

Smiles vs chewing vs speech detection by similarity matching

Wenyang Cai

August 30, 2014

Abstract

This is abstract.

Acknowledgements

This is acknowledgement

Contents

1	Introduction	2
1.1	Motivation	2
1.2	Thesis	2
1.3	Contribution	2
2	Background	3
2.1	Automatic Speech Recognition	3
2.2	Visual Front End	4
2.2.1	Face Detection	4
2.2.2	Region of Interest	4
2.2.3	Visual Feature and Postprocessing	4
2.3	Facial Expression	5
2.4	Nonlinguistic Vocalization	5
2.5	Audio-Visual Biometrics	5
3	Processing and Methodologies	6
3.1	Processing Flow	6
3.2	Face Alignment	7
3.2.1	Active Appearance Model	8
3.2.2	Trackers	9
3.2.3	Comparison	12
3.3	Remove Head-pose	15
3.4	Warping	15
3.5	Feature Extraction	17
3.5.1	Local Binary Pattern	17
3.6	Postprocessing	19
4	Experiment and Results	20
4.1	Data	20
4.1.1	Feature	20
4.2	Methodology	21
4.2.1	Find Parameter c and γ	21
4.3	Results and Analysis	21
5	Conclusion and Future Work	25
A	Classification Result	27

Chapter 1

Introduction

1.1 Motivation

Automatic Speech Recognition is a very important and popular technology in Human Computer Interactions. In recent years, device and software that support ASR system are increasing rapidly. Diversification of interacting with a system is increasing. People are not satisfied with just interacting with a computer through mouth and keyboards. Automatic Speech Recognition is a very good technology for HCI as human-human communication are mostly through speech. However, human also can understand others by reading the signal of facial expression, some non-linguistic vocalization, even small gesture. [10] Laughter are commonly used in human social conversation. However, the research in recognising the non-linguistic vocalization is rare.

1.2 Thesis

1.3 Contribution

Chapter 2

Background

In human-human communication, human would be able to understand each other through variety feedbacks. We are able to understand others through conversation, facial expression, gesture, even intonation, mood. This phenomenon was introduced into human-computer interactions(HCI). Using computer to automatically analyse human face, voice and behaviours would be helpful in improving HCI. Many technology are developed for computer to automatically recognize human actions and emotions such as automatic speech recognition [11], automatic recognize facial expression [15] and so on. These technologies not only can be used for HCI, but also can be used in other area such as avoid impostor attacks [1]. Face and voice are important and personal biometric characteristics. Compare to traditional knowledge-based and token-based person recognition method, biometric recognition technology are more convenient and safer. With the increasing in computer computing power and development in computer vision, Automatic Speech Recognition(ASR) attracts more attention in recent years, from traditional audio-only ASR to audiovisual ASR, a huge progress was made in ASR.

2.1 Automatic Speeech Recognition

The intuition of Automatic Speech Recognition is to convert a spoken sentence into to readable text in real time by computer and he ultimate goal of ASR is to let a computer 100% accurately recognise speech of any person under any environment. The beneficial of ASR is quite obvious, it can help deaf people listening by convert speech into readable text and help hard-reading people on reading by convert readable text into voice. The search engine may not be limited on text searching but also speech search. It has been studied for over 50 years, recent work has a great improvement in this area. The accuracy of recognition highly depended on robust information channel, background environment, the training data base and the adaptation of speaker to database. With decreasing price of computing power, Speech recognition techniques are widely used on mobile devices, like Siri.

People can not only know what others says by listening to speaker, they can also know what others says by watching. Deaf people is able to communicate with each other by reading mouth movements. Visual modality was proved to have positive influence on reducing noise in ASR and the history can be quantified back to 1954 [11]. [11] gave three key reasons to include vision information in human speech recognition. Firstly , it helps audio source localisation, visual information of tongue, teeth, and lips provide complementary information of articulation. Secondly, it is beneficial for distinguish confusable acoustics such as unvoiced consonants /p/ by providing information of facial muscle movements. Thirdly, facial muscle movements are robust information for ASR. This technique of using visual information to recognise speech is known as automatic lipreading or speechreading in ASR [11].

Audiovisual ASR uses both visual modality and audio modality in recognizing speech. There are two main challenge in AV-ASR from the original ASR. One problem is how to extract visual features, the other problem is how to combine it with audio features. Visual speech information mainly on speaker's face. Extracting visual feature requires techniques from other field, such as face detection,

face alignment, tracking, feature extraction and other techniques to extract useful features from image with a face. The second problem is how to make the combination of both modality produce better performance than using single modality. [11] named two area, visual front-end design and audiovisual fusion.

2.2 Visual Front End

A major problem in audiovisual automatic speech recognition is extracting visual feature from images. The process is to extract visual speech features from videos or a sequence of images. Generally, visual speech feature can be classified as three types: appearance feature, shape feature, combination of both [11]. Appearance feature usually means the image feature of Region of Interest. Image of (ROI) used to be directly used for training and classification. However, image data contains many noises and influenced by the lighting condition. Some techniques in computer vision are used to extract image features from Region of Interest. Image feature contains points, edges, texture, colors and so on. LBP feature is a very good texture descriptor. Shape feature usually means contour of speaker's face, especially speaker's lips or including jaw and cheek. Shape feature usually means geometric-type features, there are many ways to describe shape feature of a face, such as statistical shape model or image moment descriptor of mouth, these model would be able to contain the information of the height, width and other information of a mouth. Combined feature usually is the joint of both shape and appearance feature vectors or a model that include both features such as active appearance model.

In order to get appearance feature and shape feature or combined feature, there are some pre-processing steps should before extracting feature. Face detect, detecting the position of face, facial tracking, tracking facial feature points from frame to frame, face alignment, align facial feature points to a face, and ROI extracting technique extract features from ROI, these technique are all required for extracting visual speech features from videos. In order to align a face with feature points, some statistical models are used to fit the face. In order to remove head-pose, some technique are used to decompose the head points and remove head pose. If the head is not facing the front, appearance of face would not be distorted to frontal, warping technique are also needed.

2.2.1 Face Detection

Face detection is to detect the location and size of single face or several faces in one digital image. There are several important aspects would influence face detection: background, head pose and lighting condition. Many system uses traditional image processing techniques [11] , such as color segmentation, edge detection, image thresholding, template matching or motion information and some using statistical modelling and neural networks. Once a face is detected, use face alignment technique to estimate the location of several facial feature around the face.

2.2.2 Region of Interest

Usually the choose of Region of Interest is depending on the purpose of the project. In AV-ASR it usually include large part of the lower face, such as the jaw, and cheeks or even the entire face [11], as when people speak, the lower face would show some movements. In my project, the ROI only contains the grayscale values of mouth region, which is scaled to $8 * 32$ size square region. [11] report that experiments shows that including jaw and cheeks was beneficial. As the tracker I use for tracking face does not include the jaw and cheek of the face, I have to just include the mouth.

2.2.3 Visual Feature and Postprocessing

Usually the image is not directly used for classification, because of the influence of noise and brightness. There are many descriptor can be used to represent a image and reduce the influence. In addition, the dimensionality of image vector is usually very large, it is not suitable for classification. The choice of visual feature is depends on the requirements of the project. The most popular

descriptor for AV-ASR is texture feature descriptor. If the dimension of feature vector is too high, the common dimension reduction methods are traditional linear transforms. [11] gave some most commonly applied methods, Principle Component Analysis(PCA), discrete cosine transform(DCT), discrete wavelet transform, Hadamard and Haar transforms and a linear discriminant analysis based data projection. As the visual feature are extracted from variety of lighting condition and different face. These effects can be remedied by normalization, i.e. for each visual feature vector subtract it's mean and divide by the standard deviation. For different classification methods, some necessary steps may be needed before use feature vector to do classification.

2.3 Facial Expression

Generally, face would show four kinds of signals: static facial signal, slow facial signals, artificial signals and rapid facial signals [8]. Rapid facial signals underlie facial expressions. [8] indicate that rapid facial signals generally show five types of messages: affective attitudinal states and moods which means emotions, emblems, manipulators, illustrators, regulators. For example, smile belongs to regulators and chewing belongs to manipulators. Automatic analysis of facial signals such as rapid facial signals have potential applications in many areas. Layers, security, police could be use automatic analysis of facial signal system to monitoring and interpreting human facial signal and gain important information. For example, monitoring human reaction during inquisition, inquisitor would be able to tell whether a person is lie or not. Machine analysis of facial expression forms an important part of affective human-computer interface designs [8]. Research on machine analysis of facial expression mainly focuses on facial affect and facial muscle action detection [8]. Technologies used in this area are face detection, facial feature extraction, facial muscle action detection and emotion recognition [8].

2.4 Nonlinguistic Vocalization

Non-linguistic vocalization is defined as brief, discrete, non-verbal expressions of affect in both face and voice [9] such as laugh, smile. People usually show facial expression while they are talking. These facial expression usually shows speaker's affective states such as joy, sorrow, humour, distress and so on.

2.5 Audio-Visual Biometrics

Chapter 3

Processing and Methodologies

In this chapter, I will introduce the procedure of extracting facial features and the tools and methodologies I use.

3.1 Processing Flow

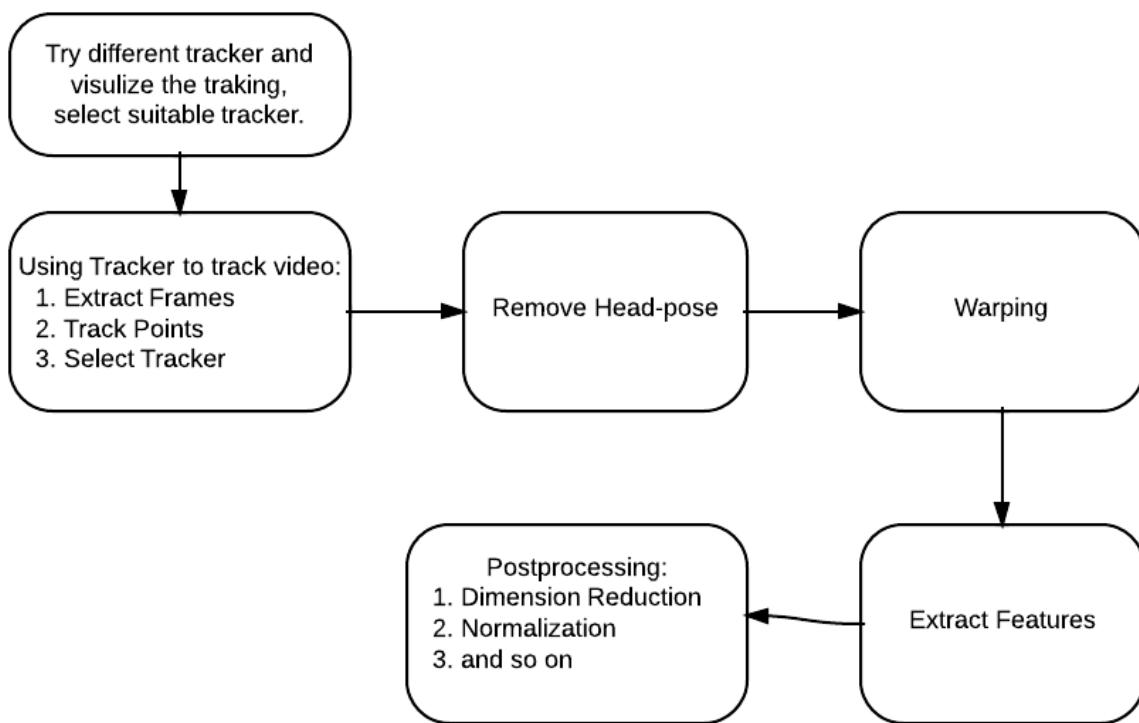


Figure 3.1: Main Procedure

Processing Chart Figure 3.1 shows main procedure of the whole project. At beginning, I tried several trackers. Intraface and DRMF are two trackers I tried and compared most. Two trackers are using different methods and also implemented in different languages. Intraface are programmed in c and matlab and has great interface for matlab. I tried two version of DRMF, DRMF programmed using CUDA which uses parallel processing is quite fast. As the programme of DRMF doesn't integrate extract frames from videos. The images are extracted using external function, then tracked using DRMF. I choose Intraface as the final choose, the reason and comparison will be given in later section. Remove head-pose seems to be a very important part for this project, as subject's head moves frequently in many videos. After having tracking points without head-pose,

each face in each frame is warped and scaled to same size grey image. Extracting features is to extract appearance feature of each face in the image. Post-processing is preprocessing before using the data for classification.

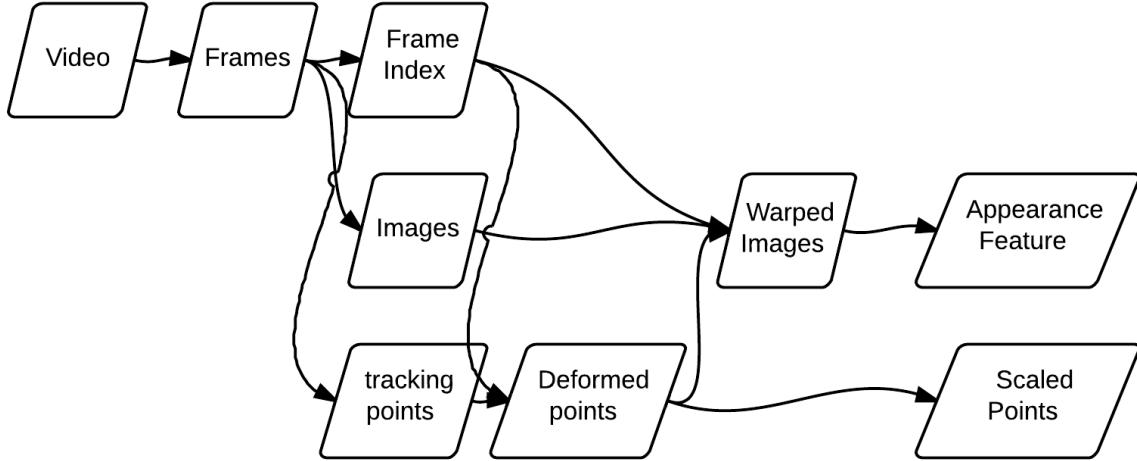


Figure 3.2: Data Flow

Data flow Chart Figure 3.2 show data I need for processing. There are two types of encoded video, one is in format of fly and the other is avi. Extracting frames from videos is proceed with Intraface and stored in formats of jpeg and mat which used for processing of matlab. There are several situations that a track is unable to track a face in the image such as no subject in the image, the head is face to a very large angle from frontal face, face is partially not show in the frame. Frame index points to those images which the tracker is able to tracker a face in a frame. Image is the frame images stored in mat format. Different tracker may tracks different number of characteristic facial points. Intraface tracks 49 facial points and DRMF tracks 66 points. Deformed points is the tracking point after removed head-pose. Warped Images is the face after remove head-pose and background which only leaves the meshes build by tracking points. Appearance feature is face feature extracted using local binary pattern (LBP). As the image size from image to warped images are changed, the points is rescaled from deformed point to scaled points.

3.2 Face Alignment

Face alignment is to align face in one image with respect to the same face in another image. Face alignment techniques are used to track characteristic facial points in image sequences. In this project, the aim of face alignment is to localise the feature points on face images. The points are usually around eyes, nose, mouth, and outline. Face alignment techniques are essential on face recognition, modelling and synthesis. There are three main different approaches Parametrized Appearance Models(PAMs), Discriminative approaches, Part-based deformable models. Parametrized appearance models contains many models such as active appearance models (AAMs), morphable models, eigentrackings, and template tracking [15]. All these models are using Principle Component Analysis(PCA) method to parametrize a face. A face could approximately decomposed as linear combination of shape basis and appearance basis. The problem of face alignment could be refer as minimising the difference between the constructed PAM and the face. Common approach is use Gauss-Newton methods [15]. Discriminative approaches are to learn the linear regression between the head move and appearance change. Part-based deformable model perform face alignment by maximising the posterior likelihood of part locations given image [15].

3.2.1 Active Appearance Model

Active Appearance Model (AAMs) is defined as a generative model of a certain visual phenomenon in [6]. AAMs are conceptually related to morphable models, constrained models and active blobs. In this project, it is refer to a model of face. As AAM is conceptually related to other parameterized appearance model, so it is introduced as an example of parameterized appearance model for understanding purpose. According to [6], there are two types of AAMs, one refers as independent shape and appearance models, which model shape and appearance independently, and the other refers as combined shape and appearance models, which parameterized shape and appearance model with a single set of linear parameters [6]. Normally AAMs appears along with a fitting algorithm. However, in the following context, it only refers to a model. [6] gave a well explain about what is an AAM, most of following theory are from [6].

Shape Shape of a face s is defined by coordinates (x, y) of v vertices of face points and the mesh they built:

$$s = (x_1, y_1, x_2, y_2, \dots, x_v, y_v)^T \quad (3.1)$$

s also can be expressed as a base shape s_0 plus linear combination of n shape vectors s_i :

$$s = s_0 + \sum_{i=1}^n p_i s_i \quad (3.2)$$

Appearance For all pixels x in the mesh s_0 , appearance $A(0)$ can be expressed by base appearance $A_0(x)$ and m appearance images $A_i(x)$.

$$A(x) = A_0(x) + \sum_{i=1}^m \lambda_i A_i(x) \quad \forall x \in s_0 \quad (3.3)$$

AAMs are usually computed by applying Principle Component Analysis (PCA) to choose images. The chosen images contains a variety of shapes. The base shape s_0 is the mean shape and the vector s_v is the eigenvector corresponding to the largest v eigenvalues. The base appearance A_0 and the appearance A_i is computed by applying Principle Component Analysis to a set of shape normalised images.

Model $W(x : p)$ is the warp from s_0 to s . Then the model M set the appearance of $W(x : p)$ to $A(x)$.

$$M(W(x : p)) = A(x) \quad (3.4)$$

Combined AAMs

Combined AAMs just use parameter $c = (c_1, c_2, \dots)^T$ to parametrize shape:

$$s = s_0 + \sum_{i=1}^l c_i s_i \quad (3.5)$$

and appearance:

$$A(x) = A_0(x) + \sum_{i=1}^l c_i A_i(x) \quad (3.6)$$

An example of AAM instantiation is clearly shown in figure 3.3.

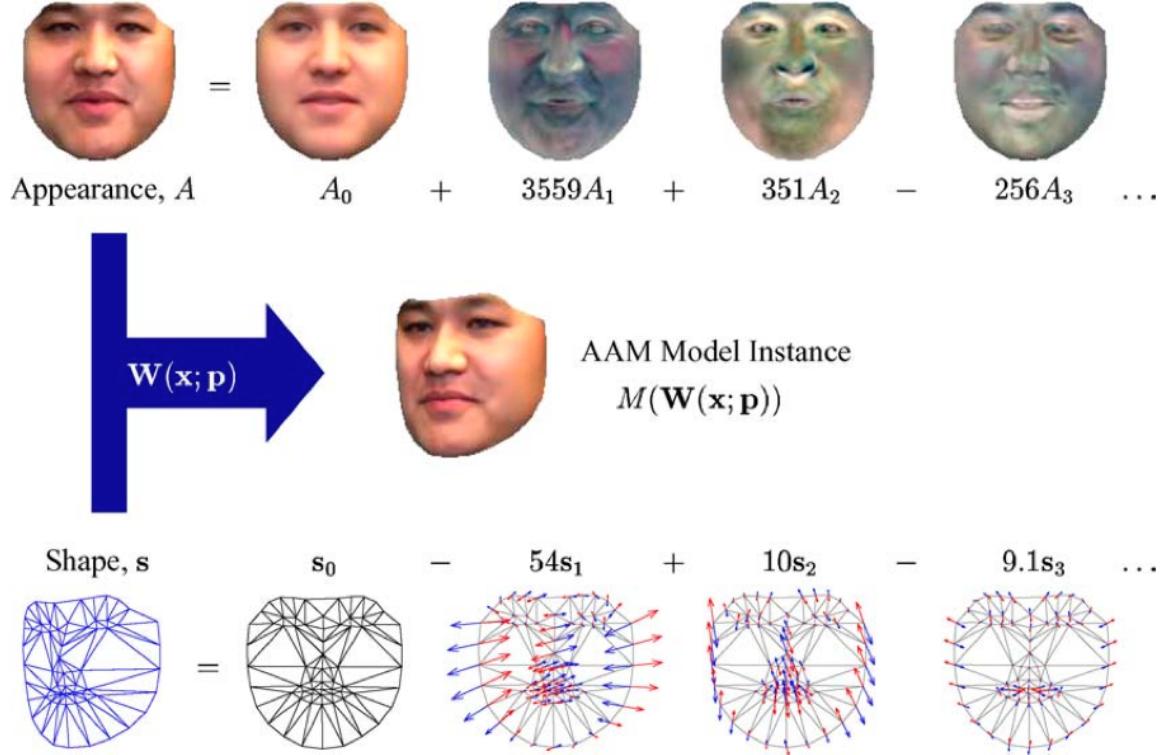


Figure 3.3: An AAM instantiation from [6]

3.2.2 Trackers

There are many different trackers for tracking facial feature points. Different tracker may using different approaches, so they are suitable for different situations. I tried two main trackers for tracking characteristic facial points, one is Intraface [15] which use supervised decent method, the other is DRMF [2] which use discriminative response map fitting. According to my using experiment, Intraface is very good at tracking motion face and DRMF is very good at fitting face model to a more standard face even with low resolutions. The number of facial points they track are also differnt.

Intraface

[15] implies image alignment can be posed as solving a nonlinear optimization problem. It uses Supervised Descent Method for minimising Non-linear Least Square(NLS) function, which avoids calculating the Hessian and the Jacobian that could be computationally expensive. For this reason, the running time of Intraface shows that the method is very effective and efficient.

Tracking Points Figure 3.4 shows the tracking points of Intraface. This tracker tracks 49 facial feature points. As you can see the eyes, nose, mouth, unfortunately the jaw and cheek may contain visual information that may help classification.

Eating and Talking Sequence Figure 3.5 shows a sequence of image of eating tracked by Intraface. The point are aligned very precisely along the face. Figure 3.6 shows a talking sequences of image tracked by Intraface. The landmark points of mouth is very accurate.

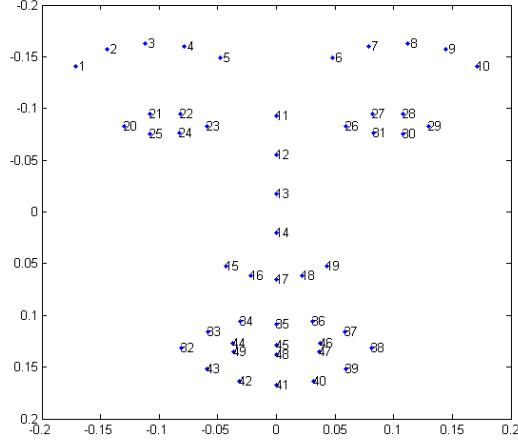


Figure 3.4: Intraface landmark points

DRMF

DRMF uses novel discriminative regression based on Constrained Local Models(CLMs) for face alignment [2]. The basic idea of DRMF is to fit a face for each frame of a video. After locating the position of a face, the tracker tries to fit a trained constrained local model to fit the face. Sometimes the fitting result is not very good and the landmark points of mouth region is not very accurate.

Tracking Points Figure 3.7 shows 66 facial feature points tracked by DRMF, the extra 17 points are the point around face bound. Other landmark points are at the same order as Intraface.

Talking and Talking Sequence Figure 3.8 and 3.9 show image sequences of eating and talking tracked by DRMF. It is easy to see that the facial feature points are not aligned as better as Intraface. However, the advantage of DRMF is that with the extract bound points of face, we are able to extract the visual information about the jaw and cheek, which may be helpful for classification.



Figure 3.5: Eating sequence tracked by Intraface



Figure 3.6: Talking sequence tracked by Intraface

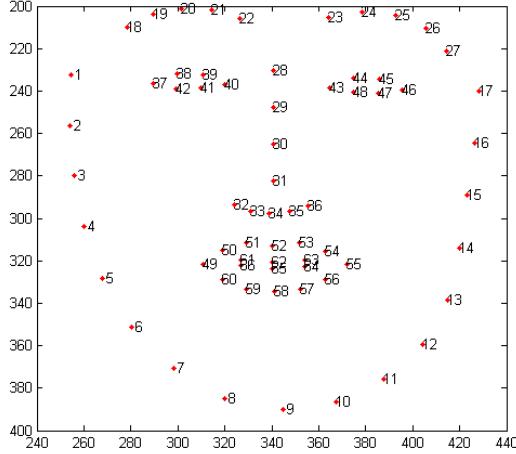


Figure 3.7: DRMF landmark points

3.2.3 Comparison

The following are some examples for comparing two trackers. Intraface is generally better than DRMF in accuracy and efficient. There are two version of DRMF tracker one is implemented by CUDA language the other is by C language. Although the C version of DRMF is very slow and very easy to run out of memory, the version implemented by CUDA is very fast as CUDA is using parallel computing. However, for face points alignment, DRMF is not as accurate as Intraface. In some situations, DRMF try to fit a face and the fitting result is awful. The images with red points are tracked by Intraface, and the images with blue points are tracked by DRMF.

In figure 3.10, compare top left and top right images, we can see Intraface does not track the face of the smaller face, DRMF tracked the smaller face instead of the large face. This shows the advantage of DRMF on tracking face of low resolution face. I tried to use Intraface to track a image with multiple face, Intraface is able to track multiply faces in one image. So this means, Intraface is not good at tracking face on low resolution face. Compare the lower two images on the right hand side, the feature points of nose is not proper aligned, this is caused by imperfect of the fitting algorithm. The feature points of face bound are also not well aligned. Comparing the alignment of eye alignment on the left hand side of both traker, Intraface is better than DRMF. The feature points of right eye brows of the mid right image are also not aligned precisely.

In figure 3.11, in the top left image, DRMF is unable to fit a face to the face as the face is facing to the left. Of course, Intraface is unable to apply face alignment to a image doesn't show half of the face like this, it choose to ignore this image. In lower two images on the left hand side shows two situations that DRMF fail to fitting face model to face because of the face is partial out of the image frame. The middle one shows that the fitting face points are forced to stay in the frame and the mouth are moved up to close nose. The lower one image shows that as most of mouth region is out of frame, the model is fitted to the left eye forcefully. The middle and lower images on the right hand side shows that Intraface would ignore those points that are out of frame. It seems using Supervised Descent Method is very good at tracking face that moves. However, there is a bug of this traker, while tracking the video, if the subject moves hand to cover the mouth, the points of mouth would be pushed upwards, and it will not come back even the hand is moved away, unless, the traker loss the face and retrack the face.



Figure 3.8: Eating sequence tracked by DRMF



Figure 3.9: Talking sequence tracked by DRMF



Figure 3.10: Tracking result: images with red points on the left hand side are tracked by Intraface and images with blue points on the right hand side are tracked by DRMF



Figure 3.11: Tracking result, images with red points on the right hand side are tracked by Intraface and images with blue points on the left hand side are tracked by DRMF

3.3 Remove Head-pose

Subjects were unrestricted while they were recording the video, their head pose are very different. The coordinate of facial feature points are different for each video frame. In order to unite the head, we need to remove the head pose from the coordinates of facial feature points. If define the horizontal direction is x, the vertical direction is y and z is the direction, subject facing camera. It is very easy to remove head-pose in x-y direction, just by rotating and scaling the points would remove the head-pose. If the subject had a head-pose in x-z and y-z direction, it would be hard to find the correct transformation matrix for tracked facial points. The algorithm of removing head-pose from tracking points is [12]. Basically, it has a deformable 3-D Constrained Local Model(CLM), minimize the error of fitting the model and the 2D points and then remove head-pose and give the new 2D points. Figure 3.12 gave two example of original track points and deformed points. From figure 3.12, we can see this method could remove head-pose without losing information of mouth shape.

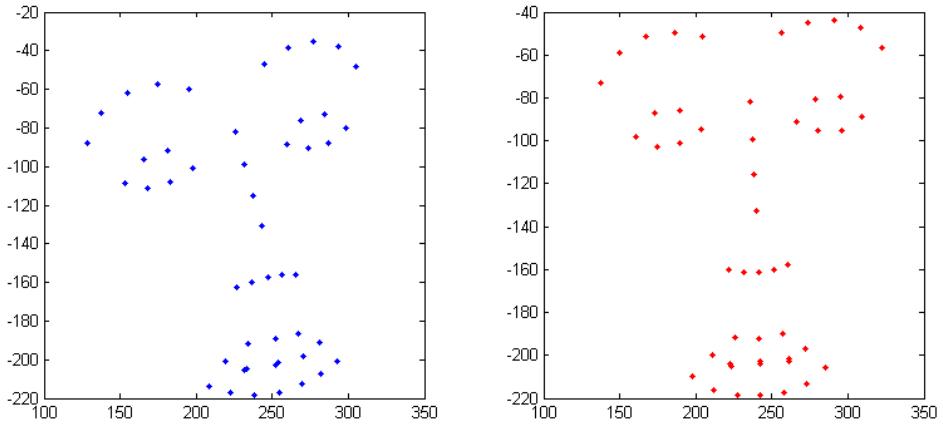


Figure 3.12: Traking points and Deformed Points Example

3.4 Warping

There are two types of features we need to extract, shape feature and appearance feature, the previous section just remove head-pose of shape feature vector. The appearance of face on image is not correctly showing a face that could be directly used for classification, as the face is not facing to frontal. In order to have head-pose free face, we need to distort the image to the one we need. In here I use Piece-wise Affine [14] to warp the image. According to the model in Active Appearance Model. We need to warping image from one to another with respect to the mapping of shape feature points set p_1, p_2, \dots, p_n into another point set p'_1, p'_2, \dots, p'_n . Each point is represented as $p = [x, y]^T$. The mapping function would be:

$$f(p_i) = p'_i \quad \forall i = 1 \dots n \quad (3.7)$$

Piece-wise Affine assume that f is locally linear. In 2-D framework like 2-D AAM, first find the mesh-grid constructed by all shape feature points. Delaunay triangulation is a good way to express it. In Delaunay triangulation, there is no points inside its circumcircle. Then the problem is to find the mapping function f to map the triangular mesh of the first point set and the second point set. For each points on each triangle of the first point set I can be mapped to unique points on each triangles of the second point set I' by affine transformation, which is combination of translation, rotation and scaling. Assume p_1, p_2 and p_3 are vertices of a triangle in I , then the points inside

the triangle can be written as:

$$\begin{aligned}
 p &= p_1 + \beta(p_2 - p_1) + \gamma(p_3 - p_1) \\
 &= (1 - \beta - \gamma)p_1 + \beta p_2 + \gamma p_3 \quad \text{assume } \alpha + \beta + \gamma = 1 \\
 &= \alpha p_1 + \beta p_2 + \gamma p_3
 \end{aligned} \tag{3.8}$$

As points p are in the triangle, so $\alpha \geq 0, \beta, \gamma \leq 1$. Then the corresponding points p' in image I' inside corresponding triangle would also satisfy the equation:

$$p' = f(p) = \alpha p'_1 + \beta p'_2 + \gamma p'_3 \tag{3.9}$$

We have the three points of a triangle, it is easy to determine the value of α, β and γ by solving two linear equation for a known points, $p = [x, y]^T$:

$$\begin{aligned}
 \alpha &= 1 - (\beta + \gamma) \\
 \beta &= \frac{yx_3 - x_1y - x_3y_1 - y_3x + x_1y_3 + xy_1}{-x_2y_3 + x_2y_1 + x_1y_3 + x_3y_2 - x_3y_1 - x_1y_2} \\
 \gamma &= \frac{yx_2 - xy_1 - x_1y_2 - x_2y + x_2y_1 + x_1y}{-x_2y_3 + x_2y_1 + x_1y_3 + x_3y_2 - x_3y_1 - x_1y_2}
 \end{aligned} \tag{3.10}$$

Equation 3.10 give the function to calculate α, β, γ , with the Equation 3.9, we can calculate all the points in image I' . Pseudo-code of piece-wise affine warp would be:

1. For each point $p = [x, y]^T$ inside image I
2. Determine which triangle p is in
3. Use equation 3.10 to calculate α, β, γ
4. Use equation 3.9 to calculate the corresponding position of p'
5. set the value of p' in I to the value of p in I
6. end

The follow is the warping result, the image was not highly distorted as the face is basically facing to the frontal. However, it is easy to notice the face is a little rotated to the right and there is a small change on the shape of mouth.

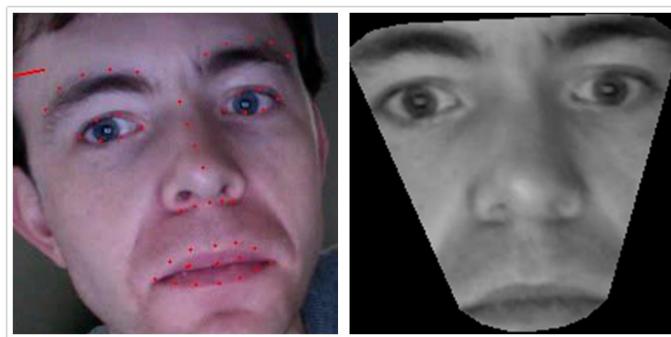


Figure 3.13: Talking sequence tracked by DRMF

3.5 Feature Extraction

The warped images are not directly used for classification, as the image value are sensitive to illumination, and noise. In addition, the dimensionality of image vector could be very large. An effective facial representation from original image is vital for classification. Experiments of [13] shwos that Local Binary Pattern(LBP) features are effective and efficient for facial expression recognition. The best performance of [13] is obtaine by combing Support Vector Machine and Boosted-LBP features. [13] also shows that LBP feature perform stably and robustly on low-resolution facial expresion recognition. The most important reasons of chosen LBP are: firstly LBP is robust to monotonic changes in illumination shown in figure 3.14; secondly, computational simplicity, the calcualtion only contain integer math and there is no need for preprocessing; thirdly, the time complex is $O(n)$.

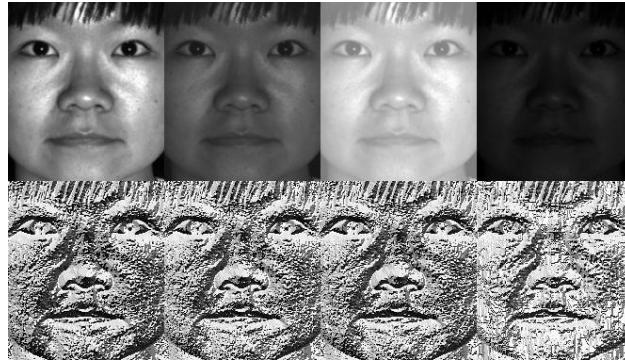


Figure 3.14: LBP Example¹

3.5.1 Local Binary Pattern

Texture is an imprtant characteristic of images and videos. 2-D surface can be charaterized by two properties: pattern and contrast. In the warping step, the image was transformed to gray scale and we need a gray scale and rotation invariant pattern to measure. LBP is invariant to any monotonic gray level change and it is easy to compute. Ojala et al(1996) introduced the fist generation of LBP operator [13], it was proved to be a powerful texture descriptor. Figure 3.15 show how to calculate LBP of a pixel and the contrast of the piexl.

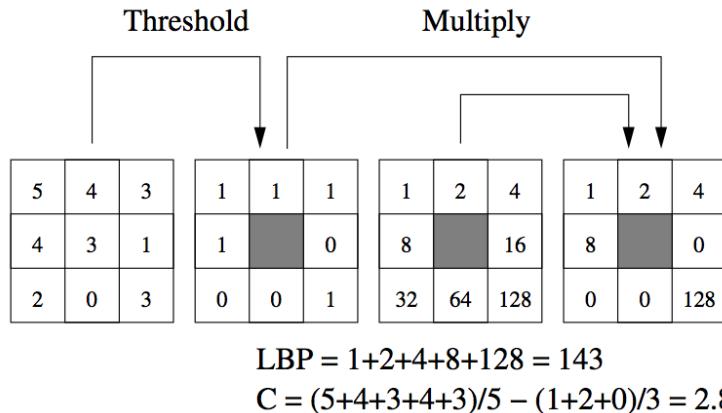


Figure 3.15: An Example of Calculating LBP code and a contrast measure of single pixel, image from [5]

The binary number could be used for coding types of curved edges, spots, flat area, etc as shown in figure 3.16. Figure 3.16 is using circular pattern. In implmentation, the on-grid points($g_0 - g_4$ points

¹LBP Example From http://docs.opencv.org/modules/contrib/doc/facerec/facerec_tutorial.html#id22

in figure 3.17) directly use sample data for calculation, but the off-grid points(the points in the second grid($P = 8, R = 1$), not in grid($P = 4, R = 1$) in figure 3.17) are calculating using interpolation. The 3X3 structure can not capture dominant feature for large scale structure [13]. So Ojala et al extend LBP to multiple scales as shown in figure 3.17.

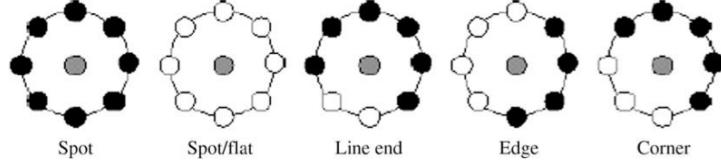


Figure 3.16: An Example of feature type coded by LBP and black circles present ones and white circle represent zeros, image from [13]

Define the operator as $LBP_{P,R}$. Define texture T in local neighborhood as the joint distribution of gray level of P image pixels, R is the spatial resolution to the central pixel.

$$T = t(g_c, g_0, \dots, g_{P-1}) \quad (3.11)$$

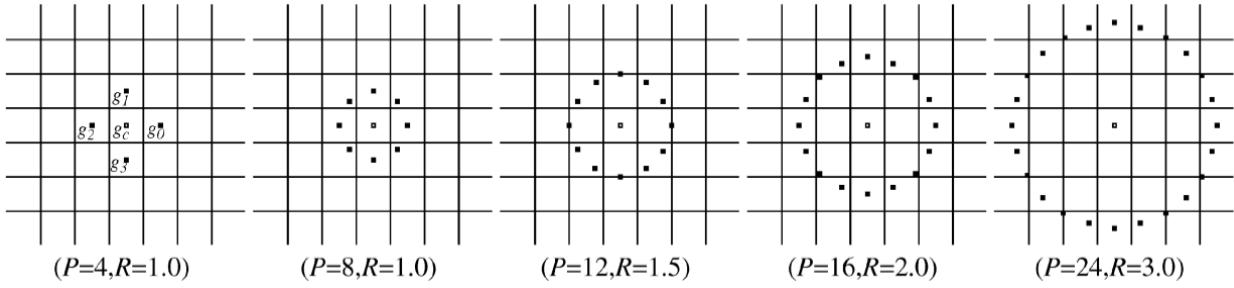
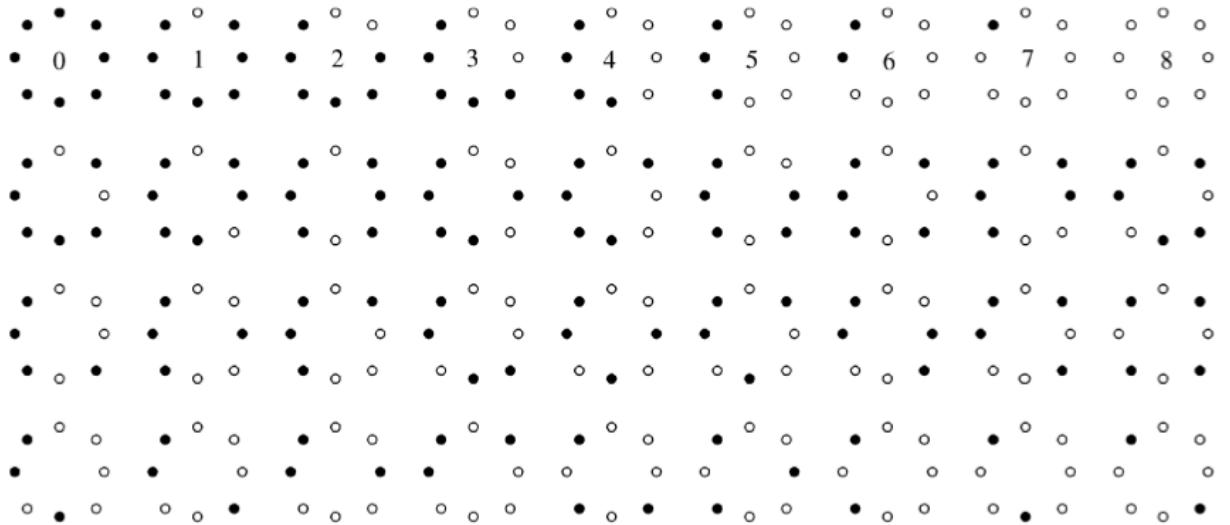


Figure 3.17: An example for multiscale circular pattern, image from [7]

Ojala et al proposed multiscale LBP in [7], which could be used for arbitrary circular neighborhoods and multiple scales; extend opertor, uniform patterns, which can help reduce the feature vector. Uniform Local Binary Pattern is defined as patterns with at most two contiguous regions. The uniform pattern is as shown in figure 3.18. Uniform Patter have significant effect on dimension reduction as the $256(2^8)$ elements can be reduced to 59 elements, with $7*8$ (uniform pattern from 1-7 in figure 3.18, as there are 8 directions) + 2 (uniform pattern 0 and 8)+ 1 (represent nonuniform patterns). In addtion, most natural images are uniform LBP and uniform pattern is more robust than non-uniform pattern that it perform better result in many applications.



*Figure 3.18: The first row are nine uniform patterns, Black represent 1, white represent 2,
image from [7]*

3.6 Postprocessing

As it is known large margin classifiers are sensitive to the way features are scaled, it's better to normalize either the data or the kernal function [3]. Feature of a image is represented by a vector, the number in the vector would influence the weight of feature in this dimension. As I would like to treat each dimension similar, I scale the number in the range of $[0, 1]$.

Normalization The performance of SVM is usually better if the data is normalized. There are two ways of applying normalization, standardizing the input features or normalizing the kernal function. As I am using the builtin function of libsvm [4], so I standardising the input features by subtracting its mean and divide by its standard deviation.

Scaling The range of appearance feature vector and shape feature vector is different. I would like to treat them as the same. So I scale all the vector into the range of $[0, 1]$, by subtract the minimum and divide by the maximum number of each dimension.

Chapter 4

Experiment and Results

4.1 Data

There are 450 videos in two formats, avi and flv, 89 of them are flv videos, 361 of them are avi videos. The time of videos are from seconds to dozens of seconds. Videos were recorded using web-cam of different PCs. Video background are vary as the video are recorded in a place chosen by subject. People are free to do anything while recording the video, I even observe a male disappear from the camera for half of the video sequence while recording. As a result, there are no faces in most of the video frames. The Intraface tracker [15] is used to track those videos. It is able to track 439 videos. 1 video is tracked, but the tracking result does not match the label. 10 videos are tracked, but unable to identify the face in the video. The face in those untracked video can be clear identify by visual. One possible may be the resolution of the image. Maximum untracked video frame is 240 * 960 pixels. Minimum tracked video frame size is 240 * 960, the same as the maximum size of untracked video frame. It is reasonable to say tracker [15] may not be good at tracking low resolution videos. The observation mentioned in comparing tracker [15] and tracker [2] also support this hypothesis. The maximum frame tracked is 600 * 2400. For each video there is a label file indicate the label of each frame, The frame number of each video is quite different, from around 200 to more than 900. The frame rate is 30 fps. Table 4.1 shows the tracking result of tracker [15].

Label Title	Normal Face	Eating	Talking	Looking Away	Occluded	Other Problem
Total	35361	10409	5623	7730	21394	8422
Tracked	33776	9460	5196	5405	19014	3884
Rate	0.9552	0.9088	0.9241	0.6992	0.8888	0.4612

Table 4.1: Frame tracking result by tracker [15]

There are six labels for each frame, normal face, eating, talking, looking away, occluded, other problem. A frame labelled as eating belong to a image sequence of eating. Label normal face, eating, talking and looking away are disjoint, but one frame can be labelled as one of them and occluded or other problem. It is very hard to track a face that face to camera from a certain angle, so the tracking rate is very small for a face looking away. I only use frame with label normal face, eating, talking in the experiment. Not all three labels are included in all videos, most of videos miss one or two or even all three labels.

4.1.1 Feature

Each face is aligned with 49 facial feature points as shown in figure 3.4. As the tracker doesn't provide face bound point, so I am unable to include jaw and cheek in the ROI. I just kept the mouth as Region of Interest. I extract Local Binary Pattern feature from ROI. The size appearance feature vector are different if I divide image of ROI into different number of blocks. I tried dividing ROI into 1 block and 1 * 3 blocks, size of appearance feature vector is 95 and 177. As there are 49 shape

feature points, so the size of shape feature points is 98.

4.2 Methodology

In the classification part, I just tried to use Support Vector Machine to do classification. I tried two different type of appearance feature vectors, their dimensionality is 95 and 177, to see whether with more detailed appearance feature vector would be better for classification. I would like to know whether apply normalisation to each video would improve the classification. I tried two different feature vectors, one is normalized by each video and then normalized together, the other is just normalized together. So there are 10 groups of experiment.

Normalisation (True if By Video)	T	T	T	T	T	F	F	F	F	F
Appearance Feature (Divide by 1 or 3 Block)	1	1	3	3	1/3	1/3	1	1	3	3
Feature A: Appearance S: Shape	A	A+S	A	A+S	S	S	A	A+S	A	A+S

Table 4.2: Experiments

I tried both linear kernel function and non-linear kernel, the Gaussian Kernel shows better result. Gaussian and polynomial kernels often leads to over-fitting in high dimensional database, while linear kernel is easier to tune because the only parameter that affects performance is the soft-margin constant [3]. The most important parameters for Gaussian Kernel is penalty parameter c and γ in equation 4.1.

$$K(x, x') = e^{-\gamma \|x - x'\|^2} \quad (4.1)$$

4.2.1 Find Parameter c and γ

A general way to find parameter c and γ is using cross-validation and grid-search. In n-fold cross-validation, first equally divide the data into n fold, leave out one fold of data as testing data and use other $n - 1$ fold of data to train the classifier. Thus all the data is predicted once and the cross-validation accuracy is the percentage of data are correctly classified. Grid-search is try various pairs of c and γ and choose the one with best cross-validation accuracy. Grid search approach is very simply and the computational time is no more than advanced method. To shorten the time of grid search, it is better to search with a coarse grid and then proceed with a more specific search in the identified grid.

4.3 Results and Analysis

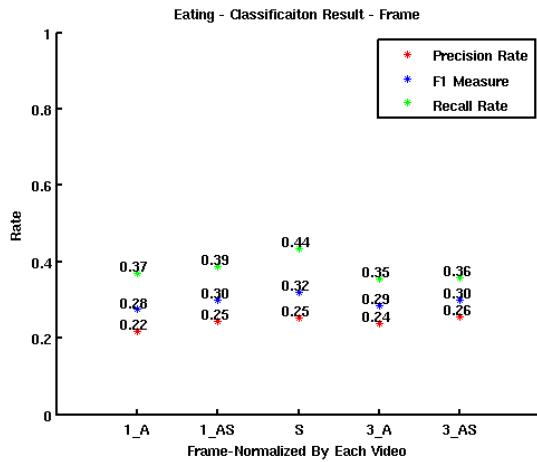


Figure 4.1: Class Eating - Classification Result of Frame - Frame nomalised by each video

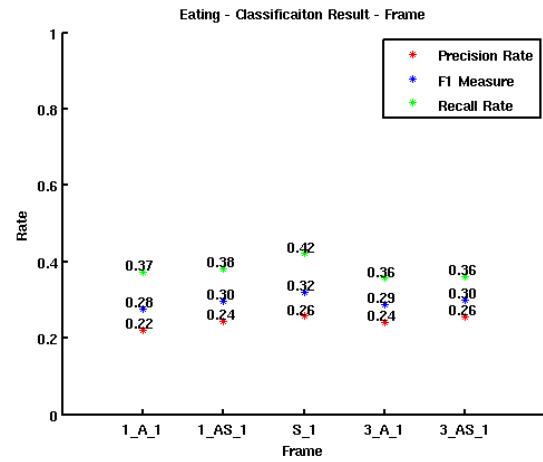


Figure 4.2: Class Eating - Classification Result of Frame - Frame NOT nomalised by each video

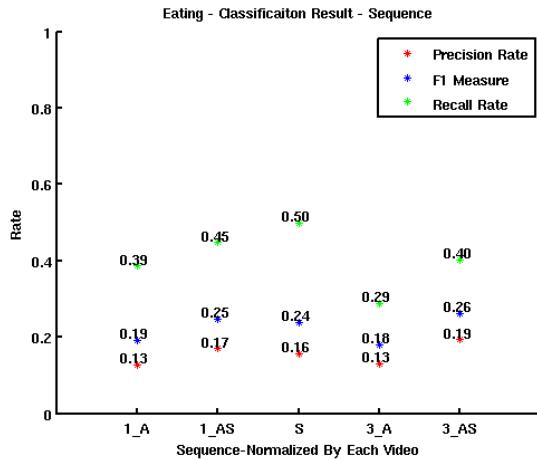


Figure 4.3: Class Eating - Classification Result of Frame - Frame nomalised by each video

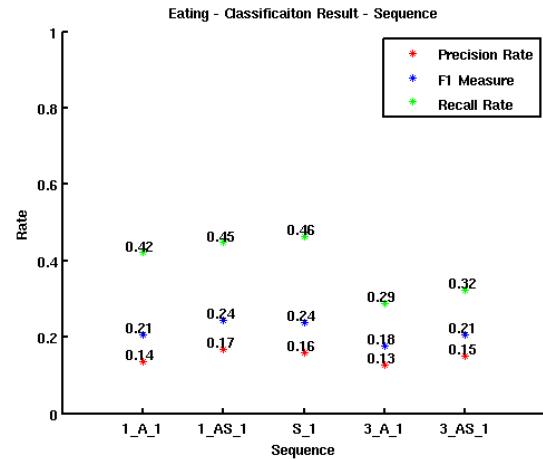


Figure 4.4: Class Eating - Classification Result of Frame - Frame NOT nomalised by each video

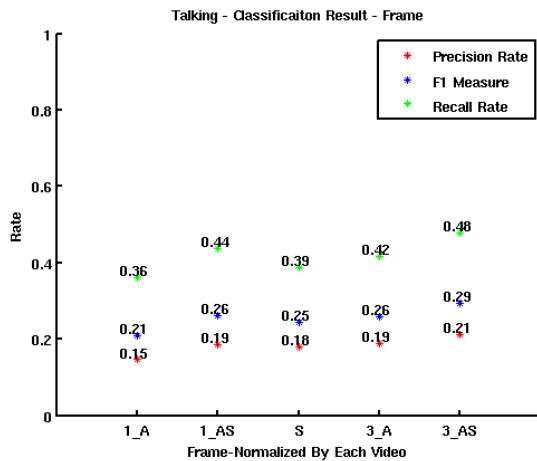


Figure 4.5: Class Talking - Classification Result of Frame - Frame nomalised by each video

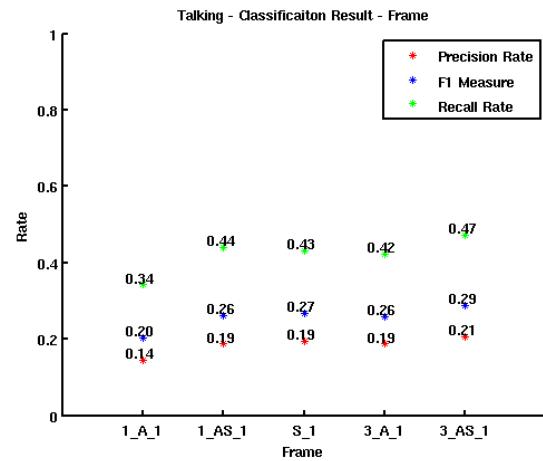


Figure 4.6: Class Talking - Classification Result of Frame - Frame NOT nomalised by each video

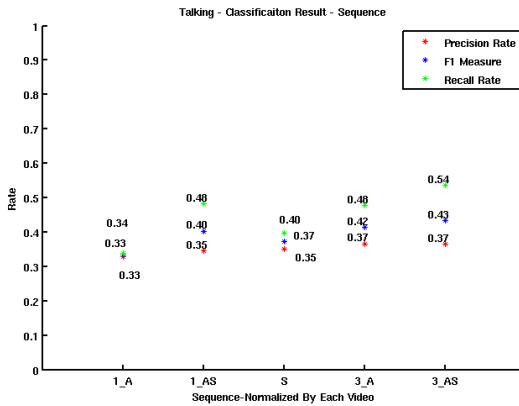


Figure 4.7: Class Talking - Classification Result of Sequence - Frame nomalised by each video

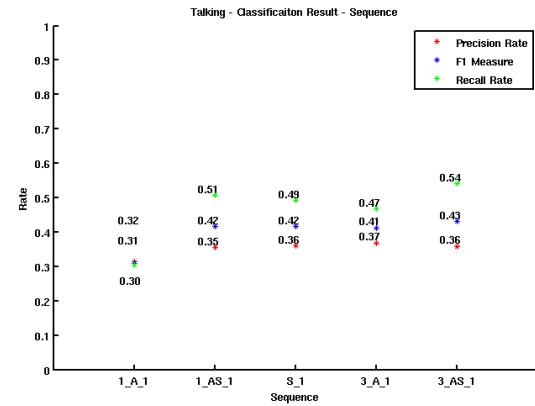


Figure 4.8: Class Talking - Classification Result of Sequence - Frame NOT nomalised by each video

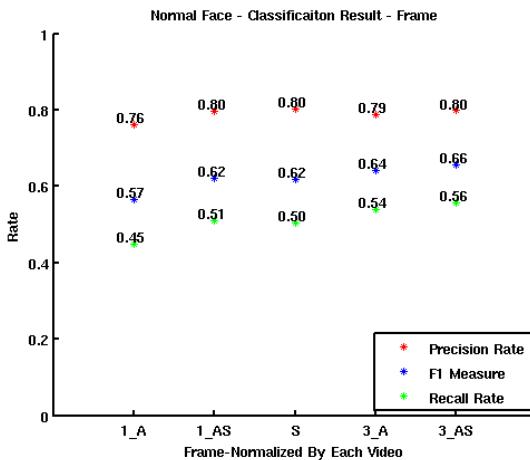


Figure 4.9: Class Normal Face - Classification Result of Frame - Frame nomalised by each video

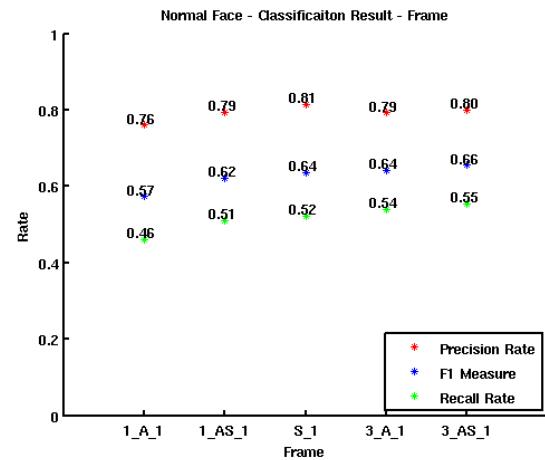


Figure 4.10: Class Normal Face - Classification Result of Frame - Frame NOT nomalised by each video

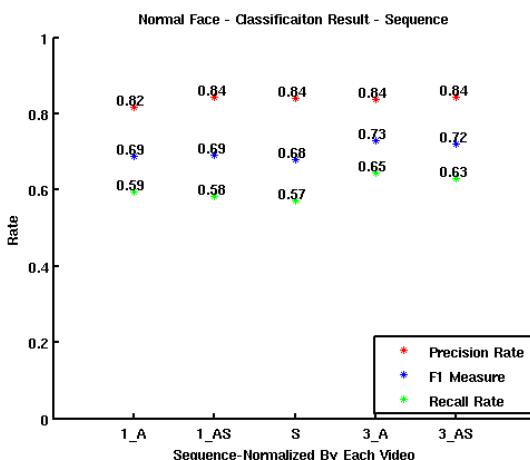


Figure 4.11: Class Normal Face - Classification Result of Sequence - Frame nomalised by each video

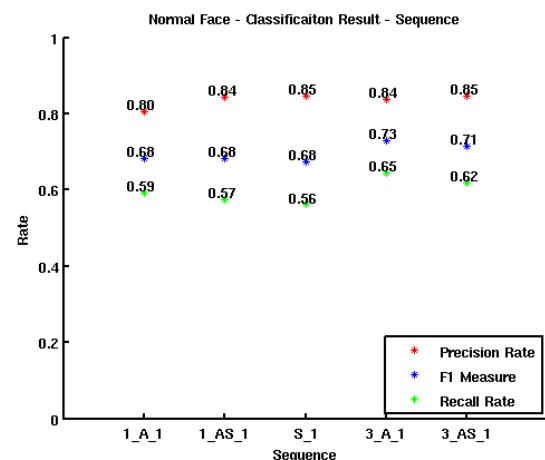


Figure 4.12: Class Normal Face - Classification Result of Sequence - Frame NOT nomalised by each video

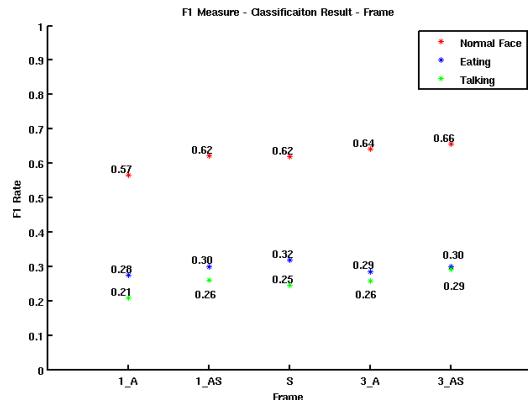


Figure 4.13: Three Class - Classification Result of frame - Frame normalised by each video

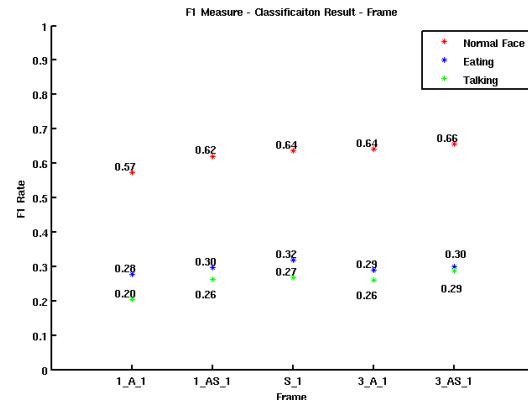


Figure 4.14: Three Class - Classification Result of frame - Frame NOT normalised by each video

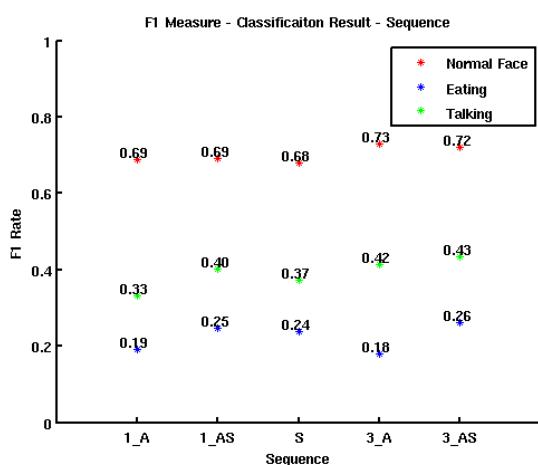


Figure 4.15: Three Class - Classification Result of frame - Frame normalised by each video

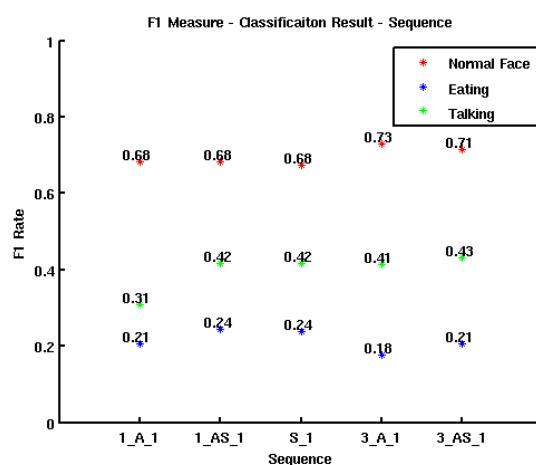


Figure 4.16: Three Class - Classification Result of frame - Frame NOT normalised by each video

Chapter 5

Conclusion and Future Work

Bibliography

- [1] Petar S Aleksic and Aggelos K Katsaggelos. Audio-visual biometrics. *Proceedings of the IEEE*, 94(11):2025–2044, 2006.
- [2] Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. Robust discriminative response map fitting with constrained local models. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3444–3451. IEEE, 2013.
- [3] Asa Ben-Hur and Jason Weston. A users guide to support vector machines. In *Data mining techniques for the life sciences*, pages 223–239. Springer, 2010.
- [4] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.
- [5] Topi Mäenpää and Matti Pietikäinen. Texture analysis with local binary patterns. *Handbook of Pattern Recognition and Computer Vision*, 3:197–216, 2005.
- [6] Iain Matthews and Simon Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.
- [7] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.
- [8] Maja Pantic and Marian Stewart Bartlett. Machine analysis of facial expressions. 2007.
- [9] Stavros Petridis and Maja Pantic. Audiovisual discrimination between speech and laughter: Why and when visual information might help. *Multimedia, IEEE Transactions on*, 13(2):216–234, 2011.
- [10] Stavros Petridis, Maja Pantic, and Jeffrey F Cohn. Prediction-based classification for audio-visual discrimination between laughter and speech. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 619–626. IEEE, 2011.
- [11] Gerasimos Potamianos, Chalapathy Neti, Guillaume Gravier, Ashutosh Garg, and Andrew W Senior. Recent advances in the automatic recognition of audiovisual speech. *Proceedings of the IEEE*, 91(9):1306–1326, 2003.
- [12] Jason M Saragih, Simon Lucey, and Jeffrey F Cohn. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91(2):200–215, 2011.
- [13] Caifeng Shan, Shaogang Gong, and Peter W McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803–816, 2009.
- [14] Mikkel Bille Stegmann. Texture model formulation, 2000.
- [15] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 532–539. IEEE, 2013.

Appendix A

Classification Result