraber bir dosyaya kaydedilmektedir. Önerilen otomatik yöntem ile Türkçe ve İngilizce duygusal klipler içeren bir ses ve görüntü içeren veritabanı oluşturulmuştur. Bu veri tabanı araştırmacıların erişimine açık olup, diğer diller için de önerilen yöntem kullanılarak kolayca genişletilebilir.

Anahtar Kelimeler: Yüz İfadesi Tanıma, Duygu Tanıma, Kısıtlı Yerel Modeller, Ses ve Görüntü İçeren Yüz Veritabanı

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AAM	: Active Appearance Model
ASM	: Active Shape Model
CK	: Cohn-Kanade Database
CLM	: Constrained Local Models
CK+	: Extended Cohn Kanade Database
GPU	: Graphical Processing Unit
HCI	: Human Computer Interaction
RGB	: Red Green Blue
SIFT	: Shift Invariant Feature Transform
SURF	: Speeded-Up Robust Feature
SVM	: Support Vector Machines
VJ	: Viola Jones (Face Detector)

LIST OF SYMBOLS

Red Green Blue	: R, G, B
Mean	: μ_i
Convolution operation	: $*$
Color skin check function	: C
Image	: I
Convolution of a variable-scale Gaussian	: G
Difference-of-Gaussian	: D
Hessian operator	: H

1. INTRODUCTION

Perception of emotions from facial expressions and voice has a central function in human-to-human communication. It is envisioned that in the near future, the ability to recognize human emotions will have an important role for human-computer interaction applications (Zeng, 2009) For example, during security checks at borders and airports; it may be possible to understand if a person lies or not in an interrogation using affective face analysis. There are also life-saving applications, such as checking whether drivers are sleepy or not.

Automatic recognition of affective facial expressions is a challenging task. However, detection and analysis of faces in an image or video is necessary to develop more effective and friendly methods for human computer interaction, surveillance and security systems. With the increased media, such systems can be used massively.

In order to develop an affective recognition system, large databases containing emotional visual or audio-visual face videos is needed. The databases available for researchers today (Zeng, 2009) are generally recorded in laboratories under controlled conditions, while the actors are acting certain emotions.

For example the CK+ database is a very popular one (Cohn, et al., 2010), which contains videos of the six basic emotions together with an additional one contempt. The videos start with a neutral facial expression and end with an apex frame, where the facial expression is extreme. Another affective database eNTERFACE’05 (eNTERFACE'05, 2011) contains audio-visual video clips of the six basic emotions. Other examples of facial expression databases with our database (BAUM-2) are listed in Table 1.1.

Table 1.1: Comparison of facial expression databases. C means controlled and CTN means close to natural.

Database	Static/Dynamic	Lab Environment	Profile View	Illumination	Occlusion	Audio
BAUM-2	Dynamic	TV & Real	Yes	CTN	Yes	Yes
AFEW (Dhall, et al., 2011)	Dynamic	Real	Yes	CTN	Yes	Yes
Belfast (Sneddon, 2012)	Dynamic	TV & Lab	No	C	Yes	No
CK+ (Patrick Lucey, 2010)	Both	Lab	Yes	C		No
Facial Age (Fg net aging database, 2012)	Static	Real	Yes	CTN	Yes	No
F.TUM (Wallhoff, 2006)	Both	Lab	No	C	No	No
GEMEP (Scherer, et al., 2010)	Dynamic	Lab	Yes	C	Yes	No
LFW (Huang, et al., 2007)	Static	Real	Yes	CTN	Yes	No
M-PIE (Ralph Gross, 2008)	Both	Lab	Yes	C	Yes	No
MMI (Maja Pantic, 2005)	Both	Lab	Yes	C	Yes	No
UT-Dallas (Alice J. O'Toole, 2005)	Both	Lab	Yes	C	Yes	No

Source: (Dhall, et al., 2011)

Since acted databases contain exaggerated expressions, the systems trained on such databases do not generalize well to real-life conditions. Therefore, databases containing spontaneous or close to spontaneous facial expressions are needed. There are a few naturalistic databases available for researchers. For example Belfast database (Sneddon, 2012) is a well-known database which is trying to create a database with genuine emotions. Instead of reading text, people were recorded discussing emotive subjects like in chat shows.

1.1 Problem Definition and Motivation

Our goal is to automatically extract and create a naturalistic affective face database from the readily available films and TV shows. We present an automatic algorithm to extract affective faces from movies and TV series towards this goal. The algorithm first finds the face and tracks its position using a Constrained Local Model based approach. After the face is detected, 66 landmarks are located on the face and tracked until a scene cut occurs or the face is occluded. A video clip is extracted from the successfully tracked frames and written to a file, together with accompanying audio and subtitles, if available. The tracked landmark positions are also saved for future use by the researchers.

1.2 Contributions and Outline of the Thesis

The contributions of this thesis can be listed as follows:

1) The first contribution of this thesis is the experimental comparison of several state of the art face tracking libraries, by pinpointing their strengths and weaknesses. The compared libraries are:
    a. FaceTracker (Saragih, et al., 2009),
    b. Stasm (Milborrow, 2008)
    c. AsmLibrary (Yao, 2011).
2) The second contribution is to increase face detection and tracking performance of FaceTracker library for color videos. First, we combined skin color post-filtering with the Haar-feature based Viola-Jones face detector. That post filter checks if the skin colored pixels are sufficient within the window that face detector algorithm detects. With that post filter we decreased the false positive rate of the face detector. Also with skin color filter we detected the minimum bounding rectangle for the FaceTracker algorithm. Then, we enhanced the tracking capabilities of FaceTracker, using an algorithm based on SURF features. Using this method we can detect the scene cuts with a higher accuracy. For every frame we obtain the SURF feature vectors and compare them with each new window, which template matching algorithm finds.

3) The third contribution of this thesis is to integrate the improved FaceTracker with audio and subtitle processing functionalities to come up with an automatic system, which can take a long film and produce short audio-visual facial clips from it.
4) The fourth and the last contribution is the creation of a naturalistic affective face database, namely BAUM-2 (BAhçeşehir University Multimodal affective database). We processed 108 movies and obtained candidate facial clips from them. The candidate clips were analyzed and the ones that are of high quality in terms of affective content are added to the database. With the database we currently provide 700 video clips for 8 emotions (neutral, angry, contempt, disgust, fear, happiness, sadness and surprise). There are clips in Turkish and in English. The database can easily be extended to include more clips from other languages.

The publications made from this thesis (so far) are as follows:

- C. Kansın, Ç. E. Erdem, "Automatic Collection of an Audio-Visual Face Databse", *Workshop on Affective Computing for Mobile HCI, Sepember 17-18, İstanbul, Turkey*.
- C. Kansın, Ç. E. Erdem, "Automatic Extraction of Affective Multilingual Audio-Visual Facial Clips: BAUM-2 Database", submitted to *IEEE Int. Conf. Automatic Face and Gesture Recognition, 2013*.

The outline of the thesis is as follows. In the first chapter, we first present general background information about the problem and our approach about face detection, tracking and landmark detection. The second chapter presents the pre-processing methods we used. The third chapter gives a literature survey on face and facial feature tracking and compares the performances of several state of the art algorithms. The fourth chapter presents the developed algorithm and software for automatic extraction of facial video clips from movies. The fifth chapter presents an audio-visual database created using the developed facial video clip extraction tool and some baseline emotion recognition experiments on the database.

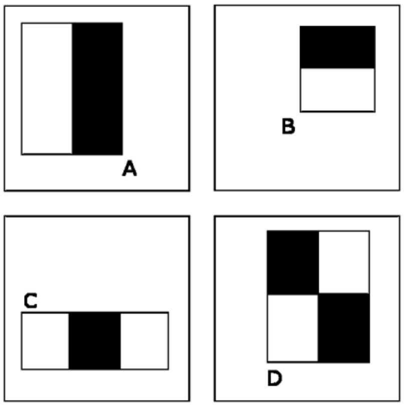
2. PRE-PROCESSING METHODS AND BASIC TOOLS USED FOR FACE DETECTION AND TRACKING

In this chapter, we give a brief overview of well-known methods and tools used for face detection and face tracking.

2.1 Viola and Jones Face Detection Algorithm

The face detection algorithm by Viola and Jones is a well-known and widely used face detector for gray-scale images due to its open implementation in OpenCV (opencv dev team, n.d.). This implementation is based on the paper “Rapid Object Detection using a Boosted Cascade of Simple Features” algorithm by “Viola and Jones" (Viola & Jones, 2001). This implementation is a widely used algorithm because of its easy use and robustness.

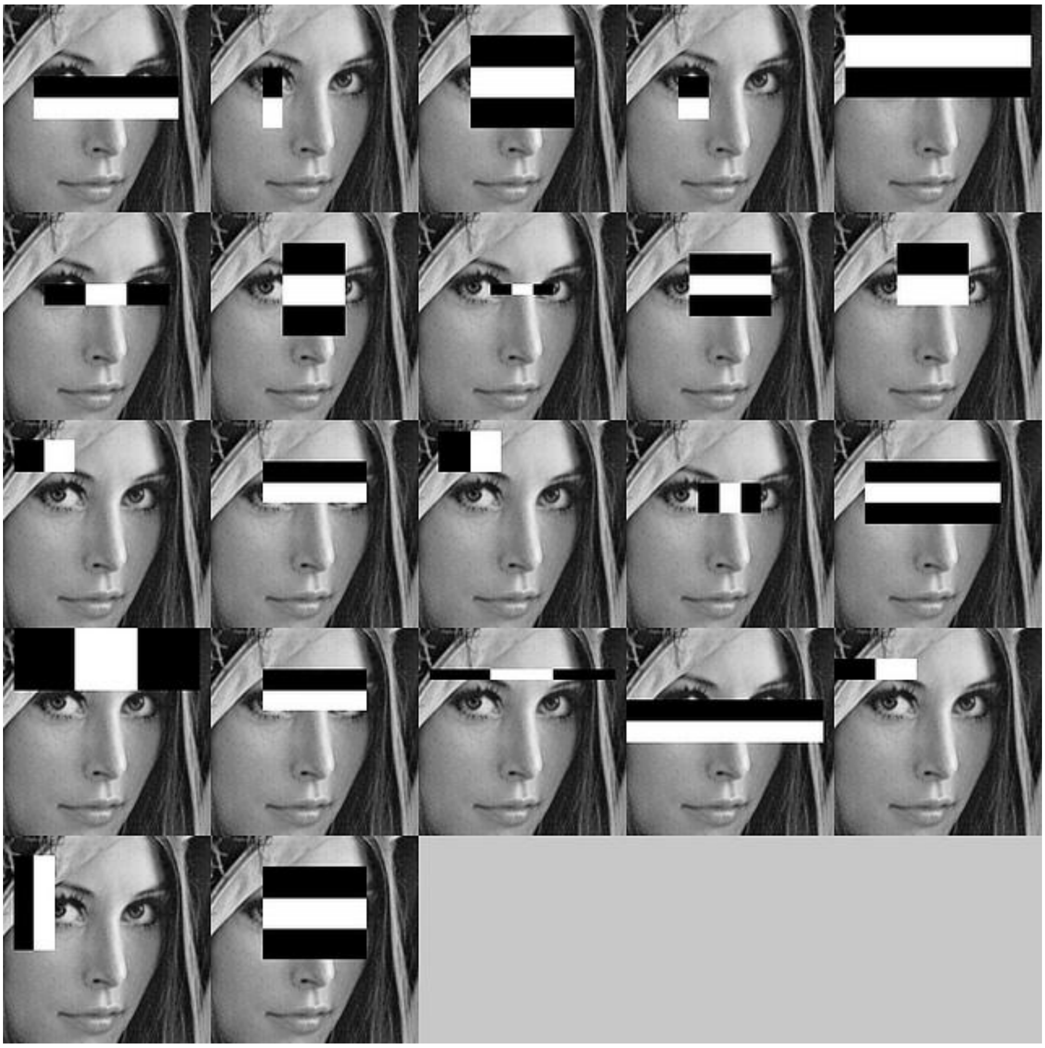
Figure 2.1: Feature types used by Viola and Jones



Source: (Wikipedia, 2012)

The main idea behind this face tracker is, instead of searching every picture pixel by pixel, categorized subsections, which were trained before is searched in a picture. The Haar like features are calculated using masks as shown in Figure 2.1. They are then classified using an Adaboost based classifier. The Haar-like feature based tracking algorithm looks for patterns for matching, which have been trained on before. Training is based on pictures, which have objects we want to detect and also without the objects. To increase the accuracy, we need to use lots of images. Algorithm’s detection window looks for adjacent rectangular areas in image to sum these area’s pixels intensities and calculate their difference. These calculations are used for categorization of the image subsections. Example of searching features within a picture shown in Figure 2.2.

Figure 2.2: An early stage in the Haar cascade



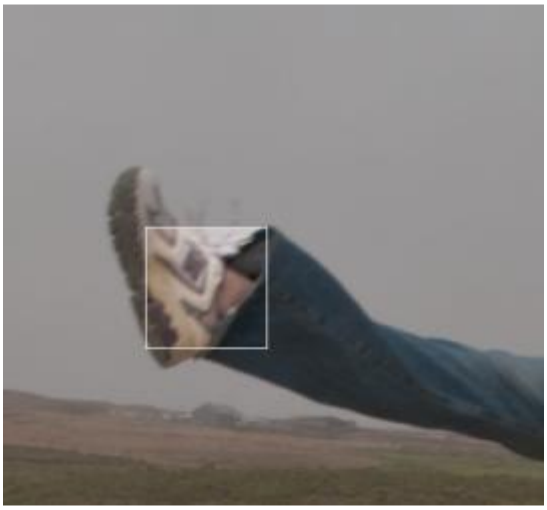
Source: (Borenstein, 2012)

After categorization step, each subsection will be compared with trained database to decide if the object that we are searching for is here or not. Because of the simplicity of this algorithm a lot of samples are needed for the training part.

2.2 Explicit Skin Color Detector Method

Skin color is an important cue for face detection in color images. Skin color detection is a difficult problem due to changes in environment, such as lighting and variation of poses, which cause occlusions. Face detection algorithms based on gray level patterns can be tricked by face like figures (see Figure 2.3 and Figure 2.5).

Figure 2.3: A face is detected within an image which has no face



Source: (Lowensohn, 2012)

In order to solve this problem, we used a method based on comparing red, green and blue values of a pixel (Ulukaya, et al., 2011). The method we used was found better than other models (Solina et al. 2003).

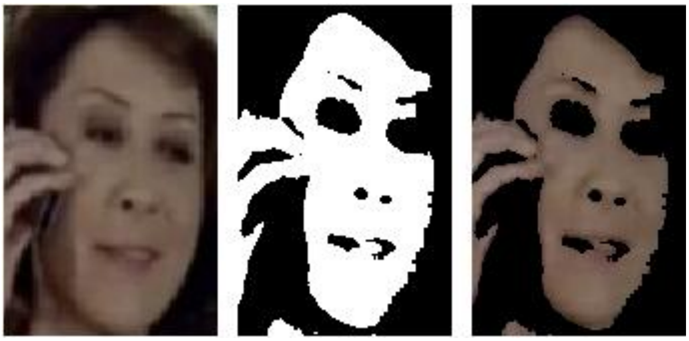
$R > 95 \ \&\& \ G > 40 \ \&\& \ B > 20$ **(2.1a)**

$(max(R, G, B) - min\ (R, G, B)) > 15$ **(2.1b)**

$max(R, G) - \ min(R, G) > 15 \ \&\& \ R > G \ \&\& \ R > B$ **(2.1c)**

(2.1a) indicates that the skin color should obey these rules and RGB ranges. Especially red is the dominant color of a skin. (2.1b/c) checks are to make sure that colors are not close, which creates greyness effect. In Figure 2.4 you can see the result of this color filter.

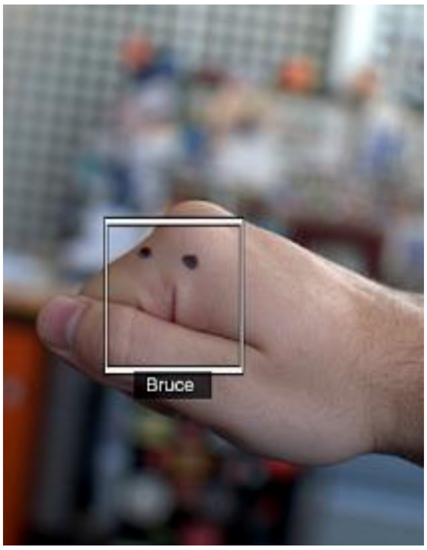
Figure 2.4: Using skin color filter for face estimation



(a) raw image (b) binary skin (c) skin-filtered

This method is not very accurate for every condition and works under some constraints. But the speed of this method provides us a robust and fast algorithm to implement and use. Despite the fact this method cannot eliminate false detections such as the skin colored face-like pictures such as the one in Figure 2.5 we use this model; because our inputs will be generally based on TV shows and movies. This type of sources should have more real faces than rare skin colored face-like objects.

Figure 2.5: An example of a face like image detected as a face.

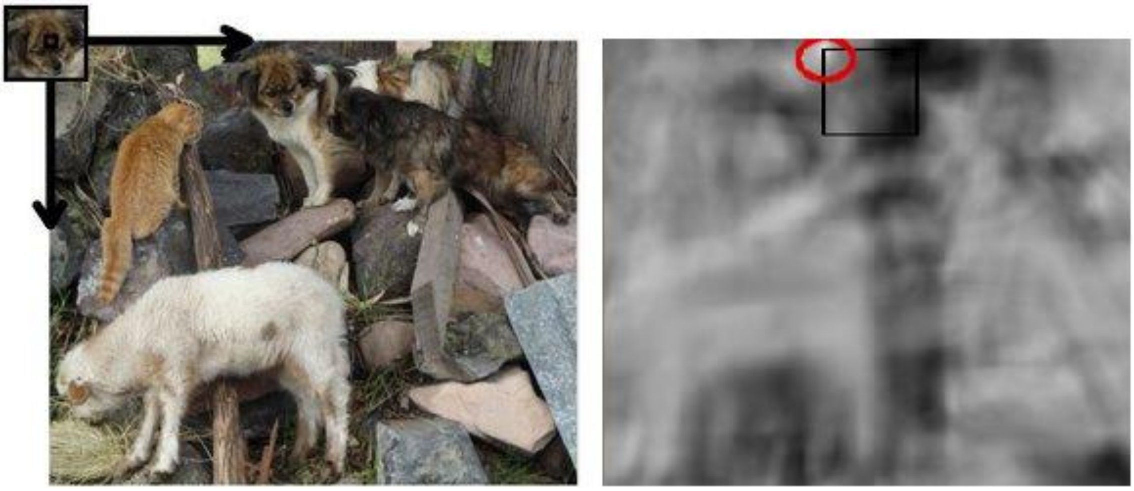


Source: (Lowensohn, 2012)

2.3 Template Matching

Template matching method is a simple method to track our faces within new frames. Basically template matching algorithm needs one source image and one template image. Template image is the image of the item we want to search for. Source image is the image we will be searching on. Logic of the template matching is sliding template image on source image pixel by pixel. For every sliding operation algorithm creates a score based on the similarity of the two images into another metric image. After searching the metric image, we can find the template image position or decide that our template image is not in that source image by using threshold values.

Figure 2.6: Example of template matching



(a)Template image and source image (b)Each location contains the match metric

Source: (opencv, 2012)

OpenCV (opencv, 2012) provides us with a function named cvMatchTemplate which can use several methods. In our project we used equation (2.2) where T means template image and I means source image.

$$R(x,y) = \frac{\sum_{x\prime,y\prime}(T'(x',y').I'(x+x',y+y'))}{\sqrt{\sum_{x\prime,y\prime}T'(x',y')^2.\sum_{x\prime,y\prime}I'(x+x',y+y')^2}} \quad \textbf{(2.2)}$$

2.4 Scale-Invariant Feature Transform

Scale Invariant Feature Transform (SIFT) is an algorithm to detect and track local feature descriptors which are greatly invariant to changes of scaling and rotation. This algorithm is first proposed by David Lowe (Lowe, 2004), and then improved. The SIFT algorithm runs on grayscale images. There are other invariant feature descriptors, which are lighter and faster like SURF (Herbert Bay, 2006). They are faster in computation while the SIFT detector has better matching performance in an empirical study of Luo Juan and Oubong Gwun, (Juan & Gwun, 2009).

SIFT feature detector algorithm has four stages:

a) Extrema detection in scale space

b) Keypoint localization

c) Estimation of Orientation

d) Calculation of descriptor.

Because of the capabilities of Gaussian function it was selected as the scaling method. To generate an image in scale space L(x,y,σ) this function used:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \qquad \textbf{(2.3)}$$

and

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \qquad \textbf{(2.4)}$$

where I means input image, L means image in scale space and G means convolution of a variable-scale Gaussian.

For an efficient detection of stable key points, extrema in the difference of Gaussian function are searched. Result is the difference between two Gaussian images separated by a multiplicative factor k:

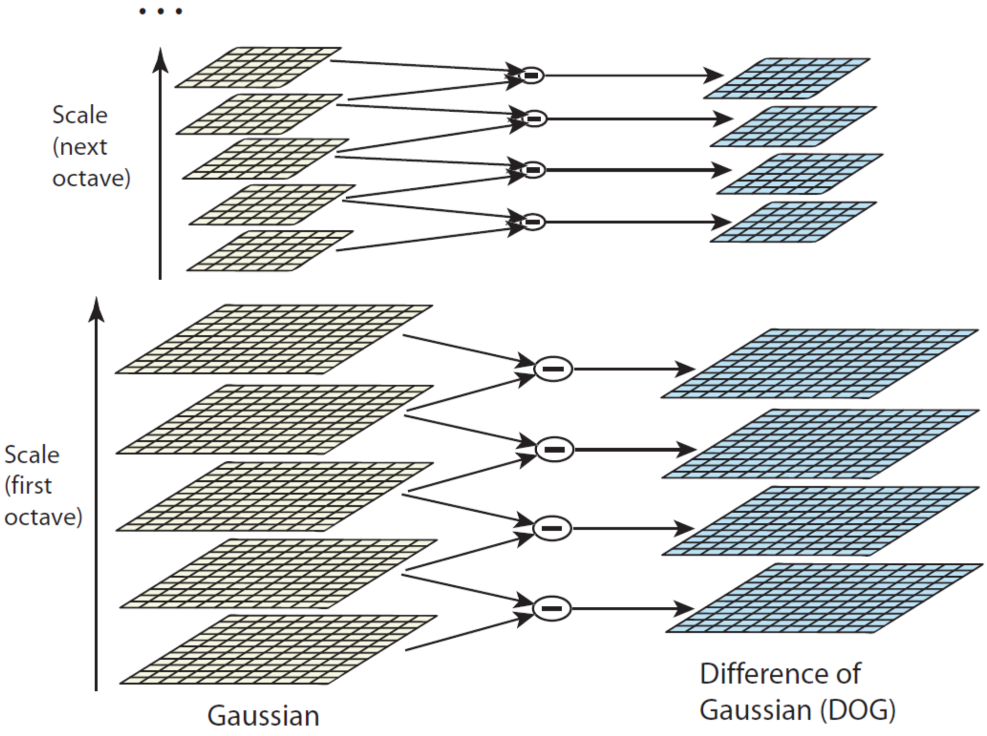
$$D(x,y,\sigma) = \big(G(x,y,k\sigma) - G(x,y,\sigma)\big) * \ I(x,y) \quad \textbf{(2.5a)}$$

$$L(x,y,k\sigma) - L(x,y,\sigma). \quad \textbf{(2.5b)}$$

where D means difference-of-Gaussian.

Figure 2.7 shows the scale procedure in detail. The input image is blurred with different values of σ which creates a Gaussian tree with finite levels.

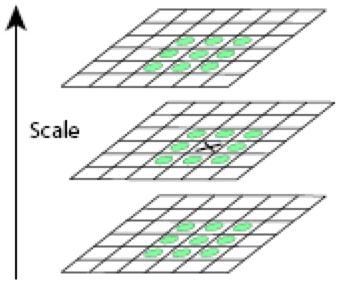
Figure 2.7: Gaussian Pyramid



Source: (Lowe, 2004)

Gaussian tree images are then checked for local extrema: A point is considered a minimum or a maximum, if it is by a given threshold higher or lower than his 26 neighbors (9+8+9) in the difference of Gaussian space:

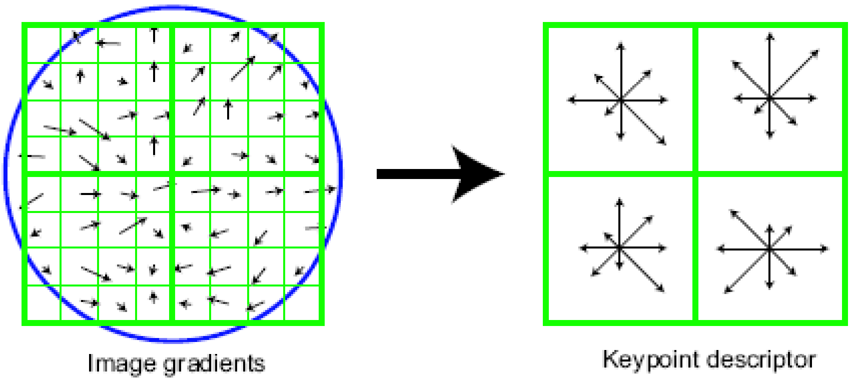
Figure 2.8: Maxima and minima of the difference-of-Gaussian images



Source: (Lowe, 2004)

SIFT descriptors are calculated for each feature point. Gradients are weighted by a Gaussian window ordered by 8 directions for each sub region around the feature location. These gradients are weighted by their distances from the central feature point by slicing 36 vectors that cover 10 degrees each (360). Orientation vector is selected with the peak value by fitting a parabola.

Figure 2.9: SIFT descriptor



Source: (Lowe, 2004)

2.5 Speeded-Up Robust Feature

SURF is a robust image detector and descriptor, first presented by Herbert Bay et al. in 2006. It generally has the same capability in terms of repeatability, distinctiveness, and robustness even in some operations outperform previously proposed methods, yet can be computed and compared much faster. This is achieved by relying on integral images for image convolutions and using the leading existing detectors and descriptors.

SURF algorithm first creates integral images with summing values between and the origin:

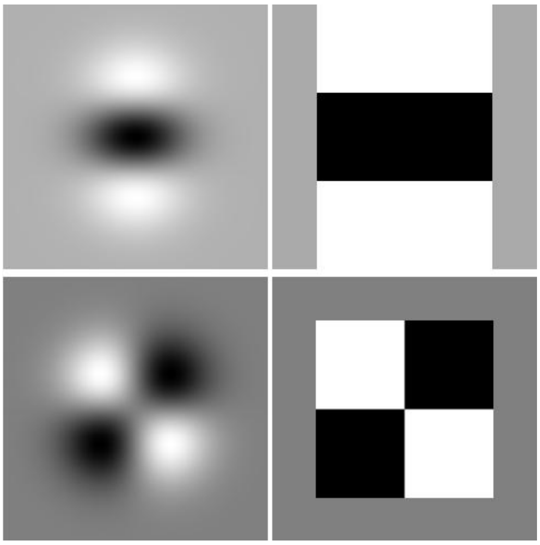
$$I_S(x,y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(x,y) \quad \textbf{(2.6)}$$

Then, algorithm generates points using Hessian operator at several scale of the images and select maxima response of the determinant of Hessian matrix.

$$\mathrm{H}(\mathrm{x},\sigma) = \begin{bmatrix} \mathrm{L_{xx}}(\mathrm{x},\sigma) & \mathrm{L_{xy}}(\mathrm{x},\sigma) \\ \mathrm{L_{xy}}(\mathrm{x},\sigma) & \mathrm{L_{yy}}(\mathrm{x},\sigma) \end{bmatrix} \quad \textbf{(2.7)}$$

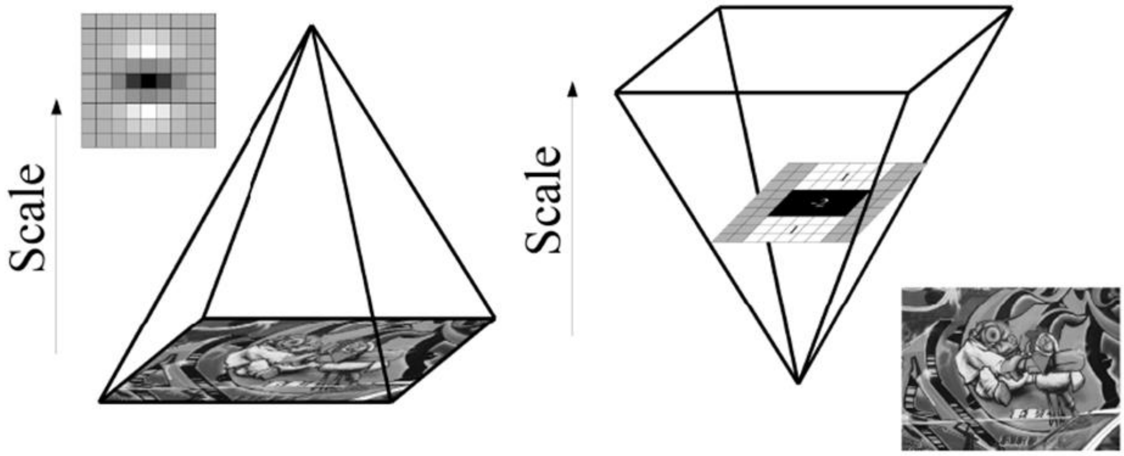
SIFT algorithm used difference of Gaussians instead of approximation of Laplacian of Gaussians to increase performance. In a similar manner, SURF algorithm used box filter representations of the respective kernels instead of approximation of Laplacian of Gaussians. Figure 2.10 shows approximated second order derivatives with box filters. Instead of traditional Gaussian pyramids SURF algorithm use these box filters for every level.

Figure 2.10: Second order derivatives of Gaussian and their approximation with box filters



Source: (Oyallon & Rabin, 2012)

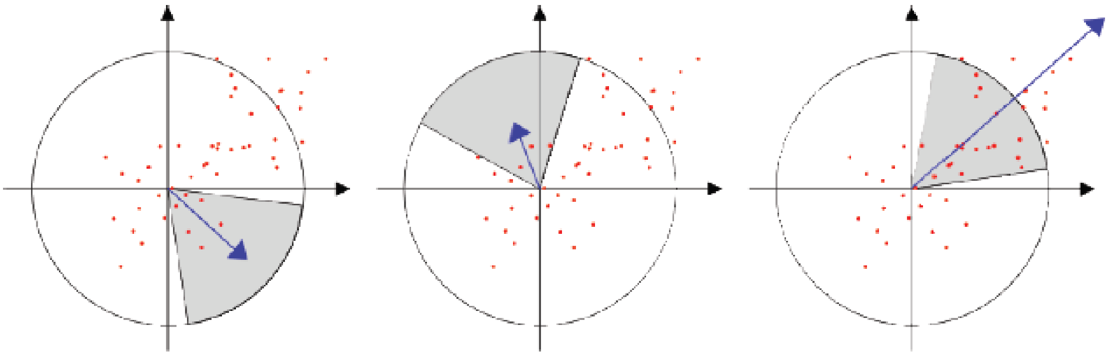
Figure 2.11: Using of integral images instead of reducing image sizes



Source: (Bay, et al., 2008)

Like SIFT, SURF also estimates the dominant orientation of each interest point by sliding an orientation window around interest point with intend of detecting dominant orientation of the Gaussian weighted Haar wavelet responses (Figure 2.12). Result is a computation of the descriptor (16 x 4 vector) corresponding to the scaled and oriented neighborhood of the interest point:

Figure 2.12: Orientation Assignment



Source: (Evans, 2012)

2.6 Support Vector Machine

A support vector machine (SVM) is a supervised learning method that analyzes data and recognizes patterns, used for classification and regression analysis (Michel, 2003). The algorithm takes a set of input data and predicts, for each given input, which of two or more possible classes forms the input, making the SVM a non-probabilistic linear classifier. Given a set of training examples, each marked as belonging to one of two or more categories, an SVM training algorithm builds a model that assigns new examples into one category or another.

For a binary classification, training examples

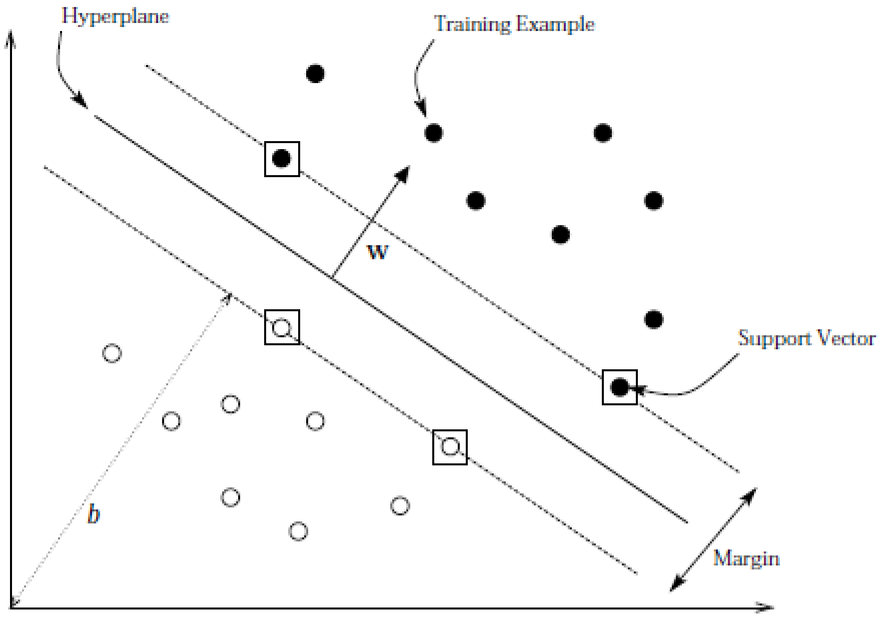
$$S = \big((x_1, y_1), \dots, (x_1, y_1)\big), y_i \in \{-1,1\} \quad \textbf{(2.8)}$$

where the x_i are input data and the y_i are labels, learning systems typically try to find a decision function of the form

$$h(x) = sgn((w.x) + b) \quad \textbf{(2.9)}$$

where w is a vector of weights and b is called the bias that yields a label ∈ $\{-1,1\}$ (for the basic case of binary classification) for a previously unseen example x.

Figure 2.13: A linear maximal margin classifier



Source (Michel, 2003)

3. LITERATURE SURVEY ON FACIAL FEATURE TRACKING LIBRARIES AND ALGORITHMS

There are numerous algorithms in the literature for face detection, face tracking and tracking the salient points on the face (facial landmark points). In this chapter, we review and compare experimentally several face tracking algorithms, which are based on statistical modeling of the shape and appearance. Even though all the compared algorithms finds the same face parts (eyebrow, eye, lips etc.), there are differences in positions of detected facial landmarks (for example finding inner and outer corners of the eye or eyebrow). The methods compared are:

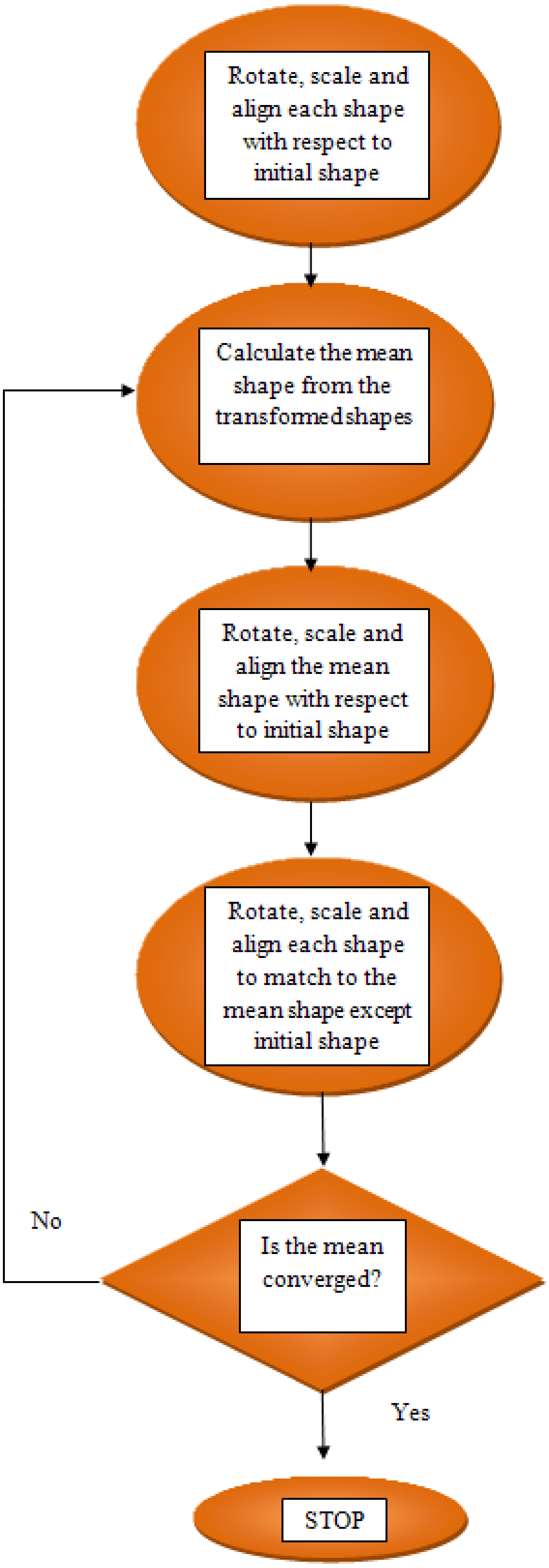
1. STASM: The method developed by Milborrow (Milborrow, 2008), based on the extension of Active Shape Models (ASM) and Stasm Software Library that belongs to this method.
2. ASMLIBRARY: Software Library based on ASM, developed by Yao Wei (Yao, 2011).
3. FaceTracker: Face tracking method based on constrained local models (CLM), developed by J. M. Saragih (Saragih, et al., 2009).

In the following, the three methods above are first explained and then compared empirically.

3.1 Active Shape Models

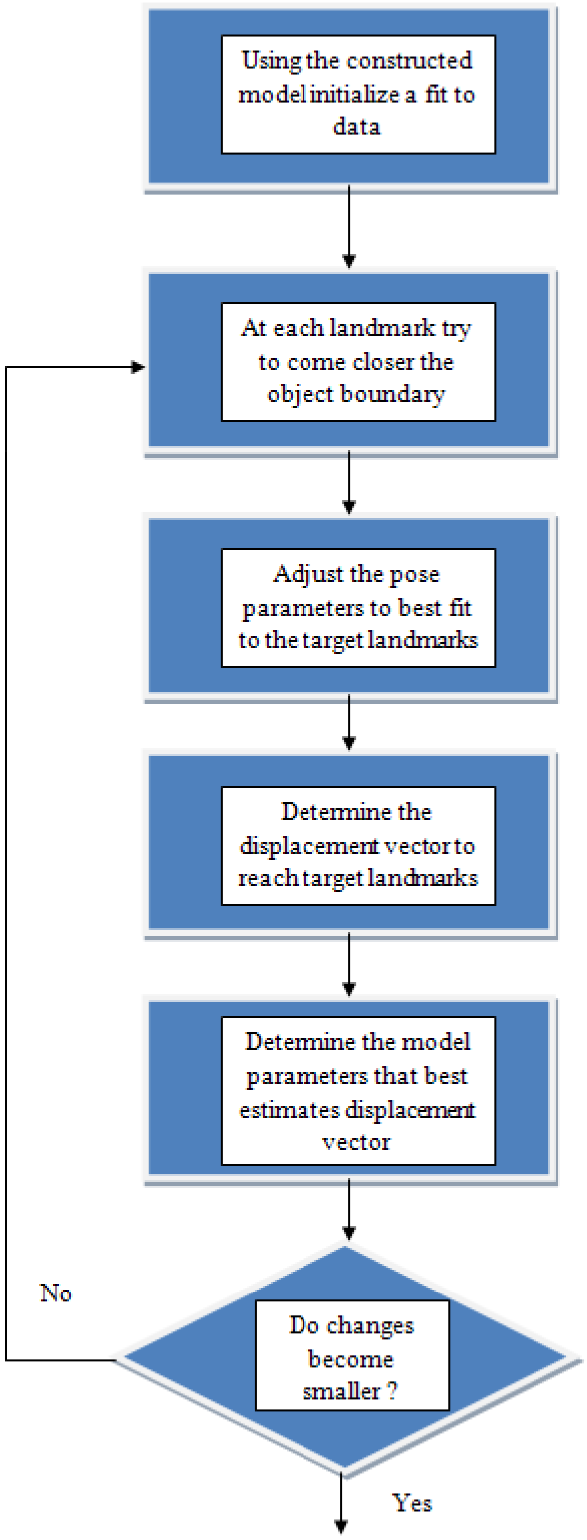
In order to use Active Shape Models (ASM) (Taylor, et al., 1994) in face tracking, the algorithm must be trained first. For the training part, we must manually mark the landmark points on many pictures. First, the faces are aligned. After that, ASM algorithm specifies the model’s main modes (components) by Principal Component Analysis (PCA) (Wikipedia, 2012). ASM algorithm also saves the texture data matrix, which is vertical to the control points to correct the position of the point in the search process. After completing ASM training, shape search process on a new image can start. Search process is an iterative process. Deformation of the first contour by the first image, and the training within the limits of the search profiles drawn perpendicular to the contour mapping process cause the positioning of contours provided to the best place.

Figure 3.1: Flowchart of ASM generation


Rotate, scale and align each shape with respect to initial shape
Calculate the mean shape from the transformed shapes
Rotate, scale and align the mean shape with respect to initial shape
Rotate, scale and align each shape to match to the mean shape except initial shape
No
Is the mean converged?
Yes
STOP

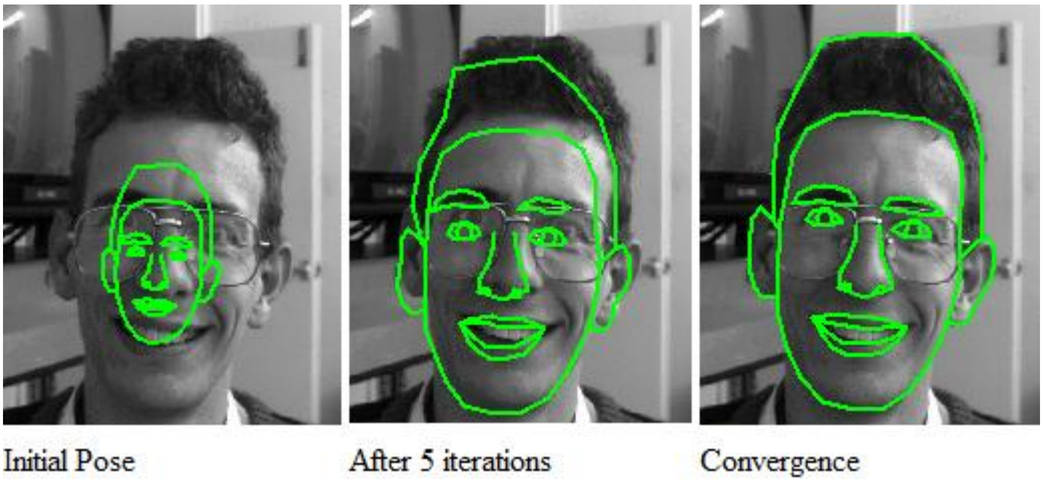
Source: (Sonka, et al., 2008)

Figure 3.2: Flowchart of ASM fitting algorithm


Using the constructed model initialize a fit to data
At each landmark try to come closer the object boundary
Adjust the pose parameters to best fit to the target landmarks
Determine the displacement vector to reach target landmarks
Determine the model parameters that best estimates displacement vector
Do changes become smaller ?
No
Yes

Source: (Sonka, et al., 2008)

Figure 3.3: Example of a multi-resolution search



Initial Pose

After 5 iterations

Convergence

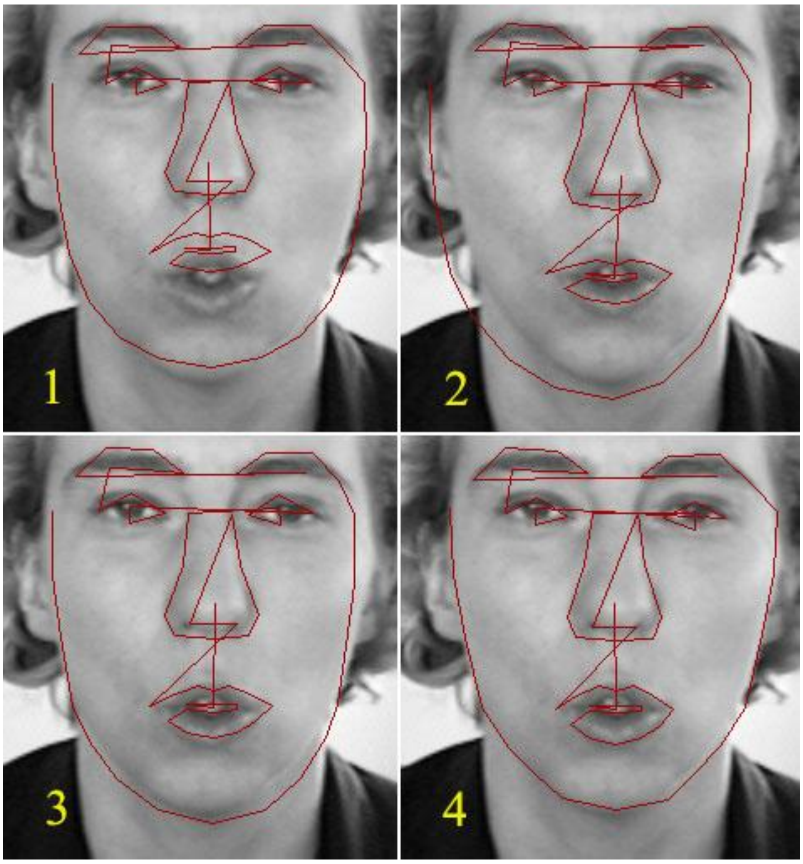
Source: (Cootes, 2004)

3.1.1 STASM Method

Stasm Library uses Active Shape Models (Milborrow, 2008). In one image, faces are first detected by the Viola Jones's method, tracked by ASM method and then the steps below are used to improve the algorithm.

- Using not one but two dimensional feature profiles to make more reliable profile comparisons.
- Using two Active Shape Models one after another to make a better starting shape.
- Increasing the number of features for a better positioning of face features and better results.
- Equalizing most of the covariance matrixes faster to zero.
- Making some changes like adding noise to training sets to make it a more robust algorithm.

Figure 3.4: An example for ASM search



Source: (Milborrow, 2008)

3.1.2 AsmLibrary

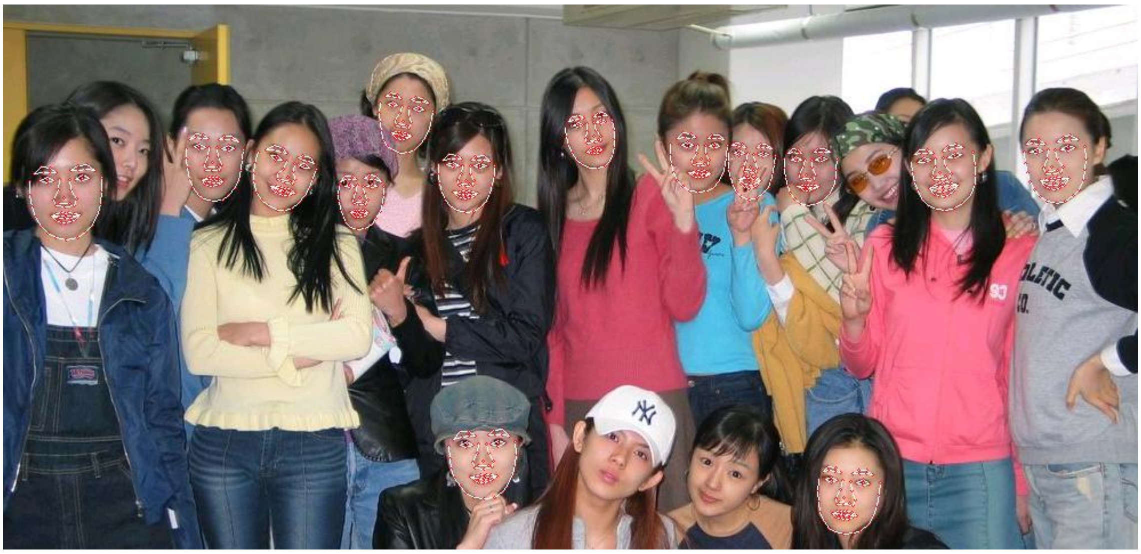
AsmLibrary is a face tracking library, which uses the ASM method. It was developed by Wei Yao, who derived it from Stasm Library (Yao, 2011). It is based on OpenCV (opencv, 2012). Compared to other libraries, it is better in finding more than one face in an image but has limited documentation. An example is seen in Figure 3.5. It has two modes:

a) Static image alignment with one person or multiple persons

b) Dynamic face tracking with only one person

- Video file
- Live camera

Figure 3.5: Result of multi face detection of AsmLibrary

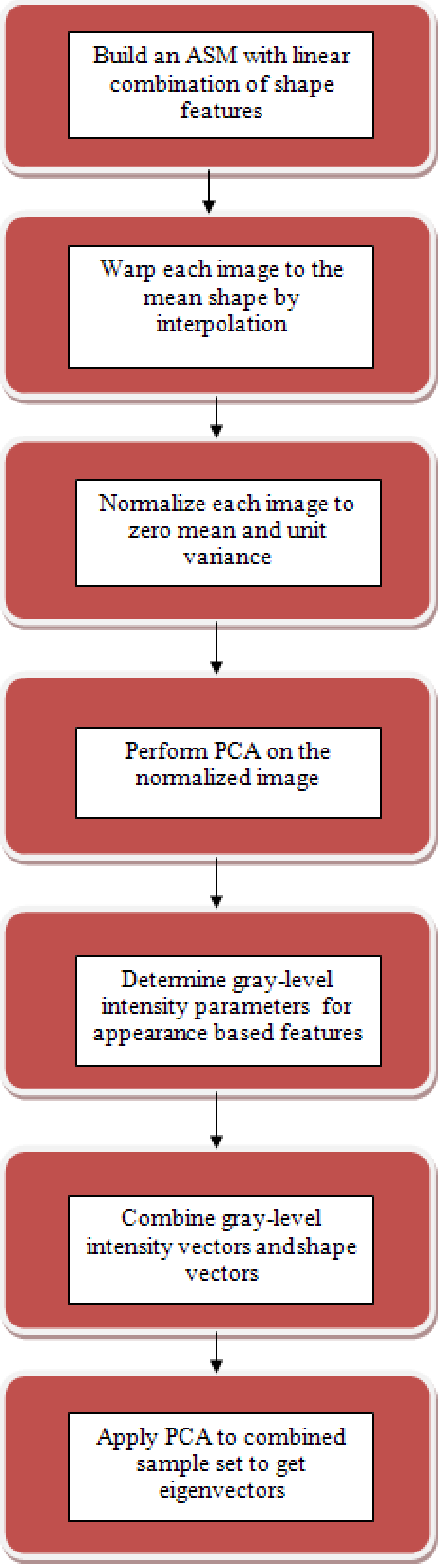


Source: (Yao, 2011)

3.2 Active Appearance Model (AAM)

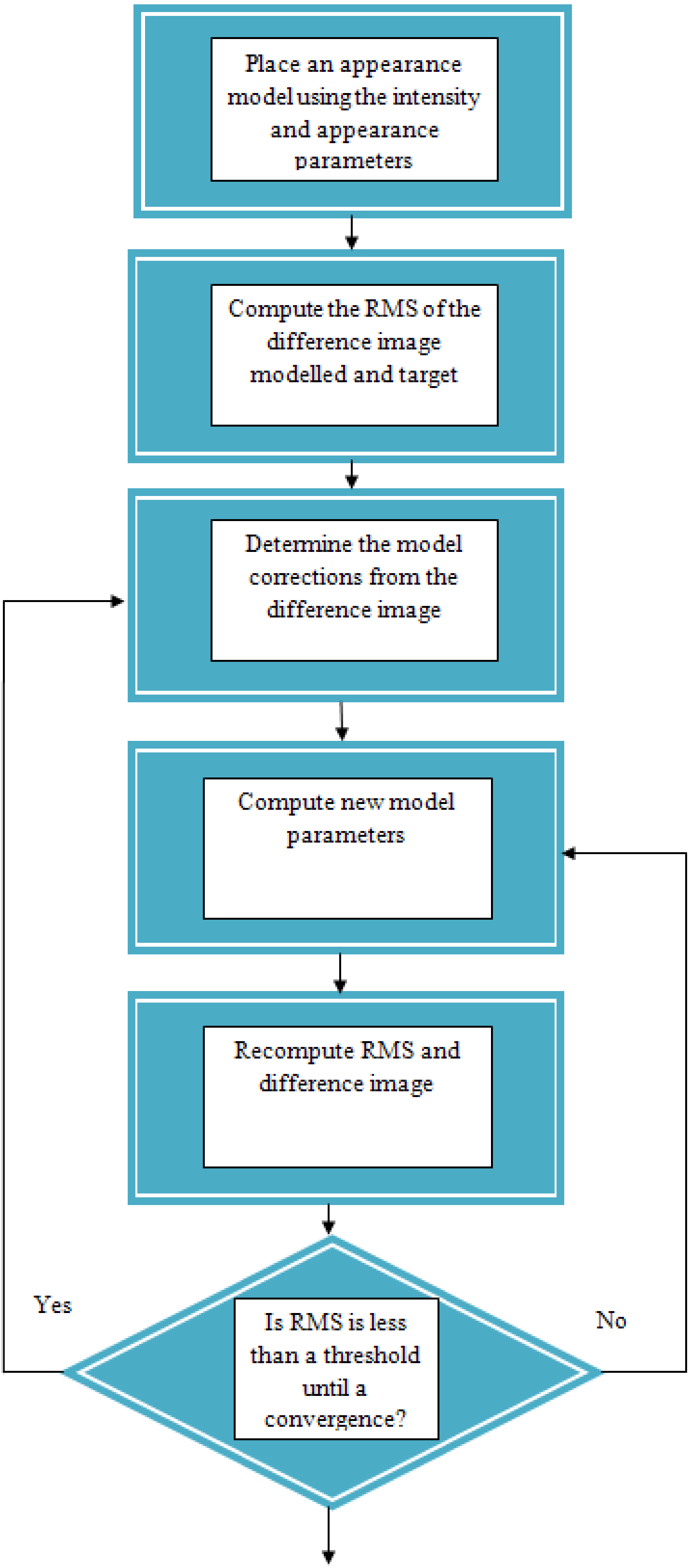
Active Appearance Model (Kittipanya-ngam & Cootes, 2002) needs a training set like ASM In training part, AAM learns a linear model. This linear model is a relationship between induced residuals and parameter displacements. To obtain a better fit logic, AAM uses this linear model to correct current parameters by measuring residuals. Even with a bad starting estimation an overall good match landmarks can be obtainable with this training. AAM algorithm's aim is not only to generate a shape model, but also to produce an appearance model. An AAM is a statistical model of the shape and grey-level appearance of the object. In AAM segmentation transformation parameters, appearance coefficients and global intensity parameters must be optimized (Sonka, et al., 2008).

Figure 3.6: Flowchart of AAM construction algorithm


Build an ASM with linear combination of shape features
Warp each image to the mean shape by interpolation
Normalize each image to zero mean and unit variance
Perform PCA on the normalized image
Determine gray-level intensity parameters for appearance based features
Combine gray-level intensity vectors and shape vectors
Apply PCA to combined sample set to get eigenvectors

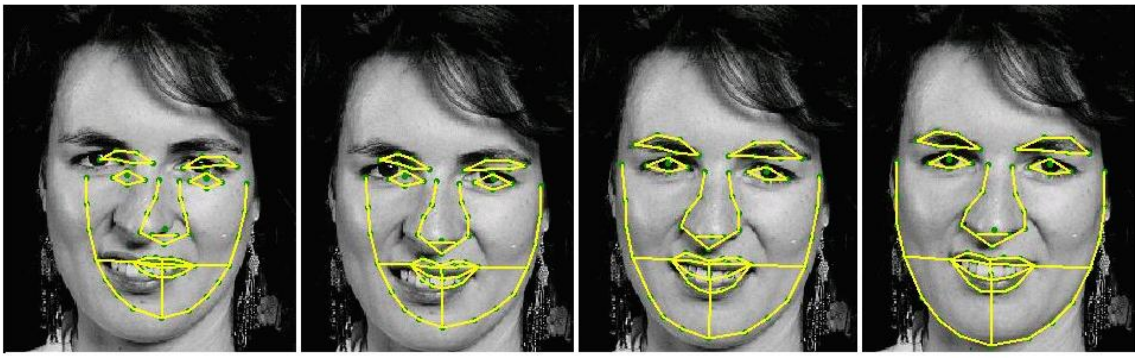
Source: (Sonka, et al., 2008)

Figure 3.7: Flowchart of AAM matching algorithm


Place an appearance model using the intensity and appearance parameters
Compute the RMS of the difference image modelled and target
Determine the model corrections from the difference image
Compute new model parameters
Recompute RMS and difference image
Is RMS is less than a threshold until a convergence?
Yes
No

Source: (Sonka, et al., 2008)

Figure 3.8: An AAM search



Initial Position

After 1 iteration

After 2 iterations

Convergence

Source: (Cootes, 2012)

3.2.1 Face Alignment through Subspace Constrained Mean-Shifts (FaceTracker)

This library was written by, Jason M. Saragih, Simon Lucey and Jeffrey F. Cohn at Carnegie Mellon University (Saragih, et al., 2009). On the contrary to the previous libraries, this library uses an algorithm closer to Active Appearance Model (AAM). In this method they indicate that algorithm is using a nonparametric representation instead of approximating the response maps for each PDM landmark using parametric models. They are using a homoscedastic kernel density estimate (KDE) with an isotropic Gaussian kernel; because of KDE has the advantage that no nonlinear optimization is required to learn the parameters of its representation. This library is superior to others because; it can find faces when they are in a different angle or even some parts occluded and can place face landmarks. The flow of this algorithm is as follows:

Require: I and p (where I is the image and p is the PDM parameters).

```
A. While not converged(p) do
      a. Compute responses
         $p(l_i = aligned \mid I, x) = \frac{1}{1+\exp\{\alpha C_i(I;x)+\beta\}}$        (2.10)
      b. Linearize shape model
         $x_i \approx x_i^c + J_i \Delta p$        (2.11)
      c. Precompute pseudo-inverse of Jacobian ($J^{\dagger}$)
      d. Initialize parameter updates: Δp ←0
      e. While not converged(Δp) do
            i. Compute mean-shifted landmarks
```

$$x_i^{\tau+1} \leftarrow \sum_{\mu_i \in \varphi_{x_i^c}} \frac{\alpha^i_{\mu_i N(x_i^{(\tau)};\mu_i,\sigma^2 I)}}{\sum_{y \epsilon \varphi_{x_i^c}} \alpha_y^i N(x_i^{(\tau)};y,\sigma^2 I)} \mu_i \qquad \textbf{(2.12)}$$

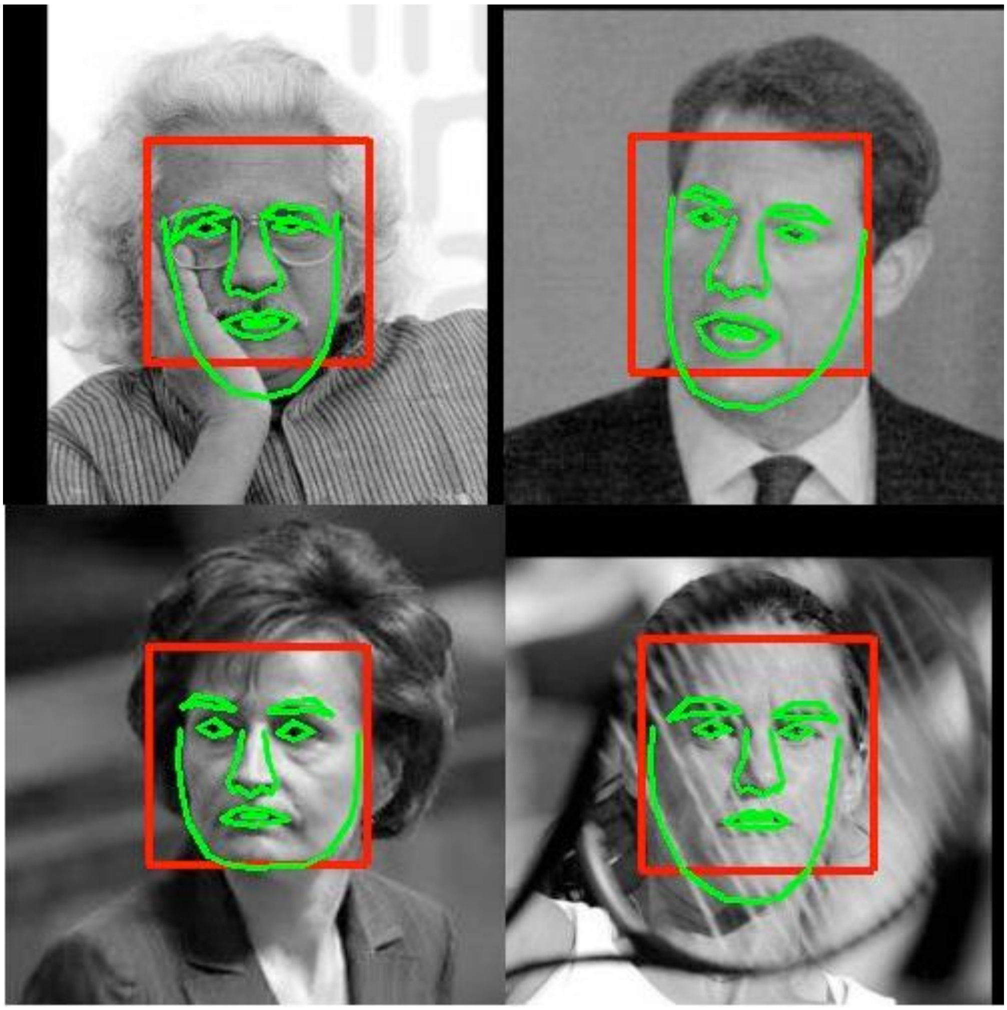
ii. Apply subspace constraint

$$\Delta p = J^{\dagger}[x_1^{(\tau+1)} - x_1^c; \dots; x_n^{(\tau+1)} - x_n^c] \qquad \textbf{(2.13)}$$

f. Update parameters: $p \leftarrow p + \Delta p$

B. return p

Figure 3.9: Example fitting results FaceTracker on the Faces



Source: (Saragih, et al., 2009)

3.3 Experimental Comparison of Face Tracking Algorithms

We used the CK+ database (Cohn, et al., 2010) to compare the three face tracking methods mentioned above briefly and will be mentioned below as Stasm, AsmLibrary and FaceTracker. The Cohn-Kanade (CK) database contains seven emotions (6 basic emotions + neutral) face mimics' acting. It is one of the most used databases since it was published. This database has been made more comprehensive, by increasing the number of people and by including new information like emotion tags, face tracking data and then opened to the use of researchers in 2010 (CK+) . CK+ database's face tracking information is generated manually in some frames and with AAM method in others, and it consists of 68 landmarks. We accept the position of these landmarks as ground truth reference and compare this reference with other 3 methods' landmark positions.

Since every algorithm tracks different number of landmarks at different positions, landmarks that are common to all the algorithms have been chosen for performance comparison. There are a total of 14 common landmarks, which are listed in Table 3.1. In Figure 3.12 you can see the common landmark points on a face image.

On Table 3.2, the arithmetic average of Euclidean distance between ground truth points and the landmark point we estimated with different libraries. In Table 3.3, all of CK+ database videos' results are shown. Face Tracker gives us the minimum average error.

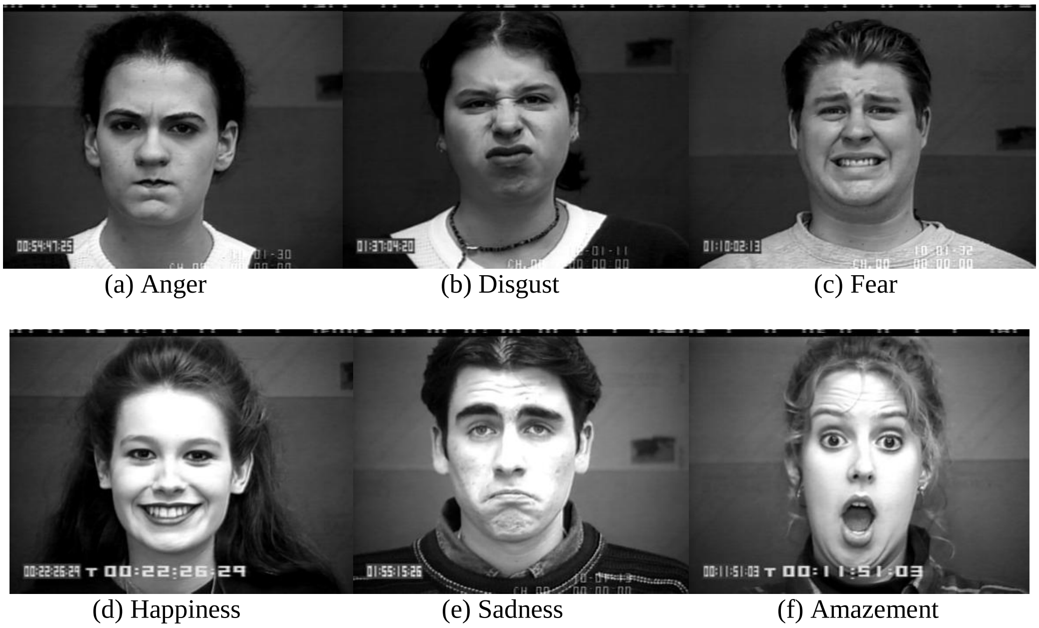
In this section, we show some examples on the CK+ database, where the compared algorithms are successful and not successful. In Figure 3.14, some frames are shown, where the STASM library makes some errors. Even if it places the chin part properly, it can't find the boundary of the mouth correctly. When the eyes are fully closed, it estimates them as eyes half open or full open. Although in our tests we have obtained the landmark points close to where we expected, in consecutive frames their places change. In Figure 3.15, we see some examples or errors for the AsmLibrary. Even though this algorithm places more stable landmarks than Stasm, we can assume that this is because it does not deviate from the main model too much. As seen in Figure 3.15, it finds many landmarks incorrectly. In Figure 3.16, we give some examples of

FaceTracker’s errors. This algorithm is more efficient than the others and it can estimate the landmark points better. This algorithm can capture the dynamics of the face better.
In order to compare the tracking accuracies of the compared algorithms, we used ground truth landmark positions provided by the CK+ dataset.

In Table 3.2 and Table 3.3, we compare the average tracking errors based on the Euclidean distance. We can see that the performance of FaceTracker is comparable to Stasm and better than the ASMLibrary.

In the light of our experiments, we can see that Stasm Library and FaceTracker Library are more incisive at finding face landmarks than the other libraries. While these two libraries are competing, it is clear that FaceTracker Library can track the face better at different angles and when it is occluded by an object (Figure 3.9). Our purpose is to find faces in challenging environments with best results and track it afterwards.

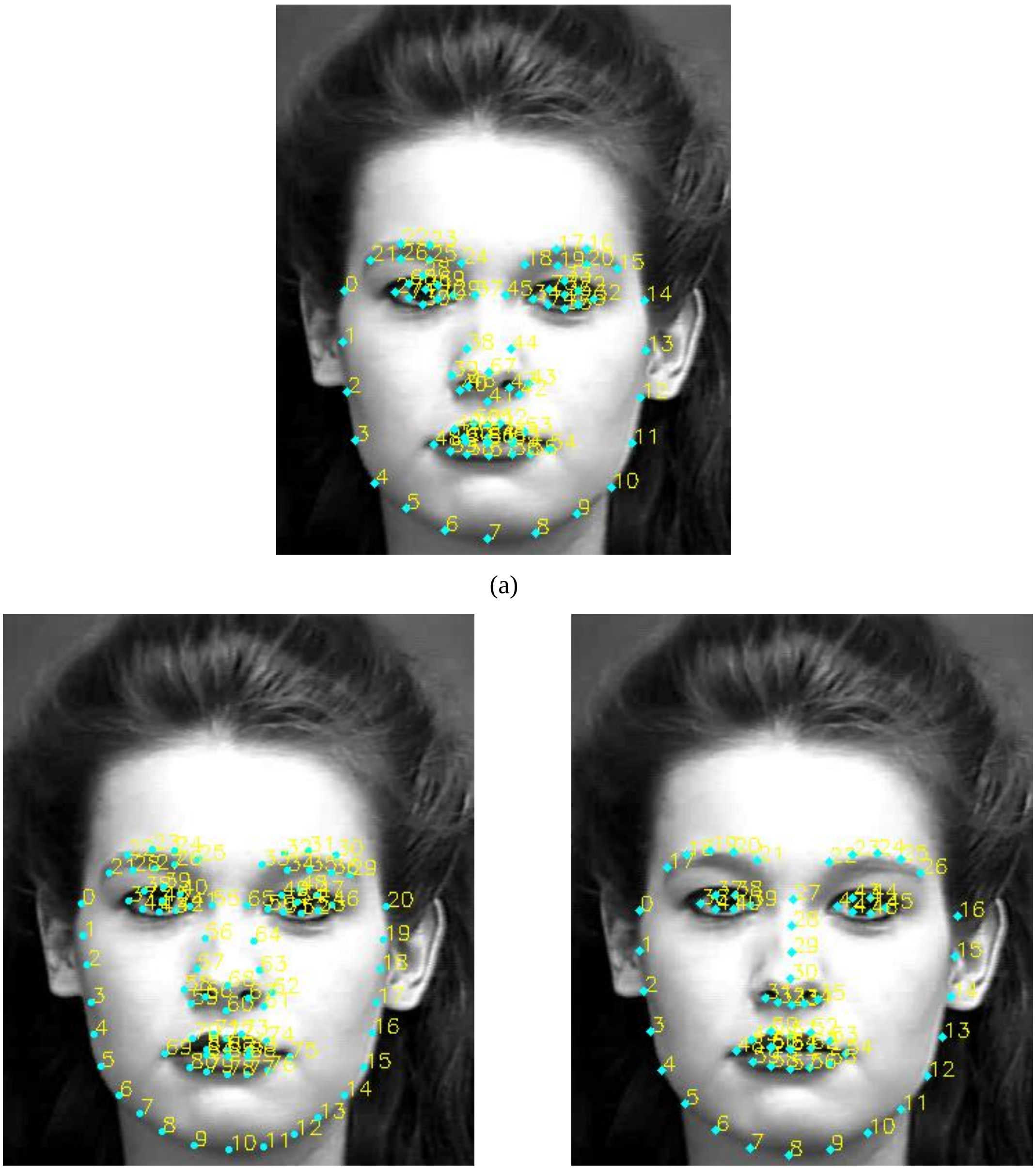
Figure 3.10: Examples from CK+ Database of face expressions that belong to 6 basic emotions.



(a) Anger (b) Disgust (c) Fear

(d) Happiness (e) Sadness (f) Amazement

Figure 3.11: (a) 66 points found by Stasm, (b) 87 points found by AsmLibrary, (c) 66 points found by FaceTracker



(a)

(b)

(c)

Table 3.1: Common Landmark points of three algorithms.

Landmark Points	CK+	Stasm	AsmLibrary	FaceTracker
Temples	0, 16	0, 14	0, 20	0, 16
Eyebrows(Right | Left)	17, 21 | 22, 26	21, 24 | 18, 15	21, 25 | 33, 29	17, 21 | 22, 26
Nose	31, 35	39, 43	58, 62	31, 35
Eyes(Right | Left)	36, 39 | 42, 45	27, 29 | 34, 32	37, 41 | 50, 46	36, 39 | 42, 45
Lips	48,54	48,54	69,75	48,54

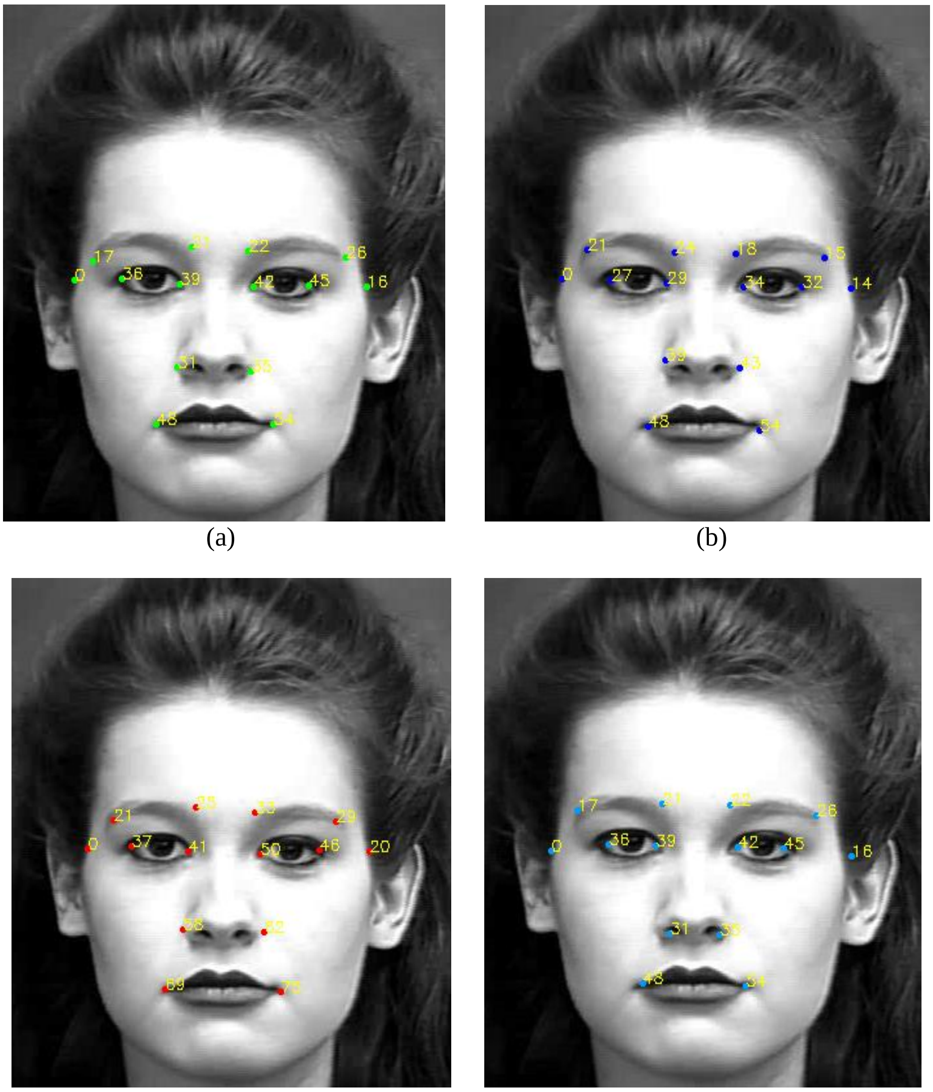
Table 3.2: Euclidian distances between common landmark points of three algorithms and CK+ database's 74. subject’s 5th video

	Stasm	AsmLibrary	FaceTracker
Temples	4.40	6.04	5.39
Eyebrows	7.53	8.75	7.44
Nose	5.97	8.08	11.59
Eyes	4.14	4.67	5.65
Lips	11.75	25.35	12.54
Average	6.75	10.57	8.52

Table 3.3: Average Euclidian distances between common landmark points of three algorithms and each of CK+ database's 10479 frame's grand truth landmark points.

	Stasm	AsmLibrary	FaceTracker
Temples	12.0316	13.8631	12.6895
Eyebrows	12.3361	12.781	10.3337
Nose	13.7367	12.0616	8.04713
Eyes	4.01898	5.86577	6.59906
Lips	6.2024	8.57294	8.99798
Average	9.66	10.62	9.33

Figure 3.12: (a) Common landmark points from CK+ database which were selected as grand truth. (b) points found by Stasm method.(c) points found by AsmLibrary method.(b) points found by FaceTracker method



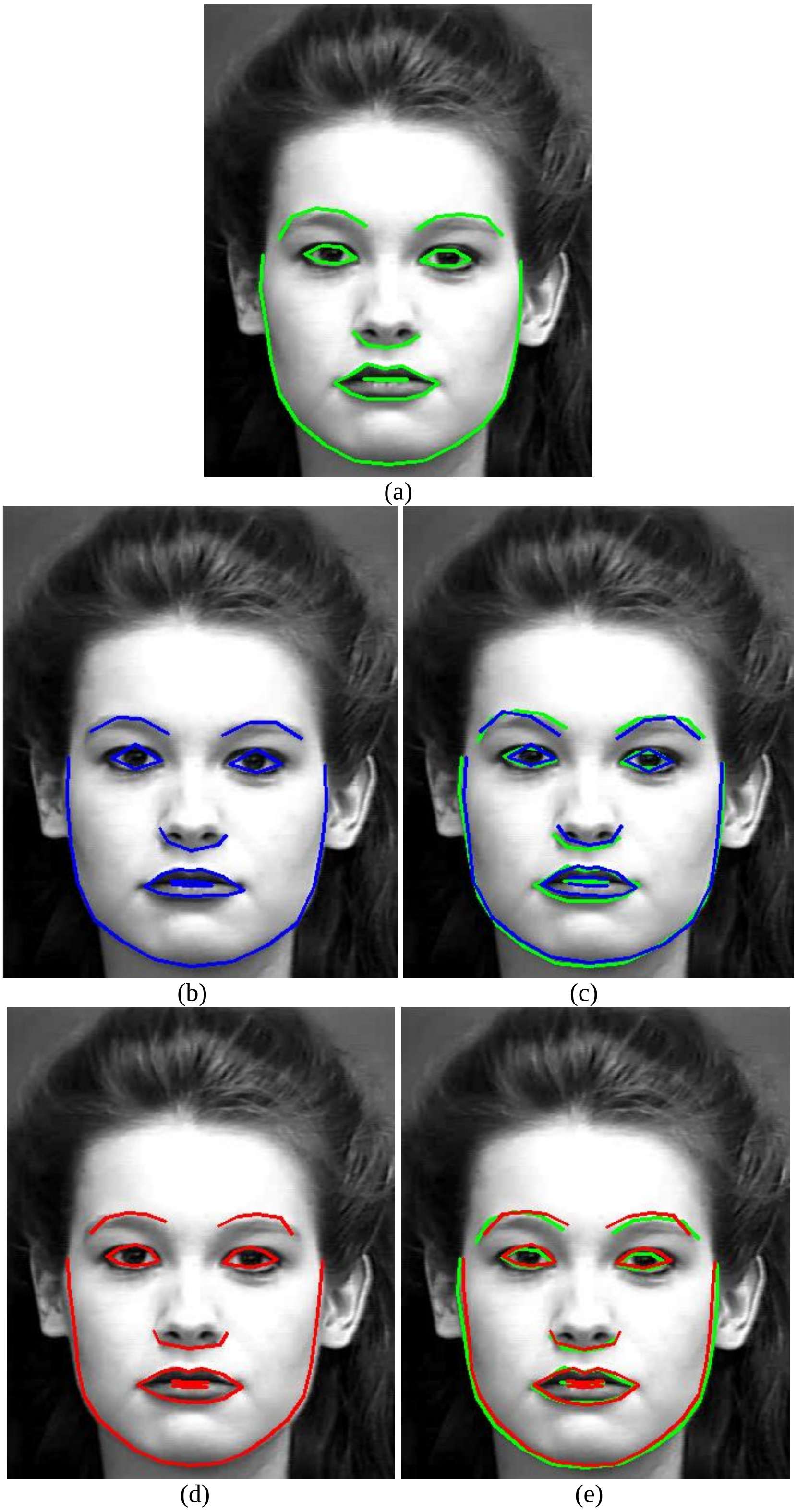
(a)

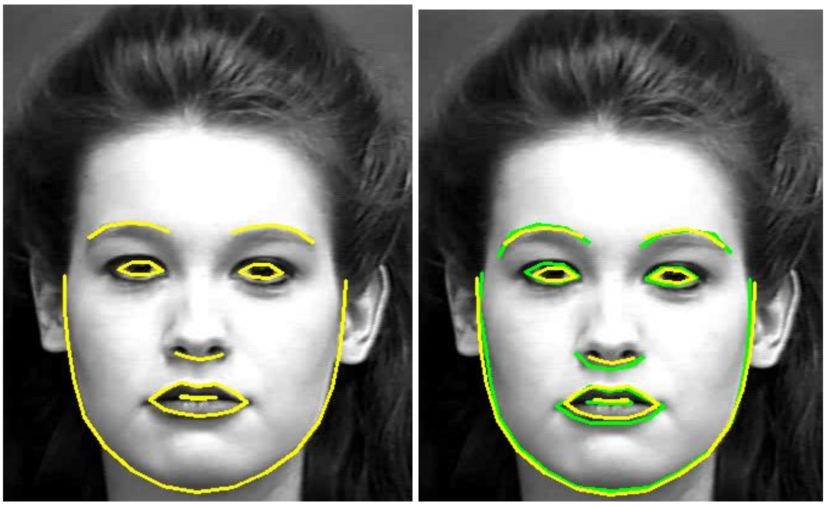
(b)

(c)

(d)

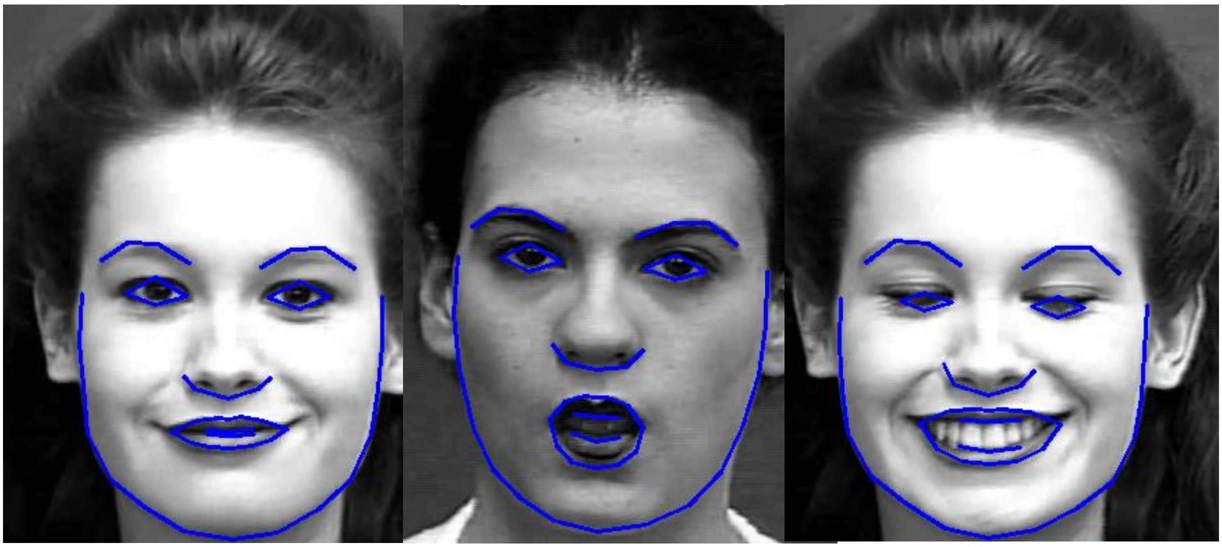
Figure 3.13: (a)CK+ database's 74th subject's 5th video's first frame drawn by ground truth points.(b)Stasm method (c) Comparison between CK+ vs Stasm (d)AsmLibrary method (e) Comparison between CK+ vs AsmLibrary (f)FaceTracker method (g) Comparison between CK+ vs FaceTracker





(f) (g)

Figure 3.14: Some mistakes of Stasm method (a) Stretched lips. (b) Mouth is bigger than expected. (c)Although the eyes are closed they are drawn as if they are open



(a) (b) (c)

Figure 3.15: Some mistakes of AsmLibrary method. (a) Lips and eyes are wrongly placed (b) Mouth is open but lips are placed with its standard version (c) Eyes are closed but drawn as open and mouth is totally wrong.

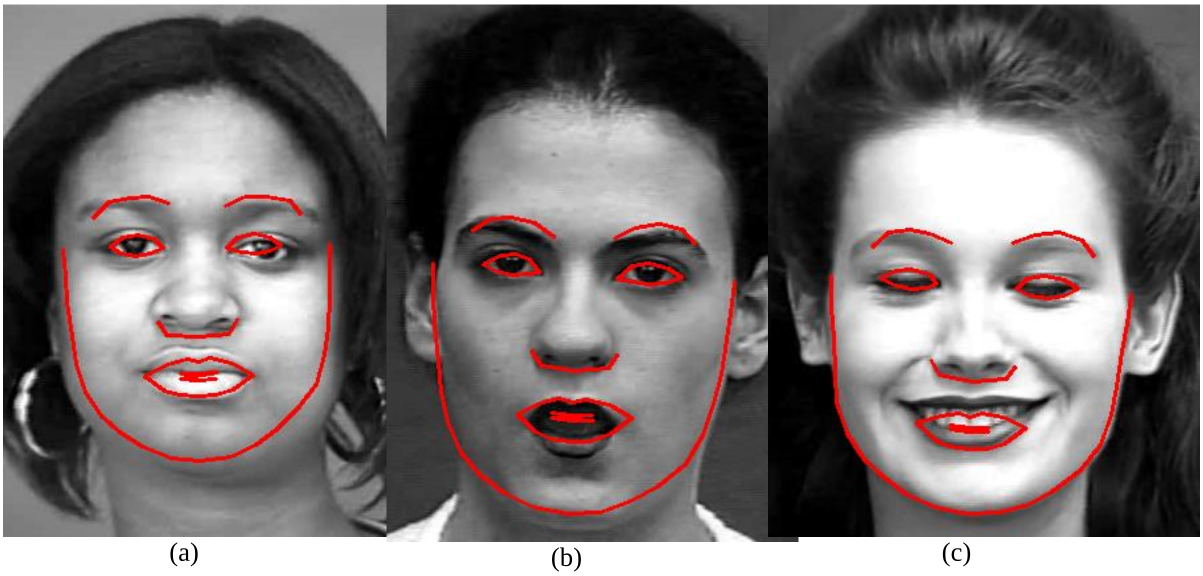
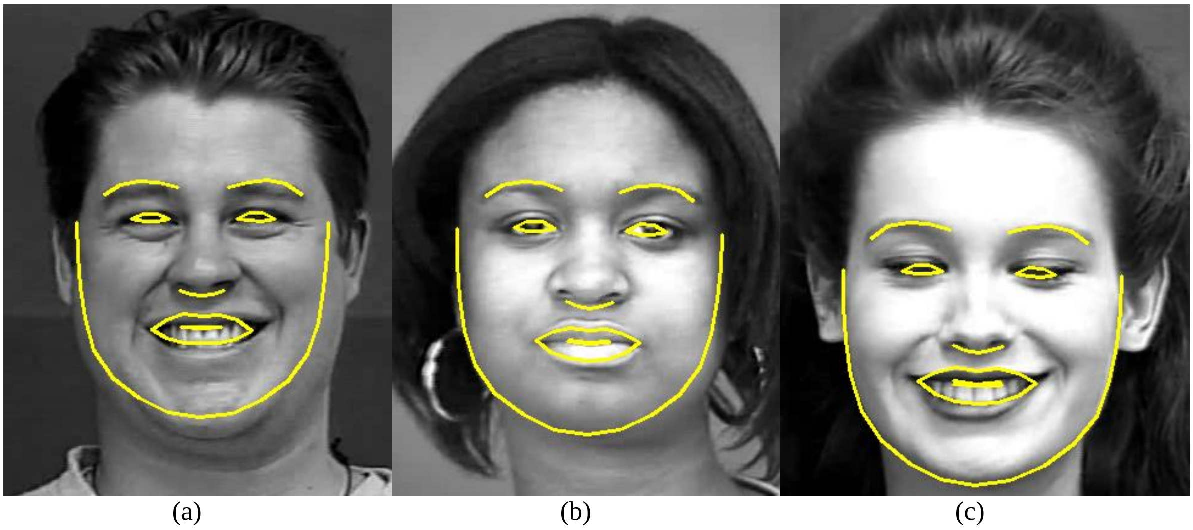


Figure 3.16: Some mistakes of FaceTracker method. (a) Lips and chin are wrongly placed (b) Eyes are not correct and same (c) Eyes are closed but drawn as open, mouth is drawn as initial and eyebrows are always symmetric.

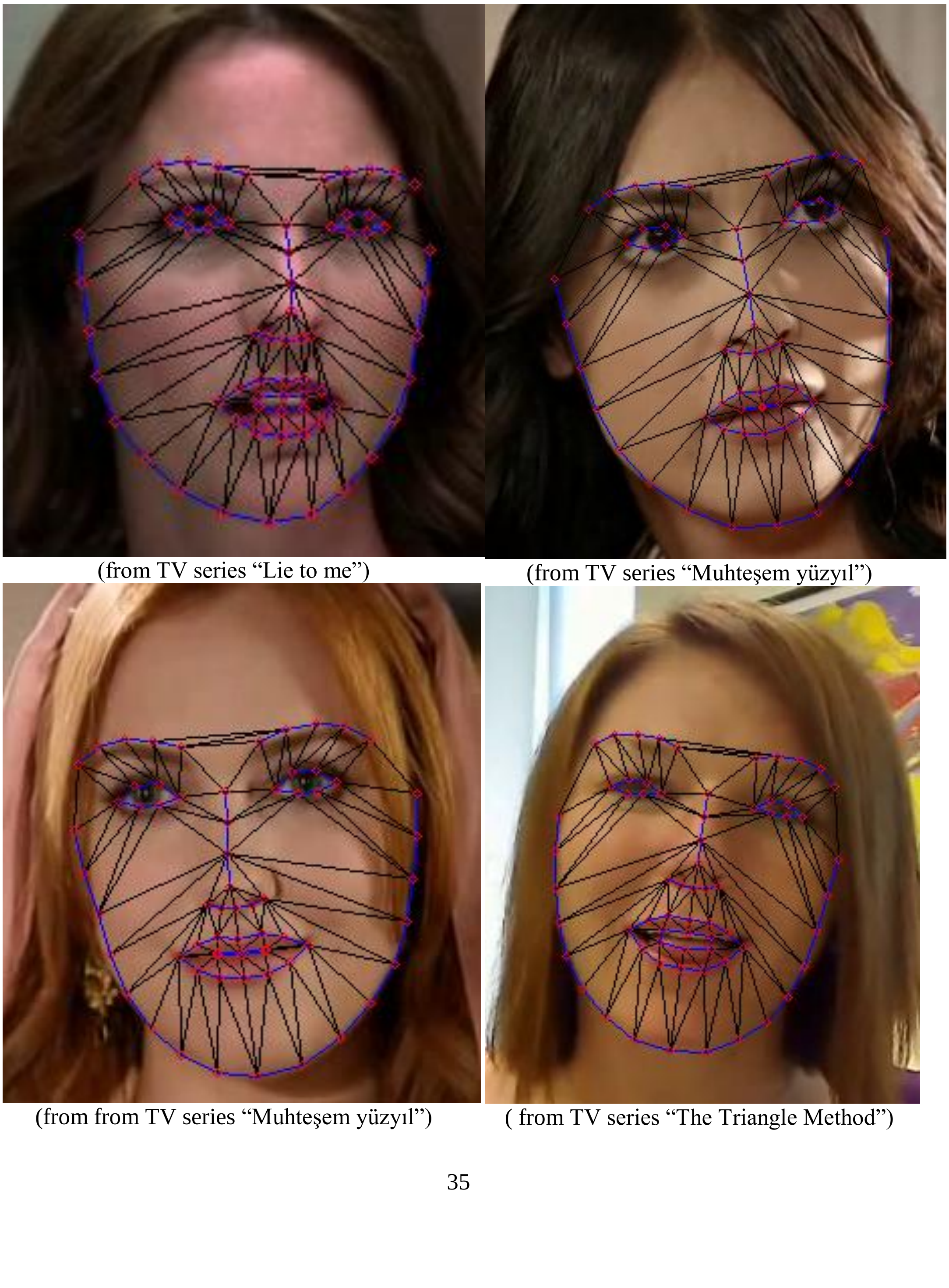


4. PROPOSED FACE DETECTION AND TRACKING METHOD

4.1 Problems of FaceTracker

The FaceTracker algorithm (Saragih, et al., 2009) automatically detects the face and tracks 66 landmarks on the face throughout the video. Although FaceTracker gives successful tracking results on frames containing some occlusion (Figure 4.1), it may give incorrect results as well (see Figure 4.2).

Figure 4.1: Successful examples of FaceTracker on occluded faces.



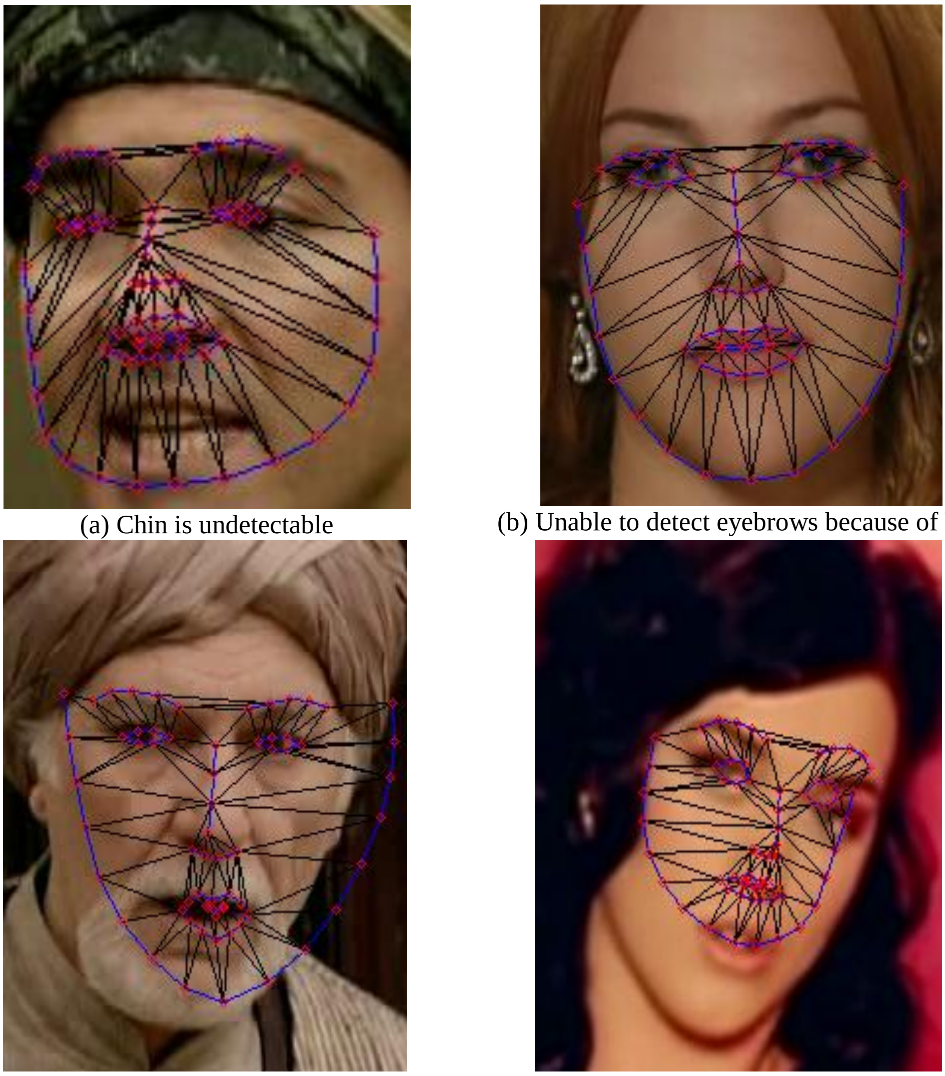
(from TV series "Lie to me")

(from TV series "Muhteşem yüzyıl")

(from from TV series "Muhteşem yüzyıl")

(from TV series "The Triangle Method")

Figure 4.2: Examples of unexpected FaceTracker results. Different types of faces, large angles and low resolution images can cause these errors.



(a) Chin is undetectable

(b) Unable to detect eyebrows because of hair

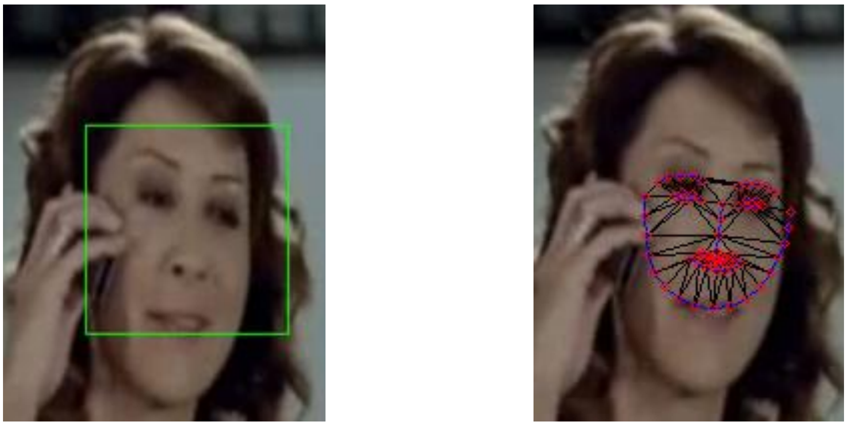
(c) Beard confuses the FaceTracker algorithm

(d) Low resolution image

During our experiments we observed that without proper modifications, we could not always get the correct tracking results we wanted to achieve. We were able to spot three causes of the problem:

1) FaceTracker uses the Viola-Jones face detection algorithm to detect the face in the first place. The face detector returns a rectangle, which includes the eyes and mouth, but excludes the chin (see Figure 4.3(a)). Exclusion of the chin causes some alignment problems when the 66 facial landmarks are detected (see Figure 4.3(b)).

Figure 4.3: (a) Viola-Jones Face Detector implemented in OpenCV gives us a rectangle around the minimal face for the given image; (b) Original Face Tracker algorithm cannot find the correct facial landmark points.

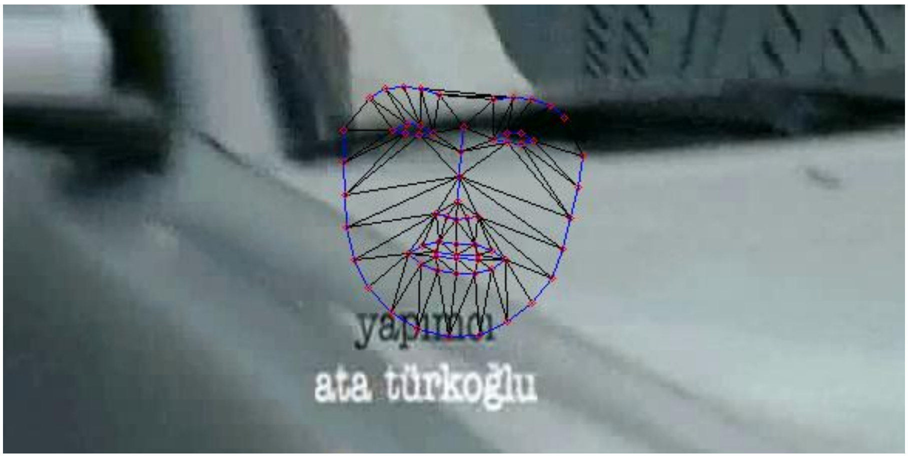


(a) Detected face

(b) Detected landmarks are not correct.

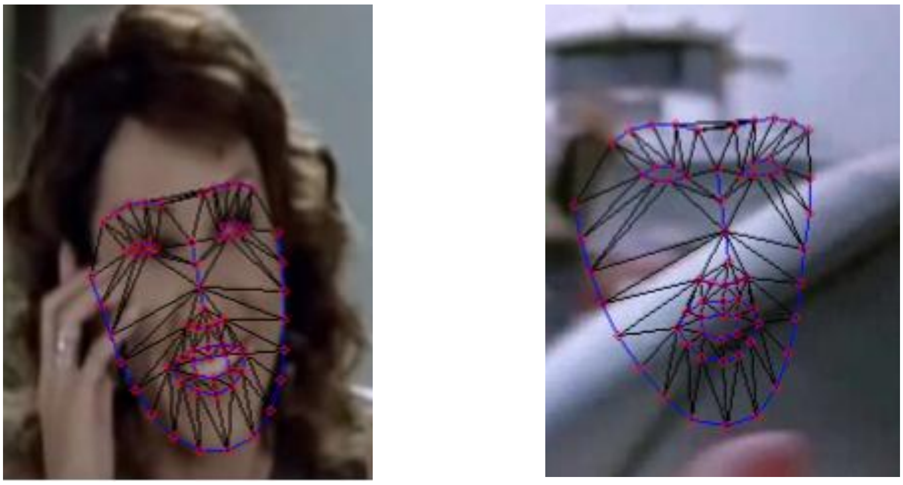
2) Another problem that we observed while using FaceTracker is as follows: All the videos we used during the experiments contain faces but also other objects. Since FaceTracker uses the Viola-Jones face detector to detect faces in a frame of a video, the FaceTracker is also affected from the errors of the face detection step. For example, the VJ face detector can detect regions that look like a face. (see Figure 4.4, where the dark stripe is seen as the eye region). Since our goal is to extract facial video clips, such errors detect and track non-face regions producing many irrelevant video clips which include no faces.

Figure 4.4 Default face tracker finds faces in unwanted places because of the errors initiated by the Viola-Jones face detector.



3) The third problem that we faced when using FaceTracker is that, FaceTracker cannot detect scene cuts successfully and continues to track the face across scene cuts. An example is shown in Figure 4.5.

Figure 4.5: FaceTracker continues to track the face across a scene cut. Two successive frames are shown below.



Since our goal is to use FaceTracker for extraction of facial video clips from movies, we improved the algorithm in several ways.

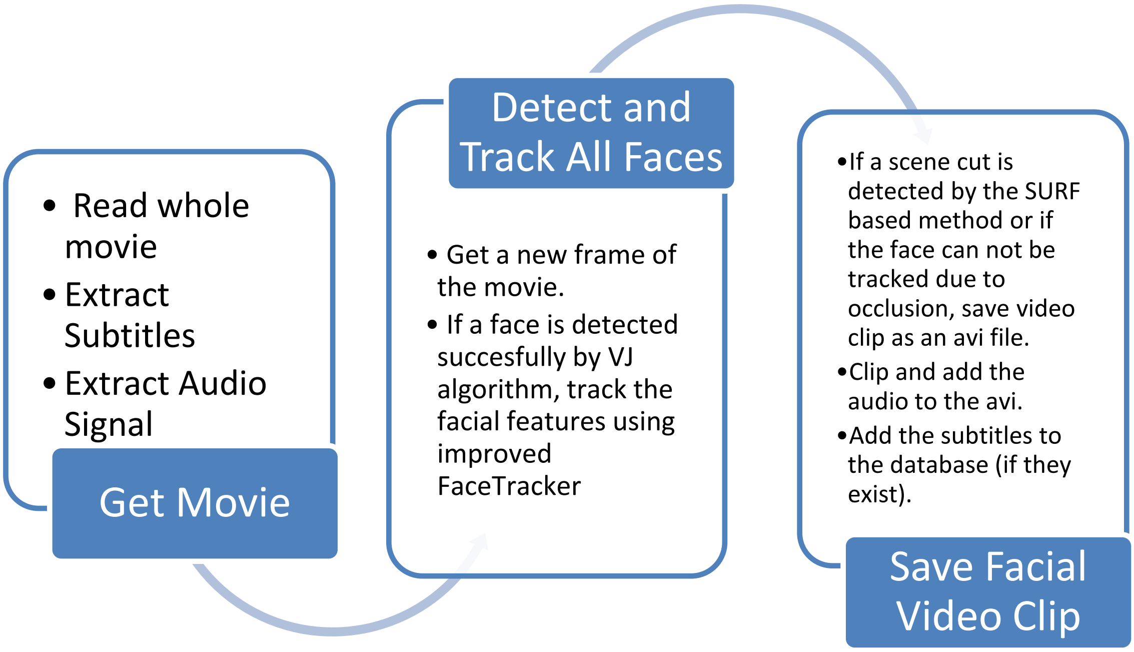
4.2 Optimizations and Improvements over FaceTracker Algorithm

Our goal is to use the FaceTracker algorithm for automatic extraction of facial clips from a movie or a film.

We can divide our problem to four phases:

- Face Detection (start of a facial clip)
- Face Landmarking
- Tracking of Facial Landmarks until face is occluded (cannot be tracked any more) or a scene change occurs (end of a facial clip).
- The flowchart of our algorithm is seen in Figure 4.6.

Figure 4.6: Flow chart of our implementation.

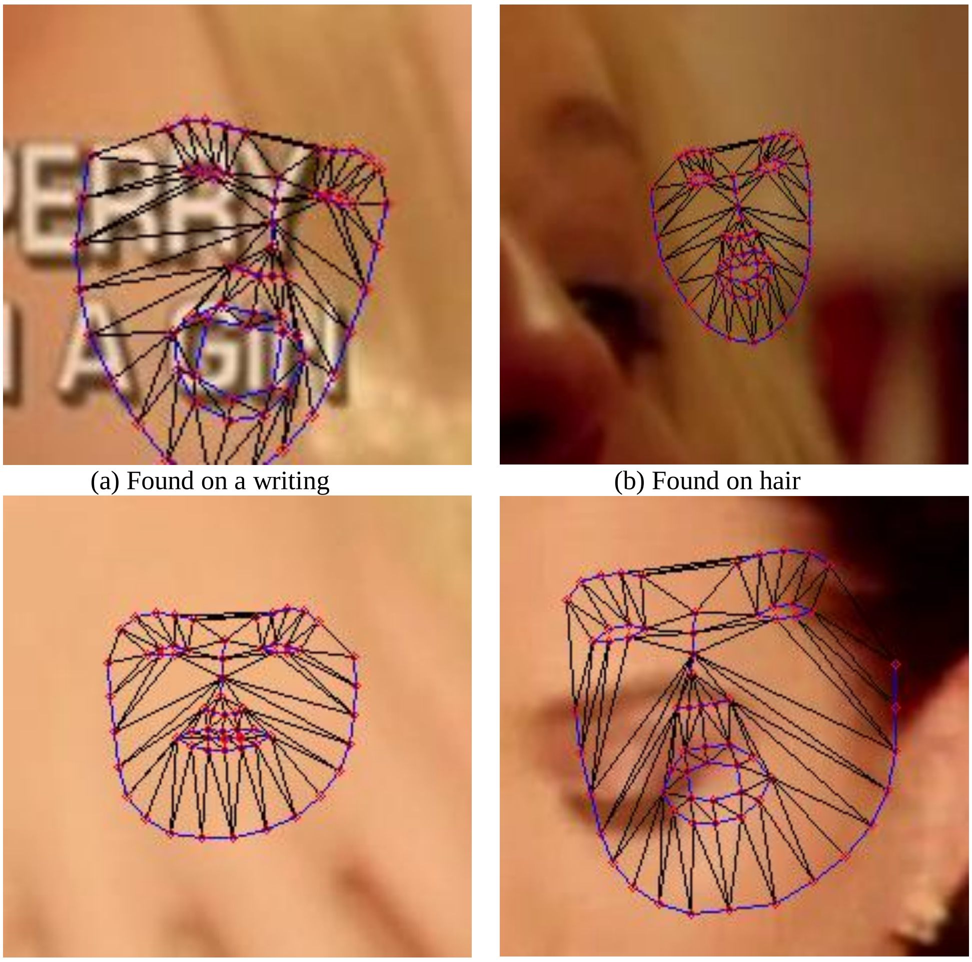


Get Movie
- Read whole movie
- Extract Subtitles
- Extract Audio Signal
Detect and Track All Faces
- Get a new frame of the movie.
- If a face is detected succesfully by VJ algorithm, track the facial features using improved FaceTracker
Save Facial Video Clip
- If a scene cut is detected by the SURF based method or if the face can not be tracked due to occlusion, save video clip as an avi file.
- Clip and add the audio to the avi.
- Add the subtitles to the database (if they exist).

4.2.1 Improvements for Face Detection

Basic implementation is the Viola–Jones face detection framework, which was implemented in OpenCV (opencv, 2012) library. This method is a good selection for our robust face detection part. However, the initial setup used with the code provided by FaceTracker is a more relaxed and error prone version of face detection (see Figure 4.7). First of all, we used a stricter version of this function by using different values for some parameters. For example, we changed the scale factor used for search windows between the subsequent scans from 1.1 to 1.16 and minimum neighbors value from 2 to 17 which helps us to eliminate face like figures. For our minimum width and length values we choose 50. This helps us to detect real face images instead of non-face objects. However, some difficulties are inevitable in low resolution images (see Figure 4.8). Low resolution faces are hard to track and find landmark points.

Figure 4.7: In our experiments we saw that different cvHaarDetectObjects parameters cause problems about face finding.



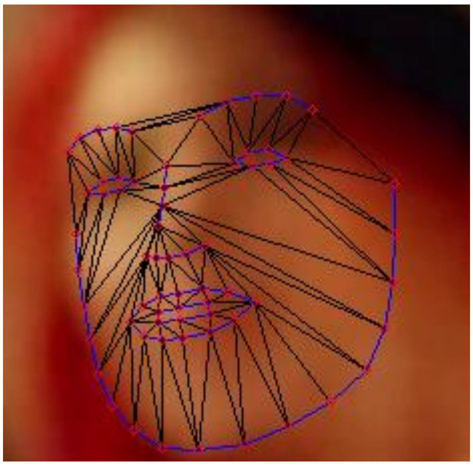
(a) Found on a writing

(b) Found on hair

(c) Found on a hand

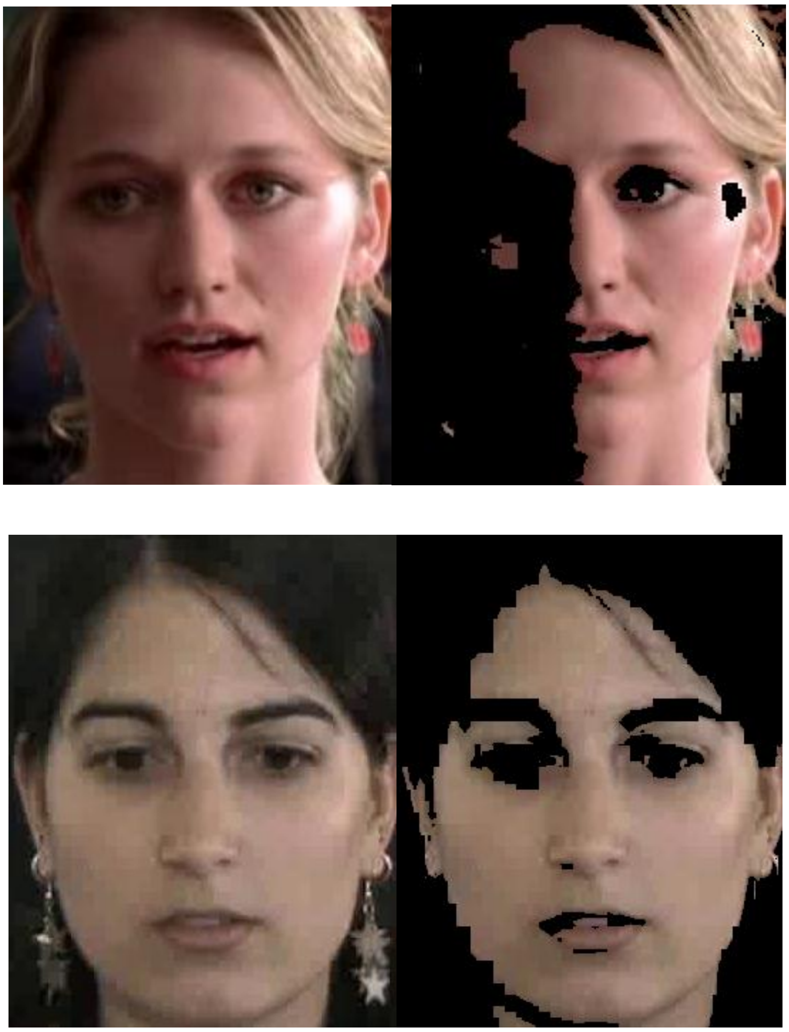
(d) Found on a part of a face

Figure 4.8: While eyes and eyebrows are located correctly, nose and mouth are landmarked incorrectly because of the low resolution.



Viola Jones face detector tries to detect the faces in gray-scale images, therefore it can falsely detect face-like objects such as a face drawn on a paper or a football, where the dark spots can be mistaken for eyes. In order to solve this problem, we used an explicit skin color filter to increase probability of finding real human faces.

Figure 4.9: Skin filter method we used is a fast but error prone for several light conditions or objects with skin color, like hair.



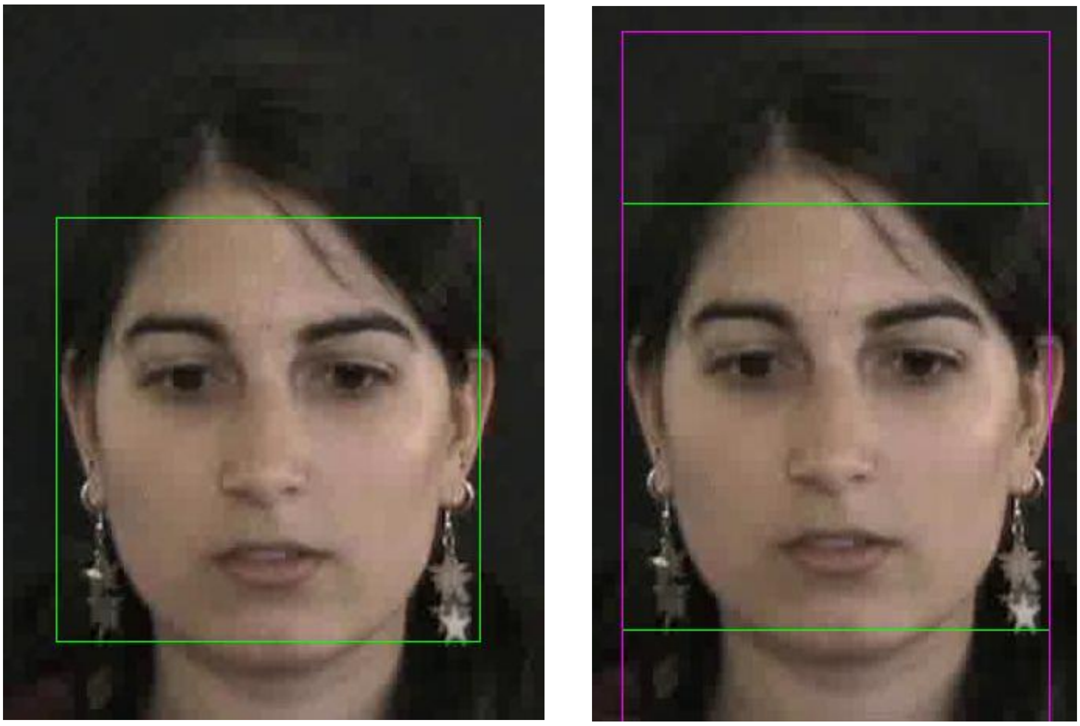
Our method for skin color detection basically searches every pixel and compares it with the skin color range. If the color is acceptable, then we flag this pixel as a skin color pixel otherwise we flag it as a non-skin color pixel. We accept the face frame if the ratio of skin-pixels to all-pixels is equal or greater than a threshold as follows:

$$\frac{\sum_{i=0}^{I_n} C(I_{i_{RGB}})}{\sum_{i=0}^{I_n} 1} \geq r \tag{3.1}$$

where C means color skin check function, I means image and r means the predefined ratio.

The VJ face detection method we are using provides us a minimalistic rectangle for the detected face (see Figure 4.10 (a)). FaceTracker multiplies this rectangle with a scale parameter to enlarge it. This window is still very small for our skin color check; causing the ratio of skin-pixel/all-pixel to decrease. To add more skin colors for the face, we also double the length of this window (more for the forehead and chin (1.8 times), less for the cheeks (1.1 times). Making that enlargement operation is very important, without it we lose lots of low resolution face images because we need to make sure we have enough skin color in our frame.

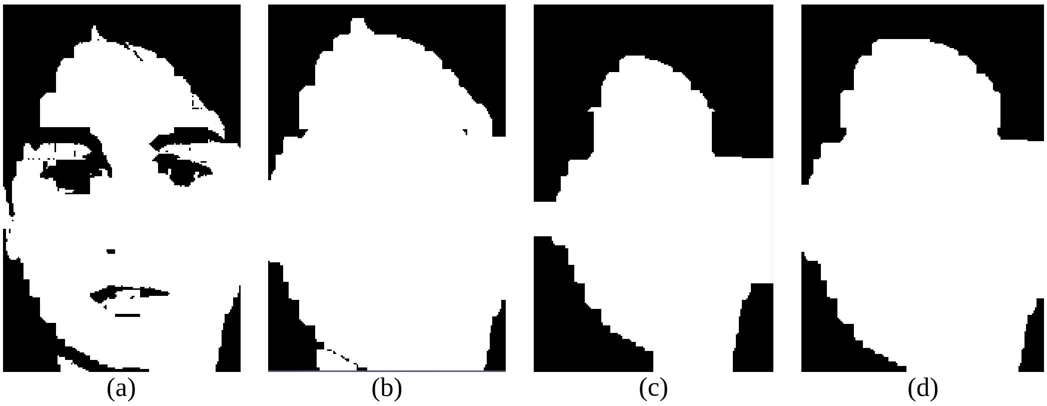
Figure 4.10: Rectangles provided by (a) OpenCV and (b) the enlarged one which we create each time.



(a) (b)

FaceTracker algorithm needs an initial face position to locate facial landmark points. However, in order to locate the facial landmarks correctly, it needs the minimal face window that contains the whole face image. Otherwise FaceTracker algorithm tries to find a face with size proportional to the given window. If FaceTracker finds a bigger face than the normal face image, the face it finds can attach to a moving object and can move with that object.

Figure 4.11: To acquire minimum bounding rectangle for face we dilated and eroded by different amounts. (a) Initial skin color filtered binary image (b) dilate by 3 (c) erode by 13 (d) dilate by 11



The face window we obtained in Figure 4.10 (b) is bigger than the face window we really wanted to achieve because we need bigger area to work on. To solve this problem we tried to estimate the real face region by using skin color filter. Our current skin color filter gives us a mask of skin color parts. By dilation and erosion we tried to eliminate small skin colored objects and bring out face's silhouette. We first dilate with a small amount to fill eyes and mouth, which are in a non-skin colored part of the image. Then we erode and dilate with larger amounts to finally get the biggest skin color blob in the image which is the face we are looking for (Figure 4.11(d)). From that point on we measure this white blob (Figure 4.12) for our final face window (Figure 4.13). We follow white pixels to the right from to origin and double that amount to find width of the face. Also we follow white pixels to the top from to origin and double that amount to find height of the face. We cannot follow to the bottom because neck is also skin colored.

Figure 4.12 Measurement of Face

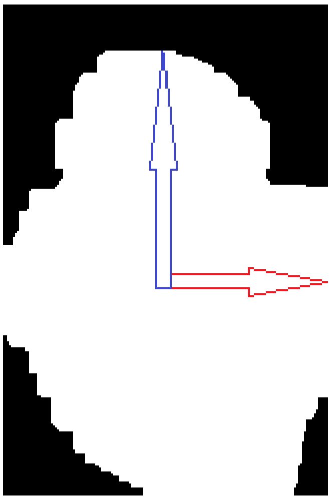
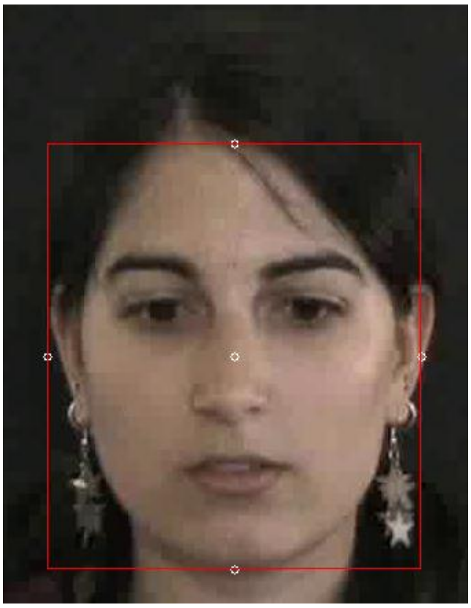


Figure 4.13: Optimum face rectangle based on our skin detector.



Only downside of this method is; if there are lots of skin colored objects around the face, our algorithm finds a bigger (which cannot be bigger than the window we are searching in) face window. Examples of this problem generally occur if the person is using his/her hand so frequently around the face or the hair is also has some skin colored parts. Our algorithm assumes most of the people use right hand, that's why in Figure 4.12 we used right side of the face to measure face. Usage of right hand like talking with a mobile phone with right hand does not affect our database; but left handed people or a vertically mirrored video may cause our algorithm to not work.

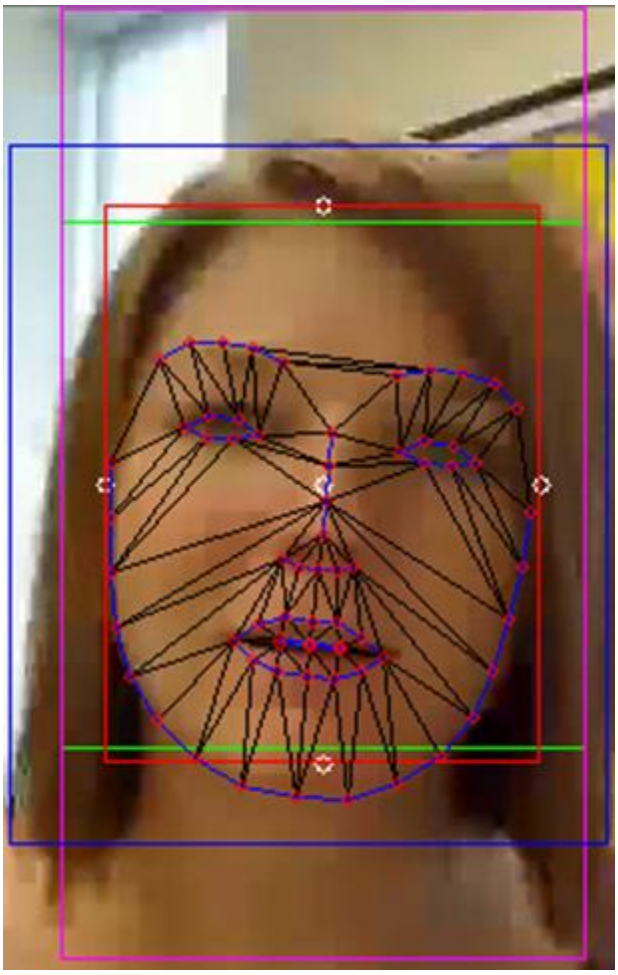
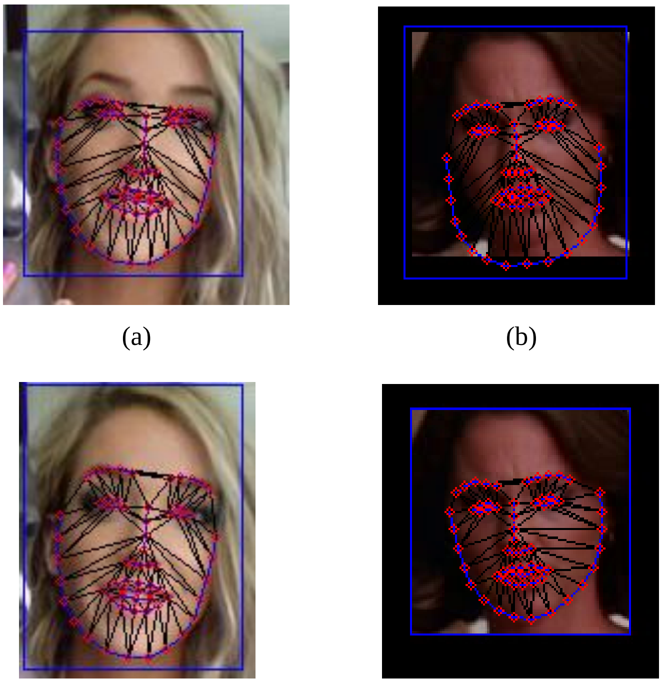


Figure 4.14: The green rectangle shows the face detection result of the Viola-Jones algorithm. Since it does not include the chin, it will problematic for the FaceTracker. The green window is extended. We enlarge it to the pink rectangle with configurable parameters. After we obtained pink rectangle we reduce it to red rectangle using skin color information. We do these operations to avoid FaceTracker's errors such as not finding eyebrows and chin.

Figure 4.15 (a),(b) If we use default rectangle eyebrows and chin may not be detectable. (c),(d) When we measure the correct face width and height we observed better results



(a) (b)

(c) (d)

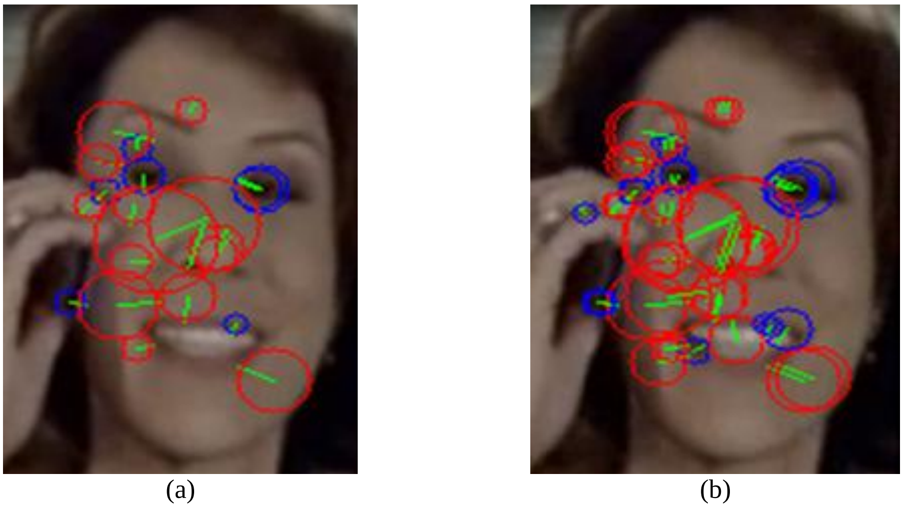
4.2.2 Improvements for Tracking Failure Detection Based on SURF based Scene-Cut Estimation

The image and the face detection window are provided to FaceTracker, which in turn gives us the positions of sixty six landmark points as an output. We record these points in a file for future use. This algorithm's model adapts itself according to face position. While this is a good ability that can detect different face angles other than profile images it has a drawback of finding wrong landmark points if face angle changes too much.

FaceTracker's face tracking algorithm uses the template matching method of OpenCV (opencv, 2012). We search the template face image we obtained from the previous frame within the current frame. Template matching method is the first method that we use to find the place of a face inside the current frame. For every new frame, template matching method searches if the template image we found in the previous frame involves in the current frame except the first frame we found the face. As we mentioned before however, this approach is open to errors and not reliable (see Figure 4.5). Therefore, FaceTracker continues to track the face even if there is a scene cut.

In order to overcome this problem, we propose a scene cut detection method based on SURF feature matching. By using the number of matched SURF descriptors between two successive, we can decide whether there is a scene cut or not. Specifically, as shown in Figure 4.16, when the number of matching SURF features between two successive frames is less than given configurable ratio, we claim that there is scene cut.
The disadvantage of using shape based template method is that when the face bends too much or rotates too much in a short time, it loses the ability to follow.

Figure 4.16: (a) SURF features detected on frame 353 of the sequence where blue circles indicate dark blobs on light backgrounds and red circles indicate light blobs on dark backgrounds (b) The SURF features of frame 354 are shown together. The numbers of matched SURF descriptors determine whether the face can be correctly tracked or not.



For each frame we create new SURF descriptors if we found a face in previous frame. If the ratio of matched descriptors from the previous frame and the current frame is more than the value we given; we accept this face and add this frame's face information to our current face video file. You can see the scene cut process in Figure 4.17; and several examples from our face detection and tracking system showed in Figure 4.18.

Figure 4.17: Top row: Six consecutive frames from a movie are shown. FaceTracker continues to track the landmarks although there is a sudden scene-cut. Middle Row: The proposed SURF-features based method helps to stops face tracking at a scene cut. Bottom Row: Green circles denote matching SURF features between the current frame and the previous frame; red circles denote features that have not matched. There are no matching SURF features at the fourth frame from left, indicating a scene cut.

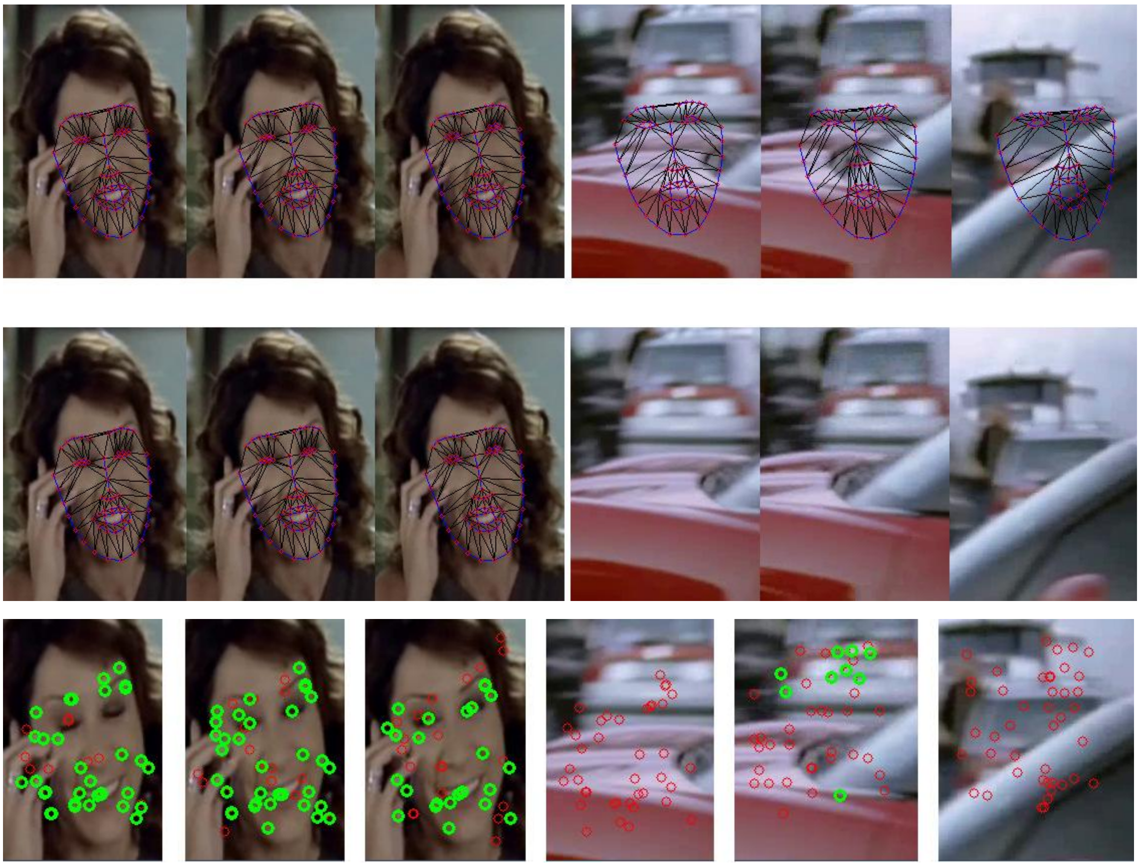
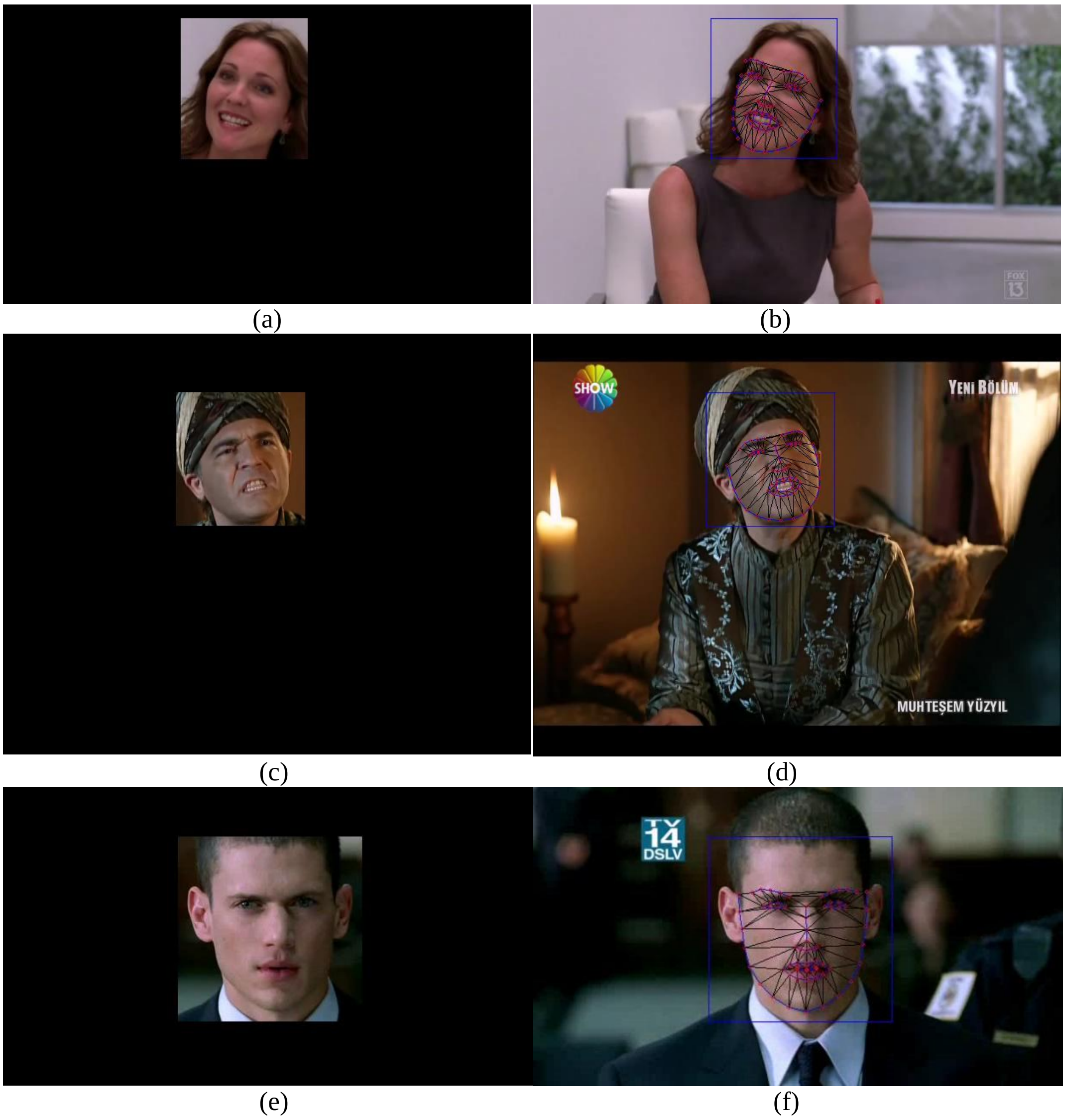


Figure 4.18: (a) An extracted happy face from “Lie to me” (b) tracking points and the rectangle we choosed to cut from this video (c) An extracted angry face from “Muhteşem Yüzyıl” (d) tracking points and the rectangle we choosed to cut from this video (e) An extracted neutral face from “Prison Break” (f) tracking points and the rectangle we choosed to cut from this video



(a) (b)

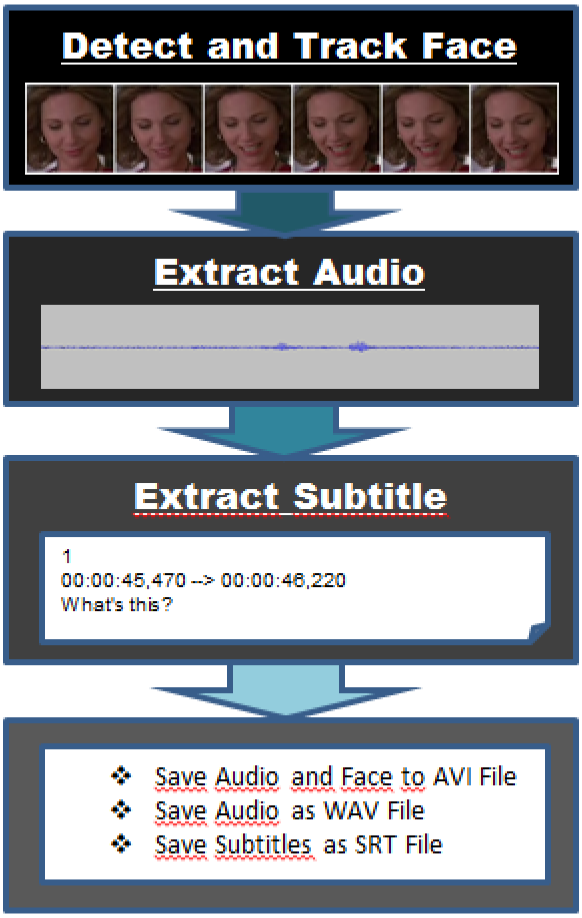
(c) (d)

(e) (f)

5. BAUM-2: AN AUDIO-VISUAL EMOTIONAL FACE DATABASE

In the previous chapters, we presented an automatic method for extraction of audio-visual facial clips from films, for the purpose of creating an audio-visual emotional facial database containing samples under diverse conditions. You can see the main flow of the proposed database generation algorithm in Figure 5.1.

Figure 5.1 Flow of the Automatic Database Creation Algorithm



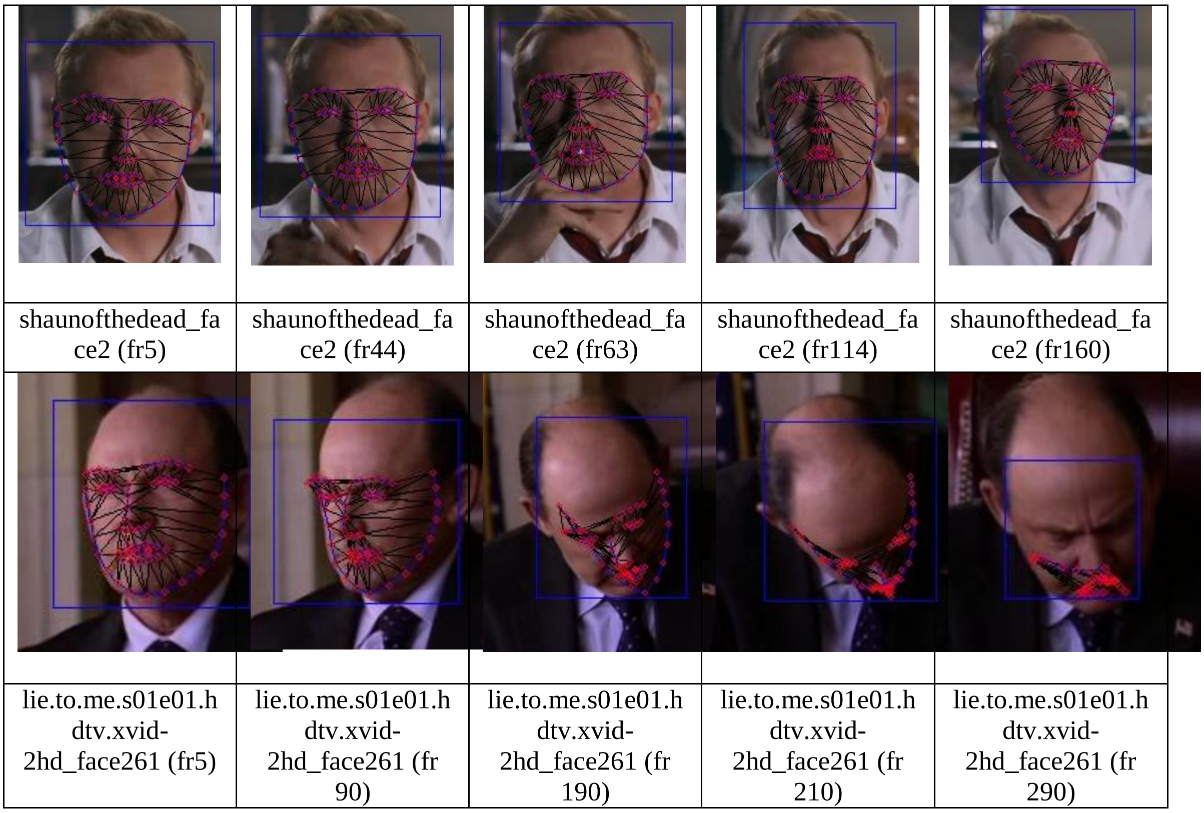
```
1
00:00:45,470 --> 00:00:46,220
What's this?
```

5.1 Automatic Processing of Movies

We tested the algorithm on 108 movies, some of which are in Turkish (e.g Devrim Arabaları, Muhteşem Yüzyıl etc.) and some are in English (e.g. Lie to Me, Shaun of the Dead, Prison Break etc.). You can see the complete list of all processed movies in Appendix B. We can extract hundreds of facial clips from a movie, but not all of them qualify to be included in an affective database. In some of the facial clips, facial landmarks cannot be tracked successfully (Figure 5.2), especially if there is too much head rotation. Some extracted clips start when a subject is smiling but end when the subject is surprised. If emotion changes within a clip, we cannot accept that as an example of a video containing a specific emotion. Such video clips can be further segmented manually, so that each segment reflects a single emotional state. We also eliminate some of the clips on which facial landmarks cannot be tracked successfully. Our approach is to choose a clip containing mostly frontal faces with a single emotion. (neutral, anger, contempt, disgust, fear, happiness, sadness, surprise).

Selected frames from several successful video clips are shown in Figure 5.3 and Figure 5.4. We have obtained a total of 21460 video clips from 108 movies and 700 of them have been manually labeled as an affective facial video clip. You can find additional examples of selected faces in Appendix A.

Figure 5.2 Undetectable Emotions



shaunofthedead_face2 (fr5)

shaunofthedead_face2 (fr44)

shaunofthedead_face2 (fr63)

shaunofthedead_face2 (fr114)

shaunofthedead_face2 (fr160)

lie.to.me.s01e01.hdtv.xvid-2hd_face261 (fr5)

lie.to.me.s01e01.hdtv.xvid-2hd_face261 (fr 90)

lie.to.me.s01e01.hdtv.xvid-2hd_face261 (fr 190)

lie.to.me.s01e01.hdtv.xvid-2hd_face261 (fr 210)

lie.to.me.s01e01.hdtv.xvid-2hd_face261 (fr 290)

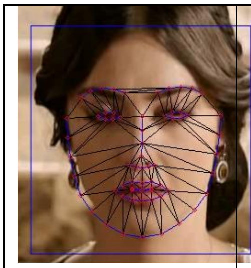
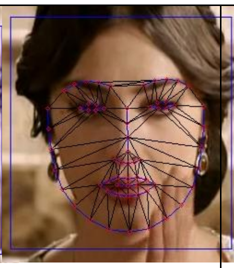
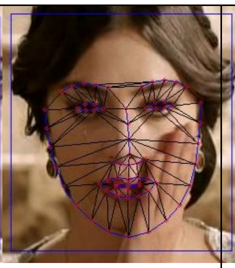
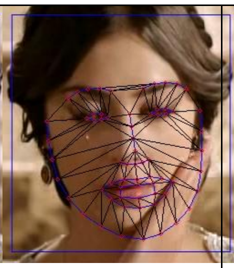
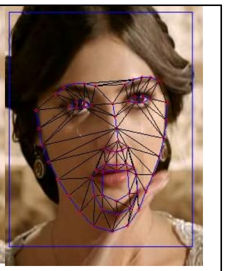
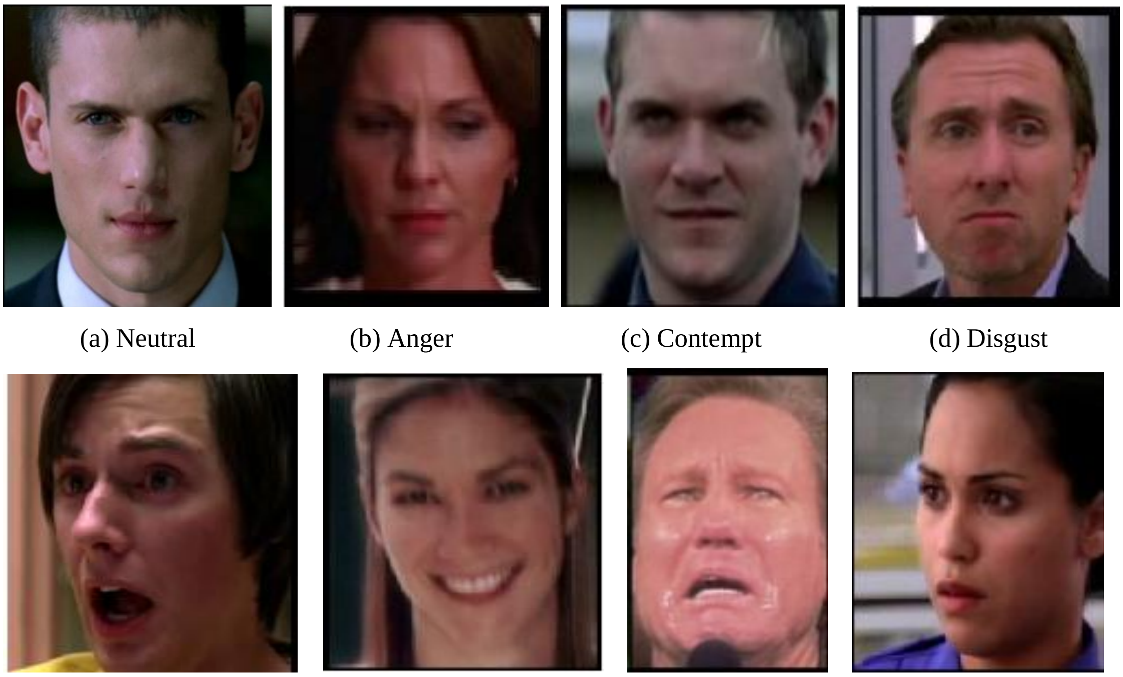
				
Muhtesem.Yuzyil. BL.32.DVBRip.X viD-OpeD_face1007 (fr5)	Muhtesem.Yuzyil. BL.32.DVBRip.X viD-OpeD_face1007 (16)	Muhtesem.Yuzyil. BL.32.DVBRip.X viD-OpeD_face1007 (20)	Muhtesem.Yuzyil. BL.32.DVBRip.X viD-OpeD_face1007 (25)	Muhtesem.Yuzyil. BL.32.DVBRip.X viD-OpeD_face1007 (30)

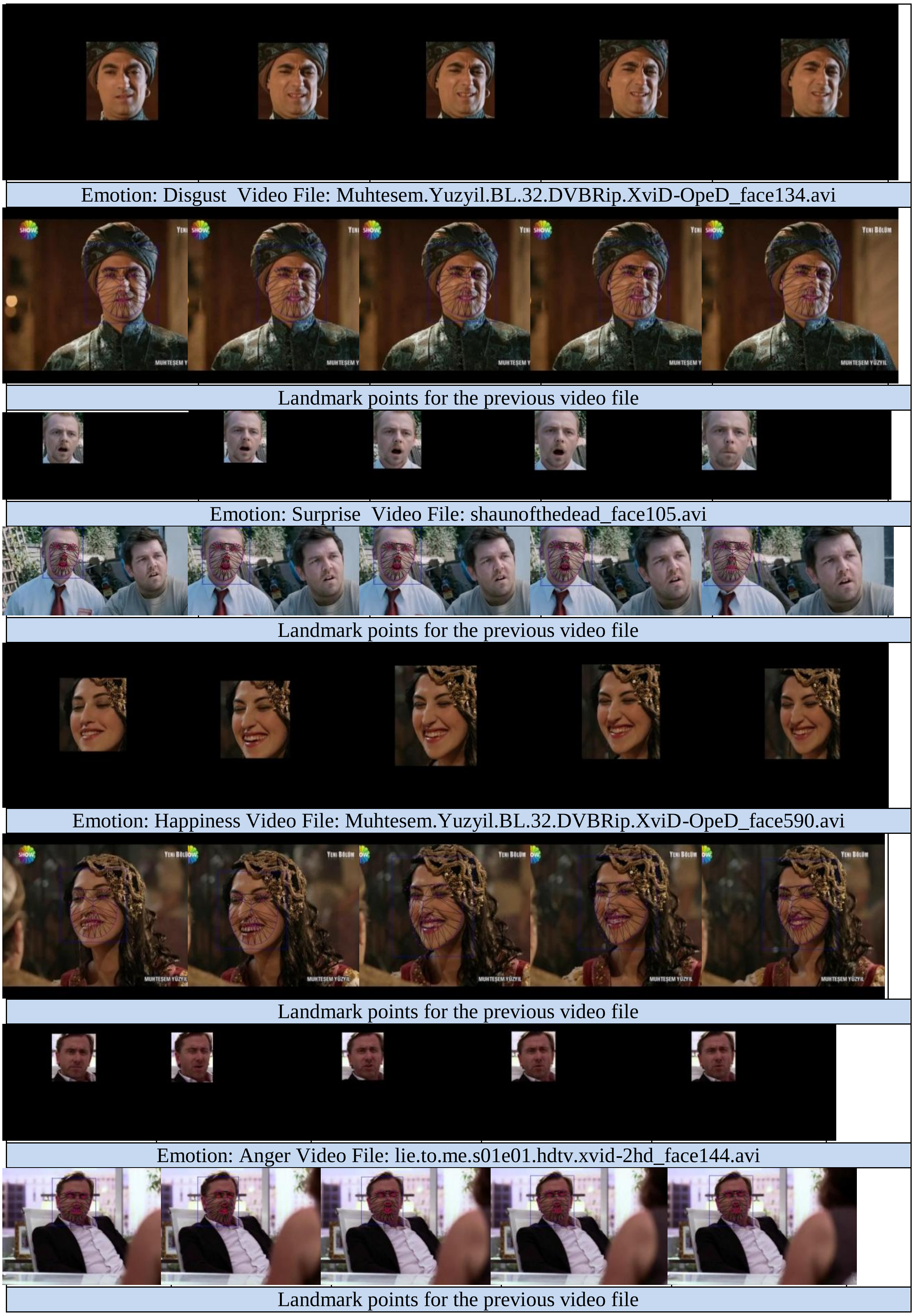
Figure 5.3: Some examples of the extracted video clips containing the eight emotions.



(a) Neutral (b) Anger (c) Contempt (d) Disgust

(e) Fear (f) Happiness (g) Sadness (h) Surprise

Figure 5.4 : Examples of Selected Faces



Emotion: Disgust Video File: Muhtesem.Yuzyil.BL.32.DVBRip.XviD-OpeD_face134.avi

Landmark points for the previous video file

Emotion: Surprise Video File: shaunofthedead_face105.avi

Landmark points for the previous video file

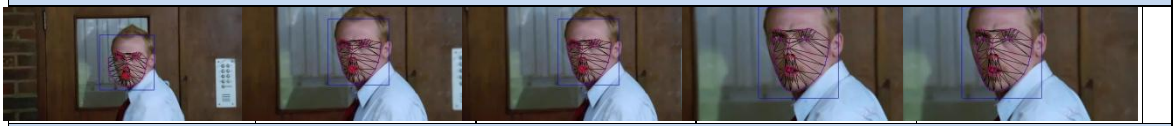
Emotion: Happiness Video File: Muhtesem.Yuzyil.BL.32.DVBRip.XviD-OpeD_face590.avi

Landmark points for the previous video file

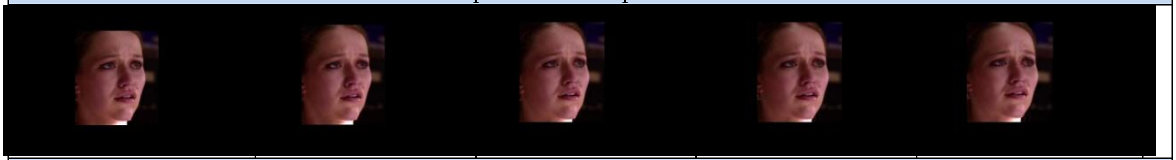
Emotion: Anger Video File: lie.to.me.s01e01.hdtv.xvid-2hd_face144.avi

Landmark points for the previous video file

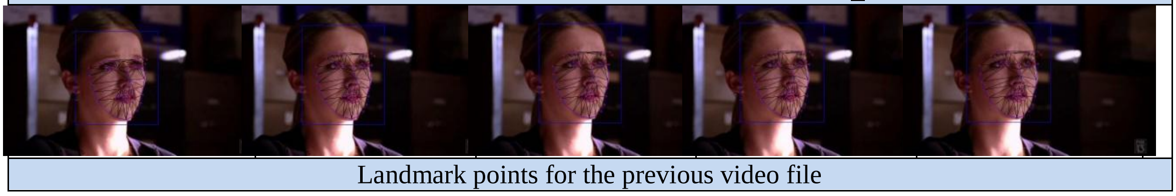
Emotion: Fear Video File:shaunofthedead_face240.avi



Landmark points for the previous video file



Emotion: Sadness Video File: lie.to.me.s01e01.hdtv.xvid-2hd_face280.avi

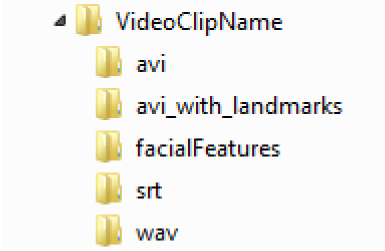


Landmark points for the previous video file

5.2 Folder Organization of the Database

Our database folder organization has a root folder named as the main avi file without file type postfix (“.avi"). Under this folder we have "wav" folder which we store extracted ".wav" sound files if main video has audio. "srt" folder consists ".srt" subtitle files if any subtitle provided. Under "avi" folder we have the facial clips merged with ".wav" files. "avi_with_landmarks" folder has the facial clips with landmark information. Users can easily observe if the FaceTracker algorithm worked correctly or not before processing the files under "avi" folder. Landmark information we obtained by using FaceTracker is under "facialFeatures" folder. You can see the folder structure in Figure 5.5.

Figure 5.5: Database folder structure

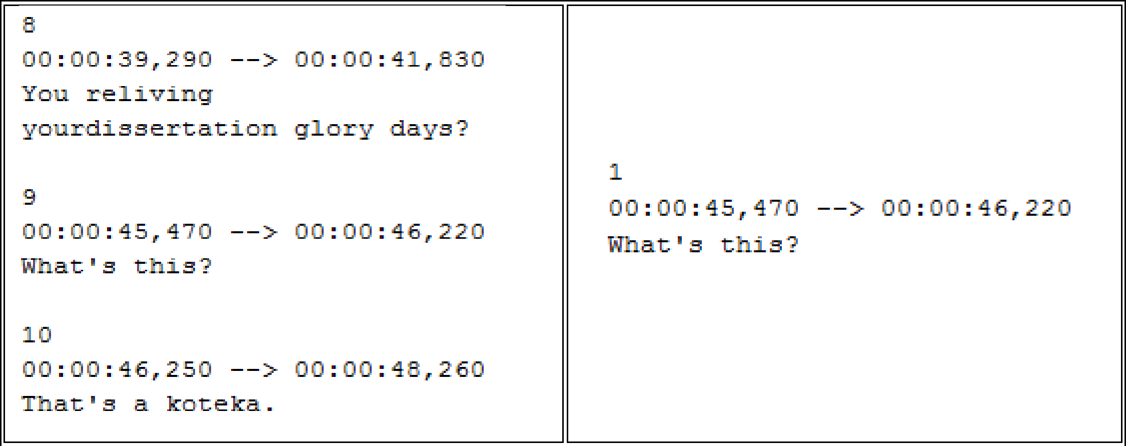


VideoClipName
avi
avi_with_landmarks
facialFeatures
srt
wav

5.3 Subtitle Parsing

We also provide subtitle information in our database if it was provided with the main video file. The file format should be SRT. When we acquire the subtitle file, first we parse it to random access string information with start and end time values. After we extract the facial clip we know the time information, with time information we get subtitle strings and we create a new SRT file with that string and time values (Figure 5.6).

Figure 5.6: (a) Part of the main srt file, (b) First facial video clip's srt file



```
8
00:00:39,290 --> 00:00:41,830
You reliving
yourdissertation glory days?

9
00:00:45,470 --> 00:00:46,220
What's this?

10
00:00:46,250 --> 00:00:48,260
That's a koteka.
```

```
1
00:00:45,470 --> 00:00:46,220
What's this?
```

(a) lie.to.me.s01e03.a.perfect.score-notv.srt

(b) lie.to.me.s01e03.a.perfect.score-notv_face1.srt

5.4 Audio Extraction

For the audio information for the facial clips, first we extract all audio data from main "avi" file in to a stream structure. After we finish tracking the face, we have the time information, with that information we pull out the amount of sound data from stream to a "wav" file. We store that data under "wav" folder. Later we merge this audio file with video file to create "avi" files.

5.5 Annotation and Features of the Database

Each video clip in the database has been annotated by seven labelers. These people were especially selected because they have no knowledge and experience in automatic emotion recognition. Their professions are architecture, mining engineering, computer engineering, psychology, philosophy and logistics. For every video clip, annotators chose an emotion (neutral, anger, contempt, disgust, fear, happiness, sadness, surprise) and a score for that emotion between 1 and 5, indicating the intensity of the emotion. A score of 1 means the emotions are shown the least, while score of 5 means emotions are at peak. In order to fuse the annotations of the labelers, we used majority voting (plurality voting). That means for a video clip, the emotion class that received the highest number of votes was selected as the label of that video clip. The average score of the winning class are averaged to assign a single score to that clip. Furthermore, we provided the average of scores for all emotions. We organized the database such that there should be at least fifty video clips for every emotion. The number of video clips for each class is given in Table 5.1 along with the average score of each class. The database is also annotated using other features as well, such as gender of the person, whether the accompanying of audio is useful or not, whether FaceTracker has worked successfully or not, approximate head pose information, approximate age of the subject, name or unique label of the subject, language of the video clip, whether gesture (head and hand) exists or not, frame number for apex expression and duration of the video clip. You can see a summary of this information in (Table 5.1). These features have been listed in an excel file which will be provided to researchers along with the database. We have named the database as Bahçeşehir Universtiy Multimodal Affective Database (BAUM-2).

Table 5.1 BAUM-2 Database Properties

Language(TR/EN)	13%
Total Video Clips	700
Gender(Female/Male)	92%
Age(Min)	7
Age(Max)	70
Age(Average)	38
Pose(Direct Looking/All Angles)	56%
Audio(Useful/All Audio)	20%
Duration(Min)(sec)	0.250
Duration(Max)(sec)	13.722
Duration(Average)(sec)	1.918
FaceTracker(Successful/All)	53%
Gesture(No Gesture/All)	94%
Count(Neutral)	102
Count(Anger)	116
Count(Contempt)	52
Count(Disgust)	50
Count(Fear)	61
Count(Happiness)	150
Count(Sadness)	77
Count(Surprise)	92
Score(Average Neutral)	3.68
Score(Average Angry)	3.31
Score(Average Contempt)	3.21
Score(Average Disgust)	3.50
Score(Average Fear)	3.68
Score(Average Happiness)	3.13
Score(Average Sadness)	2.95
Score(Average Surprise)	3.45
Score(Average All)	3.34
Standard Deviation(Average Neutral)	0.987
Standard Deviation(Average Anger)	0.998
Standard Deviation(Average Contempt)	0.996
Standard Deviation(Average Disgust)	1.092
Standard Deviation(Average Fear)	0.907
Standard Deviation(Average Happiness)	0.892
Standard Deviation(Average Sadness)	0.961
Standard Deviation(Average Surprise)	1.124
Standard Deviation(Average All)	0.985

5.6 Emotion Recognition Experiments on BAUM-2 Database

In the experiments we used the Support Vector Machine classifier (Michel, 2003). First we made preliminary experiments on the CK+ Database to verify that we are using the library correctly. Library usage has two phases, which are training SVM with emotional face landmark points and test. We used a Radial basis function kernel and C-support vector classification as the SVM type. The feature vector we used for SVM training and test phases is a 1x132 vector. This vector consists of the x and y coordinates of all the landmark points on a frame as follows:

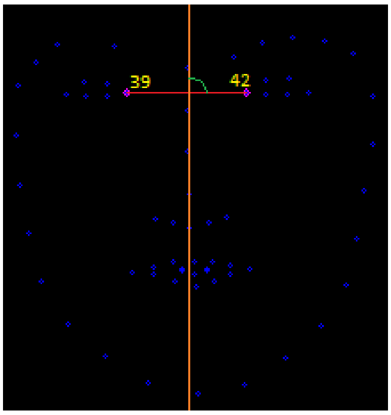
$$V = [x_1, y_1, x_2, y_2, \dots, x_{66}, y_{66}] \qquad \textbf{(5.1)}$$

For the training part on CK+, we chose 235 emotion-labeled frames which consist of 68 landmark points for each. Our face tracker generates 66 landmark points which is a subset of CK+'s landmark points. We removed these two points before testing the library for a fair comparison.

In order to align the face landmark points we used the "39th" and "42nd" points, which are the inner corners of the eyes. For a frontal pose, we assume that when we draw a line connecting these two points, it should be horizontal (that is perpendicular to the y-axis) (Figure 5.5). To get that angle we used dot product:

$$angle = \arccos(x.y/|x||y|) \qquad \textbf{(5.2)}$$

Figure 5.7: Alignment Method



To make it a fair comparison we normalized all the feature points to the range [-1, 1] for both x and y values.

$$-1 \leq 2 * \left(\frac{x-\alpha}{\beta-\alpha}\right) - 1 \leq 1 \qquad \textbf{(5.3)}$$

where α and β are the minimum and maximum values of x and y values.

Finally, we used these aligned landmark points for training (see Figure 5.8) and testing. We both used 10 fold cross validation (see Table 5.2) and LOSO (leave-one-subject-out) cross validation (see Table 5.3) methods with the SVM classifier. The average emotion recognition results are summarized in Table 5.4, which is around 95%.

Figure 5.8: (a) All landmark points for 4 emotions (neutral, disgust, happiness, surprise) from CK+ database, (b) same points after alignment and normalization.

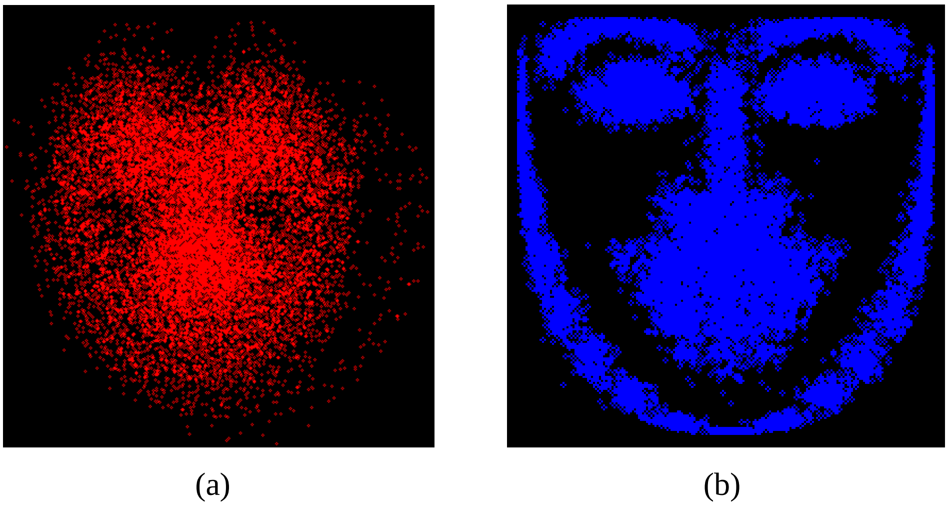


Table 5.2: Confusion matrix for 10-Fold Cross Validation using SVM on CK+ Dataset

	ESTIMATED CLASS			
TRUE CLASS	Neutral	Disgust	Happy	Surprise
Neutral	91.67	4.17	4.17	0.00
Disgust	6.78	93.22	0.00	0.00
Happy	0.00	1.45	98.55	0.00
Surprise	1.20	2.41	0.00	96.39

Table 5.3: Confusion Matrix for Leave-One-Subject-Out SVM Result of CK+ Dataset

	ESTIMATED CLASS			
TRUE CLASS	Neutral	Disgust	Happy	Surprise
Neutral	91.67	4.17	4.17	0.00
Disgust	5.08	94.92	0.00	0.00
Happy	0.00	0.00	100.00	0.00
Surprise	1.20	1.20	0.00	97.59

Table 5.4: Average 4-class emotion recognition rated on CK+ Dataset

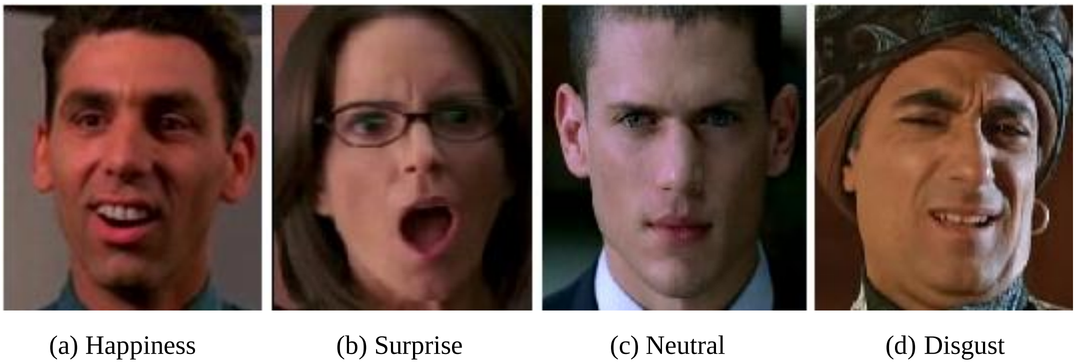
10-Fold Cross Validation Average	95
Leave-One-Subject-Out Average	96

After extracting each facial video clip using FaceTracker algorithm, we can directly access the 3D coordinate information of every landmark point for the face we are tracking. FaceTracker also provides us the 3D coordinates of facial landmarks in the frontal head pose. We used the x and y positions of the landmarks to create a 2D face which is looking directly to the screen. We use the same procedure to align these points to make sure that face is not rotated at all. After we get these aligned points we use them to train and predict the emotion.

For our experiments on BAUM-2 database, we created two image-based datasets consisting of selected frames from the video clips, namely an “easy dataset” and a “challenging dataset”. We conducted 4-class (neutral, disgust, happiness and surprise) and 7-class (neutral, anger, disgust, fear, happiness, sadness and surprise) experiments on these datasets. The “easy dataset” consists of selected frames from clips, where the emotion is at apex and the head pose is mostly frontal (see Figure 5.9 for some examples). These frames are not sequential and selected from every emotion chosen for that dataset. Every emotion has 50 images in the easy dataset. For the challenging dataset, again we selected a total of three frames, which includes the apex frame and the frames adjacent to that frame from the video clips in the database. For challenging dataset we obtained 1041 frames in total. The challenging dataset contains images with not-frontal head pose as well (see Figure 5.10), that’s why it is called as “challenging”.

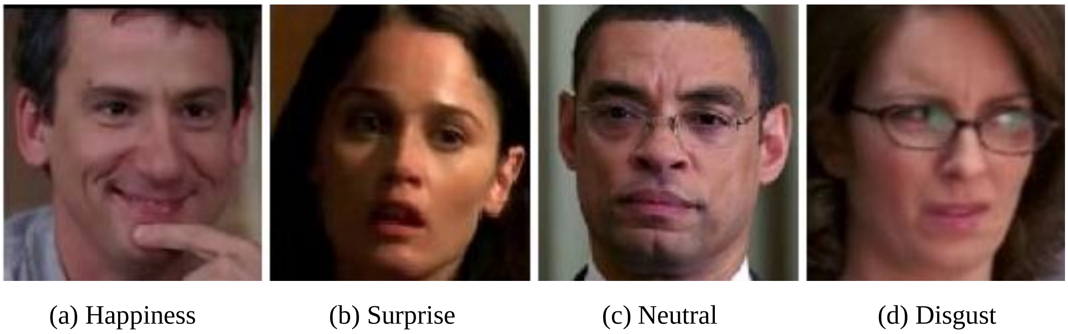
All of these frames in easy and challenging datasets have been chosen from the pool of video clips that FaceTracker algorithm tracks facial landmarks successfully. The confusion matrices for 4-class and 7-class emotion recognition experiments are provided in Tables 5.5-5.12 below, for both 10-fold stratified cross validation and leave-one-subject our cross validation.

Figure 5.9: Examples of Easy Dataset



(a) Happiness (b) Surprise (c) Neutral (d) Disgust

Figure 5.10: Examples of Challenging Dataset



(a) Happiness (b) Surprise (c) Neutral (d) Disgust

Table 5.5: Confusion matrix for Leave-One-Subject-Out Cross Validation with an SVM classifier for 7 Class Emotion Recognition on Easy Dataset

	ESTIMATED CLASS						
TRUE CLASS	Neutral	Anger	Disgust	Fear	Happy	Sadness	Surprise
Neutral	96	0	2	0	0	0	2
Anger	0	88	2	2	0	8	0
Disgust	2	4	82	0	0	4	8
Fear	0	4	0	88	0	6	2
Happy	4	0	4	0	92	0	0
Sadness	0	10	2	2	0	86	0
Surprise	2	14	0	0	2	0	82

Table 5.6: Confusion matrix for 10-Fold Cross Validation with SVM for 7 Class Emotion Recognition on Easy Dataset

	ESTIMATED CLASS						
TRUE CLASS	Neutral	Anger	Disgust	Fear	Happy	Sadness	Surprise
Neutral	90	0	4	0	0	0	6
Anger	0	88	0	4	0	8	0
Disgust	2	8	72	4	2	4	8
Fear	0	10	0	84	0	6	0
Happy	4	0	4	4	88	0	0
Sadness	0	12	4	0	0	84	0
Surprise	4	14	2	2	0	0	78

Table 5.7: Confusion matrix for Leave-One-Subject-Out with SVM for 7 Class Emotion Recognition on Challenging Dataset

	ESTIMATED CLASS						
TRUE CLASS	Neutral	Anger	Disgust	Fear	Happy	Sadness	Surprise
Neutral	79.29	2.53	0.00	1.01	6.57	5.56	5.05
Anger	10.67	56.67	6.67	0.00	10.00	0.67	15.33
Disgust	6.06	10.10	28.28	1.01	25.25	11.11	18.18
Fear	9.52	11.90	7.14	7.14	2.38	7.14	54.76
Happy	3.75	4.58	4.17	0.00	83.33	0.83	3.33
Sadness	31.53	4.50	3.60	0.90	5.41	46.85	7.21
Surprise	13.84	11.32	0.00	0.00	4.40	3.77	66.67

Table 5.8: Confusion matrix for 10-Fold Cross Validation with SVM for 7 Class Emotion Recognition on Challenging Dataset

	ESTIMATED CLASS						
TRUE CLASS	Neutral	Anger	Disgust	Fear	Happy	Sadness	Surprise
Neutral	77.78	3.03	0.00	1.01	5.56	7.07	5.56
Anger	13.33	56.00	7.33	0.67	7.33	1.33	14.00
Disgust	7.07	8.08	27.27	0.00	30.30	9.09	18.18
Fear	9.52	10.71	5.95	4.76	3.57	7.14	58.33
Happy	4.58	4.17	3.75	0.83	82.08	1.25	3.33
Sadness	31.53	5.41	4.50	0.00	3.60	49.55	5.41
Surprise	13.21	11.95	0.00	1.26	3.77	4.40	65.41

Table 5.9: Confusion matrix for Leave-One-Subject-Out with SVM for 4 Class Emotion Recognition on Easy Dataset

	ESTIMATED CLASS			
TRUE CLASS	Neutral	Disgust	Happy	Surprise
Neutral	96	2	0	2
Disgust	2	86	0	12
Happy	4	4	92	0
Surprise	0	8	0	92

Table 5.10: Confusion matrix for 10-Fold Cross Validation with SVM for 4 Class Emotion Recognition on Easy Dataset

	ESTIMATED CLASS			
TRUE CLASS	Neutral	Disgust	Happy	Surprise
Neutral	92	6	0	2
Disgust	2	88	4	6
Happy	4	4	92	0
Surprise	6	6	0	88

Table 5.11: Confusion matrix for Leave-One-Subject-Out with SVM for 4 Class Emotion Recognition on Challenging Dataset

	ESTIMATED CLASS			
TRUE CLASS	Neutral	Disgust	Happy	Surprise
Neutral	86.36	0.51	6.57	6.57
Disgust	11.11	37.37	30.30	21.21
Happy	5.42	4.17	85.83	4.58
Surprise	16.35	3.77	5.03	74.84

Table 5.12: Confusion matrix for 10-Fold Cross Validation with SVM for 4 Class Emotion Recognition on Challenging Dataset

	ESTIMATED CLASS			
TRUE CLASS	Neutral	Disgust	Happy	Surprise
Neutral	84.85	1.01	6.57	7.58
Disgust	14.14	33.33	30.30	22.22
Happy	7.50	4.17	84.17	4.17
Surprise	15.72	3.14	5.03	76.10

Table 5.13: Average Emotion Recognition Rates for 4 Class Datasets

	Easy Dataset	Challenging Dataset
10-Fold Cross Validation Average	90	69.61
Leave-One-Subject-Out Average	91.5	71.1
Average Scores for 4 Class Datasets	3.67	3.44
Standard Deviation of Scores	0.95	1.02

Table 5.14: Average Emotion Recognition Rates for for 7 Class Datasets

	Easy Dataset	Challenging Dataset
10-Fold Cross Validation Average	83.43	51.84
Leave-One-Subject-Out Average	87.71	52.6
Average Scores for 7 Class Datasets	3.51	3.33
Standard Deviation of Scores	0.93	0.99

From the above experimental results (Table 5.13 and Table 5.14), we can observe that the recognition rates on the challenging BAUM-2 dataset are much lower as compared to the easy dataset, as expected. That is because there are more diverse head pose and illumination changes and some of the videos have hand gestures in the challenging dataset. The average recognition rate for the 7 class experiment is around 52% for both 10 fold cross validation and LOSO cross- validation. This implies that the geometrical features consisting of the landmark positions are not sufficient and hence more research is needed regarding feature selection, head pose and illumination normalization for emotion recognition from images captured in more realistic conditions as compared to laboratory environments.

6. CONCLUSION AND FUTURE WORK

In this thesis, we first experimentally compared several state of the art facial landmark tracking methods on the CK+ dataset. FaceTracker [Saragih 2009] was found to be superior to other compared algorithms for tracking landmark points. We improved the performance of FaceTracker by using a skin color post-filter and a SURF–based scene cut detection approach.

In this thesis a software tool has also been developed, which takes a movie as input and gives facial video clips as output. The software detects the faces in the video frames automatically and tracks them until there is a scene cut or significant occlusion. The accompanying audio track is also extracted and merged with the extracted facial video clip. Subtitle information is also extracted, if an “srt” file has been provided.

We used the developed software to process more than 100 movies in English and Turkish and created an affective audio-visual database. The BAUM-2 (Bahçeşehir University Multimodal Affective Database) currently consists of 700 clips with 8 emotions. The database can easily be extended by processing films in other languages and it will be made available to researchers via a web site.

We also carried out emotion recognition experiments on BAUM-2 database using geometrical facial features. We used support vector machines (SVM) as a supervised learning method. We created two image datasets from BAUM-2 for emotion recognition experiments on static images, namely the “easy dataset” and the “challenging dataset”. The challenging dataset contains facial expressions with difficult cases such as non-frontal head poses. Emotion recognition experiments on BAUM-2 database indicate that emotion recognition in the-wild (on naturalistic images) is quite challenging and requires more research, since the average emotion recognition rate on the challenging dataset was only around 52%. Emotion recognition on the videos in BAUM-2 database is left as future research.

Other directions for future research may be listed as follows:

1. Appearance based features (e.g. Local binary patterns, Gabor Features) might also be used as well as geometrical features for emotion recognition.
2. Emotion can also be recognized from the audio channel using prosodic features of speech (if the audio track is useful). The results can be merged with results of emotion recognition from images.
3. Subtitles can also be useful in recognizing the emotion.
4. Emotion can be recognized by using all modalities: visual, audio and subtitle information.
5. The software can be improved in terms of speed, by using a parallel programming techniques.

REFERENCES

Books

Sonka, M., Hlavac, V. & Boyle, R., 2008. *Image Processing, Analysis, and Machine Vision, Third Edition.* s.l.:Thomson Engineering.

Scherer, K., Bänziger, T. & Roesch, E., 2010. Introducing the Geneva Multimodal Emotion Portrayal. In: T. B. a. E. R. K.R. Scherer, ed. *A Blueprint for Affective Computing: A sourcebook.* England: Oxford university Press, pp. 271-294.

Journals

Bay, H., Ess, A., Tuytelaars, T. & Gool, L. V., 2008. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding 110,* Volume 1, p. 346–359.

Cohn, P. L. J. F., Kanade, T., Saragih, J. & Ambadar, Z., 2010. The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit. *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on,* Volume 1, pp. 94 - 101.

Gross, R. et al., 2008. Multi-PIE. *Proceedings of the Eighth IEEE International Conference on Automatic Face and Gesture Recognition,* Volume 1, p. 1–8.

Juan, L. & Gwun, O., 2009. A Comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing IJIP,* 3(4), pp. 143-152.

Lienhart, R. & Maydt, J., 2002. An Extended Set of Haar-like Features for Rapid Object Detection. *Image Processing. 2002. Proceedings. 2002 International Conference on,* Volume 1, pp. I-900 - I-903.

Lowe, D. G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision 60,* Volume 1, pp. 91-110.

O'Toole, A. J., Harms, J., Snow, S. L. & Hurst, D. R., 2005. A Video Database of Moving Faces and People. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Volume 1, pp. 812-816.

Saragih, J. M., Cohn, J. & Lucey, S., 2009. Face Alignment through Subspace Constrained Mean-Shifts. *Computer Vision, 2009 IEEE 12th International Conference on,* Volume 1, pp. 1034 - 1041.

Sneddon, I., 2012. The Belfast Induced Natural Emotion Database. *IEEE Transactions on Affective Computing,* pp. 32 - 41.

Ulukaya, S., Erdem, C., Karaali, A. & Erdem, A., 2011. Combining Haar Feature and Skin Color Based Classifiers for Face Detection. *Acoustics, Speech and Signal Processing (ICASSP),* Volume 1, pp. 1497 - 1500.

Viola, P. & Jones, M., 2001. Rapid Object Detection using a Boosted Cascade of Simple Features. *Computer Vision and Pattern Recognition, 2001. CVPR 2001.*

Proceedings of the 2001 IEEE Computer Society Conference on, Volume 1, pp. I-511 - I-518.

Taylor, C., Lanitis, A. & Cootes, T., 1994. Active Shape Models : Evaluation of a Multi-Resolution Method for Improving Image Search. *Proc. British Machine Vision Conference,* Volume 1, pp. 327-336.

Zeng, Z., 2009. A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* pp. 39 - 58.

Other

Abhinav Dhall, Roland Goecke, Simon Lucey, Tom Gedeon, 2011. *Acted Facial Expressions In The Wild Database,* Australia: Research School of Computer Science, Australian National University.

Anon., 2012. *belfast naturalistic database.* [Online]
Available at: http://sspnet.eu/2010/02/belfast-naturalistic/
[Accessed 20 6 2012].

Borenstein, G., 2012. *Makematics.* [Online]
Available at: http://www.makematics.com/research/viola-jones/
[Accessed 1 6 2012].

Cootes, T., 2004. *Active Shape Models.* [Online]
Available at:
http://personalpages.manchester.ac.uk/staff/timothy.f.cootes/Models/asms.html
[Accessed 1 6 2012].

Cootes, T., 2012. *Active Appearance Models.* [Online]
Available at: http://personalpages.manchester.ac.uk/staff/timothy.f.cootes/Models/aam.html
[Accessed 1 6 2012].

eNTERFACE'05, 2012. *Emotional database.* [Online]
Available at: http://psy.ff.uni-lj.si/Katedre/PM/speech_emotion/clanki/Audiovisual%20emotion%20database.pdf
[Accessed 1 5 2011].

Evans, C., 2012. *opensurf.* [Online]
Available at: https://opensurf1.googlecode.com/files/OpenSURF.pdf
[Accessed 3 6 2012].

Fg net aging database, 2012. *Face and Gesture Recognition Research Network.* [Online]
Available at: http://www.fgnet.rsunit.com/
[Accessed 1 6 2012].

Herbert Bay, T. T. L. V. G., 2006. SURF: Speeded Up Robust Features.

Huang, G. B., Ramesh, M., Berg, T. & Learned-Miller, E., 2007. *Labeled faces in the wild: A database for studying face recognition in unconstrained environments,* Amherst: University of Massachusetts.

Kittipanya-ngam, P. & Cootes, T., 2002. *Comparing Variations on the Active Appearance Model Algorithm.* BMVC

Lowensohn, J., 2012. *cnet news.* [Online]
Available at: http://news.cnet.com/8301-27076_3-10363727-248.html
[Accessed 1 6 2012].

Michel, P., 2003. *Support Vector Machines in Automated Emotion Classification*

Milborrow, S., 2007. *Locating Facial Features with Active Shape Models*

opencv, 2012. *itseez.* [Online]
Available at: http://opencv.itseez.com/doc/tutorials/imgproc/histograms/template_matching/template_matching.html
[Accessed 1 6 2012].

opencv, 2012. *OpenCV.* [Online]
Available at: http://sourceforge.net/projects/opencvlibrary/
[Accessed 13 6 2012].

Oyallon, E. & Rabin, J., 2012. *Image Processing On Line.* [Online]
Available at: http://www.ipol.im/pub/algo/or_speeded_up_robust_features/
[Accessed 1 6 2012].

Wallhoff, F., 2006. *Facial expressions and emotion database.* [Online]
Available at: http://www.mmk.ei.tum.de/ waf/fgnet/feedtum.html

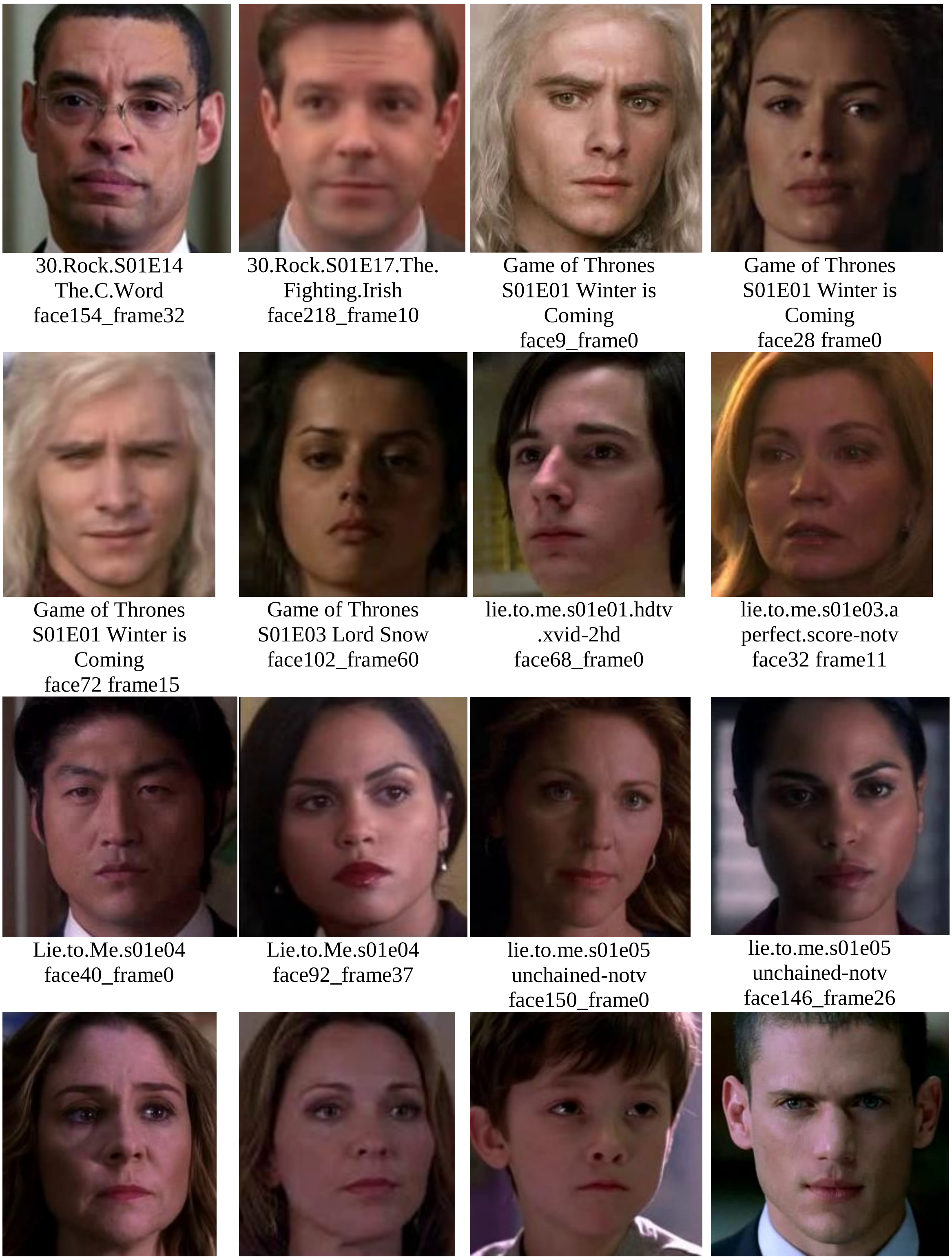
Wikipedia, 2012. *Wikipedia.* [Online]
Available at: http://en.wikipedia.org
[Accessed 1 3 2012].

Yao, W., 2011. *AsmLibrary.* [Online]
Available at: http://code.google.com/p/asmlibrary/
[Accessed 1 3 2012].

APPENDICES

Appendix A: Examples from BAUM-2 Database

Figure A.1: Examples of neutral facial expression



30.Rock.S01E14
The.C.Word
face154_frame32

30.Rock.S01E17.The.
Fighting.Irish
face218_frame10

Game of Thrones
S01E01 Winter is
Coming
face9_frame0

Game of Thrones
S01E01 Winter is
Coming
face28 frame0

Game of Thrones
S01E01 Winter is
Coming
face72 frame15

Game of Thrones
S01E03 Lord Snow
face102_frame60

lie.to.me.s01e01.hdtv
.xvid-2hd
face68_frame0

lie.to.me.s01e03.a
perfect.score-notv
face32 frame11

Lie.to.Me.s01e04
face40_frame0

Lie.to.Me.s01e04
face92_frame37

lie.to.me.s01e05
unchained-notv
face150_frame0

lie.to.me.s01e05
unchained-notv
face146_frame26

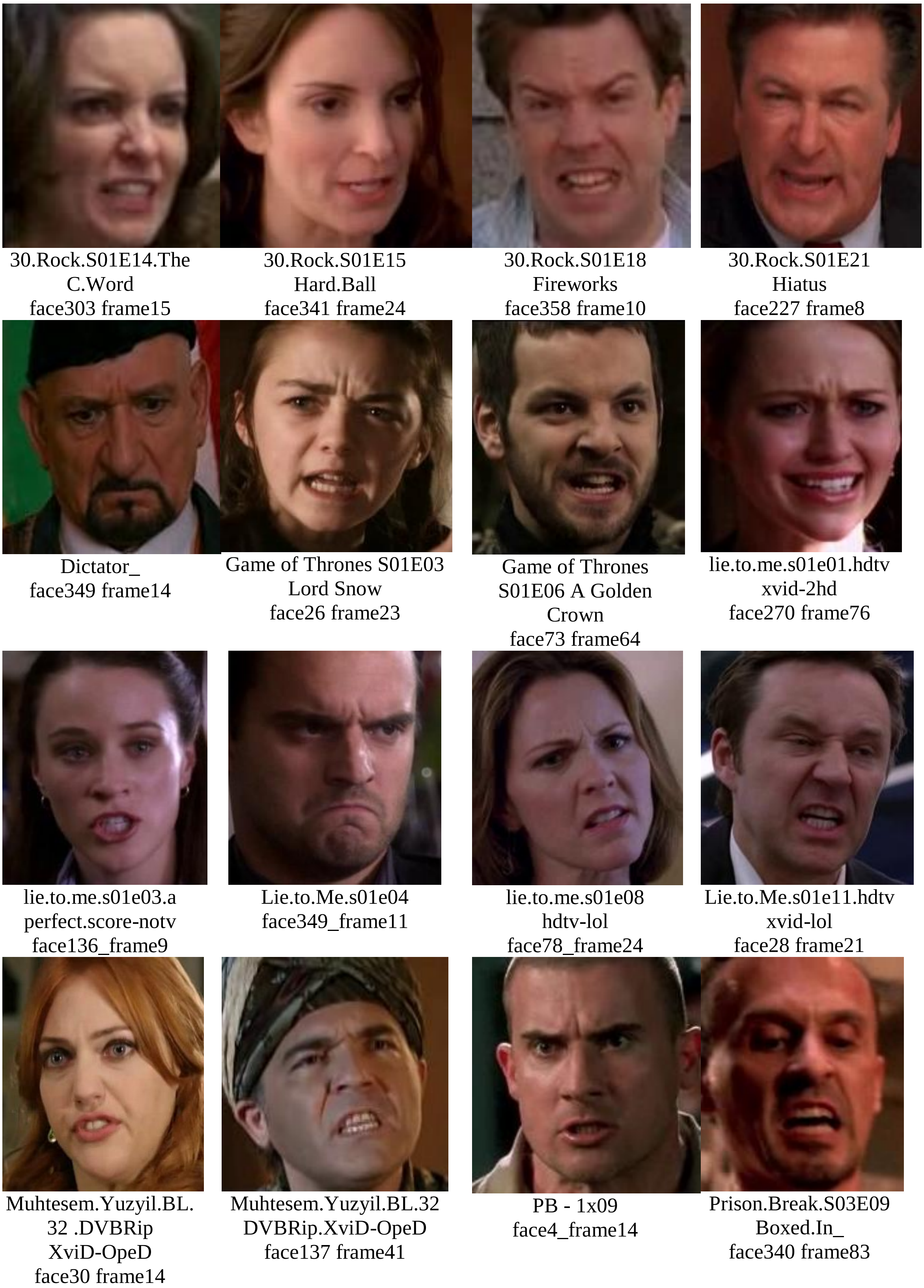
lie.to.me.s01e06
hdtv-lol
face8_frame23

lie.to.me.s01e08
hdtv-lol
face75_frame0

lie.to.me.s01e10
hdtv-lol
face125 frame29

PB 1x01
face16_frame0

Figure A.2: Examples of anger facial expression



30.Rock.S01E14.The C.Word face303 frame15

30.Rock.S01E15 Hard.Ball face341 frame24

30.Rock.S01E18 Fireworks face358 frame10

30.Rock.S01E21 Hiatus face227 frame8

Dictator_ face349 frame14

Game of Thrones S01E03 Lord Snow face26 frame23

Game of Thrones S01E06 A Golden Crown face73 frame64

lie.to.me.s01e01.hdtv xvid-2hd face270 frame76

lie.to.me.s01e03.a perfect.score-notv face136_frame9

Lie.to.Me.s01e04 face349_frame11

lie.to.me.s01e08 hdtv-lol face78_frame24

Lie.to.Me.s01e11.hdtv xvid-lol face28 frame21

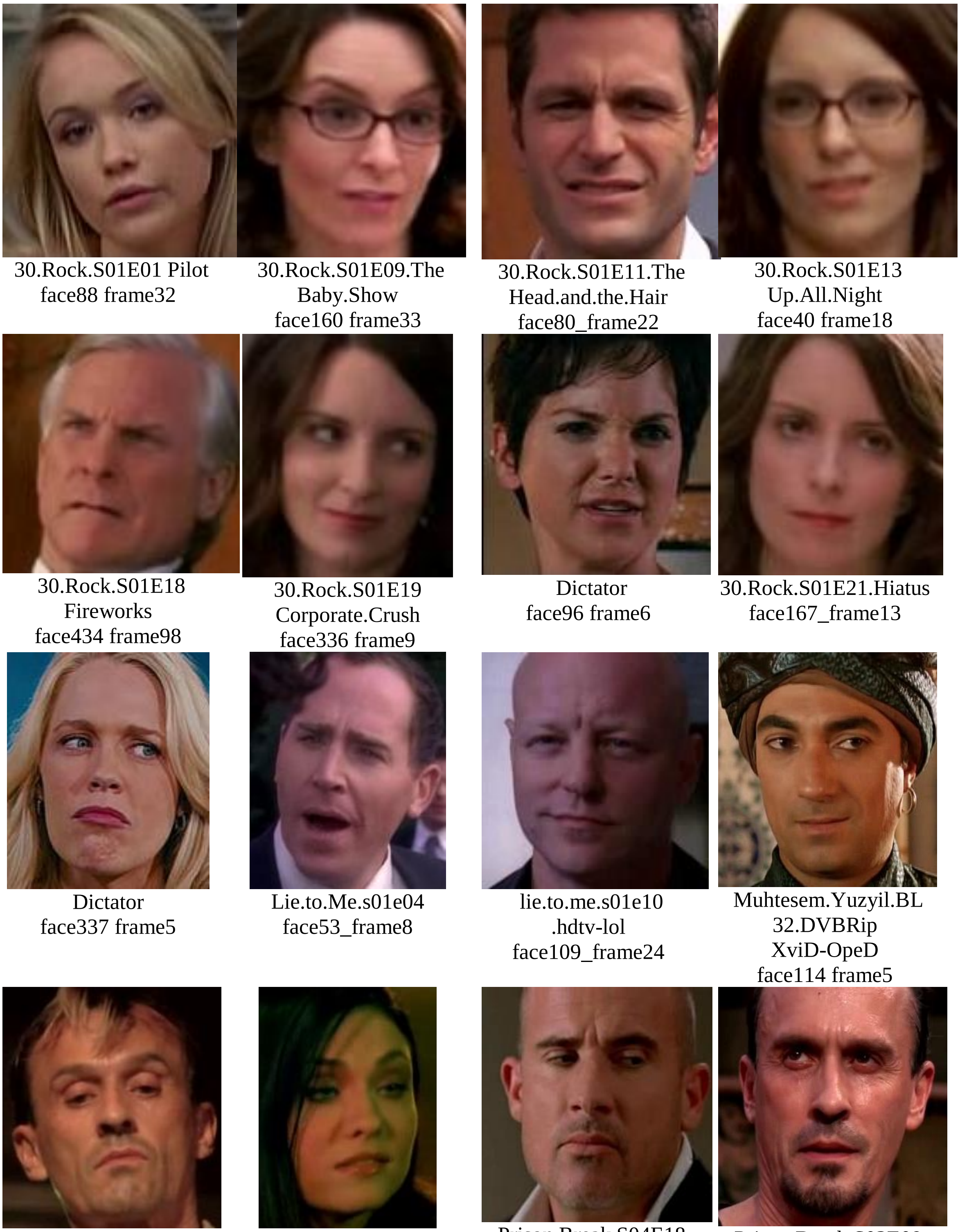
Muhtesem.Yuzyil.BL. 32 .DVBRip XviD-OpeD face30 frame14

Muhtesem.Yuzyil.BL.32 DVBRip.XviD-OpeD face137 frame41

PB - 1x09 face4_frame14

Prison.Break.S03E09 Boxed.In_ face340 frame83

Figure A.3: Examples of contempt facial expression



30.Rock.S01E01 Pilot
face88 frame32

30.Rock.S01E09.The Baby.Show
face160 frame33

30.Rock.S01E11.The Head.and.the.Hair
face80_frame22

30.Rock.S01E13 Up.All.Night
face40 frame18

30.Rock.S01E18 Fireworks
face434 frame98

30.Rock.S01E19 Corporate.Crush
face336 frame9

Dictator
face96 frame6

30.Rock.S01E21.Hiatus
face167_frame13

Dictator
face337 frame5

Lie.to.Me.s01e04
face53_frame8

lie.to.me.s01e10 .hdtv-lol
face109_frame24

Muhtesem.Yuzyil.BL 32.DVBRip XviD-OpeD
face114 frame5

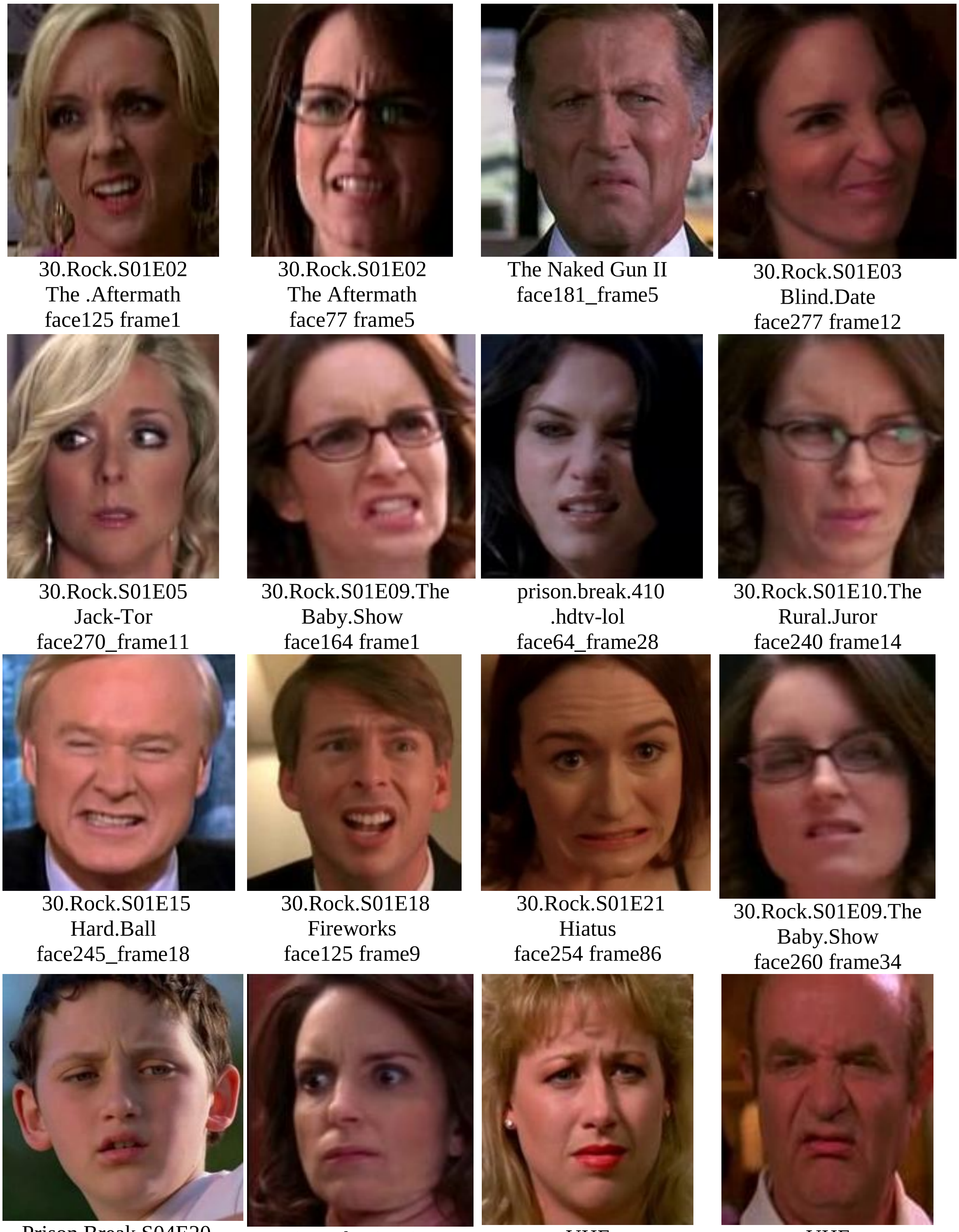
PB 3x01
face306 frame25

Prison.Break.S03E11.Under And.Out
face82 frame14

Prison.Break.S04E18 HDTV.XviD-LOL
face86 frame5

Prison.Break.S03E09 Boxed.In
face415 frame40

Figure A.4: Examples of disgust facial expression



30.Rock.S01E02
The .Aftermath
face125 frame1

30.Rock.S01E02
The Aftermath
face77 frame5

The Naked Gun II
face181_frame5

30.Rock.S01E03
Blind.Date
face277 frame12

30.Rock.S01E05
Jack-Tor
face270_frame11

30.Rock.S01E09.The
Baby.Show
face164 frame1

prison.break.410
.hdtv-lol
face64_frame28

30.Rock.S01E10.The
Rural.Juror
face240 frame14

30.Rock.S01E15
Hard.Ball
face245_frame18

30.Rock.S01E18
Fireworks
face125 frame9

30.Rock.S01E21
Hiatus
face254 frame86

30.Rock.S01E09.The
Baby.Show
face260 frame34

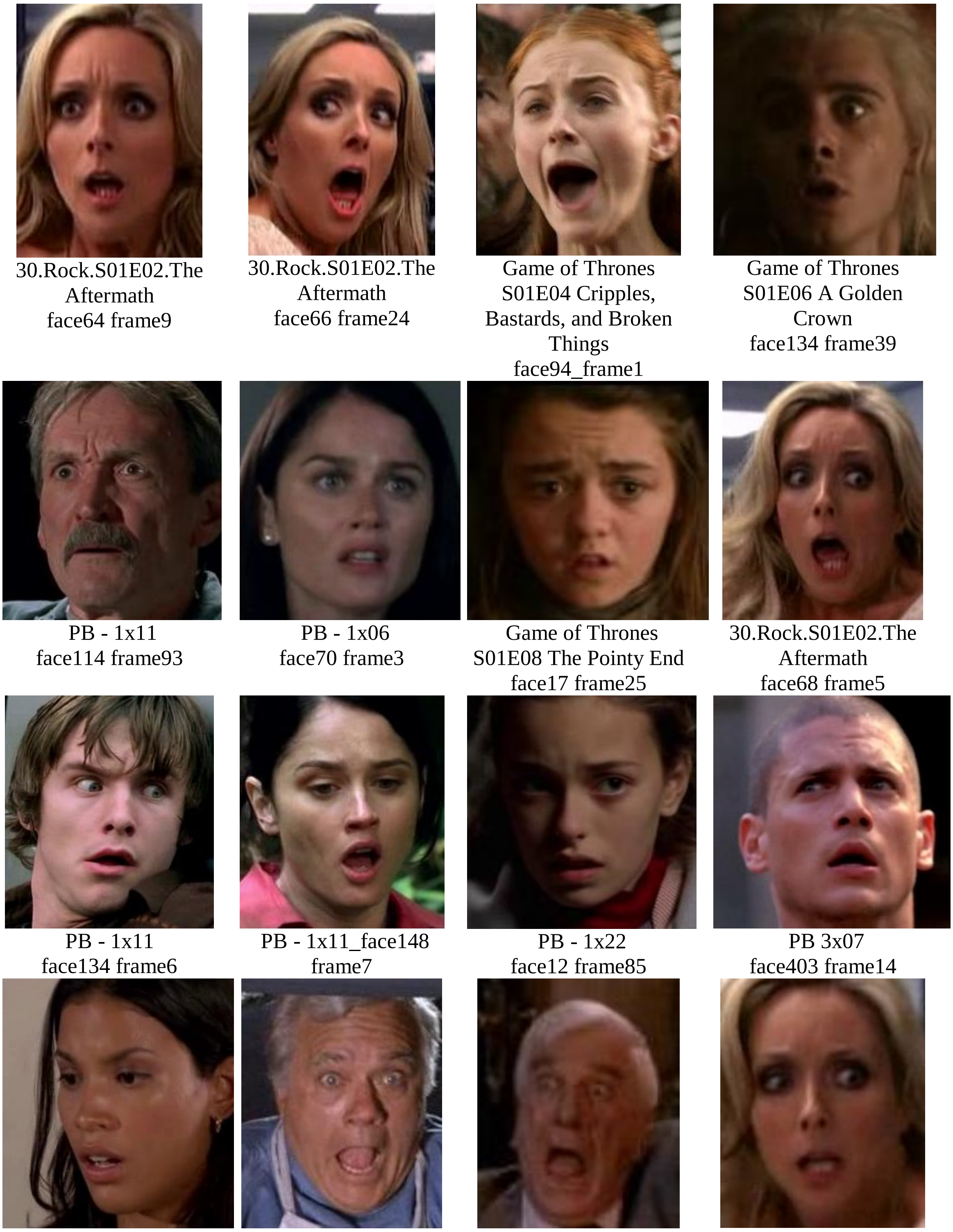
Prison.Break.S04E20
HDTV.XviD-LOL
face268 frame30

30.Rock.S01E03
Blind.Date
face55 frame9

UHF
face357_frame37

UHF
face27_frame30

Figure A.5: Examples of fear facial expression



30.Rock.S01E02.The Aftermath
face64 frame9

30.Rock.S01E02.The Aftermath
face66 frame24

Game of Thrones S01E04 Cripples, Bastards, and Broken Things
face94_frame1

Game of Thrones S01E06 A Golden Crown
face134 frame39

PB - 1x11
face114 frame93

PB - 1x06
face70 frame3

Game of Thrones S01E08 The Pointy End
face17 frame25

30.Rock.S01E02.The Aftermath
face68 frame5

PB - 1x11
face134 frame6

PB - 1x11_face148
frame7

PB - 1x22
face12 frame85

PB 3x07
face403 frame14

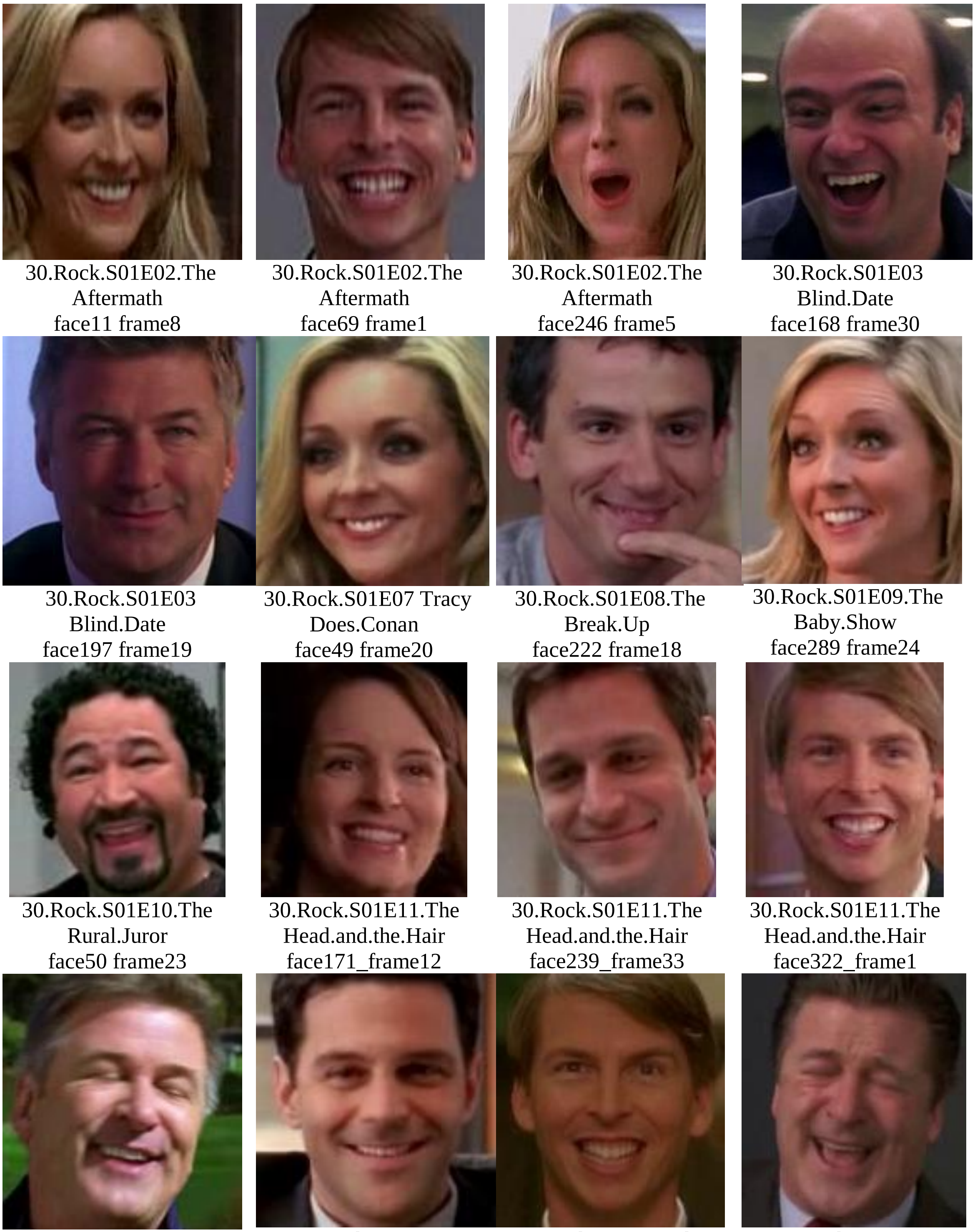
Prison.Break.S03E10 Dirt.Nap
face188 frame22

The Naked Gun I
face171 frame14

The Naked Gun I
face218 frame9

30.Rock.S01E02.The Aftermath
face67_frame9

Figure A.6: Examples of happiness facial expression



30.Rock.S01E02.The Aftermath face11 frame8

30.Rock.S01E02.The Aftermath face69 frame1

30.Rock.S01E02.The Aftermath face246 frame5

30.Rock.S01E03 Blind.Date face168 frame30

30.Rock.S01E03 Blind.Date face197 frame19

30.Rock.S01E07 Tracy Does.Conan face49 frame20

30.Rock.S01E08.The Break.Up face222 frame18

30.Rock.S01E09.The Baby.Show face289 frame24

30.Rock.S01E10.The Rural.Juror face50 frame23

30.Rock.S01E11.The Head.and.the.Hair face171_frame12

30.Rock.S01E11.The Head.and.the.Hair face239_frame33

30.Rock.S01E11.The Head.and.the.Hair face322_frame1

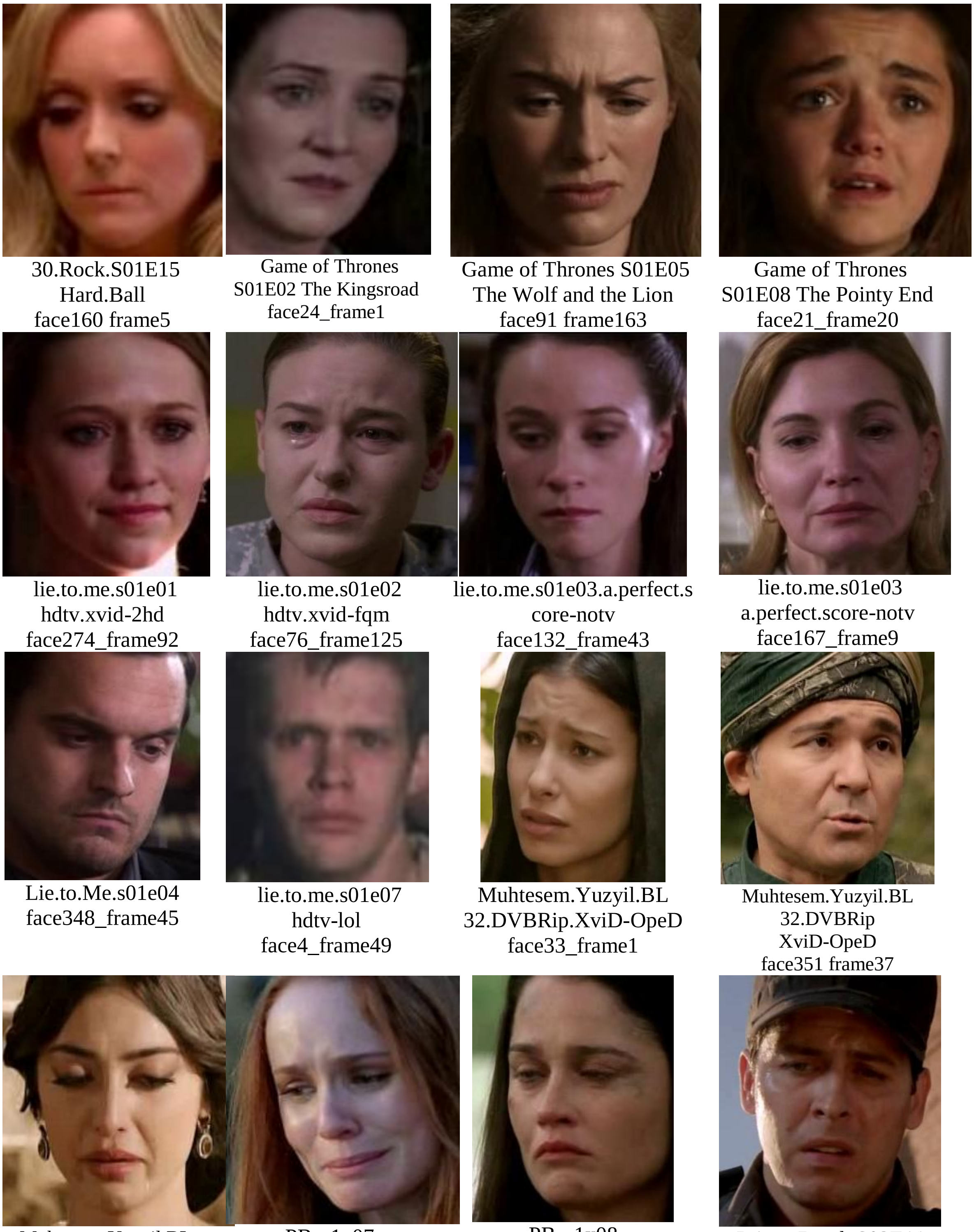
30.Rock.S01E14.The C.Word face208_frame1

30.Rock.S01E15 Hard.Ball face201 frame14

30.Rock.S01E17.The Fighting.Irish face144 frame1

30.Rock.S01E19 Corporate Crush face330 frame1

Figure A.7: Examples of sadness facial expression



30.Rock.S01E15
Hard.Ball
face160 frame5

Game of Thrones
S01E02 The Kingsroad
face24_frame1

Game of Thrones S01E05
The Wolf and the Lion
face91 frame163

Game of Thrones
S01E08 The Pointy End
face21_frame20

lie.to.me.s01e01
hdtv.xvid-2hd
face274_frame92

lie.to.me.s01e02
hdtv.xvid-fqm
face76_frame125

lie.to.me.s01e03.a.perfect.s
core-notv
face132_frame43

lie.to.me.s01e03
a.perfect.score-notv
face167_frame9

Lie.to.Me.s01e04
face348_frame45

lie.to.me.s01e07
hdtv-lol
face4_frame49

Muhtesem.Yuzyil.BL
32.DVBRip.XviD-OpeD
face33_frame1

Muhtesem.Yuzyil.BL
32.DVBRip
XviD-OpeD
face351 frame37

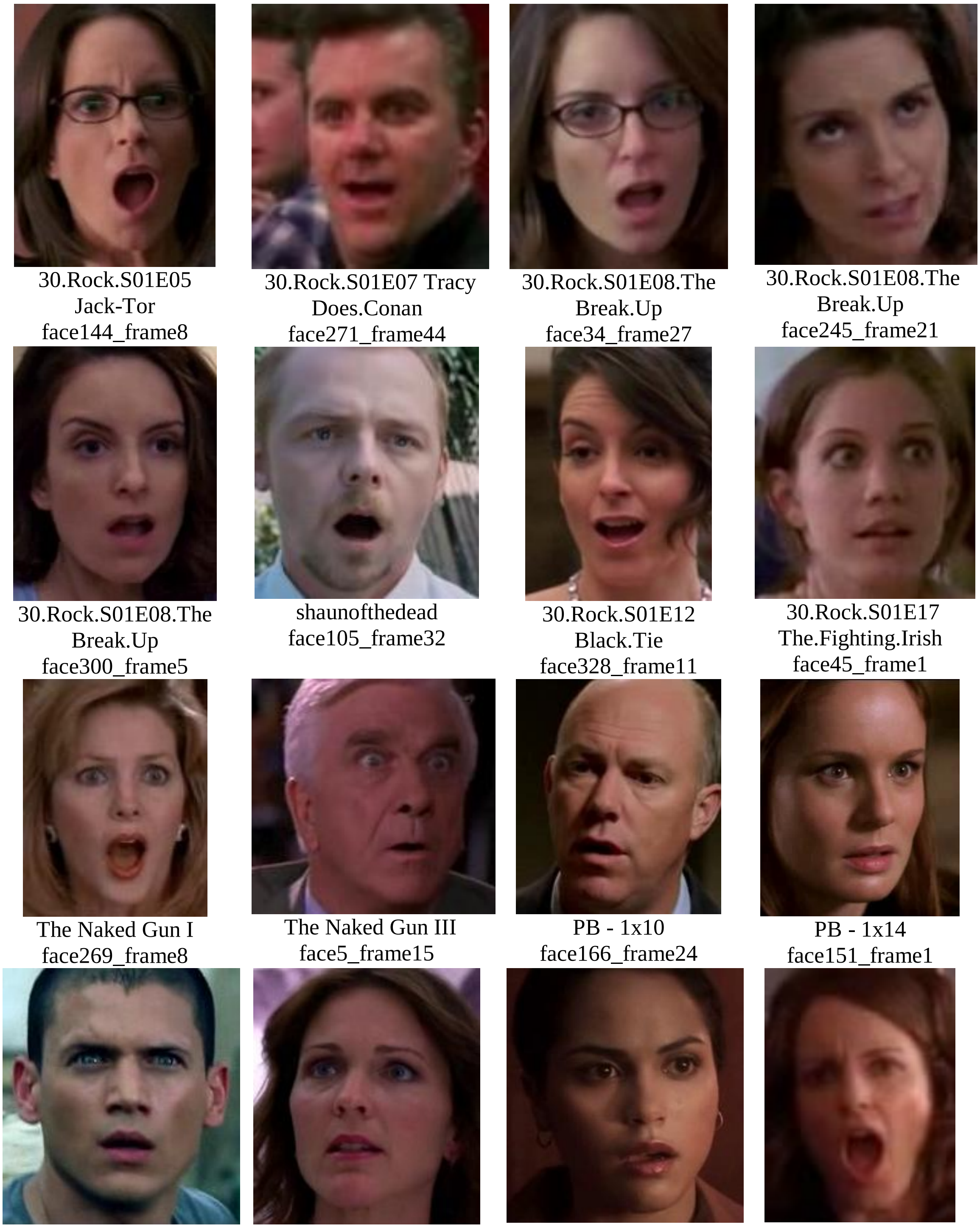
Muhtesem.Yuzyil.BL
32.DVBRip.XviD-OpeD
face1006 frame1

PB - 1x07
face50 frame8

PB - 1x08
face85_frame1

Prison.Break.S03E11
Under.And.Out
face166_frame5

Figure A.8: Examples of surprise facial expression



30.Rock.S01E05 Jack-Tor face144_frame8

30.Rock.S01E07 Tracy Does.Conan face271_frame44

30.Rock.S01E08.The Break.Up face34_frame27

30.Rock.S01E08.The Break.Up face245_frame21

30.Rock.S01E08.The Break.Up face300_frame5

shaunofthedead face105_frame32

30.Rock.S01E12 Black.Tie face328_frame11

30.Rock.S01E17 The.Fighting.Irish face45_frame1

The Naked Gun I face269_frame8

The Naked Gun III face5_frame15

PB - 1x10 face166_frame24

PB - 1x14 face151_frame1

PB - 1x07 face7_frame11

lie.to.me.s01e01 hdtv.xvid-2hd face239_frame63

lie.to.me.s01e03 a.perfect.score-notv face38_frame34

30.Rock.S01E20 Cleveland face305_frame12

Appendix B : List of All Processed Movies

1. 30 Rock Session 1 Episode 1
2. 30 Rock Session 1 Episode 2
3. 30 Rock Session 1 Episode 3
4. 30 Rock Session 1 Episode 4
5. 30 Rock Session 1 Episode 5
6. 30 Rock Session 1 Episode 6
7. 30 Rock Session 1 Episode 7
8. 30 Rock Session 1 Episode 8
9. 30 Rock Session 1 Episode 9
10. 30 Rock Session 1 Episode 10
11. 30 Rock Session 1 Episode 11
12. 30 Rock Session 1 Episode 12
13. 30 Rock Session 1 Episode 13
14. 30 Rock Session 1 Episode 14
15. 30 Rock Session 1 Episode 15
16. 30 Rock Session 1 Episode 16
17. 30 Rock Session 1 Episode 17
18. 30 Rock Session 1 Episode 18
19. 30 Rock Session 1 Episode 19
20. 30 Rock Session 1 Episode 20
21. 30 Rock Session 1 Episode 21
22. Game of Thrones Session 1 Episode 1
23. Game of Thrones Session 1 Episode 2
24. Game of Thrones Session 1 Episode 3
25. Game of Thrones Session 1 Episode 4
26. Game of Thrones Session 1 Episode 5
27. Game of Thrones Session 1 Episode 6
28. Game of Thrones Session 1 Episode 7
29. Game of Thrones Session 1 Episode 8
30. Game of Thrones Session 1 Episode 9
31. Game of Thrones Session 1 Episode 10
32. Lie To Me Session 1 Episode 1
33. Lie To Me Session 1 Episode 2
34. Lie To Me Session 1 Episode 3
35. Lie To Me Session 1 Episode 4
36. Lie To Me Session 1 Episode 5
37. Lie To Me Session 1 Episode 6
38. Lie To Me Session 1 Episode 7
39. Lie To Me Session 1 Episode 8
40. Lie To Me Session 1 Episode 9
41. Lie To Me Session 1 Episode 10
42. Lie To Me Session 1 Episode 11
43. Lie To Me Session 1 Episode 12
44. Prison Break Session 1 Episode 1
45. Prison Break Session 1 Episode 2
46. Prison Break Session 1 Episode 3

47. Prison Break Session 1 Episode 4
48. Prison Break Session 1 Episode 5
49. Prison Break Session 1 Episode 6
50. Prison Break Session 1 Episode 7
51. Prison Break Session 1 Episode 8
52. Prison Break Session 1 Episode 9
53. Prison Break Session 1 Episode 10
54. Prison Break Session 1 Episode 11
55. Prison Break Session 1 Episode 12
56. Prison Break Session 1 Episode 13
57. Prison Break Session 1 Episode 14
58. Prison Break Session 1 Episode 15
59. Prison Break Session 1 Episode 16
60. Prison Break Session 1 Episode 17
61. Prison Break Session 1 Episode 18
62. Prison Break Session 1 Episode 19
63. Prison Break Session 1 Episode 20
64. Prison Break Session 1 Episode 21
65. Prison Break Session 1 Episode 22
66. Prison Break Session 3 Episode 1
67. Prison Break Session 3 Episode 2
68. Prison Break Session 3 Episode 3
69. Prison Break Session 3 Episode 4
70. Prison Break Session 3 Episode 5
71. Prison Break Session 3 Episode 6
72. Prison Break Session 3 Episode 7
73. Prison Break Session 3 Episode 8
74. Prison Break Session 3 Episode 9
75. Prison Break Session 3 Episode 10
76. Prison Break Session 3 Episode 11
77. Prison Break Session 3 Episode 12
78. Prison Break Session 3 Episode 13
79. Prison Break Session 4 Episode 1
80. Prison Break Session 4 Episode 2
81. Prison Break Session 4 Episode 3
82. Prison Break Session 4 Episode 4
83. Prison Break Session 4 Episode 5
84. Prison Break Session 4 Episode 6
85. Prison Break Session 4 Episode 7
86. Prison Break Session 4 Episode 8
87. Prison Break Session 4 Episode 9
88. Prison Break Session 4 Episode 10
89. Prison Break Session 4 Episode 11
90. Prison Break Session 4 Episode 12
91. Prison Break Session 4 Episode 13
92. Prison Break Session 4 Episode 14
93. Prison Break Session 4 Episode 15
94. Prison Break Session 4 Episode 16

95. Prison Break Session 4 Episode 17
96. Prison Break Session 4 Episode 18
97. Prison Break Session 4 Episode 19
98. Prison Break Session 4 Episode 20
99. Prison Break Session 4 Episode 21
100. Prison Break Session 4 Episode 22
101. Muhteşem Yüzyıl Episode 32
102. Dictator
103. Shaun of the Dead
104. Naked Gun 1
105. Naked Gun 2
106. Naked Gun 3
107. The Big Year
108. UHF

CURRICULUM VITAE

Name Surname : Can Kansın

Birthplace and Date : İstanbul, 1986

Second Language : Turkish (native), English (fluent)

High School : Vefa Anadolu Lisesi

Bachelor School : Kadir Has University, Computer Engineering

Graduate School : Bahçeşehir University, Electrical and Electronics Engineering

Institute Name : Graduate School of Natural and Applied Sciences

Work Life : Huawei January 2010 - Today