Rochester Institute of Technology

RIT Digital Institutional Repository

Theses

7-19-2011

Semantic Analysis of Facial Gestures from Video Using a Bayesian Framework

Gati Vashi

Follow this and additional works at: https://repository.rit.edu/theses

Recommended Citation

Vashi, Gati, "Semantic Analysis of Facial Gestures from Video Using a Bayesian Framework" (2011). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by the RIT Libraries. For more information, please contact repository@rit.edu.

Semantic Analysis of Facial Gestures from Video Using a Bayesian Framework

Gati Vashi

gdv7368@cs.rit.edu
http://www.cs.rit.edu/~gdv7368

This M.S. Thesis is submitted to the faculty of the Department of Computer Science, in partial fulfillment of the requirements for the degree of Master of Computer Science in the Golisano College of Computing and Information Science

Rochester Institute of Technology
Rochester, New York

July 19, 2011

Approved by: _____
Dr. Roxanne Canosa, Thesis Advisor

Approved by: _____
Dr. Leon Reznik, Reader

Approved by: _____
Dr. Zack Butler, Observer

Abstract

The continuous growth of video technology has resulted in increased research into the semantic analysis of video. The multimodal property of the video has made this task very complex. The objective of this thesis was to research, implement and examine the underlying methods and concepts of semantic analysis of videos and improve upon the state of the art in automated emotion recognition by using semantic knowledge in the form of Bayesian inference. The main domain of analysis is facial emotion recognition from video, including both visual and vocal aspects of facial gestures. The goal is to determine if an expression on a person's face in a sequence of video frames is happy, sad, angry, fearful or disgusted. A Bayesian network classification algorithm was designed and used to identify and understand facial expressions in video. The Bayesian network is an attractive choice because it provides a probabilistic environment and gives information about uncertainty from knowledge about the domain. This research contributes to current knowledge in two ways: by providing a novel algorithm that uses edge differences to extract keyframes in video and facial features from the keyframe, and by testing the hypothesis that combining two modalities (vision with speech) yields a better classification result (low false positive rate and high true positive rate) than either modality used alone.

Acknowledgement

This thesis could not have been written without Dr. Canosa who guided and encouraged me throughout my research work. I am very much thankful for her kindness and guidance. I have no words to express my gratitude to her.

I am also thankful to Dr. Reznik and Dr. Butler for their valuable support and for reviewing my thesis report.

I would like to take this opportunity to thank my parents for their love and support without them I would not have been part of RIT. I thank my husband for his love and encouragement throughout my academic program. I could not have been completed my report with my husband's encouragement and support. I thank my sister, my friends and all my family members for their continuous support and understanding.

Table of Contents

List of Figures.....IV

List of Tables.....V

Chapter 1. Overview.....1

Chapter 2. Thesis Objectives.....2

Chapter 3. Background.....3

3.1 Shot boundary detection from Video.....5
3.2 Keyframe extraction.....9
3.3 Face detection.....10
3.4 Facial expression segmentation.....14
3.5 Vocal feature segmentation.....23
3.6 Bayesian Network.....25
3.7 Bayesian Inference Techniques.....27

Chapter 4. Approach 30

4.1 Facial expression classification using a Bayesian network.....30
4.2 Voice classification using a Bayesian Network.....35
4.3 Evaluation.....40

Chapter 5. Results and Discussion.....42

5.1 Overall Results of the Three Types of Classifiers.....42
5.2 Comparing the Results Using Histogram Differences to Edge Differences.....47
5.3 Comparing the Results to Existing Research.....47

Chapter 6. Conclusion & Future Work.....50

Appendix A. Source Code.....51

Appendix B. Video Frames.....72

References 77

List of Figures

Figure 1: Process flow diagram of emotion classification.....4
Figure 2: Binary image 1.....7
Figure 3: Binary image 2.....7
Figure 4: Imposed image.....7
Figure 5: Original Face.....11
Figure 6: Skin area segmentation.....12
Figure 7: Filtered image.....13
Figure 8: Final face detection.....14
Figure 9: Face after edge detection.....16
Figure 10: Face sections.....16
Figure 11: Facial features detection.....17
Figure 12: Tracking points on upper and lower lips.....18
Figure 13: Points on left and right eyebrow.....18
Figure 14: Tracking points on face.....19
Figure 15: Points on upper and lower lips on keyframe.....19
Figure 16: Points on left and right eyebrows of keyframe.....19
Figure 17: Tracking points on face in keyframe.....20
Figure 18: Edge detection before the nose wrinkles are formed.....21
Figure 19: Frame image when nose wrinkles are formed.....22
Figure 20: Frame image where nose zone is having wrinkles.....22
Figure 21: Edge detection after nose wrinkles are formed.....23
Figure 22: Singly Connected Poly Tree.....27
Figure 23: Bayesian network for Facial expression classification.....35
Figure 24: Bayesian network for Vocal expression classification.....37
Figure 25: ROC curve for Visual feature classifier.....45
Figure 26: ROC curve for Vocal feature classifier.....46

List of Tables

Table 1: Action Units associated with emotions as per FACS guide.....31
Table 2. Facial feature action unit measurements.....32
Table 3. CPT for lip corner.....33
Table 4. CPT for cheek.....33
Table 5. CPT for nose.....33
Table 6. CPT for lower lip.....33
Table 7. CPT for upper lip.....34
Table 8. CPT for inner eyebrow.....34
Table 9. CPT for outer eyebrow.....34
Table 10. Summary of human vocal effects most commonly associated with the emotions by Murray and Arnott (1993).....36
Table 11. CPT for speech rate.....38
Table 12. CPT for pitch average.....38
Table 13. CPT for pitch changes.....39
Table 14. Vocal feature action unit measurement.....40
Table 15. Summary of classification results.....42
Table 16. Confusion matrix of facial feature classification results.....43
Table 17. Confusion matrix of vocal features classification results.....44
Table 18. Area under ROC curve values per Classifier.....47
Table 19. Summary of classification results based on Edged difference.....47
Table 20. Summary of classification results based on Histogram difference.....48
Table 21. Summary of classification results mentioned in Datcu and Rothkrantz (2008).....48
Table 22. Summary of classification results.....49

Chapter 1: Thesis Overview

An increasingly important topic of interest in recent years is the study of comprehending and fully "understanding" digital videos. The growing interest in this area correlates to the substantial inventory of digital videos that are pervasive in contemporary culture. Advancing technology, such as the introduction of broadband, has contributed to the prevalence of videos easily accessible on the web. In addition, applications like web-based learning, digital libraries and on-demand video technology have increased in use and, therefore, require a growing database of videos to enable their functions. As a result, videos have begun growing in prevalence and, accordingly, perceptive analysis of images, within the context of videos, has been a significant area of recent study. However, analysis pertaining to the specific content of video is an area that is yet to be explored—at least sufficiently. This research undertakes facial gesture video content analysis.

Emotion recognition can be done by extracting verbal and non-verbal expression. Facial emotion analysis has been an active research area. It requires higher level knowledge of visual and vocal information. Researchers are still struggling to provide a complete automated computerized system which can identify emotion regardless of gender, age, context and culture. This thesis is an investigation of and a contribution to the field of facial emotion analysis using a Bayesian network. This research can be useful in numerous fields such as emotion analysis of human-agent interaction, paralinguistic communications, advanced psychiatry and lie detection.

Detailed information of this thesis work has been divided into various sections. Chapter 1 provides an overview of the thesis. Chapter 2 contains the thesis objective, Chapter 3 provides some background information on the technical details, Chapter 4 outlines the approach taken, Chapter 5 gives the results found, and Chapter 6 concludes the results discussion and gives information about future work.

Chapter 2: Thesis Objectives

Contributing to the current body of literature, this thesis has extracted and analyzed facial gestures from the contents of videos, utilizing the visual and vocal properties of the videos to achieve this. The intent is to resolve the issue of facial mood classification by employing a Bayesian network. This is done by instituting four phases: *feature extraction, classification of features using a Bayesian network, fusion of vocal and visual cues* and, finally, *mapping of classification results to high-level semantics.* Overall, feature extraction can be considered the base process used to gather features from the input video. A Bayesian network can then be utilized to analyze these features and interpret facial gestures. Finally, while visual cues are necessary for inferring semantic meaning, the added element of vocal information, such as voice intensity and pitch, can provide better information and, therefore, provide better results as a final outcome of the study. The main objective is to compare and derive strengths and weaknesses of vocal and visual classifier.

In conducting this type of study, many researchers have followed Ekman and Friesen (1978) Facial Action Coding System [FACS] to detect facial expressions. According to Ekman and Friesen (1978), there are seven common emotion expressions which are universal in nature, meaning that they are common across most cultures. These are: *neutral, anger, sad, happy, fear, surprise* and *disgust.* This thesis will seek to achieve recognition of anger, sad, happy, fear and disgust key expression from the videos and evaluate comparison of both classifiers.

RIT LIBRARY

Chapter 3: Background

Semantic analysis of video is a current topic of interest. Many recent research endeavors have made efforts to address the issue of semantic video analysis. One of these was by Wei Yue, et al. (2007), in which the researchers presented a generic framework based on statistics theory. More specifically, they combined the components of visual and vocal semantics, applying pattern classification and Fourier transforms to bridge semantic gaps. Another notable effort was by Liang Lao, et al. (2007), in which integrated low level features and video content analysis algorithms were integrated into the ontology. Discovering a single object of interest by proposing compact probabilistic representation was the objective of Liu, et al. (2008), while Hari, Sundaram, et al. (2003) utilized segmentation, event analysis and then summarization to execute experimental analysis with audio/visual content. Yet another inventory of papers has sought to employ a specific set of videos, such as sports or news videos, in order to address this issue. These include the work of Chen Jianyun, et al. (2003) and Hangzai Fan, et al. (2007). Finally, Alan Hanjalic (2004) dedicated a book to the topic, in which he presents affective video content analysis for mood extraction as a means of enabling the personalization of video content.

The means by which the semantic content of videos is currently evaluated is predominantly based on annotations. However, because videos consist of a collaboration of information modes, including audio, motion and text components, analyzing semantic information can be a complex task. Quite simply, it is one that can be somewhat subjective in nature, resulting in various interpretations of any given video, based on the person viewing it. In essence, there is a significant semantic gap between what video frames represent and the meaning that we extract from them when they are viewed (Naphade, et al., 2001). Such discrepancies continue to pose an ongoing problem. In response to this, the research proposed here will investigate the underlying concepts and methods of semantic analysis of videos and their content. The main domain pertaining to this research will be the analysis of facial gestures of the people presented in the context of the videos, proposing an approach that can efficiently extract facial gestures and define parameters that can help decipher and promote a better understanding of this information in a semantic capacity.

In the current body of literature, many researchers have attempted to achieve these objectives by following the work of psychologists Ekman and Friesen's (1978) facial action coding system [FACS] to detect facial expressions. Yafei Sun (2004) and his research associates have assessed several machine learning algorithms for facial expression classification which included techniques such as Bayesian networks, support vector machines, and decision trees. In yet another study involving facial gestures that are typical for speech articulation, facial gesture recognition was interpreted based on rules related to facial muscle action units. Teol, De Silva and Vaddakkepatt (2004) used both a neural network approach and a statistical approach to recognize facial expression. Finally, facial expression decision analysis was used in Raouzaiou and associates (2004), and was based on evidence theory--an alternative to Bayesian theory.

Facial Action Coding System (FACS) is a comprehensive system which has derived all the possible distinguishable facial muscle movements. It is an anatomically based tool to measure facial behavior. It can be used to provide a standard baseline against which scientists

can evaluate theories pertaining to facial communication. Each observable component of facial muscle movement is called an AU (Action Unit). All facial expressions can be described by a combination of AUs. There are 44 unique Action Units. Out of 44 Action Units, 12 describe the muscles movements of the upper part of the face and 18 describe the lower part. Dr Ekman initially started with various different cultural groups and began to suspect universality of facial expression. This interpretation was controversial because all of the cultures that he researched had exposure to conventional facial expressions. Later, he experimented with two different isolated tribal groups who had minimal exposure to other literate cultures. These groups were able to associate expressions based on story photograph trial. His research reconciled with Charles Darwin's belief on universal expression (Thuy Nguyen, 2005). FACS is used as the theoretical basis of most modern facial animation work and in current psychological research on facial communication.

The main goal of this research is to execute an analysis of facial gestures which can promote a better understanding of the videos from which they are derived. The primary tasks in achieving this goal are depicted in Figure 1, and are briefly summarized here. Additional details are given in the following sections.

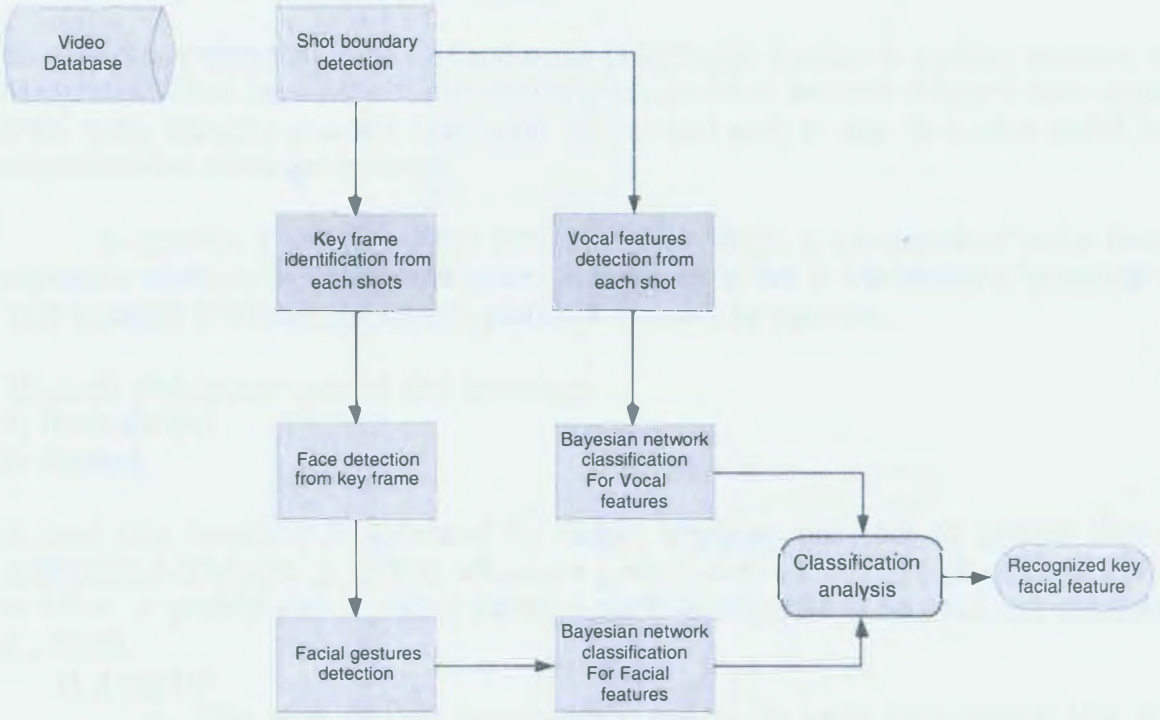


Figure 1. Process flow diagram of emotion classification

The first step is to detect the shot boundary from the video. Each shot would represent a particular facial gesture. From each shot, a keyframe is extracted. This keyframe is a main frame which should represent all the features of the shot. Each shot has frames similar to each other, so

to avoid extra overhead, it is better to do further analysis on the keyframe only, rather than on each and every frame of the shot. In the next step, the face region needs to be segmented. From this face region we can easily obtain facial features which are input to the Bayesian network classification for facial features. In parallel with the facial feature recognition system, speech features are analyzed from each shot and become input to the Bayesian network classification for speech. In the final stage, the output from both classifiers is merged together and analyzed to determine the most probable facial gesture label.

This thesis takes a probabilistic approach to resolving the primary tasks of facial feature classification and vocal feature classification. The Bayesian network is an accepted and very popular approach for classification problems that rely on a probabilistic approach (Russell S. and Norvig P., 1995). The following sections elaborate further on the implementation details, the results, and the application to a larger population of videos.

Evaluation of the entire system is performed using the Database of Facial Expression (DaFEx) videos created by Alberto Battocchi (http://i3.fbk.eu/en/dafex). This database contains a total of approximately 1000 short videos of all six Ekman's emotion expressions and neutral expressions.

3.1 Shot boundary detection from video:

Shot boundary detection is the first and major prerequisite in order to perform semantic analysis. Many researchers have delved into resolving this problem because efficient shot segmentation from video provides a useful foundation for the next task or step. It is also useful for video summarization and video indexing.

In general, a shot is a basic unit in a video which is composed of video frames, and represents some action in time and space. In this case, a shot is a sequence of frames in a video. This sequence of frames represents a particular emotion by a person.

There are mainly two types of shot boundary:
1) Hard (abrupt)
2) Gradual

A hard shot boundary is generated by simply attaching one shot to another shot without modification (Hanjalic, A., 2004), whereas a gradual shot is a slow modification of a shot using an effect. A gradual shot boundary can have the following effects between two shots (Hanjalic, A., 2004).

1) Dissolve
    a. First shot darkens progressively and at the same time second shot gradually lightens
2) Fades
    a. Progressive darkening (light to black) of the shot is referred to as fade-out.
    b. Progressive lighting (black to light) of the shot is referred to as fade-in.
3) Wipe
    a. Any pattern which separates the two shots visually. This pattern moves across the

RIT LIBRARY

video frame and in the end the second shot replaces the first shot.

This thesis has assumed that shots in a video are hard cuts and so there are no effects between them.

While performing shot boundary detection, one can assume that the frames of each shot are significantly different from the frames in all other shots of the video. In facial expression analysis video domains, it can be assumed that each shot represents a particular person, each expressing a particular emotion. In this respect, one can also conclude that each shot may represent a particular expression depicted by a particular person. However, because it can be assumed that two consecutive frames are likely to contain most of the same visual and vocal information, in order to minimize computation overload, calculating only odd numbered frames will take place, as opposed to unnecessarily calculating each and every frame. The most common approach to finding a shot boundary is the utilization of a color histogram difference. It is easy to detect hard cuts by finding a color variation between successive frames. Basically, if there is no significant color difference between frames, then an algorithm is used to find edge difference ratios between frames. If this also renders a matching result, then the frame within the video can be considered to belong to the shot. A significant contribution of this thesis is to improve upon the technique for detecting shot boundaries, as color histogram differences have been found to be inadequate for this purpose. This new technique uses edge difference ratios instead of color differences for finding shot boundaries and for discriminating between facial expressions. The edge difference algorithm developed for this thesis is described in detail in the next section, and is evaluated in the chapter on results.

Equation 1, adopted from (Leinhart, R., 1998), shows how to calculate the color histogram difference (CHD) assuming the RGB color space:

$$CHD_i = \frac{1}{N} \sum_{r=0}^{2^B-1} \sum_{g=0}^{2^B-1} \sum_{b=0}^{2^B-1} |p_i(r,g,b) - p_{i-1}(r,g,b)| \qquad (1)$$

Where $p_i(r,g,b)$ = number of pixels of color (r,g,b) in frame I(i) of N pixels.
r,g,b $\in$ (0 , $2^B - 1$)
B = bits per pixel

This approach works well when each shot is of a different person because there is a significant color difference available between each shot, but it doesn't work well when the consecutive shots have the same person with the same appearances, but showing different emotions. In this case, color values don't change much; therefore edge differences between frames are calculated.

RIT LIBRARY

Using edge differences to improve shot boundary detection and facial features:

Edges are detected when image brightness values get changed abruptly. There are many edge detection algorithms available but the Canny edge detector (Canny, 1986) is the optimal edge detector, in the sense that it is less prone to detect noise and more likely to detect weak edges. The Canny edge detector is used here to detect edges from the video frame image. Once edges are detected, the whole frame image becomes a binary image. This binary image has a black background. Next, the total number of edge pixels is calculated from the binary frame image. This binary image frame is inverted by subtracting the image values from a pure white image. The resulting inverted image now has a white background because after inversion, the white pixels become black and the black pixels become white. Edges are then dilated. The output of an AND operation between the dilated frame n-1 and the inverted frame n is called entering edge pixels and similarly between the edge frame n and the inverted frame n-1 is termed "exiting edge pixels". Basically the edge pixels in one image that have edge pixels very close by in the second image are not regarded as entering edge pixels to compensate for motion (Lienhart, 2001). Entering edge pixels appear far from existing edge pixels and the count of entering edge pixels would have a high value during a hard cut.

For example, (as explained by Lienhart, 2001) suppose we have two binary successive image frames as shown in Figure 2 and Figure 3. If we overlay binary image 1 onto binary image 2, we can find that entering edge pixels are a segment of edge pixels in image 2 which is far from the closest edge pixels of image 1 as shown in Figure 4.

Figure 2. Binary image 1

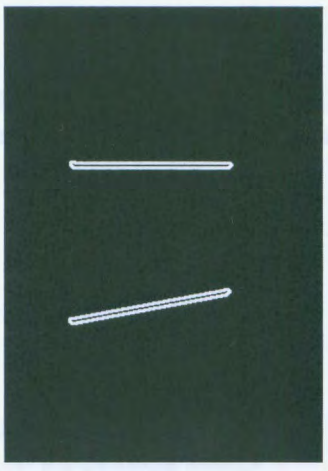


Figure 3. Binary image 2

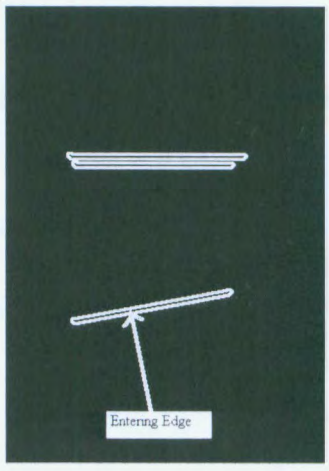


Figure 4. Imposed image

The Edge Change Ratio (ECR) between frame n-1 and n can be calculated using Equation 2,

RIT LIBRARY

adopted from Leinhart, (1998):

$$ECR_n = \max\left(\frac{X^{in}(n)}{\sigma(n)}, \frac{X^{out}(n-1)}{\sigma(n-1)}\right) \quad (2)$$

Where $\sigma(n)$ = number of edge pixels in the frame n,
$\sigma(n-1)$ = number of edge pixels in the frame n-1,
$X^{in}(n)$ = number of entering edge pixels in frame n,
$X^{out}(n-1)$ = number of exiting edge pixels in frame n – 1,
ECR_n = edge change ratio between frame n-1 and n. it ranges from 0 to 1.

Sometimes frames will change abruptly in the video. This indicates a high frame difference value because frame color values or edges may have changed significantly in a very short time. This type of change is known as a "hard shot boundary". The above calculations can efficiently detect hard shot boundaries because they are based on frame differencing.

Shots can be detected once the color histogram difference and edge difference value is above some threshold value, however, using a common threshold value cannot always give the most accurate results. For example, suppose the threshold value is high. In this case, more misdetection is possible. If the threshold value is low, then more false positives are possible. (Dugad, et al., 2002 and Yusoff, et al., 2000) have proposed the approach of adaptive thresholding to overcome this issue. Adaptive thresholding is essentially a two step procedure to confirm the shot boundary:

1) Create dissimilarity measure to compare histograms
   a. Histogram difference values are calculated using Equation 1. The dissimilarity value is calculated using Equation 3, based on the Minkowski distance:

$$d(H(n), H(n+1)) = \sqrt{\sum_{i=0}^{255} (H(n)(i) - H(n+1)(i))^2} \quad (3)$$

H(n) = Histogram of current frame
H(n+1) = Histogram of successive frame

   b. The dissimilarity measure for the edge feature is based on the ECR (edge change ratio) which is calculated using Equation (2).

2) Apply local thresholding schema on dissimilarity measure values:

Generally, a difference caused by motion does not change much from one frame to the next. So a shot cut will be detected with a sudden increase of this difference. To identify the shot cut,

RIT LIBRARY

examine the difference values using a sliding window of 10-15 frames and compute statistical parameters. This will be used to decide if there is a shot cut between the two frames in the middle of the window. The adaptive thresholding technique is adopted from (Dugad et al., 1998).

a. Use the dissimilarity measure within the sliding window of 11 frames width. The accuracy of shot detection is better when the window size is small, but too small of a value (such as five) can cause excessive computation.
b. Means and standard deviations of the left and right side of the window are calculated
c. Shot boundary is detected if
    i. Middle value is maximum in the window
    ii. Middle value is greater than

$$\max\left(\mu(\text{left}) + T_d\sqrt{\sigma(\text{left})}, \mu(\text{right}) + T_d\sqrt{\sigma(\text{right})}\right) \quad (4)$$

Where
T_d = threshold value;
μ (left) = mean value of the left side of the dissimilarity measure value of window
σ (left) = standard deviation of the left side of the dissimilarity measure value of window
μ (right) = mean value of the right side of the dissimilarity measure value of window
σ (right) = standard deviation of the right side of the dissimilarity measure value of the window

3.2 Key frame extraction

The keyframe is the main frame in a particular shot, which should present all the aspects of the shot. It is not wise to segment facial features from all the frames in the video due to computational costs. It is best to segment facial features using the keyframe only.

The keyframe detection algorithm was initially adopted from Kim and Park (2004), where the cumulative measure is used, based on histogram differences, as shown in Equation (5):

$$C = \sum_{t}^{t+k}\left(\sum_{j} | H_{t+1}(j) - H_t(j) |\right) \quad (5)$$

Where k = total number of accumulated frames in the shot,
H_t (j) = histogram in the j^{th} bin $0 <= j <= 255$,
$t = t^{th}$ frame.

As discussed by Kim and Park (2004), there are two criteria to detect the key frame: (1) the

RIT LIBRARY

cumulative measure value C between the current frame and the previous key frame is larger than the given threshold, and (2) the histogram difference between the previous key frame and the current frame is larger than the threshold. For initialization, the first frame is used as the first key frame.

This method did not work well with the DaFEx video shots because this algorithm is based on how color histogram values are changing over time. We have only one person showing emotion in the same setting and lighting. The only cases when cumulative histogram values differ are during lip movement when teeth are shown. This is not sufficient and did not provide good results. Finding an accurate keyframe is very essential as all subsequent processing is dependent on it. To get better results edge difference values, as discussed above, were used instead of the histogram difference value. Edge change values are calculated based on Equation 2. Key frame is detected as soon as maximum change value is found. This worked well because our assumption of key frame is based on the fact that the key frame should represent emotion. This works well because facial feature movements are common during emotion.

3.3 Face detection

The DaFEx database videos (described in detail in Chapter 4, Approach) were filmed in a normal lighting condition and do not have shadows. All videos were filmed with normal lighting variations.

The face detection algorithm is applied on the first frame and the key frame of each shot to extract the face region. This face region is the input region for extracting the facial features. The skin area is an important cue to detect the face. The first step is to transform the entire frame into the skin region image. A skin color model is used to segment the skin area. A sample skin color range is calculated from the training videos. From the RGB color space, all areas in the image where the R, G and B values are in the sample skin color range are considered as face candidate blobs. The largest blob area gives a rough estimate of the face location in the keyframe.

RIT LIBRARY

Figure 5 shows an original frame from a selected video. First, all the pixels are detected as skin pixels whose RGB values are within the sample skin color range. All the detected skin pixel values are then set to 255 and the rest of the pixel color values are set to 0. This results in Figure 6.

Figure 5. Original Face

RIT LIBRARY

Figure 6 still has some unwanted pixels which are not the skin pixels. This is much like "salt and pepper" noise. Median filtering is applied on the grayscale frame to reduce the "salt and pepper" noise. A median filter is very effective in this case because the goal is to simultaneously reduce noise and preserve edges. Median filtering works by applying 3 x 3 window mask and the center value is replaced by the median value of all neighboring pixels.

Figure 6. Skin area segmentation

RIT LIBRARY

Figure 7 is an image frame after applying median filtering. It has significantly less noise than Figure 6.

Figure 7. Filtered Image

The goal of face detection is to detect just the face without another body part. The problem with the sample skin color range detection is that even the neck part is getting detected which can add error to the next stage where we need to detect facial features. To avoid that, the facial height/width ratio is used. If the detected face region height and width ratio is greater than the threshold value then the detected face region height will get reduced to ¾ of the total height. This threshold limit value has been determined using facial anatomy knowledge and by testing on the various data set of DaFEx videos. This limit worked well with 100% success rate.

RIT LIBRARY

The final detected region is only the face area, which is shown in Figure 8

Figure 8. Final face detection

3.4 Facial expression segmentation:

There are 4 main features which are needed to be extracted from the face region.

1. Eye Brow contour
2. Cheek area
3. Nose area
4. Lip contour

These facial features have horizontal edges in the face. Kun Peng, et al. (2005) have used horizontal and vertical projections of the gradient image to determine the face width and the eye locations.

Eye locations, face size, and facial anatomy knowledge were combined to derive facial features such as lips, eyebrows, nose and cheek. Once facial feature areas are identified, tracking point positions on features are calculated.

The horizontal projection of the binary image detects the number of foreground pixels in each

RIT LIBRARY

row and the vertical projection detects the number of foreground pixels in each column. As seen in Figure 6, the horizontal projection would have a large count because the number of foreground pixels around the eye areas has a high value.

According to facial anatomy, eyes are located on the upper part of the face, and from face detection, the face height is known. By applying a horizontal projection on the gradient image of the face, the eye zone can be identified, because the pixels around the eye area will be more changeful. The peak of the horizontal projection is identified as the eye zone. Once we know the eye location and face width, we can determine the cheek zone, nose zone and lip zone based on the facial anatomy assumption and the vertical and horizontal projections. As seen in Figure 7, the vertical projection would have a large count because the number of foreground pixels around the nose and cheek area has a high value. Once the projections are found, the next step is to determine the motion in these zones.

Initially, optical flow analysis was used to get the motion vector. For a given sequences of image frames, optical flow estimates the local image motion based upon the local derivatives (Barron et al. 2005). Optical flow is calculated between the first frame and the key frame. The eye brow zone is further equally divided into two sections. The one which is near the nose zone would be the lower brow zone, and the zone farther from the nose zone would be the upper brow zone. Before calculating motion in these zones, a smoothing filter is applied to reduce noise. The Lucas-Kanade two frame differential method was used to determine the optical flow in the facial feature zones. This motion should simulate the physical facial feature muscle movements. For example, if the eyebrow zone motion direction is upward then we can store the result as a raised eyebrow. Experiments revealed that optical flow is not an efficient solution in this case because facial feature motion is more like a shape change, rather than rigid object motion. Optical flow is more efficient in the cases where shape changes are not allowed and rigid object motion is occurring. So rather than calculating optical flow between frames it would be better to track some interesting points on the eyebrows and lips. The interesting points for the eyebrows are located on the outer eyebrow, inner eyebrow and between the outer and inner sides of eyebrows. Each point's x and y positions are tracked as shown in the following figures.

RIT LIBRARY

Figure 9 shows the face image after the edge detection. The Sobel edge detection algorithm was used to detect edges. As seen in the Figure 9, edges are more prominent around the eye area and lip area. This face image is equally divided into upper half and lower half since we know the fact that eyes are on the upper part of the face and lips are on the lower part of the face. The upper portion of the face is further equally divided so that the left portion shows the left eye and the right portion shows the right part. The lower portion of the face shows the lips. This is shown in Figure 10.

Figure 9. Face after edge detection

Figure 10. Face sections

RIT LIBRARY

Figure 11 shows all of the facial features that are detected on a frame.

Figure 11. Facial features detection

The left eyebrow, right eyebrow and lips are detected using the horizontal and vertical projections on the binary edge images of the upper left, upper right and lower half section. Eyebrow detection is also based on the following three facts:

1) On the upper part of the face there are two main regions: 1) eyes, 2) eyebrow. The eyebrow region is the first region to have many edges
2) An eyebrow's length along the y dimension is a little less than the width of the upper left face or the width of the upper right face.
3) There is gap between the eyes and the eyebrows

For lip detection, simply applying a horizontal projection on the lower half of the face does not work effectively in every case. In order to isolate the lip area in a better way we should first find the cheek boundary. This is done based on counting the number of white pixels for every column starting with the first row of the binary image of the lower half face. As soon as the count is greater than the height of the lower face region, that column value is termed the cheek boundary. The lip horizontal projection should be applied for the region after the cheek boundary.

RIT LIBRARY

There are five points on the lip as shown in Figure 12: 1) one on the outer left corner of the lip 2) one on the outer right corner of the lip 3) one on the middle part of the upper lip 4) two on the lower part of the lip.
There are total 3 points on each of the eyebrows as shown in Figure 13: 1) one on the outer left corner of the eyebrow 2) one on the outer right corner of the eyebrow 3) one in the middle of the eyebrow.

Figure 12. Points on upper and lower lips

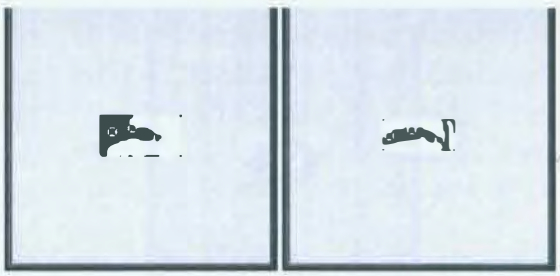


Figure 13. Points on left and right eyebrow

RIT LIBRARY

Figure 14 shows a frame from the video with the tracking points superimposed. Figures 15 and 16 show the feature points that were found during tracking of an actual video sequence, and

Figure 14. Tracking points on face

Figure 15. Points on upper and lower lips

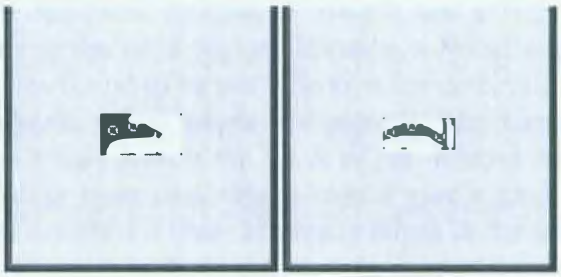


Figure 16. Points on left and right eyebrows

RIT LIBRARY

Figure 17 shows the corresponding video frame from which the features were extracted and tracked.

Figure 17. Tracking points on face of a key frame

The nose and cheeks are detected based on the location of the eyebrows and lips. The nose region is detected by applying horizontal and vertical projections on the area above the lip region. The left cheek region is located below the left eye region and left side of the nose region. The right cheek region is located below the right eye region and right side of the nose region. Cheeks are always located below eyes and right and left side of a nose. The face width is also known, so based on face size, the nose location, and the eyebrow location, the cheek area is defined. The cheek area is a square area along with the nose in horizontal space, and below the eye area in vertical space.

Nose wrinkles are detected from the nose zone. Motion tracking doesn't detect nose wrinkles since nose wrinkles do not represent motion. A simple and efficient way to detect the nose wrinkles is to find the edges in the nose region. Initially, a Sobel edge detection algorithm was used to detect edges but it was found to be not effective for detecting weak edges. In the case of nose wrinkles and raised cheek, subtle edges are present. The Laplace edge detector is more sensitive to noise and hence it also detects the weak edges. If nose wrinkles are available then it would create more edges in the nose zone which would give a clue to estimate the location of wrinkles. Nose wrinkles are detected if there are many edges in the nose region.

RIT LIBRARY

Figure 18 shows the face edges before nose wrinkles are formed.

Figure 18. Edge detection before the nose wrinkles are formed

RIT LIBRARY

Figure 19 shows the frame where nose wrinkles are formed. Figure 20 shows the nose region.

Figure 19. Frame image where nose wrinkles are formed

Figure 20. Frame image where nose zone is having wrinkles

RIT LIBRARY

Figure 21 shows the face edges after nose wrinkles are formed. It is obvious that there are more edges in the nose region of Figure 21 than of Figure 18. These frames show that the likelihood of nose wrinkles occurring during a facial expression is high.

Similarly, the cheek area should also have edges if the cheeks are raised. So instead of applying motion tracking, edge detection is again used to check for raised cheeks.

Figure 21. Edge detection after nose wrinkles are formed

3.5 Vocal feature segmentation

There are three key voice features that need to be identified from a shot: speech rate, pitch average, and pitch changes.

Pitch determination:

Pitch is an inherent part of the human voice. It is generally defined as the rate of vibration of the vocal chords. Pitch rises when vocal chords start to vibrate quickly. These vibrations and speed are dependent on the thickness and length of the vocal chords. Women tend to have higher pitches than men. Pitch is commonly referred to as sensations of frequencies; a high frequency sound generally has a high pitch.

Initially, pitch analysis was done using a simple time domain approach based on autocorrelation. The autocorrelation function is the correlation of a signal to itself (Parr 1999) by delaying some time lag which is represented by:

RIT LIBRARY

$$Xor_p = \log \sum_{i=0}^{n} x_{i+p}\, x_i \quad (6)$$

Where $x_{i+p} = (i+p)^{th}$ signal in that frame
n = number of frames

Equation 6 is adopted from Zeng et al. (2004).

Pitch is detected from a window of the signal. This window is determined based on a frame rate and a sampling rate. The sampling rate is the number of signals of audio per second. For example, if a frame is captured every 0.04 seconds and the sampling rate is 22050 signals per second, then the window length is 0.04 * 22050 = 882.

From this window of signal values the autocorrelation value is calculated based on Equation 6. The pitch value is determined based on the highest value of XOR from this window. This method proved to be naïve and did not work well. It will yield better results if the autocorrelation is applied on signals which the human auditory system perceives. To do this, Lyon's cochlear model (available in the MATLAB Auditory Toolbox developed by Malcolm Slaney) was used (cobweb.ecn.purdue.edu/~malcolm/interval/1998-010/). The various parts of the human auditory system interact with each other and convert sound waves to neural firing (Lyon R. F., 1982). Lyon's auditory model describes the most important part of the human ear cochlea. The Auditory Toolbox contains *LyonPassiveEar* to implement this model. The output of this model is a vector proportional to the firing rate of neurons at each point in the cochlea. The output is stored as a two-dimensional array. The rows of this array represent one neuron's firing probability, and the columns represent the firing probability on the auditory nerves at one time, as described in the Auditory Toolbox guide. The cochlear output is a repetitive waveform, and we need to have a better representation of this waveform. A correlogram is an excellent way to summarize this periodicity. The Toolbox has a method called *CorrelogramArray* to compute the correlogram array. Pitch is computed from this correlogram array based on the autocorrelation method. This is also provided in the Auditory Toolbox as method *CorrelogramPitch* and is based on Equation 6.

Speech rate is calculated based upon the energy value of sound per each frame.

The logarithm of the energy is the value actually used and is computed by

$$E = \log \sum_{i=1}^{n} x_i^2 \quad (7)$$

Where N = number of frames
$X_i = i^t$ signal in the frame

RIT LIBRARY

Equation 7 is adopted from (Zeng, Z et al.,2004).

As discussed by Zeng et al. (2004) a frame is considered as speech if for any frame the log energy (E) is greater than the threshold value. This threshold value is derived based on training data.

3.6 Bayesian Network

A Bayesian network is a popular modeling tool to analyze complex problems. Understanding human emotion is a complex task which can help researchers in the area of human computer interaction. A Bayesian network is a valuable tool which involves probabilistic reasoning. Bayesian networks are very useful for predicting outcomes when a system has uncertain and/or incomplete information.

A Bayesian network is a probabilistic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph. A Bayesian network is useful when you have certain knowledge (prior probabilities) about your domain and you need to know uncertain values (posterior probabilities). It is based on Bayes' rule. Bayes' rule was named after Thomas Bayes who first provided the mathematical base for probability inference in "Essay Towards Solving a Problem in the Doctrine of Chances" (1763).

Bayes' rule is an expression of marginal probability and conditional probability, as shown in Equation 8. Conditional probability P(A|B) is the probability of some event A given the occurrences of some event B. Marginal probability is an unconditional probability P(A) of the event A regardless of whether event B occurred or not.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{8}$$

Where

- P(A) = prior or marginal probability of A.
- P(A|B) = posterior or conditional probability of A, given B.
- P(B|A) = posterior or conditional probability of B given A.
- P(B) = prior or marginal probability of B.

As a simple example, suppose we need to estimate the probability of person having an allergy provided he has a fever. Assume we do not have any particular information about this scenario but we do know the probability of a person having a fever given he has an allergy. We also know the prior independent probabilities of a person having a fever and a person having an allergy

RIT LIBRARY

based on his medical records. Based on this information, we can derive the probability of a person having an allergy given that he has a fever.

The posterior probability is –

Posterior probability = (likelihood * prior) / marginal likelihood

Definition: A Bayesian network consists of the following (Russel, R and Norvig, P., 1995):

- A set of variables and a set of directed edges between variables
- Each variable has a set of mutually exclusive states
- The variables and the directed edges form a directed acyclic graph (DAG). (A directed graph is acyclic if there is no directed path $A1 \rightarrow \ldots \rightarrow An$ such that $A1 = An$)
- To each variable A with parents B1,…,Bn, there is attached conditional probability table P(A|B1,…,Bn)

There are several advantages of using a Bayesian network for classification. The most important advantage of the Bayesian network is that it provides a probability framework, also, it is easy to interpret and unambiguous. A Bayesian network can also handle different data types like binary, discrete, and continuous. Another advantage is that it is easy to improve the Bayesian framework. One can always add more information in the network just by adding more nodes. The whole framework can be represented by a graph so that a human or a computer can easily understand the problem domain and dependencies. The disadvantage is that one needs to have perfect prior information of the problem domain. If the prior knowledge is wrong then it affects the whole network and the result becomes unreliable. It is also not sensible to build the Bayesian network for a very large application because as the number of nodes grows it becomes very expensive to calculate the posterior probability of the domain.

How to construct Bayesian network for emotional recognition:

Using Bayes' rule as show in Equation 8 we can calculate the posterior probability of specific facial expressions:

$$P(FE_{(i)} \mid AU_{(j)}) = P(AU_{(j)} \mid FE_{(j)}) \cdot P(FE_{(i)}) / P(AU_{(j)})$$

Where

$P(FE_{(i)} \mid AU_{(j)})$ = posterior probability of facial expression (FE) given an action unit (AU)

$P(AU_{(j)} \mid FE_{(j)})$ = likelihood of action unit given a facial expression

$P(FE_{(i)})$, $P(AU_{(j)})$ = prior probability of facial expression and action unit.

$FE_{(i)}$ = { Happy , Sad , Angry , Fear , Disgust }

RIT LIBRARY

In case of visual features :
$AU_{(j)}$ = { inner eyebrow movement , outer eyebrow movement , upper lip movement , lower lip movement , lip corner , cheek , nose }
In case of vocal features :
$AU_{(j)}$ = { Speech rate , Pitch average , Pitch changes }

Given the prior and posterior probabilities, Bayesian analysis can be used to determine an emotional state; for example, the 'happy state'. Based on the facial action coding system (FACS), there are two features that can determine the happy state. 1) lip corners are pulled 2) cheek is raised.

The basic task here is to compute the posterior probability of the emotion given some observed events like lip corners are pulled and/or cheeks are raised. In this case, P(Happy|lip_corner = true, cheek raised = true) needs to be calculated.

There are different algorithms to calculate the posterior probabilities. These algorithms are known as inference techniques.

Different inference techniques:

1) Inference by enumeration
2) Variable elimination algorithm

The following section discusses these inference techniques in more detail.

3.7 Bayesian Inference Techniques

For general cases, exact inference in a Bayesian network is an NP-Hard problem. But if the network is a singly connected polytree, as shown in Figure 22, then it is possible to calculate the exact inference in time linear in the number of the nodes ((Russell S. and Norvig P., 1995). For this thesis, both networks (visual and audio) are singly connected since any two nodes in the graph have a maximum of one connection between them; therefore, exact inference in polynomial time is possible.

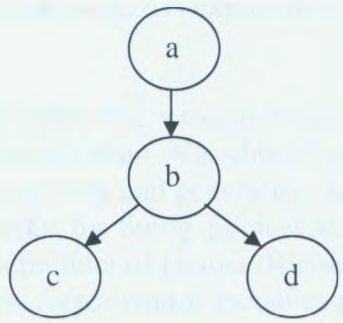


Figure 22. Singly Connected Poly Tree

RIT LIBRARY

According to Equation 9, a query P (X/e) to the network can be answered by,

$$P\left(\frac{X}{e}\right) = \alpha \sum_{Y} P(X, e, y) \tag{9}$$

where e = particular observed event
y = nonevidence variable
X = query variable
α = normalization constant needed to make the entries in P(Y|X) sum to 1.

It is known that the full joint distribution represents the complete Bayesian network (Russell S. and Norvig P., 1995), as shown in Equation 10:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i / parents(X_i)) \tag{10}$$

where $P(x_1, \ldots, x_n) = P(X_1 = x_1 \wedge .. \wedge X_n = x_n)$
parents(X_i) = specific values of the variables in Parents(X_i)

Equations 9 and 10 are adopted from (Russell, S and Norvig, P., 1995).

P(x,e,y) in the joint distribution can be written as products of conditional probabilities from the network. Therefore we can say that a query can be answered by computing sums of products of conditional probabilities from the network (Russell, S and Norvig, P.,1995).

To calculate the joint probability, the nodes must be ordered in the expression in such a way that if node X2 is dependent on another node X1, then X2 appears after X1 in the joint. Ordering is important so that connections between nodes make logical sense. This requires starting with the causes and then adding the events they cause, and then treating those events as causes etc. This significantly reduces the complexity of computation.

Inference by enumeration is an exact method which requires summing over the joint distribution. This joint distribution size is exponential in the number of nodes. This is an expensive way to do inference.

This thesis will apply the variable elimination inference method to the Bayesian networks used for emotion classification. Variable elimination is a form of dynamic programming. The main advantage is simplicity and generality. Another aspect is context specific independence. Variable elimination works by doing point-wise production of a pair of factors and summing out a variable from a product of factors (Russell, S and Norvig, P.,1995). The point-wise production f1 x f2 gives a new factor whose variables are the union of the variables in f1 and f2. Summing out a variable has a simple mathematical rule as xy+xz+xw = x(y+z+w). Any leaf node that is not a query variable or evidence variable would also be removed because they are extraneous and irrelevant to the query (Russell, S and Norvig, P.,1995). Variable elimination therefore removes

RIT LIBRARY

all these variables before processing the query. Variable elimination can help avoid duplicate computation which makes it more efficient than inference by enumeration.

Below are the basic steps to construct a Bayesian network:

1) Random variables (graph nodes) are defined to represent domain.
2) Variable values are evaluated.
3) Using the basic knowledge about the domain define the dependency between the variables. (Draw arcs between nodes in the graph).
4) Define joint probability distribution if the network is really small. The joint probability distribution has all the information about the domain, but as the network grows, the joint probability distribution tends to become large and so avoid calculating needless dependency between nodes. Find out conditional independence between nodes.
5) Use different inference techniques (inference by enumeration, variable elimination algorithm) to calculate posterior probabilities.

RIT LIBRARY

Chapter 4: Approach

This thesis uses bimodal information (visual and vocal) to improve upon existing emotion recognition systems. The basic approach is to feed facial feature movements and audio feature information to their respective Bayesian networks. Output from both classifiers is used to make the final classification. The following sections provide more information on how facial features and audio features are measured and evaluated.

4.1 Facial Expression Classification Using A Bayesian Network

Below are the set of variables and their values for the facial expression classification Bayesian network. The variables represent the action units defined in Facial Action Coding System (FACS). FACS is a popular method for estimating and understanding facial behaviors. It was developed by Paul Ekman and W.V. Friesen in the 1970s. They examined and determined how the contraction of each muscle alone or in combination with other facial muscles changes the appearance of the face. Action units are the measurement units of FACS.

1. Inner Brow movement
    a. Brow raised
    b. Brow lowered
    c. Unaffected
2. Outer Brow movement
    a. Brow raised
    b. Brow lowered
    c. Unaffected
3. Upper Lip movement
    a. Upper lip raised
    b. Unaffected
4. Lower Lip movement
    a. Lower lip lowered
    b. Unaffected
5. Lip corner
    a. Lip corner pulled
    b. Lip corner depressed
    c. Unaffected
6. Nose
    a. Nose wrinkle
    b. Unaffected
7. Cheek
    a. Cheek raised
    b. Unaffected

The domain knowledge is established using the action unit classifications as defined in the FACS

RIT LIBRARY

guide, as shown in Table 1 below. For example, if the lip corners are pulled and the cheek is raised then the probability of the person laughing is high, therefore that person is happy.

Table 1 shows action units associated with emotions as per FACS guide

	Fear	**Anger**	**Sadness**	**Happiness**	**Disgust**
Inner eyebrow movement	raised	lowered	raised	unaffected	unaffected
Outer eyebrow movement	raised	lowered	lowered	unaffected	unaffected
Upper lip movement	raised	unaffected	unaffected	unaffected	raised
Lower lip movement	lowered	unaffected	unaffected	unaffected	lowered
Lip corner	unaffected	depressed	depressed	pulled	unaffected
Cheek	unaffected	unaffected	unaffected	raised	unaffected
Nose	unaffected	unaffected	unaffected	unaffected	wrinkled

Table 1. Action Units associated with emotions as per FACS guide

These action units are measured by comparing key frame tracking points with the first frame (neutral frame). The position of tracking points of both the neutral frame and the key frame is stored in a structure. This structure has an x and a y position for each point from the left and right eyebrows and the upper and lower lips.

Table 2 shows how action unit movements were determined. For example, when the eyebrow is raised, the tracking points on the eyebrow will have a vertical movement when compared to the neutral frame, and hence the y position will be changed. Threshold values are in pixels, and were derived based on the assumption that heads are not moving or tilted during shot.

RIT LIBRARY

Facial Feature	Tracked Point	Threshold	Result
lower lip	y-position	>2	lowered
lip corner	x-position	>4	pulled
lip corner	x-position	<2	depressed
upper Lip	y-position	<2	raised
inner Eyebrow	y-position	>2	lowered
inner Eyebrow	y-position	<2	raised
outer Eyebrow	y-position	>2	lowered
outer Eyebrow	y-position	<2	raised
nose	Number of white pixels	>5	wrinkled
cheek	Number of white pixels	>3	raised

Table 2. Facial feature action unit measurements

Threshold values are determined based on training data. A total 12 DaFEx emotional and 5 neutral videos were chosen. For each emotion video facial feature, movements were measured in pixels. 95% of total emotion videos showed that movements are within this threshold range. All neutral videos showed that movements do not meet threshold requirement. Count of white pixels from edge detected image area of nose is also stored and compared against neutral frame. Probability of wrinkle is high if count of white pixels in nose area is greater than then the count of white pixels from edge detected image area of nose of neutral frame. In order to consider nose wrinkle value true the count difference should be higher than 5. Similarly for cheek area the count of white pixels should be higher than 3.

Conditional probability values for each facial feature are described below. It uses the convention that false=1 and true=2.

Conditional probability values are derived based on 20 training videos. For each training video feature motion is measured and probability is calculated manually. For example if all happy emotion videos have lip corner pulled feature as true then the probability of lip corner being pulled for happy emotion is 1. If 4 videos out 5 disgust emotion videos has wrinkle feature has true then probability of having nose wrinkle for disgust emotion is 0.8.

RIT LIBRARY

Table 3 shows conditional probability table (CPT) for lip corner (LC).

Emotion	Probability of LC being pulled	Probability of LC being depressed	Probability of LC being unaffected
Angry=True	0.15	0.75	0.1
Angry=False	0.85	0.25	0.9
Sad=True	0	0.75	0.25
Sad=False	0	0.25	0.75
Happy=True	0.9	0.1	0
Happy=False	0.1	0.9	0

Table 3. CPT for lip corner

Table 4 shows conditional probability table for cheek.

Emotion	Probability of Cheek being raised	Probability of Cheek being unaffected
Happy=False	0.35	0.65
Happy=True	0.65	0.35

Table 4. CPT for cheek

Table 5 shows conditional probability table for nose.

Emotion	Probability of nose having wrinkled	Probability of nose having no wrinkles
Disgust=False	0.25	0.75
Disgust=True	0.75	0.25

Table 5. CPT for nose

Table 6 shows conditional probability table for lower lip.

Emotion	Probability of lower lip lowered	Probability of lower lip unaffected
Fear=False	0.15	0.85
Fear=True	0.85	0.15
Disgust=False	0.25	0.75
Disgust=True	0.75	0.25

Table 6. CPT for lower lip

RIT LIBRARY

Table 7 shows conditional probability table for upper lip.

Emotion	Probability of upper lip raised	Probability of upper lip unaffected
Fear=False	0.15	0.85
Fear=True	0.85	0.15
Disgust=False	0.25	0.75
Disgust=True	0.75	0.25

Table 7. CPT for upper lip

Table 8 shows conditional probability table for inner eyebrow (IE).

Emotion	Probability of IE being raised	Probability of IE being lowered	Probability of IE being unaffected
Angry=True	0.2	0.7	0.1
Angry=False	0.8	0.3	0.9
Fear=True	0.8	0.1	0.1
Fear=False	0.2	0.9	0.9
Sad=True	0.7	0.2	0.1
Sad=False	0.3	0.8	0.9

Table 8. CPT for inner eyebrow

Table 9 shows conditional probability table for outer eyebrow(OE).

Sad	Probability of OE being raised	Probability of OE being lowered	Probability of OE being unaffected
Angry=True	0.3	0.7	0
Angry=False	0.7	0.3	0
Fear=True	0.7	0.3	0
Fear=False	0.3	0.7	0
Sad=True	0	0.6	0.4
Sad=False	0	0.4	0.6

Table 9. CPT for outer eyebrow

RIT LIBRARY

Figure 23 defines the Bayesian network for facial expression classification:

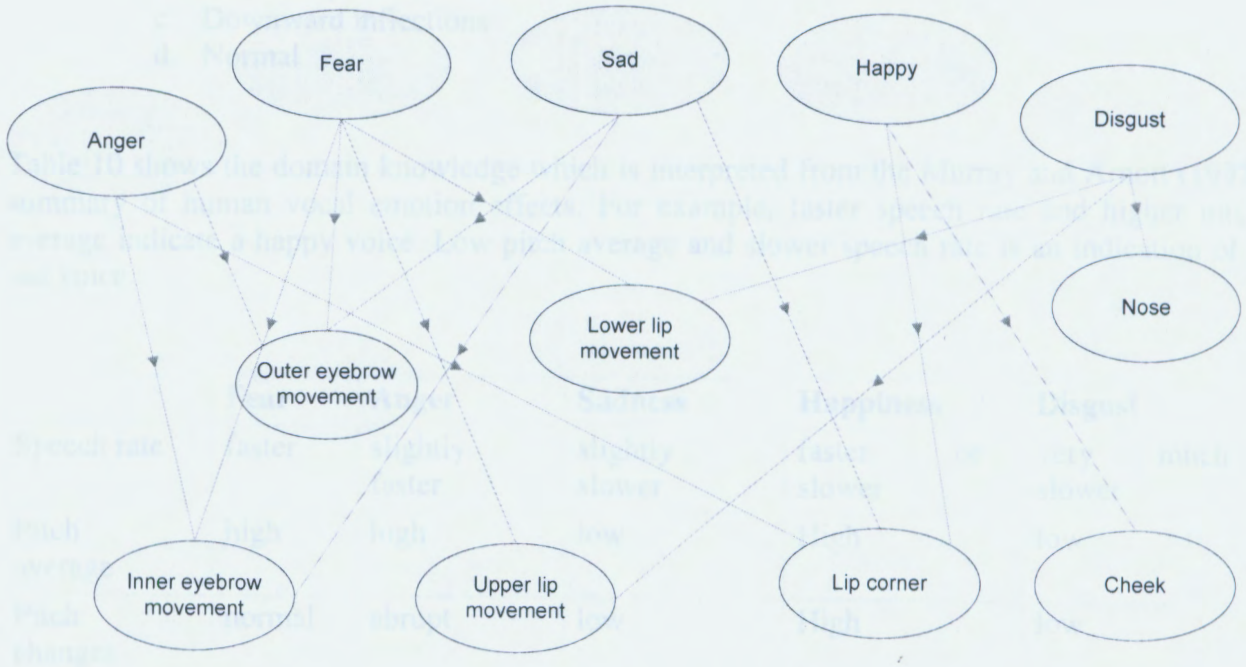


Figure 23. Bayesian network for facial expression classification

This thesis used the MATLAB Bayesian network toolbox which was written by Kevin Murphy (www.cs.ubc.ca/~murphyk/Software/BNT/bnt.html). This is an excellent tool to implement a probabilistic model. Also, it is very useful for research and rapid prototyping since the source code is very well documented.

4.2 Voice Classification Using A Bayesian Network

Below are the set of variables and their values for the speech emotion classification Bayesian network. Murray and Arnott (1993) have described human vocal emotion effects. The random variables in the Bayesian network are chosen based on their classification and are relative to neutral speech.

1. Speech Rate
    a. Slightly faster
    b. Much faster
    c. Very much slower
    d. Slightly slower

RIT LIBRARY

2. Pitch average
    a. High
    b. Low

3. Pitch changes
    a. Abrupt
    b. Smooth upward inflections
    c. Downward inflections
    d. Normal

Table 10 shows the domain knowledge which is interpreted from the Murray and Arnott (1993) summary of human vocal emotion effects. For example, faster speech rate and higher pitch average indicate a happy voice. Low pitch average and slower speech rate is an indication of a sad voice.

	Fear	**Anger**	**Sadness**	**Happiness**	**Disgust**
Speech rate	faster	slightly faster	slightly slower	faster or slower	very much slower
Pitch average	high	high	low	High	low
Pitch changes	normal	abrupt	low	High	low

Table 10. Summary of human vocal effects most commonly associated with the emotions by Murray and Arnott (1993).

RIT LIBRARY

The Bayesian network representation used for speech recognition is shown in Figure 24.

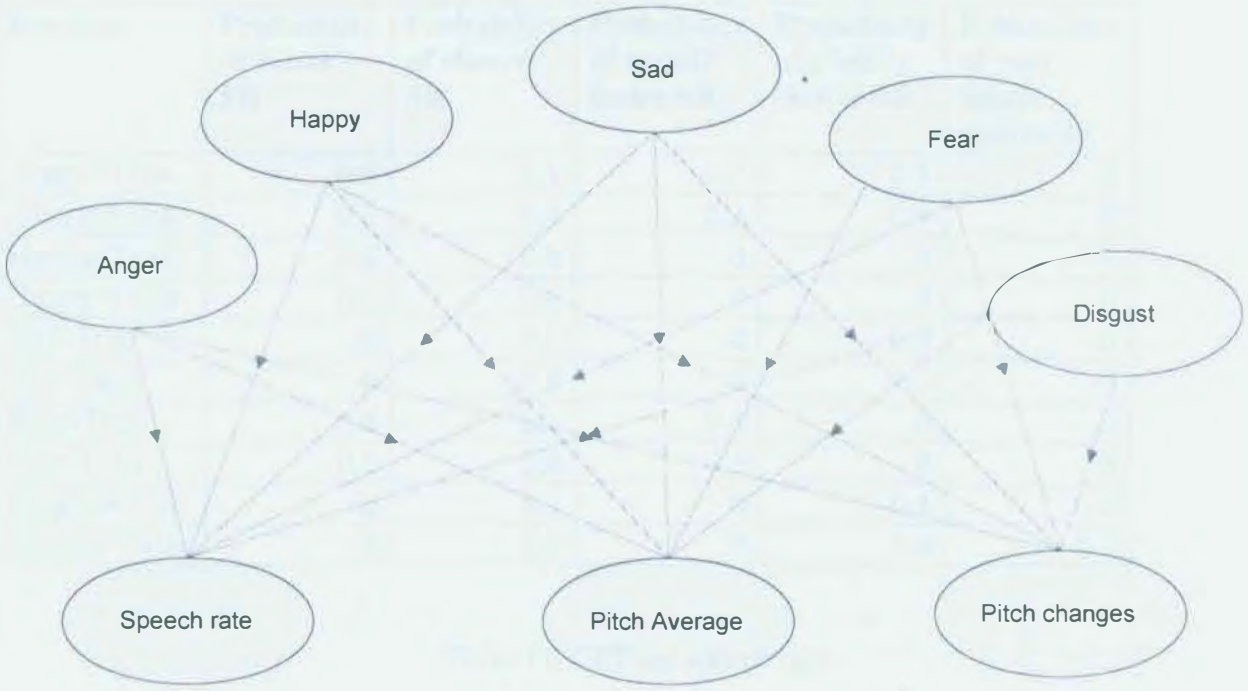


Figure 24. Bayesian network for speech expression classification

Using this network we can compute the posterior probability of features nodes based on the vocal effects, and classify a sample facial expression according to which has the highest probability.

Conditional probability of each speech feature is calculated based on 15 training videos. Each speech feature value is calculated based upon the method described in above section. Probability is calculated based on each speech feature value and which emotion it belongs to. For example, if 4 out of 5 sad emotion video is having pitch average value as low then probability of pitch average feature for sad emotion is 0.8.

RIT LIBRARY

Table 11 shows conditional probability table (CPT) for speech rate (SR).

Emotion	Probability of faster SR	Probability of slower SR	Probability of slightly faster SR	Probability of slightly slower SR	Probability of very much slower SR
Angry=True	0.1	0.1	0.7	0.1	0
Angry=False	0.9	0.9	0.3	0.9	0
Happy=True	0.5	0.5	0	0	0
Happy=False	0.5	0.5	0	0	0
Sad=True	0	0.2	0	0.8	0
Sad=False	0	0.8	0	0.2	0
Fear=True	0.8	0.1	0.1	0	0
Fear=False	0.2	0.9	0.9	0	0
Disgust=True	0	0.1	0	0.1	0.8
Disgust=False	0	0.9	0	0.9	0.2

Table 11. CPT for speech rate

Table 12 shows conditional probability table for pitch average (PA).

Emotion	Probability of PA being high	Probability of PA being low
Angry=True	0.9	0.1
Angry=False	0.1	0.9
Happy=True	0.9	0.1
Happy=False	0.1	0.9
Sad=True	0.1	0.9
Sad=False	0.9	0.1
Fear=True	0.9	0.1
Fear=False	0.1	0.9
Disgust=True	0.1	0.9
Disgust=False	0.9	0.1

Table 12. CPT for pitch average

RIT LIBRARY

Table 13 shows conditional probability table for pitch changes (PC).

Emotion	Probability of PC being high	Probability of PC being low	Probability of PC being abrupt
Angry=True	0	0	1
Angry=False	0.5	0.5	0
Happy=True	0.9	0.1	0
Happy=False	0.1	0.9	0
Sad=True	0.1	0.9	0
Sad=False	0.9	0.1	0
Disgust=True	0.1	0.9	0
Disgust=False	0.9	0.1	0

Table 13. CPT for pitch changes

The DaFEx database contains neutral (non-emotion) videos as well. Audio data of these neutral videos are used as a baseline to measure the sound feature values. Each person is different and can have different pitch and speech rate values even for a neutral emotion. These feature values are pre-calculated for each actor and supplied while measuring audio features of emotional videos. Pitch values are quantified using a statistical analysis. Statistical parameters such as mean, variance and standard deviation are calculated from the pitch vector. By comparing the standard deviation and variance of pitch values of emotional audio to the neutral base line value we can determine whether a person's pitch changes are high or low. The mean value of an emotional pitch is compared to the mean value of a neutral audio. Similarly, the speech rate values are also compared to neutral.

The pitch average can be high or low. If the mean value of an emotional pitch is higher than the mean value of a neutral pitch of the same actor, then the pitch average feature is marked as high. If the mean value of an emotional pitch is lower than the mean value of a neutral pitch of the same actor, then the pitch average feature is marked as low. The pitch value is computed based on the method described in section 3.5.

Pitch changes are measured by calculating a standard deviation. Standard deviation shows variability and it could indicate a pitch of high, low or abrupt. Abrupt is defined as a standard deviation of pitch values changing from high to low rapidly. If the standard deviation of pitch values of an emotional video is greater than the standard deviation of pitch values of a neutral video of the same actor, then pitch change is marked as high. If the standard deviation of pitch values of an emotion video is less than the standard deviation of pitch values of a neutral video of the same actor, then pitch change is marked as low.

Speech rate can be fast, slightly fast, slow, slightly slow, and very slow. If the difference between the speech rate of an emotional video and a neutral video is greater than 5% of the

RIT LIBRARY

speech rate of a neutral video but less than 20% of the speech rate of a neutral video, then speech rate is marked as slightly fast. If the difference between the speech rate of an emotional video and a neutral video is greater than 20% of the speech rate of a neutral video, then speech rate is marked as faster.

If the difference between speech rate of neutral video and emotional video is less than 20% of speech rate value of neutral video then speech rate is marked as slower. If the difference between speech rate of neutral video and emotional video is less than 30% speech rate value of neutral video then speech rate is marked as very much slower. These threshold values are determined based on doing analysis on training videos where speech rate values were derived and compared to determine the threshold value.

Table 14 shows how vocal feature action units were measured.

Vocal Feature	**Threshold Criteria**	**Result**
Speech Rate	>20%	Fast
Speech Rate	Between 5% to 20%	Slightly fast
Speech Rate	< neutral video value	Slow
Speech Rate	Between 20% to 30%	Slightly slow
Speech Rate	< 30%	Very slow
Pitch Average	Higher than neutral video value	High
Pitch Average	Lower than neutral video value	Low
Pitch Changes	Higher than neutral video value	High
Pitch Changes	Lower than neutral video value	Low

Table 14. Vocal feature action unit measurement

4.3 Evaluation

This thesis has used the DaFEx (Database of Human Facial Expressions) database (http://i3.fbk.eu/en/dafex). DaFEx is composed of 10

08 short videos containing emotional facial expressions of the six Ekman emotions (fear, anger, sad, happy, disgust, and surprise) and the neutral expression. Facial expressions were recorded by 8 Italian professional actors (4 male and 4 female) on 3 intensity levels (low, medium, high)

RIT LIBRARY

and in utterance and non utterance condition. This thesis does not use the non-utterance videos, the low and medium intensity videos, or the surprise videos.

Below are the main characteristics of DaFEx videos:

1) Videos have all frontal views and only frontal view faces are analyzed.
2) Videos have one person at a time showing emotion in the video.
3) Videos are taken under normal lighting condition.
4) All are indoor shots.
5) While taking video, position the subject so that only face and neck is seen.
6) Video format is .avi.
7) Videos are already labeled by having 80 subjects classify the emotion expressed in the videos
8) There are 25 frames per second captured in the videos.
9) Image resolution is 360 x 288.

Probability values from both classifications (facial and vocal) were derived as shown above. The DaFEx video database ranged in gender and age are tested. Results are assessed by counting the true positive, false positive and false negative classification. Ground truth is provided by the author of the DAFEX database. Ground truth data is stored for the verification purposes. Actual results are compared to the ground truth already labeled in the database.

A confusion matrix is a helpful visualization tool which can show the true performance of the classifier. The confusion matrices will have results from the 1) facial feature classifier, 2) vocal classifier and 3) combined visual and vocal classifier.

An ROC curve is a graphical representation of true positives vs. false positives. Each visual and vocal classification result and the combined visual and vocal classification result is plotted as an ROC curve and analyzed.

RIT LIBRARY

Chapter 5: Results and Discussion

5.1 Overall Results of the Three Types of Classifiers

This thesis found that emotion recognition through the use of vocal and facial features depends upon the type of classifier used (facial, vocal, or both). From Table 15 we can see that happy and disgust emotions are classified better with just facial features, while sad, angry and fear emotions are classified better with vocal features.

Percent Correct Classification of Emotions by Classifier Type

Classifier	Happy	Sad	Disgust	Angry	Fear
Facial Feature	**80%**	**60%**	**70%**	**50%**	**60%**
Vocal Feature	**70%**	**85%**	**40%**	**70%**	**70%**

Table 15. Summary of classification results

Table 15 shows summary of classification results by each classifier. Happy and disgust emotion was detected at least 10% better by facial classifier while sad, fear and angry emotion was detected at least 10% better by vocal classifier. This shows it is not possible to detect all the emotions using just one modality and additional semantic information can help improve classification of certain emotions.

RIT LIBRARY

Table 16 shows classification results by facial feature classifier.

Actual expression (Ground truth)

Classified As	Happy	Sad	Disgust	Angry	Fear
Happy	80%	-	3%	20%	14%
Sad	-	60%	7%	25%	17%
Disgust	5%	5%	70%	-	-
Angry	10%	30%	10%	50%	28%
Fear	5%	5%	10%	5%	60%

Table 16. Confusion matrix of facial feature classification results

A Bayesian network is an optimal classifier provided it is properly trained and modeled. The happy emotion is detected when the lip corner is pulled and the cheek is raised. Lip corner movement detection and cheek detection rate is 80% and for all these cases, the happy emotion was detected. It did not work when the nose region overlapped with the cheek region, and so the action unit was classified as nose wrinkle. Hence, the emotion was misclassified as disgust instead of happy. Angry was also misclassified due to subtle cheek movements and the fact that lip corners were not identified correctly on the key frame.

The sad emotion is based on eyebrow movements and lip corner depressed action units. The sad and anger emotions have very similar visual features, where the only difference is with the inner eyebrow movement. In the case of sadness, the inner eyebrow is raised sometimes while in the case of anger, the inner eyebrow is lowered. So if inner eyebrow movements are not accurately identified, then the classifier can get confused between these two emotions.

Disgust emotion has very unique facial feature – nose wrinkles. No other emotion is supposed to exhibit this action unit. Nose wrinkle detection rate is 80% but in some cases this expression lasted briefly, only for 2-3 frames. It can go undetected if the key frame is not captured during this brief expression.

The facial feature classifier has an average performance of 50% and 60% for the angry and fear emotions respectively. Their detection is based mostly on eyebrow movements. In some cases actors did not display any eyebrow movements at all and some cases movements were pretty subtle to track. Also threshold values are determined based on the assumption that there should not be any head movement. In some cases heads were tilted and it can interrupt with tracking facial feature points. Some cases eyebrow tracking points were not identified accurately due to hair falling off over forehead and eyebrows.

This raises a need to incorporate more features and more advanced ways to identify eyebrow

RIT LIBRARY

movements. One way to improve the model would be to create a 3D wire frame model that captures the possible motions of facial muscles (Dornaika F. and Davoine F., 2008).

Actual expression (Ground truth)

Classified As	Happy	Sad	Disgust	Angry	Fear
Happy	70%	-	10%	15%	5%
Sad	-	85%	28%	-	5%
Disgust	10%	5%	40%	10%	15%
Angry	10%	5%	10%	70%	5%
Fear	10%	5%	12%	5%	70%

Table 17. Confusion matrix of vocal features classification results

The vocal classifier detection rate for the sad emotion is 85%. This emotion has low energy and slower speech rate. Both features were detected 90% of the time successfully for this emotion. Sad and disgust both has similar pitch features which is why the classifier confused these emotions with each other.

The vocal classifier detection rate for the happy, angry and fear emotion is 70%. The happy emotion was confused with angry because both have similar speech rate and pitch average.

The detection rate for the disgust emotion by the vocal classifier is just 40%. Disgust is a complex emotion and its vocal feature measurements are similar to the sad emotion. The only difference is that disgust has speech rate lower than sad. Even just by hearing disgust speech without knowing the language it was hard to conclude this emotion. This suggests the need to derive linguist information as well from the speech. It may be desirable to tag certain words with certain emotions; for example, 'sorry' with sad or 'wow' with happy. This is another research area where future work may enhance the performance of the classifiers. The more features computers can extract, the better the emotion recognition would likely be. In the current scope of this research we can conclude that key features like speech rate and pitch values can derive emotions like sad. Adding more features may help recognize certain complex emotions such as disgust.

The disgust emotion is sometimes misclassified as sadness because both of them have low pitch feature. The angry emotion is also detected by the vocal classifier but also sometimes misclassified as happy. They are mutually confused because there are some acoustic similarities between happy and angry. They have higher pitch rate and energy values. Fear was also misclassified as angry sometimes since both of them share similar speech rate and pitch.

RIT LIBRARY

Actual expression (Ground truth)

Classified As	Happy	Sad	Disgust	Angry	Fear
Happy	75%	-	6.5%	17.5%	9.5%
Sad	-	72.5%	17.5%	25%	11%
Disgust	7.5%	5%	55%	10%	15%
Angry	10%	17.5%	10%	60%	16.5%
Fear	7.5%	5%	11%	5%	65%

Table 18. Confusion matrix of facial feature and vocal features combined

ROC curves below illustrate the results obtained for the classifiers used separately, and then combined. The ROC curve shows the false positive rate (FPR) on X axis and true positive rate (TPR) on Y axis. The closer the curve follows the Y axis the more accurate the test since it means we have more true positive tests than false positives. The area under the curve also provides valuable information about the classifier. Classifier accuracy is higher if area under ROC curve is close to 1. Table 19 shows comparison between AUC values of difference classifier.

Figure 25 shows ROC curve for visual classifier. Area under ROC curve (AUC) for visual classifier is 0.77.

Figure 25. ROC Curve for visual classifier

RIT LIBRARY

Figure 26 shows ROC curve for vocal classifier. Area under curve is 0.80.

Figure 26. ROC curve for the vocal classifier

	Facial Feature Classifier	Vocal Feature Classifier
AUC Value	0.77	0.80

Table 19. Area under ROC curve values per Classifier

By comparing the AUC values of the three classifiers we can conclude that bi-modal information provides more true positives than false positives, hence the overall accuracy is better with combined information.

Overall, the results of both the classifier are quite similar however the recognition rate of some of the emotion is higher than the other. For example, sadness is better recognized by the vocal classifier while happiness is better recognized by the visual classifier. The combined results of both classifiers together seem promising because it also decreases true negative results of individual classifier.

5.2 Comparing the Results Using Histogram Differences to Edge Differences

Key frame detection was initially based on histogram differences. This did not work well as explained in Chapter 3.2.

Percent Correct Classification of Emotions by Classifier Type

Classifier	Happy	Sad	Disgust	Angry	Fear
Facial Feature	80%	60%	70%	50%	60%
Vocal Feature	70%	85%	40%	70%	70%

Table 20. Summary of classification results based on Edged difference

Percent Correct Classification of Emotions by Classifier Type

Classifier	Happy	Sad	Disgust	Angry	Fear
Facial Alone	68%	40%	60%	38%	41%
Vocal Alone	50%	65%	30%	35%	48%

Table 21. Summary of classification results based on Histogram difference

Comparison between Table 20 and Table 21 shows great improvement in overall classification due to the fact that correct key frame was identified. Classification rate is increased by 20% at least. This is a very promising result and shows that idea of detecting edge changes rather than histogram changes worked well and can provide reliable semantic information about changes in the videos.

5.3 Comparing the Results to Existing Research

To the best of my knowledge there is no emotion recognition research has been presented on DaFEx database. Dragos Datcu and Leon J.M. Rothkrantz have done similar research (Datcu, D. and Rothkrantz, L, 2008) on combining audio and facial feature information to derive better

classification results using the Cohn-Kanada database. Their facial expression method uses Active Appearance Model (AAM) for the extraction of face shape, and Support Vector Machines (SVM) for the classification of feature patterns in one of the prototypic facial expressions. They also used the Praat (Boersma and Weenink, 2005) tool for extracting the features from each sample from all generated data sets. Similar to this thesis work, classification is done using a dynamic Bayesian network. They also considered the surprised emotion along with happy, sad, anger, disgust and fear. Their classification results are shown in Table 22.

Percent Correct Classification of Emotions by Classifier Type

Classifier	Happy	Sad	Disgust	Angry	Fear
Facial Alone	**72.64%**	**82.79%**	**80.35%**	**75.86%**	**84.70%**
Vocal Alone	**60.87%**	**64.71%**	**62.50%**	**60%**	**66.67%**

Table 22. Summary of classification results mentioned in Datcu and Rothkrantz (2008)

Percent Correct Classification of Emotions by Classifier Type

Classifier	Happy	Sad	Disgust	Angry	Fear
Facial Feature	**80%**	**60%**	**70%**	**50%**	**60%**
Vocal Feature	**70%**	**85%**	**40%**	**70%**	**70%**

Table 23. Summary of classification results for this thesis

By comparison with Table 23 it is evident that this thesis work shows a higher classification accuracy for the happy emotion, and similar accuracy for the sad emotion. Datcu and Rothkrantz shows higher classification accuracy for disgust, anger, and fear. The main difference between Datcu and Rothkrantz is that their input to system is directly a shot so they do not need shot detection algorithm. They extract facial features based on Active Appearance Model (AAM) (Cootes et al., 1998) and extraction is done on each and every frame instead of key frame. They are also using face tracking algorithm to detect possible move or inclination of the face. This method worked well and their classification results by facial classifier are better than facial classifier results by this thesis. Although vocal classifier alone detected sad and happy emotion almost 15% better than Datcu and Rothkrantz research. This thesis is using Auditory toolbox model which is based on how human perceive pitch changes which proved to be very useful to classify differences between basic emotions such as happy and sad.

RIT LIBRARY

Chapter 6: Conclusion and Future Work

This research contributes to current knowledge by providing a novel algorithm that uses edge differences for extracting key frames and facial features for semantic processing of human emotions in videos. This thesis also tested the hypothesis that comparison of the two modalities of vision and speech will yield a better classification result than either used alone. It analyzed the strengths and weaknesses of a visual and vocal emotion classifier. As discussed, classification results provide some evidence that the certain emotion can be classifier better with particular classifier. Comparison of classifiers provides more information about emotion recognition event. This information can result in better semantic analysis of the human emotion recognition videos. While testing this hypothesis, this thesis also implemented edge difference tracking as a novel way of finding a key frame, and effectively used image processing techniques to derive facial features. It also explored audio features statically. This research shows potential for improved human computer interfaces that might automatically understanding human response.

This thesis analyzed five emotions: happy, angry, sad, fear, disgust. An extension of this thesis by others could consider additional emotions such as neutral or surprise. By detecting a neutral emotion, the system could determine if a person is in an emotional state or not. Additional facial features such as a wink or a blink could also be extracted to perform facial feature classification. For example, closing an eye for some time might indicate sleepiness or tiredness. This research has considered only facial and vocal features for human emotion analysis. Even though facial and vocal features provide substantial information to detect emotions, further study can also investigate hand movements combined with the facial gestures. More vocal cues like intensity and articulation can also be extracted to perform voice classification. Linguistic information may also greatly improve vocal classification. Identifying certain words from speech can help derive more information about the emotion. This thesis detects emotions on frontal faces only. Future study can also take profile faces into consideration. Also, this study assumes that there is only one person at a time during each shot. Future research work might consider multiple persons showing different emotions.

PITTSBURGH

Appendix A. Source code

```
%This function takes video as an input file and provides output of total
%frames,keyframe and shot boundary frame
function [shotB,totalFrames,keyframe]=shotDetection(inputFile)

shotB = 0;
video = aviread(inputFile);
totalFrames = size(video,2);

for count=1:2:totalFrames
   fileName = strcat(num2str(count),'.bmp');
   imwrite(video(count).cdata,fileName,'bmp');
end

HistList = histogramValues(totalFrames);
HistShot = shot(HistList,5,11);
EdgeList = EdgeValues(totalFrames);
EdgeShot = shot(EdgeList,5,11);

if (size(HistShot) == 0 & size(EdgeShot) == 0 )
   [valE , iE ] = max(EdgeList(:,2)); %only single shot so return with keyframe
   keyframe = EdgeList(iE,1);
   return;
end

shotBH = HistShot(1);
shotBE = EdgeShot(1);

if (shotBH >= shotBE )
   shotB = shotBE-1;
end
if (shotBH <= shotBE)
   shotB = shotBH-1;
end


%Get histogram dissimilarity measure
function [histvalue] = histogramValues(frame_num)

histvalue = [];
```

```
fileName = strcat(num2str(1),'.bmp');
firstframe = imread(fileName);
firstfrm = rgb2gray(firstframe);
[f, f1] = imhist(firstfrm);

for i = 1:2:frame_num
        fileName = strcat(num2str(i),'.bmp');
        nextframe = imread(fileName);
        nextfrm = rgb2gray(nextframe);
        [f2, fn] = imhist(nextfrm);
   diffvalue = (f2-f).*(f2-f);
   diffvalue = sum(diffvalue);
   sqrtvalue = sqrt(diffvalue);
   histvalue = [histvalue; i sqrtvalue];
   f = f2;
end

%

% Get edge dissimilarity measure

function [edgevalue] = EdgeValues(frame_num)

edgevalue = [];

fileName = strcat(num2str(1),'.bmp');
firstframe = imread(fileName);
firstfrm = rgb2gray(firstframe);
firstfrmbw = edge(firstfrm, 'canny');

for i = 1:2:frame_num
   sizef = size(find(firstfrmbw),1);
        nextfrm = strcat(num2str(i),'.bmp');
        nf = imread(nextfrm);
        nfd = rgb2gray(nf);
        nextfrmbw = edge(nfd, 'canny');
   sizen = size(find(nextfrmbw),1);
        nextfrmbwInv = 1 - nextfrmbw;
   firstfrmbwInv = 1 - firstfrmbw;
   se = strel('square',3);
   df = imdilate(firstfrmbw, se);
   dn = imdilate(nextfrmbw, se);
   edgein = df & nextfrmbwInv;
   edgeout = dn & firstfrmbwInv;
   edgeRationIn = size(find(edgein),1)/sizen;
   edgeRatioOut = size(find(edgeout),1)/sizef;
```

```
    edgeratio = max(edgeRatioIn, edgeRatioOut);
    edgevalue = [edgevalue; i edgeratio];
    firstfrmbw = nextfrmbw;
end
return;
%This function is based on adaptive thresholding method

function [shotBoundary] = shot(list,t,w)
shotBoundary = [];
i = 1;
while i<size(list,1)
    m = i;
    l = m - w;
    if l < 1
        l = 1;
    end
    r = i+w;
    if r>size(list,1)
        r = size(list,1);
    end
    maxwin = max(list(l:r,2));
    if  list(m,2) < maxwin
        pos = find(list(:,2)==maxwin);
        if i<pos(1)
            i = pos(1);
        else
            i = i+1;
        end
        continue;
    end

    lmean = mean(list(l:m-1,2));
    lstd = std(list(l:m-1,2));
    rmean = mean(list(m+1:r,2));
    rstd = std(list(m+1:r,2));
    if list(m,2)>max(lmean+Td*lstd, rmean+Td*rstd)
        shotBoundary = [shotBoundary;list(m,:)];
    end
    i = r+1;
end

return;

%
function [keyframe] = keyframe (shotB,totalFrames)
```

```
EdgeList = EdgeValues(shotB);

[valE , iE ] = max(EdgeList(:,2));

keyframe = EdgeList(iE,1);


%

function [facelip,facenose,faceRighteye, faceLefteye , leftCheek , rightCheek] =
Face_Detection(frame)
original = frame;
face_original = original;

for j=1:size(original,1)%Read row pixels
   for i=1:size(original,2)%Read column pixels

    if (original(j,i,1)>160 && original(j,i,2)>120 && original(j,i,3)>88)
        face_original(j,i,1)=255;
        face_original(j,i,2)=255;
        face_original(j,i,3)=255;

      else
        face_original(j,i,1)=0;
        face_original(j,i,2)=0;
        face_original(j,i,3)=0;

      end
   end
end

c = rgb2gray(face_original);
L = medfilt2(c, [ 3 3 ]);
figure, imshow(L)
[lable,n] = bwlabel(L,8);
   region = regionprops(lable,'all');

    for i=1:size(region,1)

      % consider only those objects which are of suffiecient size
      if(region(i).Area > 4000 && region(i).MajorAxisLength > 35 &&
region(i).MinorAxisLength > 25)

        % maintain a location
        rect = region(i).BoundingBox;
```

```
        % rectangle('curvature',[0 0],'position',rect,'edgecolor',[0 1 0],'LineWidth',[1]);
         % figure,imshow(region(i).Image);
          [facelip,facenose,faceRighteye, faceLefteye , leftCheek , rightCheek ] =
locateFacialFeature(region(i),original,rect);

    end
  end


%
%
function [lipRect,noseRect,RightEyeRect, leftEyeRect , leftCheekRect , rightCheekRect] =
locateFacialFeature(face,original,faceLoc)


rect = face.BoundingBox;

height = face.MajorAxisLength;
width = face.MinorAxisLength;

if ( (height/width) > 1.7 )
  rect(4) = rect(4)*(3/4);
end

rectOriginal = rect;

%figure, imshow(original)

rectangle('curvature',[0 0],'position',rectOriginal,'edgecolor',[0 1 0],'LineWidth',[1]);

cmin = floor(rect(1));
cmax = floor((rect(1)+rect(3)));
rmin = floor(rect(2));
rmax = floor((rect(2)+rect(4)));

rect(4) = rectOriginal(4)/2;

edgeIm = edge(rgb2gray(original),'canny');

%7/7
%
%binIm = im2bw(face.Image);
%binImage = medfilt2(binIm, [ 3 3 ]);
imCropped = imcrop(original,rectOriginal);
binIm = im2bw(imCropped);
figure, imshow(binIm);
```

```
upperHalfFace = imcrop(edgeIm,rect);

  %lower half portion
   lowerrect(2) = ((rectOriginal(4)/2) + rectOriginal(2));
   lowerrect(1) = rectOriginal(1);
   lowerrect(3) = rectOriginal(3);
   lowerrect(4) = (rectOriginal(4)/2);

lowerHalfFace = imcrop(edgeIm,lowerrect);

lowerHalfFaceRGB = imcrop(original,lowerrect);

% upper left half portion
rect(1) = rectOriginal(1);
rect(2) = rectOriginal(2);
rect(4) = rectOriginal(4)/2;
rect(3) = rectOriginal(3)/2;

upperLeftFace = imcrop(edgeIm,rect);

upperLeftFaceRGB  = imcrop(original,rect);
% upper right half portion

uRrect(1) = rectOriginal(1) + (rectOriginal(3)/2)
uRrect(2) = rect(2);
uRrect(3) = rectOriginal(3)/2;
uRrect(4) = rect(4);

upperRightFace = imcrop(edgeIm,uRrect);

% figure ,imshow( upperHalfFace ), title('upperHalfFace')
% figure ,imshow( lowerHalfFace ), title('lowerHalfFace')
% figure ,imshow( upperLeftFace ), title('upperLeftFace')
% figure ,imshow( upperRightFace ), title('upperRightFace')

%7/7find binary area for all face regions

uLeftBinFace = im2bw(imcrop(original,rect));
uRightBinFace = im2bw(imcrop(original,uRrect));
lowerBinFace = im2bw(imcrop(original,lowerrect));

% horizontal projection for left eyebrow

 uLeftBinFace = 1 - uLeftBinFace;
 uRightBinFace = 1 - uRightBinFace;
 lowerBinFace = 1 - lowerBinFace;
```

```
% ------------------------
%imview(uLeftBinFace);
%
%
[uLrow , uLcol] = size(uLeftBinFace);
sumRow = zeros(uLrow,1)

%skip forehead area

for j=uint8(uLrow/4):uLrow
   sumRow(j) = sum(uLeftBinFace(j,:));
end

[val , pos ] = max(sumRow)

%pos = pos+uint8(uLrow/4);

for j=uLcol:-1:5
   if(uLeftBinFace(pos,j)==1)
    lEyebrowx=j
    lEyebrowy=pos
    break;
   end
end

leftEyebrowLen = uint8(uLcol/2)

leftEyeRectx = j-leftEyebrowLen
leftEyeRecty = pos - 10
leftEyeRectw = leftEyebrowLen+1
leftEyeRecth = 20


% first get the eye rect of the upperLeft face
leftEyeRect = [leftEyeRectx,leftEyeRecty,leftEyeRectw,leftEyeRecth]

rectangle('curvature',[0 0],'position',leftEyeRect,'edgecolor',[0 1 0],'LineWidth',[1]);

%=============================

% eye rect of the upperRight face

[uRrow , uRcol] = size(uRightBinFace);
sumRow = zeros(uRrow,1)
```

```
%skip forehead area

for j=uint8(uRrow/4):uRrow
   sumRow(j) = sum(uRightBinFace(j,:));
end

[val , pos ] = max(sumRow)

%pos = pos+uint8(uLrow/4);

for j=1:1:uRcol
   if(uRightBinFace(pos,j)==1)
     REyebrowx=j
     REyebrowy=pos
     break;
   end
end

RightEyebrowLen = uint8(uRcol/2)

RightEyeRectx = j
RightEyeRecty = pos - 10
RightEyeRectw = RightEyebrowLen+1
RightEyeRecth = 15


% first get the eye rect of the upperLeft face
RightEyeRect = [RightEyeRectx,RightEyeRecty,RightEyeRectw,RightEyeRecth]

RightEyeRectFace = [RightEyeRectx+rect(3),RightEyeRecty,RightEyeRectw,RightEyeRecth]

rectangle('curvature',[0 0],'position',RightEyeRectFace,'edgecolor',[0 1 0],'LineWidth',[1]);


%==========================================

%lower half face

%figure,imshow(lowerBinFace);

invertLowerBinFace = 1 - lowerBinFace;

imview(invertLowerBinFace);

%se = strel('disk',1);
%c = imerode(c,se);
```

```
[lrow , lcol] = size(lowerBinFace);
sumCol = zeros(lcol,1)

for j=1:lcol
    sumCol(j) = sum(invertLowerBinFace(:,j));
    if sumCol(j) > (lcol/2)
       j
       break;
    end
end

j = j + 1;

%remove chin

lowerFaceWoChin = uint8(lrow - (lrow/3))-10;

rectChin = [ 1 , 1 , lcol , lowerFaceWoChin ]

lowerBinFaceC = imcrop(lowerBinFace , rectChin);

sumlipCol = zeros(lcol,1)
for k=j:(lcol-20)
    sumlipCol(k) = sum(lowerBinFaceC(:,k));
end

[val , pos] = max(sumlipCol);

lipx = pos;

[lcrow,lccol ] = size(lowerBinFaceC);

sumlipRow = zeros(lcrow,1);

for l=1:lcrow
   sumlipRow(l) = sum(lowerBinFaceC(l,:));
end

[val , pos] = max(sumlipRow)

lipy = pos;

if ( lipx > (lccol/2))
   lipxA = lipx-20
else
```

```
   lipxA = lipx-10
end

lipRect = [ lipxA, (lipy-5)+lrow, 45 , 15 ];

 rectangle('curvature',[0 0],'position',lipRect,'edgecolor',[0 1 0],'LineWidth',[1]);


%==========
% nose area

noseRect = [ lipRect(1), lipRect(2)-40, 30, 25];
rectangle('curvature',[0 0],'position',noseRect,'edgecolor',[0 1 0],'LineWidth',[1]);


% nose area end
%------
% Cheek area

leftCheekRect = [ leftEyeRect(1), noseRect(2), 30,25];
rectangle('curvature',[0 0],'position',leftCheekRect,'edgecolor',[0 1 0],'LineWidth',[1]);

rightCheekRect = [ RightEyeRectFace(1), noseRect(2), 30,25];
rectangle('curvature',[0 0],'position',rightCheekRect,'edgecolor',[0 1 0],'LineWidth',[1]);

%=======

function
[lipFirstx,lipFirsty,lipLastx,lipLasty,lipMidx,lipMidy,lowerLip1x,lowerLip1y,lowerLip2x,lower
Lip2y,REyeFirstx,REyeFirsty,REyeLastx,REyeLasty,REyeMidx,REyeMidy,LEyeFirstx,LEyeFir
sty,LEyeLastx,LEyeLasty,LEyeMidx,LEyeMidy] = findpoints(facelip, faceRighteye,
faceLefteye)


facelipRow  = facelip(4); % how many rows
facelipCol  = facelip(3); % how many col

lipBW = im2bw(imcrop(ipframe,facelip),0.6);

%for initial rows find out the first white point to track on lip

flag = 'N';

for i = 3:facelipCol
   for j = 3:facelipRow
      if ( lipBW(j,i) == 0 )
```

```
            lipFirstx = i;
            lipFirsty = j;
            flag = 'Y';
            break;
        end
    end
    if (flag == 'Y')
        break;
    end
end

figure , imshow( lipBW)
%rectangle('curvature',[1 1],'position',[lipFirstx lipFirsty 2 2 ],'edgecolor',[0 1
0],'LineWidth',[1],'FaceColor','r');

flag = 'N';

for i = (facelipCol-3):-1:1
    for j = 3:facelipRow
        if ( lipBW(j,i) == 0 )
            lipLastx = i;
            lipLasty = j;
            flag = 'Y';
            break;
        end
    end
    if ( flag == 'Y')
        break;
    end
end

mid = round((lipLastx + lipFirstx)/2)-2;

flag = 'N';

for i = mid:facelipCol
    for j = 1:facelipRow
        if ( lipBW(j,i) == 0 )
            lipMidx = i;
            lipMidy = j;
            flag = 'Y';
            break;
        end
    end
    if ( flag == 'Y')
        break;
```

```
    end
end

[ m n ] = size ( lipBW);
lowerLip1x = lipMidx - 4;
lowerLip1y = m - 4;

lowerLip2x = lipMidx + 4;
lowerLip2y = m - 4;

rectangle('curvature',[1 1],'position',[lipFirstx lipFirsty 2 2 ],'edgecolor',[0 1
0],'LineWidth',[1],'FaceColor','r');
rectangle('curvature',[1 1],'position',[lowerLip2x lowerLip2y 2 2 ],'edgecolor',[0 1
0],'LineWidth',[1],'FaceColor','r');
rectangle('curvature',[1 1],'position',[lowerLip1x lowerLip1y 2 2 ],'edgecolor',[0 1
0],'LineWidth',[1],'FaceColor','r');
rectangle('curvature',[1 1],'position',[lipMidx lipMidy 2 2 ],'edgecolor',[0 1
0],'LineWidth',[1],'FaceColor','r');
rectangle('curvature',[1 1],'position',[lipLastx lipLasty 2 2 ],'edgecolor',[0 1
0],'LineWidth',[1],'FaceColor','r');


%
%      %for initial rows find out the first white point to track on right eye brow
%
rightEyeBW = im2bw(imcrop(ipframe,faceRighteye));

rightEyeBW = medfilt2(rightEyeBW, [ 3 3 ]);

figure , imshow ( rightEyeBW )

faceREyeRow  = faceRighteye(4); % how many rows
faceREyeCol  = faceRighteye(3); % how many col


flag = 'N';

for i = 3:faceREyeCol
   for j = 1:faceREyeRow
      if ( rightEyeBW(j,i) == 0 )
         REyeFirstx = i;
         REyeFirsty = j;
         flag = 'Y';
         break;
      end
   end
```

```
    if ( flag == 'Y')
        break;
    end
end


flag = 'N';

for i = (faceREyeCol-7):-1:1
    for j = 1:faceREyeRow
        if ( rightEyeBW(j,i) == 0 )
            REyeLastx = i;
            REyeLasty = j;
            flag = 'Y';
            break;
        end
    end
    if ( flag == 'Y')
        break;
    end
end
%
mid = round((REyeLastx + REyeFirstx)/2)-2;
%

flag = 'N';

for i = mid:faceREyeCol
    for j = 1:faceREyeRow
        if ( rightEyeBW(j,i) == 0 )
            REyeMidx = i;
            REyeMidy = j;
            flag = 'Y';
            break;
        end
    end
    if ( flag == 'Y')
        break;
    end
end
%

r = 2 ;
c = 2 ;

rectangle('curvature',[1 1],'position',[REyeMidx REyeMidy r c ],'edgecolor',[0 1
```

```
0],'LineWidth',[1],'FaceColor','r');
rectangle('curvature',[1 1],'position',[REyeLastx REyeLasty r c ],'edgecolor',[0 1
0],'LineWidth',[1],'FaceColor','r');
rectangle('curvature',[1 1],'position',[REyeFirstx REyeFirsty r c ],'edgecolor',[0 1
0],'LineWidth',[1],'FaceColor','r');


%
%      %for initial rows find out the first white point to track on left eye brow
%
leftEyeBW = im2bw(imcrop(ipframe,faceLefteye));
leftEyeBW = medfilt2(leftEyeBW, [ 3 3 ]);
figure , imshow ( leftEyeBW )

faceLEyeRow  = faceLefteye(4); % how many rows
faceLEyeCol  = faceLefteye(3); % how many col


flag = 'N';

for i = 6:faceLEyeCol
   for j = 1:faceLEyeRow
      if ( leftEyeBW(j,i) == 0 )
         LEyeFirstx = i;
         LEyeFirsty = j;
         flag = 'Y';
         break;
      end
   end
   if ( flag == 'Y')
      break;
   end
end


flag = 'N';

for i = (faceLEyeCol-1):-1:1
   for j = 1:faceLEyeRow
      if ( leftEyeBW(j,i) == 0 )
         LEyeLastx = i;
         LEyeLasty = j;
         flag = 'Y';
         break;
      end
   end
```

```
   if ( flag == 'Y')
      break;
   end
end
%

flag = 'N';
mid = round((LEyeLastx + LEyeFirstx)/2)-2;

for i = mid:faceLEyeCol
   for j = 1:faceLEyeRow
      if ( leftEyeBW(j,i) == 0 )
         LEyeMidx = i;
         LEyeMidy = j;
         flag = 'Y';
         break;
      end
   end
   if (flag == 'Y')
      break;
   end
end


r = 2 ;
c = 2 ;

rectangle('curvature',[1 1],'position',[LEyeMidx LEyeMidy+5 r c ],'edgecolor',[0 1
0],'LineWidth',[1],'FaceColor','r');
rectangle('curvature',[1 1],'position',[LEyeLastx LEyeLasty r c ],'edgecolor',[0 1
0],'LineWidth',[1],'FaceColor','r');
rectangle('curvature',[1 1],'position',[LEyeFirstx LEyeFirsty+7 r c ],'edgecolor',[0 1
0],'LineWidth',[1],'FaceColor','r');




%
function [pitchvar, pitchavg] = findPitch(inputfile)

[y, Fs, nbits] = wavread(inputFile);

coch=LyonPassiveEar(y,22050,1,4,.5);
cor=CorrelogramArray(coch,22050,50,256);
```

```
[pf,s]=CorrelogramPitch(cor,256,22050);
[x,y] = size(pf);

for i = 1:y-1
   diffPf(i) = pf(i+1)-pf(i);

end

varfear = var(pf)
avgfear = mean(pf)
devfear = std(pf)



% bnet for facial

function [] = dag_facialfeature1(llP,oeP,ieP,lcP,ulP,cP,nP)

cd C:\MATLAB7\FullBNT-1.0.4\
addpath(genpathKPM(pwd))

N = 12;
dag = zeros(N,N);

A = 1;  % angry
IE = 2; % inner eyebrow
OE = 3; % outer eyebrow
F = 4;  % fear
UL = 5 % upper lip
LL = 6; % lower lip
S = 7;  % sad
LC = 8; % lip corner
H = 9; % happy
C = 10; % cheek
D = 11; % disgust
N = 12; % nose

dag ( [ A,F,S] , [IE,OE]) = 1;
dag ( [ A,S,H] , LC) = 1;
```

```
dag ( H , C) = 1;
dag ( D , N) = 1;
dag ( [F,D] , [UL,LL]) = 1;

%draw_graph(dag)

discrete_nodes = 1:N;
node_sizes = [ 2 3 3 2 2 2 2 3 2 2 2 2 ];

bnet = mk_bnet(dag, node_sizes,'names', {'angry','inner_eb','outer_eb','fear','upp_lip',
'low_lip','sad','lip_corner','happy','cheek','disgust','nose'}, 'discrete', discrete_nodes);

a = bnet.names('angry');
bnet.CPD{a} = tabular_CPD(bnet, a, [0.5 0.5]);
f = bnet.names('fear');
bnet.CPD{f} = tabular_CPD(bnet, f, [0.5 0.5]);
s = bnet.names('sad');
bnet.CPD{s} = tabular_CPD(bnet, s, [0.5 0.5]);
h = bnet.names('happy');
bnet.CPD{h} = tabular_CPD(bnet, h, [0.5 0.5]);
d = bnet.names('disgust');
bnet.CPD{d} = tabular_CPD(bnet, d, [0.5 0.5]);

lc = bnet.names('lip_corner');
bnet.CPD{lc} = tabular_CPD(bnet, lc, [0.5  0.15  0     0     0.9   0     0     0
        0.5   0.75  0.75  0     0.1   0     0     0     0     0.1   0.25  0
        0     0     0     0]);

ch = bnet.names('cheek');
bnet.CPD{ch} = tabular_CPD(bnet, ch , [0.35       0.65 0.65 0.35]);

n = bnet.names('nose');
bnet.CPD{n} = tabular_CPD(bnet, n, [0.25  0.75   0.75   0.25]);

ll = bnet.names('low_lip');
bnet.CPD{ll} = tabular_CPD(bnet, ll, [0.5 0.85 0.75 1 0.5 0.15 0.25 0]);

ul = bnet.names('upp_lip');
bnet.CPD{ul} = tabular_CPD(bnet, ul, [0.5 0.85 0.75 1 0.5 0.15 0.25 0]);

oe = bnet.names('outer_eb');
bnet.CPD{oe} = tabular_CPD(bnet, oe, [0   0.3    0.7   0     0     0     0     0
        0     0.7   0.3   0     0.6   0     0     0     0     0     0     0
        0.4   0     0     0 ]);

ie = bnet.names('inner_eb');
```

```
bnet.CPD{ie} = tabular_CPD(bnet, ie, [0    0.2    0.8    0    0.7    0    0    0
      0    0.7    0.1    0    0.2    0    0    0    0    0.1    0.1    0
      0.1    0    0    0]);

evidence = cell(1,N);

var_elim_engine = var_elim_inf_engine(bnet);
if(llP ~= 2)
evidence{ll} = llP;
end
if(oeP ~= 3)
evidence{oe} = oeP;
end
if(ieP ~= 3)
evidence{ie} = ieP;
end
if(lcP ~= 3)
evidence{lc} = lcP;
end
if(ulP ~= 2)
evidence{ul} = ulP;
end
if(cP ~= 2)
evidence{ch} = cP;
end
if(nP ~= 2)
evidence{n} = nP;
end

[var_elim_engine, loglik] = enter_evidence(var_elim_engine, evidence);

marg_varElimA = marginal_nodes(var_elim_engine, a);
marg_varElimH = marginal_nodes(var_elim_engine, h);
marg_varElimS = marginal_nodes(var_elim_engine, s);
marg_varElimF = marginal_nodes(var_elim_engine, f);
marg_varElimD = marginal_nodes(var_elim_engine, d);

emotion = size(5);

emotion(1) = (marg_varElimA.T(2))
emotion(2) = (marg_varElimH.T(2))
emotion(3) = (marg_varElimS.T(2))
emotion(4) = (marg_varElimF.T(2))
emotion(5) = (marg_varElimD.T(2))

[var pos] = max(emotion);
```

```
if( pos == 1 )
   fprintf('Emotion detected by Facial Feature classifier is Angry\n');
end

if( pos == 2 )
   fprintf('Emotion detected by Facial Feature classifier is Happy\n');
end

if( pos == 3 )
   fprintf('Emotion detected by Facial Feature classifier is Sad\n');
end

if( pos == 4 )
   fprintf('Emotion detected by Facial Feature classifier is Fear\n');
end

if( pos == 5 )
   fprintf('Emotion detected by Facial Feature classifier is Disgust\n');
end

clear

% bnet for speech

function [] = dag_speech(srP,paP,pcP)

addpath('C:\MATLAB7\FullBNT-1.0.4\')

N = 8;
dag = zeros(N,N);

A = 1;  % angry
SR = 2;  % speech rate
H = 3; % happy
PA = 4;  % pitch avg
S = 5; % sad
F = 6; % fear
D = 7; % disgust
PC = 8; % pitch changes

dag ( [ A,H,S,F,D] , [SR,PA]) = 1;
dag ( [ A,H,S,D] , [SR,PC]) = 1;

%draw_graph(dag)
```

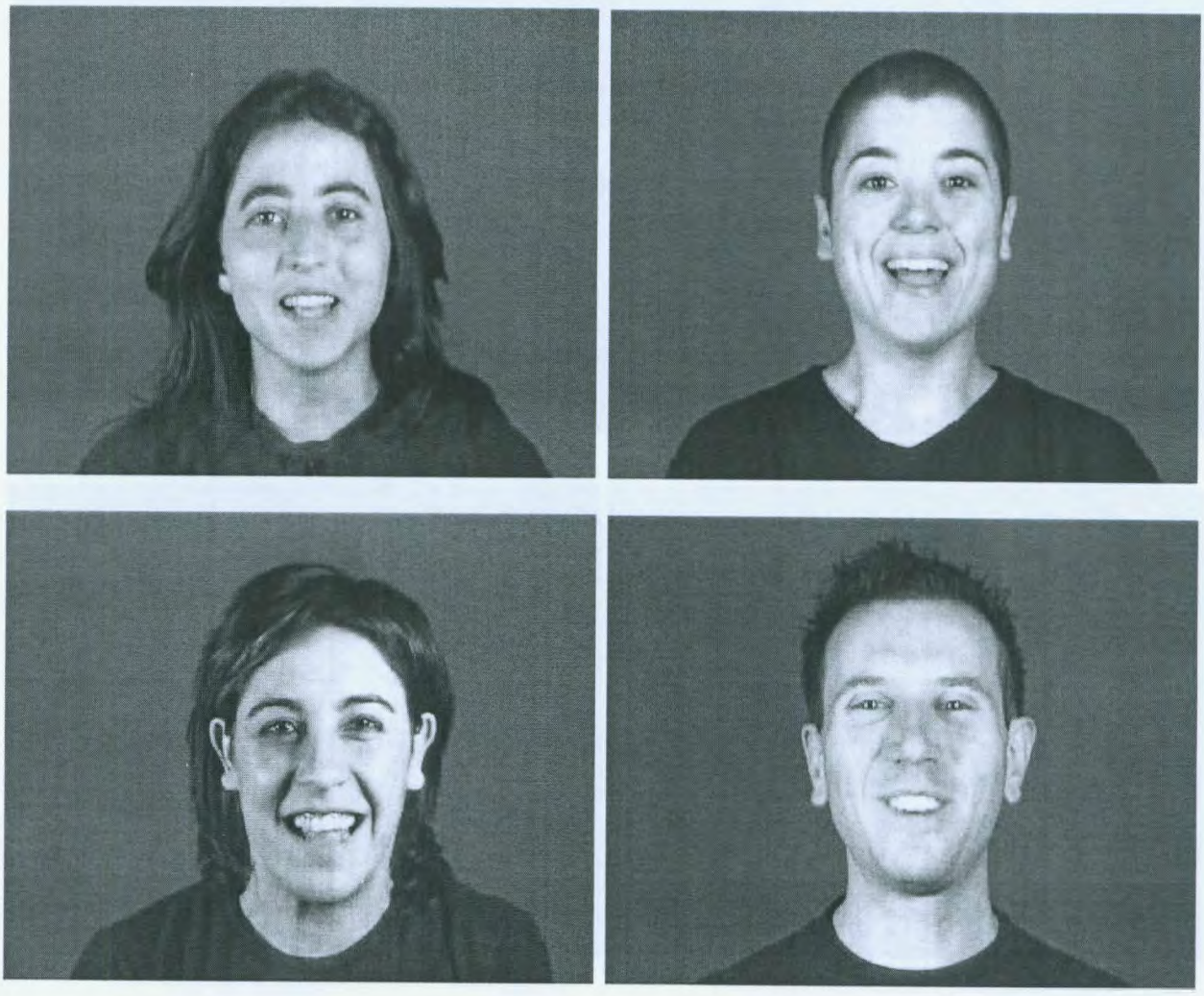
```
discrete_nodes = 1:N;
node_sizes = [ 2 5 2 2 2 2 2 3 ];
bnet = mk_bnet(dag, node_sizes,'names', {'angry','speech_rate','happy','pitch_avg','sad',
'fear','disgust','pitch_var'}, 'discrete', discrete_nodes);

a = bnet.names('angry');
bnet.CPD{a} = tabular_CPD(bnet, a, [0.5 0.5]);
f = bnet.names('fear');
bnet.CPD{f} = tabular_CPD(bnet, f, [0.5 0.5]);
s = bnet.names('sad');
bnet.CPD{s} = tabular_CPD(bnet, s, [0.5 0.5]);
h = bnet.names('happy');
bnet.CPD{h} = tabular_CPD(bnet, h, [0.5 0.5]);
d = bnet.names('disgust');
bnet.CPD{d} = tabular_CPD(bnet, d, [0.5 0.5]);
pa = bnet.names('pitch_avg');
bnet.CPD{pa} = tabular_CPD(bnet, pa, [0 0.9 0.9 0 0.1   0   0   0   0.9   0
        0   0   0   0   0   0   0.1   0   0   0   0   0
        0   0   0   0   0   0   0   0   0   0   0   0.1
        0.1   0   0.9   0   0   0   0.1   0   0   0   0   0
        0   0   0.9   0   0   0   0   0   0   0   0   0
        0   0   0   0   0   0 ]);
pc = bnet.names('pitch_var');
bnet.CPD{pc} = tabular_CPD(bnet, pc, [0   0.5   0.9   0   0.1   0   0   0
        0.1   0   0   0   0   0   0   0   0   0.5   0.1   0
        0.9   0   0   0   0.9   0   0   0   0   0   0   0 0
        0   0   0   0   0   0   0   0   0   0   0
        0   0   0 ]);
sr = bnet.names('speech_rate');
bnet.CPD{sr} = tabular_CPD(bnet, sr , [0   0.1   0.5   0   0   0   0   0
        0.8   0   0   0   0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0   0   0   0   0
        0   0.1   0.5   0   0.2   0   0   0   0.1   0   0   0
        0   0   0   0   0.1   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0   0   0.7   0   0
        0   0   0   0   0.1   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0   0   0   0   0
        0   0   0   0   0   0.1   0   0   0.8   0   0   0
        0   0   0   0   0   0   0   0   0.1   0   0   0
        0   0   0   0   0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0   0   0   0   0
        0   0   0   0   0.8   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0]);

%engine = jtree_inf_engine(bnet);
```

```
evidence = cell(1,N);

var_elim_engine = var_elim_inf_engine(bnet);

evidence{pa} = paP;
evidence{pc} = pcP;
evidence{sr} = srP;

[var_elim_engine, loglik] = enter_evidence(var_elim_engine, evidence);
marg_varElimA = marginal_nodes(var_elim_engine, a);
marg_varElimH = marginal_nodes(var_elim_engine, h);
marg_varElimS = marginal_nodes(var_elim_engine, s);
marg_varElimF = marginal_nodes(var_elim_engine, f);
marg_varElimD = marginal_nodes(var_elim_engine, d);


emotion = size(5);

emotion(1) = (marg_varElimA.T(2));
emotion(2) = (marg_varElimH.T(2));
emotion(3) = (marg_varElimS.T(2));
emotion(4) = (marg_varElimF.T(2));
emotion(5) = (marg_varElimD.T(2));

[var pos] = max(emotion);

if( pos == 1 )
   fprintf('Emotion detected by Speech classifier is Angry\n');
end

if( pos == 2 )
   fprintf('Emotion detected by Speech classifier is Happy\n');
end

if( pos == 3 )
   fprintf('Emotion detected by Speech classifier is Sad\n');
end

if( pos == 4 )
   fprintf('Emotion detected by Speech classifier is Fear\n');
end

if( pos == 5 )
   fprintf('Emotion detected by Speech classifier is Disgust\n');
end
```

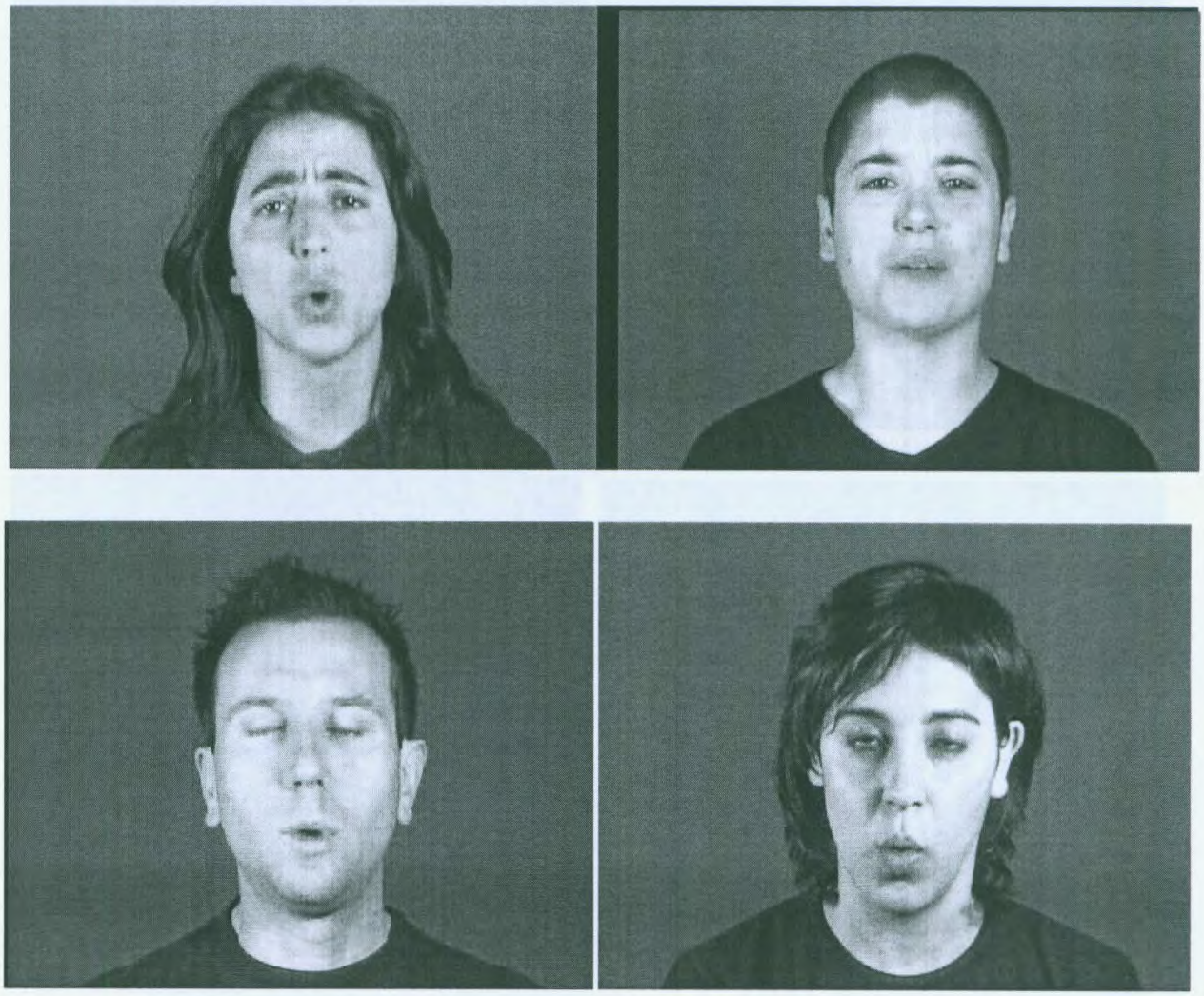
clear

Appendix B. Video Frames

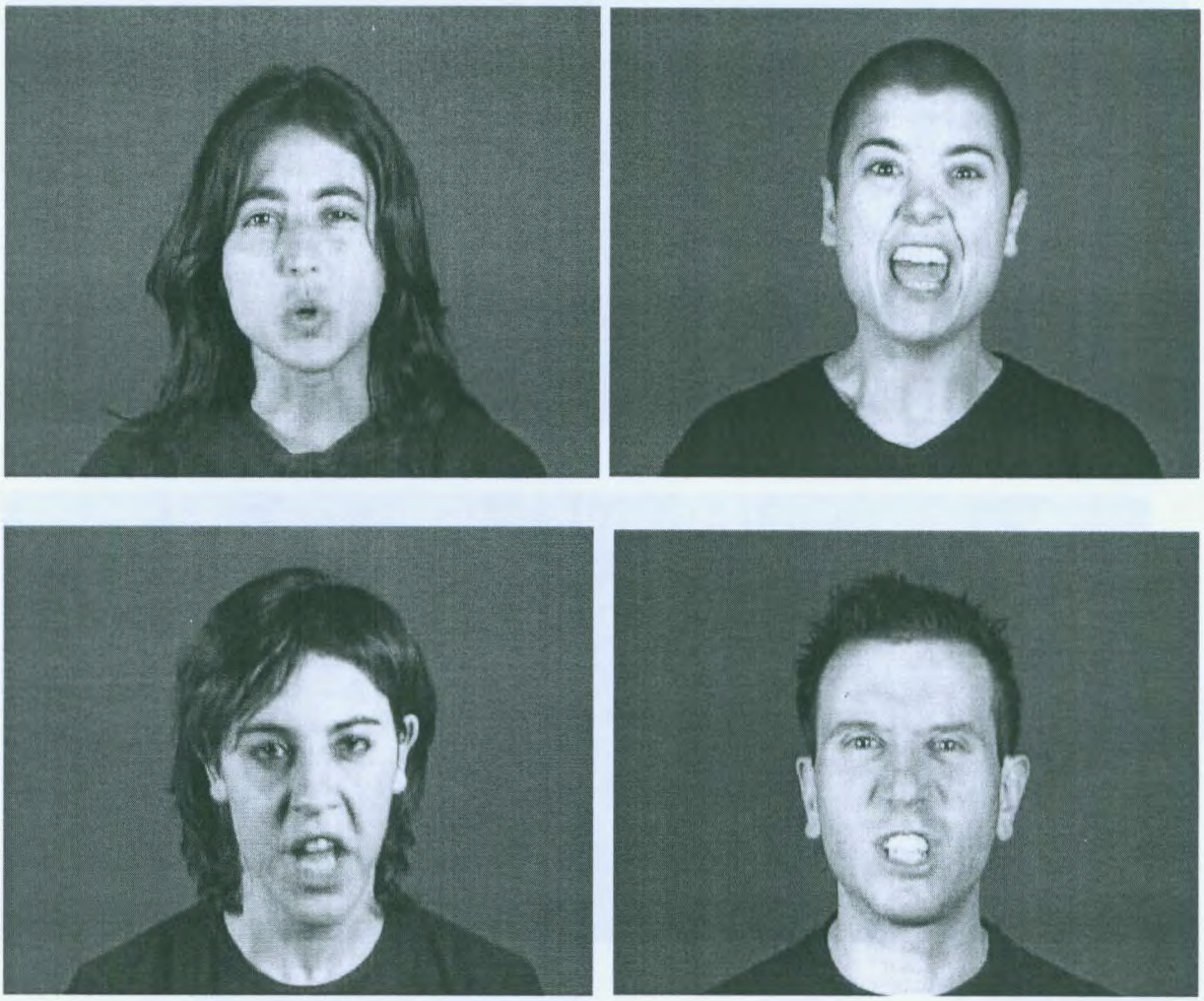
Happy emotion frames from various actors:



Sad emotion frames from various actors

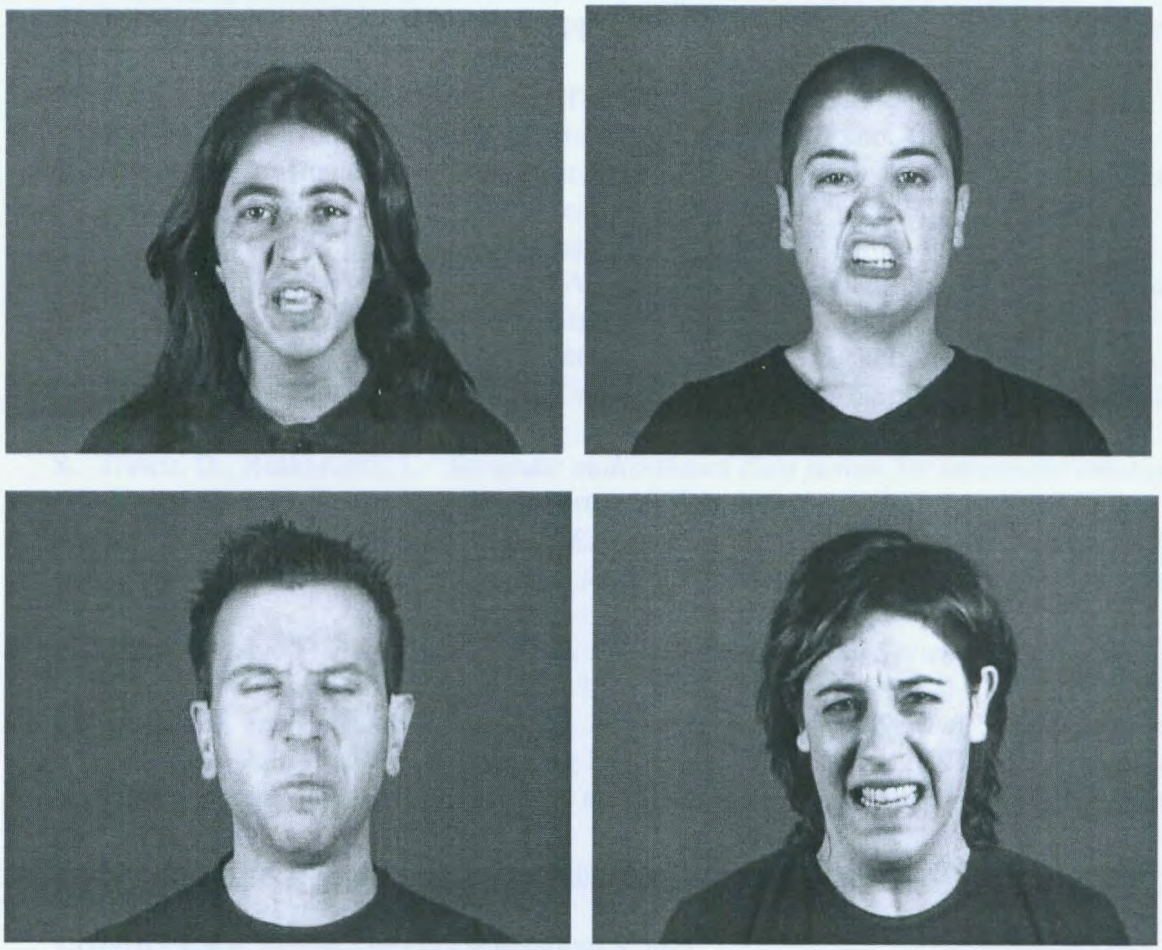


Angry emotion frames from various actors



Fear emotion frames from various actors

Disgust emotion frames from various actors:



References

1. Alan Hanjalic *Content-Based Analysis of Digital Video* ISBN-10: 1402081146
2. Artificial Intelligence: A Modern Approach (Second Edition) by Stuart Russell and Peter Norvig Chapter 13,14,15
3. Auditory Toolbox Malcolm Slaney Technical Report #1998-010
4. B. Chanda D. Dutta Majumder Digital Image Processing and Analysis (2000)
5. Battocchi, A.; Pianesi, F.; Goren-Bar, D.. DaFEx: *Database of Facial Expressions.* In Proceedings of INTETAIN'05, November 30 – December 2, 2005 – Madonna di Campiglio (Italy). http://tcc.itc.it/research/i3p/dafex/
6. Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 8(6).
7. Chew, M., and Tygar, J.D. Image Recognition CAPTCHAs. In *Proceedings of the 7th International Information Security Conference (ISC 2004)*, Springer, September, 2004, 268-279.
8. Datcu, D., Rothkrantz, L.: *Semantic audio-visual data fusion for automatic emotion recognition.* In: Euromedia 2008, Porto (2008).
9. Dornaika F. and Davoine F., *Facial Expression Recognition In The Presence Of Head Motion*, published by InTech - Open Access Publisher, ISBN 978-3-902613-23-3.. (2008).
10. Ekman, P. and Friesen, W.V. (1978). *Facial Action Coding System: Manual.* Palo Alto, CA: Consulting Psychologists Press, Inc., 1978, ch. 1-2, pp. 1-2, 4.
11. Ekman, P. and Friesen, W.V. (1978). *Facial Action Coding System: Investigator's Guide, Part Two.* Palo Alto, CA: Consulting Psychologists Press, Inc., 1978, ch. 9, pp. 144.
12. Finn V. Jensen, "Bayesian Networks and Decision graphs", Springer - Verlag, New York, 2001
13. http://math.uc.edu/~siva/mdbayes/chap2p.pdf
14. http://www.bayesit.com
15. http://www.bayesit.com/docs/advantages.html
16. http://www.grin.com/e-book/115799/universals-in-facial-expression
17. J.L. Barron and N.A. Thacker, *Tutorial: Computing 2D and 3D Optical Flow*, Tina Memo No. 2004-012, 2005
18. Jhanwar N, Raina A *Pitch correlogram clustering for fast speaker identification Full text* EURASIP Journal on Applied Signal Processing Volume 2004 , Issue 1 (January 2004) ISSN:1110-8657
19. Kim, S.H. and Park, R.H. (2004). *A Novel Approach to Video Sequence Matching Using Color and Edge Features With the Modified Hausdorff Distance.* Circuits and Systems, 2004. ISCAS apos;04. Proceedings of the 2004 International Symposium on Volume 2, Issue , 23-26 May 2004 Page(s): II - 57-60 Vol.2
20. Kun Peng, Liming Chen, Su Ruan and Georgy Kukharev (2005) *A Robust Algorithm for Eye Detection on Gray Intensity Face without Spectacles* Lecture Notes in Computer

Science Springer Berlin / Heidelberg ISSN0302-9743 (Print) 1611-3349 (Online) Volume 3687/2005
21. Lecture notes by Philip Loizou https://www.utd.edu/~loizou/ee6362/lec4.pdf
22. Lienhart, R. (1998). *Comparison of Automatic Shot Boundary Detection Algorithms.* Microcomputer Research Labs, Intel Corporation, Santa Clara, CA 95052-8819
23. Lyon R. F., "*A Computational Model of Filtering, Detection, and Compression in the Cochlea.*" Proc IEEE-ICASSP, 1982, 1282-1285.
24. N. Sebe; I. Cohen; T. Gevers; T.S. Huang *Emotion Recognition Based on Joint Visual and Audio Cues* Pattern Recognition, ICPR 2006. 18th International Conference on Volume 1, Issue , 2006
25. Narayanan, S.; Wang, D. *Speech Rate Estimation via Temporal Correlation and Selected Sub-Band Correlation* Acoustics, Speech, and Signal Processing, 2005. Proceedings. (lCASSP apos;05). IEEE International Conference on Volume 1, Issue , March 18-23, 2005
26. Pantic, M. Rothkrantz, L.J.M. (2002). *Facial Gesture Recognition in Face Image Sequences: A Study on Facial Gestures Typical for Speech Articulation.* Systems, Man and Cybernetics, 2002 IEEE International Conference on Publication Date: 6-9 Oct. 2002
27. R. Dugad, K. Ratakonda, and N. Ahuja. *Robust video shot change detection.* IEEE Workshop on Multimedia Signal Processing, Dec 1998.
28. Rakesh Dugad, Krishna Ratakonda and Narendra Ahuja *Robust video shot change detection*
29. Raouzaiou, A., Ioannou, S., Akrivas, G., Karpouzis, K., and Kollias, S. (2004). *Adaptation of Expression Analysis Based on Evaluation Principles.* Fourth European Symposium on Intelligent Technologies and their implementation on Smart Adaptive Systems (Eunite 2004), Aachen, Germany, June 2004.
30. Teol, W.K., De Silva, L.C., and Vadakkepat, P. (*2004*). *Facial Expression Detection and Recognition System.* Journal of The Institution of Engineers, Singapore Vol. 44 Issue 3 2004.
31. www.cis.temple.edu/~latecki/Courses/CIS750-03/Lectures/Heshan_ShotDet.ppt
32. www.cs.ubc.ca/~murphyk/Software/BNT/bnt.html
33. Xuejing Sun (2002) *Pitch Determination And Voice Quality Analysis Using Subharmonic-To-Harmonic Ratio* Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, Orlando, Florida, May 13-17, 2002.
34. Y. Yusoff, W. Christmas and J. Kittler *Video Shot Cut Detection Using Adaptive Thresholding* Centre for Vision, Speech and Signal Processing University of Surrey 2000
35. Yafei S , Nicu S ,Lew M S. ,Gevers T *Authentic emotion detection in real-time video* Computer vision in human-computer interaction : ECCV 2004 workshop on HCI, Prague , TCHEQUE, REPUBLIQUE 2004, vol. 3058, pp. 94-104, ISBN 3-540-22012-7 ;
36. Zeng, Z., Tu, J., Liu, M., Zhang, T., Rizzolo, N., Zhang, Z.,Huang, T., Roth, D., and Levinson, S., *Bimodal HCI-related affect recognition*, in Proc. *ICMI,* 2004.