Circuits and Systems
Mekelweg 4,
2628 CD Delft
The Netherlands
http://ens.ewi.tudelft.nl/

CAS-2018-4618335

M.Sc. Thesis

Audio-visual authentication for mobile devices

Lucas Montesinos

Abstract

Authentication is becoming an increasingly important application in the connected world and is driven by the growing use of mobile and IoT devices that use an increasing number of applications that require transactions of sensitive data. Security usually relies on passwords and/or two-factor authentication which are too intrusive for daily use. Biometric solutions such as fingerprints, voice, iris and retina are a good alternative to overcome previous problems. In this project an audio-visual identity verification is presented, where the use of multiple modes that can already be captured from most IoT devices (microphone and camera) make authentication robust to adverse conditions. End-factor analysis (i-vectors) with cosine distance is implemented as the main classification algorithm which takes into account variations within and between speakers. Mel Frequencies Cepstrum Coefficients (MFCC) are used as audio features, 2D-DCT coefficients of a single snapshot and Motion Vectors (MV) of the lips are extracted for visual features. Improvements combining different modes are shown using VidTimit dataset where the proposed algorithm achieves 0.7% of Half Total Error (HTER) in the test set outperforming single modes audio and visual by 9.5% and 6.4%, respectively.

TU Delft

Faculty of Electrical Engineering, Mathematics and Computer Science

Delft University of Technology

Audio-visual authentication for mobile devices

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Electrical Engineering

by

Lucas Montesinos
born in Santiago, Chile

This work was performed in:

Circuits and Systems Group
Department of Microelectronics & Computer Engineering
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Delft University of Technology

Delft University of Technology
Department of
Microelectronics & Computer Engineering

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled **"Audio-visual authentication for mobile devices"** by **Lucas Montesinos** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 29/08/2018

Chairman: ______________________________
prof.dr.ir. R. Heusdens

Advisors: ______________________________
prof.dr.ir. R. Heusdens

____
dr. N. Gaubitch

Committee Members: ______________________________
dr. F.R.S.T. D. Tax

____

Preface

The thesis "Audio-visual authentication for mobile devices" has been written to obtain the degree of Master of Science at TU Delft. The project was done together with Pindrop, a phone anti-fraud and authentication technology company, between November 2017 and August 2018. It was supervised by prof.dr.ir. R. Heusdens and Ph.D. student Andreas Koutrouvelis from TU Delft and dr. N. Gaubitch from Pindrop.

Acknowledgments

First, I would like to thank my advisor dr. N. Gaubitch , prof.dr.ir. R. Heusdens and Andreas Koutrouvelis for our weekly meetings which keep me on focus and organized along the whole project. Those meetings were essential for my progress given that most of the ideas, to not say all, presented in this thesis came from them. Nick always had new ideas to try or a interesting paper from which we could work on. He also spent last weeks preparing me for my final presentation and making sure the report was all right. Richard made me question every algorithm I was using and why I was using it which help me to have a critical vision on what I was implementing. Andreas always had kind words for my work which encourage me to continue every week. I also want to thank him for his time even when he was finishing his Ph.D. thesis at the same time.

Looking back to my experience in Delft I can only see good moments. The first year was tough, but I had never learned that much in my life. I want to thank all my professors whose teaching help me to understand and love the field where I am right now. Not only my academic life was good, friends I met here make my life easier. Coffee breaks and lunches made my life happier inside the university where I learned from different cultures and visions of the world. I want to thank all my friends with whom I share this two years, the barbacues and picnics in Delftse Hout, the birthdays and pre-birthdays parties we had and every time we met to have a couple of beers and talk about life.

I don't want to miss the opportunity to thank my family back in Chile, my parents Ramiro and Maria Cristina and my sister Celeste. They gave me the necessary support and courage to came to this part of the world to study what I love.

Finally, and most important, I would like to thank my wife Loreto, who did everything to make my life easier during this process. She is the reason I am here now, she convinces me to apply to this master even when she knew she couldn't work here. I can't imagine going trough this two years without her. Walking this path together is one of the best decisions we have made.

Lucas Montesinos
Delft, The Netherlands
29/08/2018

Contents

List of Figures

List of Tables

Nomenclature

2D-DCT	Two dimensional Discrete Cosine Transform
MFCC	Mel Frequency Cepstrum Coefficients
MV	Motion Vectors
STFT	Short Time Fourier Transform
UBM	Universal Background Model
VAD	Voice Activity Detector
ROC	Receiver Operating Characteristic
DET	Detection Error Trade-off
FAR	False Accpetance Rate
FRR	False Rejection Rate
EER	Equal Error Rate
HTER	Half Total Error Rate
AUC	Area Under the Curve
GMM	Gaussian Mixture Model
λ_{UBM}	GMM-UBM
λ_{Target}	GMM target speaker
F	Number of Features
C	Number of Gaussian
x	Feature vector
X	Feature space
Σ	Covariance Matrix
m	Mean supervector UBM
V	Eigenvoice Matrix
U	Eigenchannel Matrix
T	Total variability matrix
ω	Identity vectors

Introduction 1

Authentication is becoming an increasingly important application in the connected world. Some of the reasons for this are the growing use of mobile devices which provide access to personal information over the Internet [4] in order to perform payments or many other types of voice control for applications such as Amazon Echo, Siri, and generally in the space of Internet of Things (IoT), whereby 2020 there will be over 50 billion connected devices according to Cisco [5].

Authentication is the process of verifying if a person is who they claim to be. It consists of two main steps: enrolment and validation. First, each person is modelled using a target signal which is supposed to be unique to that person (password, voice, etc.) and in the second step a test signal is given and used to verify if the test signal matches the target signal.

There are three traits in verification:

- Something you know
- Something you have
- Something you are

All of them have some pros and cons. Passwords (something you know) have been the main method of authentication in the last decades, but the number of sites that need authentication increases every year which makes it impractically to remember (if a different password is used for each site) or risky (if the same password is used for all sites). Password management products appear as a practical solution but they are not attack-free which can expose information of more than a thousand people. Banks and other institution relies on secure-id tokens or devices (something you have) that are constantly changing codes that allow you to transfer money and pay bills, which can be slow and intrusive as you always need to carry the device with you. Biometric authentication (something you are) have obtained more significance due to previous problems, which extract features that are ideally unique for each person and hard to imitate. Examples of biometric authentication are fingerprints, face images, behavioural signals and voice.

In 2017 56% of all fraud in the UK were identity frauds which represent an increase of 5% from 2016. Most of these fraud incidents are bank related (accounts and cards) with more than 50.000, but also increasing in sectors like telecommunications (9.000) and on-line retail (5.000) with over 50% more compared to 2016 [6]. Improved voice security products have been one of the solutions to prevent attacks over the phone channel, where call centres fraud has increased from 1 in 2.000 calls to 1 in 937 from 2015 to 2016 [7].

1.1 Research Statement and Outline

This study investigates authentication by using voice and facial information from video sequences. This means that given a video of a certain person there are two possible answers: the person is the claimed identity (genuine) or is a different person (impostor). Using a multi-modal approach prevents that low quality in one of the modes affects the overall performance of the algorithm. On one hand, having low audio quality such as high level of noises and reverberation should make the algorithm rely more on the visual signal. On the other hand, if low video quality is present such as poor light conditions, face obstruction or blurred image the algorithm should rely more on audio information. Also, dealing with multiple sessions can be challenging: enrolment and validation can be done in different environments such as location, types of noises and device used to record the videos which will change the quality and values of the features extracted from both modes. Current voice solutions can suffer from factors like noise and reverberation that can degrade the performance of verification which can be overcome by including visual information of the person.

In this project, the implementation and evaluation of state-of-the-art algorithms are presented for voice (from an audio stream) and visual verification (from the corresponding video stream). The evaluation will consider each mode of authentication and the improvement that can be achieved by combining them. It is envisaged that the appropriated fusion of audio and visual queues will result in improved performance and reduced sensitivity to environmental effects such as acoustic noise and illumination variations.

The objectives can be summarized in two points:

- Study the improvement of combining audio and visual information for person authentication over single-mode algorithms.

- Analyse the combination of modes using feature fusion and late fusion (score).

Combining audio and visual features has shown improvements over single-mode algorithms [4] [8]. We introduce a three-mode authentication (obtained from audio, a single image of the face and motion vectors of the lips) that outperforms any combination of two methods and single mode algorithms.

Voice Activity Detection (VAD) plays a fundamental role in any voice-based system and especially in voice verification. We present a robust audio-visual VAD based on [9] and [10] which outperforms single mode VAD.

The remainder of this thesis is organized as follows: **Chapter 2** introduces an overview of the state of the art of voice and face verification. **Chapter 3** reviews several available data sets for this study and highlights the scarce availability of suitable data. **Chapter 4** describes how to combine audio and visual information to construct a robust Voice Activity Detection. **Chapter 5** shows experiments with VidTimit dataset [11] for single mode and multi-modal verification. Finally, conclusions and future work are presented in **Chapter 6**.

Background 2

Audio-visual person verification can be divided in two: voice and facial authentication. The state of the art algorithms for voice verification can be reduced to techniques using Mel Frequency Cepstrum Coefficients (MFCC) together with the energy of each frame as features and Gaussian Mixtures Model (GMM) for feature representation. Initially, a likelihood ratio detector was used [12] which was followed by more sophisticated solutions that use Factor Analysis to deal with session variabilities [13] [14] [15] and Deep Neural Networks (DNN) [16] [17]. End-factor analysis (i-vectors) is one of the most popular solutions which will be preferred in this work over DNN given that achieves similar results and is more flexible in terms of training size.

For visual id verification, there are several features and models that can be used. Most common features are pixel intensity [18], Local Binary Patterns (LBP) [19], 2D-DCT coefficients [2] and landmark position [1]. Classification can be done using histogram distances, Factor Analysis and classical classifiers like Logistic, Neural Networks and Support Vector Machine (SVM) together with dimensionality reduction techniques such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) [20]. For large datasets (over 100M images) DNN can achieve excellent results [21] and for small datasets (less than 1000 images) techniques using GMM and session variability [2] are preferred which will be used in this project.

2.1 Voice verification

2.1.1 Feature extraction

Feature extraction plays a fundamental role in any classification problem. Without appropriate features that are uncorrelated between classes, no classifier can achieve a good performance. In the case of audio signals, several features can be extracted, known as the energy of the signal, the pitch and MFCCs which have been widely studied and show the most robust performance for speech id verification.

The speech is usually segmented into frames of 20-30ms which are assumed to be wide sense stationary (overlapped windows are useful). Features are extracted from each frame where a Voice Activity Detector (VAD) is applied to discard non-speech frames from target voice frames. More details about VAD used for this project are presented in Chapter 4.

As mentioned before, MFCCs will be used as the main features for each frame given that they tend to provide better results. They are based on Mel scale that is a perceptual scale based on perceived pitches to be equal. This means that a pair of sounds which are perceptually equidistant in pitch are separated by an equal number of Mel [22]. A schematic for extraction of the MFCC is shown in Fig. 2.1 [23].

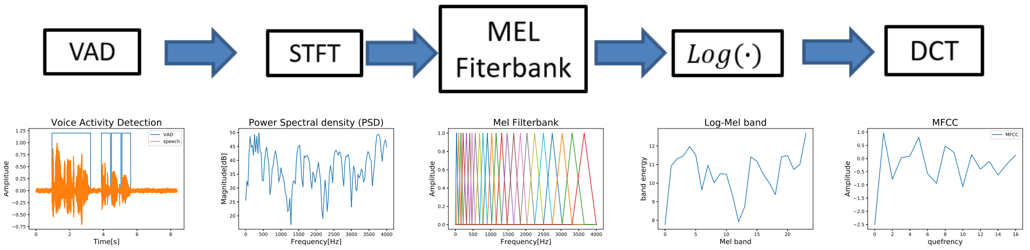

Figure 2.1: MFCC extraction steps

First, VAD is performed over the speech. For each voice frame, a pre-emphasis filter is applied and a Short Time Fourier Transform (STFT) is performed to obtain the energy over frequency bands. Then, a Mel filter bank is used followed by the log operation (human response to signal level is logarithmic [22]). Finally, the inverse Fourier Transform (IFFT) is applied which can be replaced with a discrete cosine transform (DCT) which takes only the real part leading to less computational effort and produce uncorrelated features. Several works [15] include the energy of the frame in the features which may improve the results [24]. Delta features are used (first and second order) which are calculated as the difference between the values of neighbouring frames and describe the temporal dynamics of the speech.

2.1.2 Channel dependent algorithms

2.1.2.1 Likelihood ratio test

Having a segment of speech Y, from a target speaker S the task is to determine if Y was spoken from S. This can be seen as a hypothesis test between:

H_0: Y is from the target speaker S.

H_1: Y is not from the target speaker S.

The optimum test when likelihood functions are known, to decide between these two hypotheses is a likelihood ratio test:

$$\frac{P(Y|H_0)}{P(Y|H_1)} \begin{cases} \geq \theta \ \textbf{Accept} \ H_0 \\ < \theta \ \textbf{Reject} \ H_0 \end{cases} \tag{2.1}$$

Where $p(Y|H_i)$, $i \in \{0, 1\}$ is the likelihood of the hypothesis H_i given the speech Y [12].

The model of this hypothesis test is shown in Fig. 2.2, where H_0 is fully represented by a model λ_{target} that characterizes the target speaker S by a feature space X which is compose of features vectors x_t for each frame t in Y ($X = [x_1, x_2, \ldots, x_T]$).

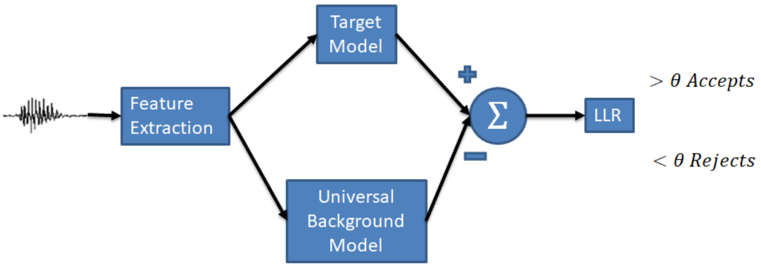

Figure 2.2: Log Likelihood Ratio (LLR) Detector for an unknown signal

For the target model λ_{target}, a Gaussian distribution is assumed over the feature vectors x_t, explained by the mean vector and the covariance matrix of the distribution.

The hypothesis H_1 is explained by the Universal Background Model (UBM) $\lambda_{\overline{target}}$ or λ_{UBM}, which represents the space of all the alternatives to speaker S. This approach performs better than comparing each target model, in terms of computational time for estimating each target model and differentiating them from the UBM. The UBM can be obtained in several ways, a commonly used approach is to train a single model from a large number of speakers (around 200 speakers and 3 to 6 hours of recordings) to represent general characteristics of speech [12].

One of the most successful likelihood functions for text-independent speaker recognition are GMMs [12]. The mixture model for a F dimensional feature vector x is given by the mixture density:

$$p(x|\lambda) = \sum_{i=1}^{C} \omega_i p_i(x) \tag{2.2}$$

Where C is the number of uni-modal Gaussian with mean vector μ_i $(F \times 1)$ and covariance matrix Σ_i $(F \times F)$ which is usually a diagonal matrix. ω_i is a weight parameter where $\sum_{i=1}^{C} \omega_i = 1$ and $p_i(x)$ is the uni-modal Gaussian density function:

$$p_i(x) = \frac{1}{(2\pi)^{F/2}|\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right] \tag{2.3}$$

The parameters of the model are denoted as $\lambda = \{\omega_i, \mu_i, \Sigma_i\}$ for $i = 1, \ldots, C$ and are estimated by maximum likelihood using an Expectation Maximization

(EM) algorithm which is an iterative algorithm where after each step k we have that $p(X|\lambda^{k+1}) > p(X|\lambda^{k})$. The main critical aspects of GMM are the order C of mixtures and the initialization of the EM algorithm [25], where K-means can be used. Assuming independence in the features vectors in X, the average of the log likelihood presented in equation 2.4 of the model λ can be used dividing by T which is the number of frames for a given utterance.

$$\log\left[p(X|\lambda)\right] = \sum_{t=1}^{T} p(x_t|\lambda) \tag{2.4}$$

2.1.2.2 Universal Background Model (UBM) and Speaker Model Adaptation

Training a GMM-UBM using a single model could be done by using several recordings and estimate the parameters via EM. Another approach is to use a combination of two models, one for male recordings and one for female [12]. For the target model, a common approach is to adapt the UBM by updating the parameters instead of training an independent model. While doing this, each target model will differ from the UBM enough to distinguish between genuine and impostor. The adaptation is done in a similar way as the EM algorithm. Expectation step is as in EM, where the expected value of the log-likelihood from equation 2.4 is calculated based on the current values of the parameters using Bayesian adaptation and sufficient statistics are obtained: [12].

$$Pr(i|x_t) = \frac{\omega_i p_i(x_t)}{\sum_{i=1}^{C} \omega_i p_i(x_t)} \tag{2.5}$$

$$n_i = \sum_{t=1}^{T} Pr(i|x_t) \tag{2.6}$$

$$E_i(x) = \frac{1}{n_i}\sum_{t=1}^{T} Pr(i|x_t)x_t \tag{2.7}$$

$$E_i(x^2) = \frac{1}{n_i}\sum_{t=1}^{T} Pr(i|x_t)x_t^2 \tag{2.8}$$

Then the MAP estimator is calculated for the GMM maximizing equation 2.4 with respect to each parameter and using old parameters to update its values:

$$\hat{\omega}_i = \left[\frac{\alpha_i^{\omega} n_i}{T} + (1-\alpha_i^{\omega})\omega_i\right]\gamma \tag{2.9}$$

$$\hat{\mu}_i = \alpha_i^{m} E_i(x) + (1-\alpha_i^{m})\mu_i \tag{2.10}$$

$$\hat{\sigma}_i^2 = \alpha_i^{\nu} E_i(x^2) + (1 - \alpha_i^{\nu})(\sigma_i^2 + \mu_i^2) - \hat{\mu}_i^2 \tag{2.11}$$

Where $\alpha_i^k = n_i/(n_i + \tau^k)$ for $k = \{\omega, m, v\}$ (weights, means and variance) and τ^k is a fixed factor for the parameter k. Parameter γ in equation 2.9 is a scale factor that ensure the weights to sum up to 1 [12]. A graphical representation of the adaptation model in 2-D is presented in Fig. 2.3, where each axis represents a feature, the dots are the means of each Gaussian (4 in the example) and the circles represent the standard deviation. Adaptation of only the means μ_i has shown better results [12]. In real applications, the number of Gaussian will be between 16 and 2048 which will depend on the number of speakers in the training set, and the number of features will be in the order of 20-70.

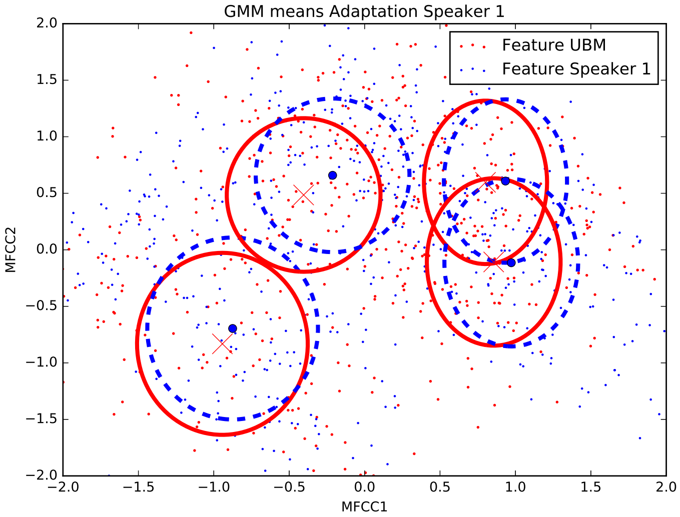

Figure 2.3: Voxforge GMM example. 4 GMM for the UBM (red) and speaker model adaptation of the mean (blue).

Finally, the match score for a given utterance is calculated using both models via a log likelihood ratio:

$$LLR(X, \lambda_{speaker}, \lambda_{UBM}) = \frac{1}{T} \sum_{t=1}^{T} \left[log(p(x_t|\lambda_{speaker})) - log(p(x_t|\lambda_{UBM})) \right] \tag{2.12}$$

There are some techniques which improve the computational time by evaluation the log likelihood ratio only in the top L scoring Gaussian of the UBM [12] or with linear approximations using first-order Taylor series [26].

2.1.3 Channel Independent algorithms

The models described in Section 2.1.2, target and UBM, can be explained completely by means, weights and variances of each Gaussian and feature $(\omega_i, \mu_i, \Sigma_i)$. It was explained that only the means are different between UBM and target model. This can lead to a new formulation of the same problem using *supervectors*, where the model can be explained by a vector M of length $CF \times 1$ (with C Gaussian and F features), having the information of all the Gaussian and all the features in one vector, therefore the name *supervector*.

Having the mean of C Gaussian and F features in one vector:

$$m = [\mu_{11}, \mu_{12}, ..., \mu_{1F}, \mu_{21}, ..., \mu_{2F}, ..., \mu_{CF}]^T \tag{2.13}$$

The model of speaker i, s_i, can be written as [27] [2]:

$$s_i = m + d_i \tag{2.14}$$

Where m are the means of the UBM and d_i is the client-specific offset.

2.1.3.1 Joint Factor Analysis (JFA) and Session Variability

One of the main problems in speaker verification is the information of the channel that is part of the speaker model which is not taken into account in the MAP estimation described in Section 2.1.2. The channel represents any variation that leads to changes in each model which is usually related to the session where recordings of a speaker were taken. Differences in terms of noises and reverberation could lead to a different model for the same speaker.

It can be assumed that the speaker and channel-dependent information can be represented in a supervector M, which can be seen as a speaker supervector s plus a channel supervector c, with normal distribution and independent of each other [13]:

$$M = s + c \tag{2.15}$$

The main idea is that decomposition techniques can find the directions of variability within each speaker and between speakers, and separate both spaces in the solution. The speaker model s can be decomposed as:

$$s = m + Vy + Dz \tag{2.16}$$

Where m is a CFx1 supervector (C components of the GMM model with F features), V is a rectangular matrix of low rank called *eigenvoice matrix* which is speaker dependent and represent the variation within speakers for different utterances and y is the speaker factor which is a normally distributed hidden random vector, D is a diagonal matrix of CFxCF and z is a normally distributed random vector of $CF \times 1$. Then, s is normally distributed with mean m and covariance matrix $D^2 + VV^T$. The channel supervector c is decomposed as:

$$c = Ux \tag{2.17}$$

Where U is a rectangular matrix of low rank called *eigenchannel matrix* which is channel dependent and represent the variations between sessions. Vector x is the channel factors which is a normally distributed hidden random vector analogue to y in the speaker space. Assuming $V = 0$ and $U = 0$ we are in the same case as the MAP described in Section 2.1.2.2 [12].

To estimate the parameters V, U and D the Baum-Welch statistics of the GMM posterior probabilities of each mixture i using the UBM model λ_{UBM} (weights ω_i, means μ_i and covariance matrix Σ_i) are used as described in [14]. First, the estimation of V is done assuming $U = 0$ and $D = 0$. This is done in an iterative way, were two main steps are done as explained in [15]:

1. For each speaker in the training set, and using the current state of V and covariance matrix Σ, we find the supervector y, that maximize the likelihood given the training features $X(s)$ of speaker s. This step is called "maximum likelihood decomposition".

$$y(s) = \arg\max\{P(X(s)|m + Vy, \Sigma)\} \tag{2.18}$$

2. Using the features of all speakers in the training set $X(s)$ we update V and Σ by maximizing the joint likelihood. This is called "maximum likelihood eigenspace".

$$\arg\max \prod_s P(X(s)|m + Vy(s), \Sigma) \tag{2.19}$$

U is computed using V and finally, the estimation of D is done using the previous results as explained in [13]. The posterior distributions of y, z and x (which are speaker and session dependent) are calculated using the MAP estimate of $m + Vy + Dz$. A more detailed explanation could be found in [14] [28].

2.1.3.2 Front-end Factor Analysis

Over last years, a variation of JFA combining both, speaker and channel variability (total variability space), has been used. This new method was developed after realizing that the channel information in JFA, also contains speaker information [15]. Here instead of finding the direction of variation of the speakers and channels, we find the direction of variation over all the utterances independently, having a matrix T that represents the new subspace. In this method, instead of having the model of equation 2.15, one new space is estimated called *"total variability space"* with the speaker and channel variability defined in matrix T:

$$M = m + T\omega \tag{2.20}$$

Where m is the speaker and channel independent supervector (taken from means μ_i of the UBM model). T is a rectangular matrix of low rank that represents the direction of variability of the utterances over all train speakers and ω is a random vector having a standard normal distribution $N(0, I)$. The vector ω is called the identity vector or i-vector, which is assumed to be unique for each speaker. This means that the speaker model M differs from other speaker models only by these vectors [15].

With this approach, the training of T is done in the same way we estimate V in JFA but now every utterance of a speaker is assumed to be produced by different speakers. The Baum-Welch statistics needed are the same as before and presented in equations 2.21 to 2.23, where x_t represents the feature vector of frame t for a given utterance and the UBM model λ_{UBM} is composed of C mixture components in a space feature of dimension F with means μ_i and covariance Σ_i.

$$N_i = \sum_t P(i|x_t, \lambda_{UBM}) \tag{2.21}$$

$$F_i = \sum_t P(i|x_t, \lambda_{UBM})x_t \tag{2.22}$$

$$\tilde{F}_i = \sum_t P(i|x_t, \lambda_{UBM})(x_t - \mu_i) \tag{2.23}$$

The expected value of the i-vector for a given utterance u is:

$$E\big[\omega(u)\big] = (I + T^t\Sigma^{-1}N(u)T)^{-1}T^t\Sigma^{-1}\tilde{F}(u) \tag{2.24}$$

Which is obtained by the expectation step on EM and proved in Appendix A.1. Maximization step is done as in JFA [14] to obtain final values T and Σ.

Where $N(u)$ is a diagonal matrix of dimension $CF \times CF$ with diagonal blocks N_iI. $\tilde{F}(u)$ is a supervector of dimension $CF \times 1$ which is the concatenation of $\tilde{F}_i$ for a given utterance u. Σ is a diagonal covariance matrix of dimension $CF \times CF$ with covariance matrix blocks Σ_i for each Gaussian.

2.1.3.3 Scoring and Channel compensation

The main method to use the i-vectors for speaker verification is with cosine similarity and the channel compensation can be done using Within Class Covariance Normalization (WCCN).

The cosine similarity between the target speaker i-vector ω_{target}, which correspond to the claimed identity, and the test i-vector ω_{test} is presented in equation 2.25 where θ is the decision threshold to accept or reject the speaker test [15] calculated in the development set.

$$score(\omega_{target}, \omega_{test}) = \frac{\langle \omega_{target}, \omega_{test} \rangle}{\|\omega_{target}\| \|\omega_{test}\|} \gtrless \theta \tag{2.25}$$

WCCN aims to minimize the expected error rate of false acceptance and false rejections [29] used for the cosine distance. The solution is given by the kernel:

$$k(\omega_1, \omega_2) = \omega_1^t R \omega_2 \tag{2.26}$$

Where R is the inverse of the within class covariance matrix W from equation 2.27, calculated over all speakers on the training background. This solution uses the W matrix to normalize the cosine kernel described in equation 2.26 [15].

$$W = \frac{1}{S} \sum_{s=1}^{S} \frac{1}{n_s} \sum_{i=1}^{n_s} (\omega_i^s - \bar{\omega}_s)(\omega_i^s - \bar{\omega}_s)^T \tag{2.27}$$

$$\begin{aligned} \omega_i^s &: \text{i-vector speaker s} \\ \bar{\omega}_s &: \text{mean i-vectors speaker s} \\ n_s &: \text{Number of utterances speaker s} \\ S &: \text{Number of speakers} \\ W^{-1} &= BB^t \\ k(\omega_1, \omega_2) &= \omega_1^t BB^t \omega_2 \end{aligned}$$

The final form of the kernel which compensate for intersession variability without changing the direction in space is [15]:

$$k(\omega_1, \omega_2) = \frac{(B^t\omega_1)(B^t\omega_2)}{\sqrt{(B^t\omega_1)^t(B^t\omega_1)}\sqrt{(B^t\omega_2)^t(B^t\omega_2)}} \tag{2.28}$$

A graphical representation of i-vectors in 2D using Linear Discriminative Analysis (LDA) dimensionality reduction is shown in Fig. 2.4 where five speakers are tested. Speakers are not completely separated in the graphs due to the projection in 2D space, which is been used just for visualization. The dimensionality reduction in real applications can vary from 10 to 300 features depending on the number of speakers.

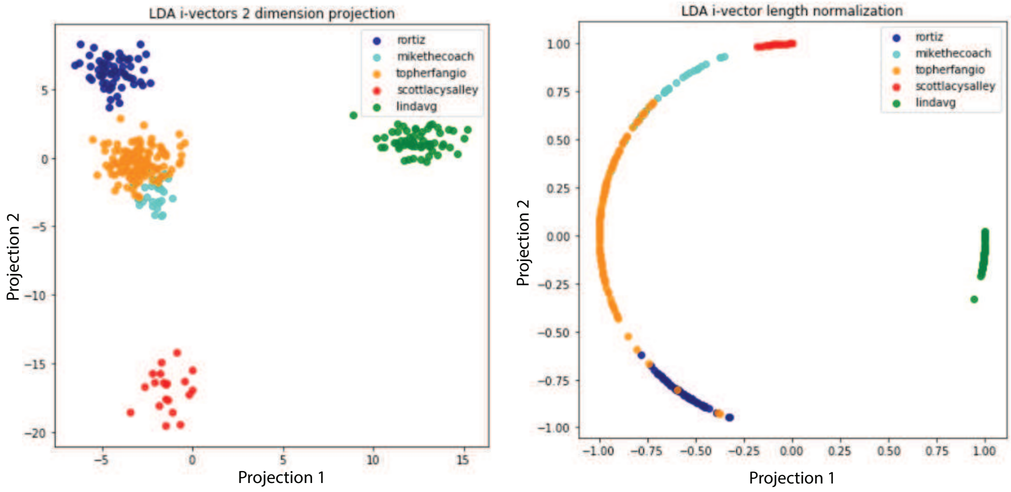

Figure 2.4: (left) LDA i-vectors 5 speakers. (right) LDA i-vector normalization.

2.2 Visual verification

The goal of face verification is to validate if a given face image matches a claimed identity. Similar to the case of voice verification, two signals are given, one face target image and one face test image and the algorithm should check if they came from the same person.

The main difference with speaker verification is the features that can be extracted from the images. In this case, we have two dimensions and there exist more types of features to select from.

Before doing face verification one needs to detect the face and process the image when having different lighting conditions and textures, known as normalization. After that, features are extracted and classification is done using different algorithms.

2.2.1 Face Detection and Normalization

The first step in face verification is to isolate the face in an image. This can be done in several ways, from algorithms that include every pixel and scale to boosting algorithms [30]. Different approaches can be taken depending on the problem, which includes multiple faces or no faces in one image, occlusion, poor illumination, face rotation, etc. For face verification applications using mobile phones, it can be assumed that only one face is always present.

In this project appearance and template models are used to detect and find key points (such as eyes, mouth and nose) on the face, respectively. The appearance model scan overlapping segments of the image searching for face candidates. This is done using different scales and combining with a cascade of classifiers. This technique was developed by Viola and Jones in 2004 [31] which use boosting [32] combining simple classifiers blending the outputs [30]. This is one of the most used techniques for face detection due to fast computation. Instead of using pixels directly, Local Binary Patterns (LBP) [33] are preferred as features which achieve better results and are fast to calculate [34] [35]. Template models, known as Active Appearance Models (AAMs), find key points on the face, dealing with differences in shape and texture using a training set of images [30]. Techniques based on this approach have been tested in mobile phones with good results [4]. Examples of face landmarks positions are shown in Fig. 2.5. More details about how to obtain AAMs are described in Appendix A.2.

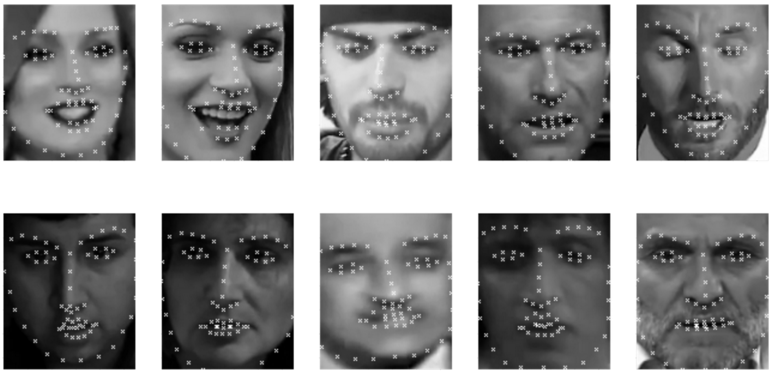

Figure 2.5: 68 landmark face position fitted with AAMs using Menpo [1] and Bob toolbox [23] .

Once the face is detected an isolated from the rest of the image is sometimes necessary to deal with changes in illumination. This will depend on the algorithm used to verify the identity of the person, where some of them are more susceptible to this differences from one sample video to another.

One tested solution for difficult lighting conditions is the work of Tan and Triggs [36], where a gamma correction is applied, followed by a difference of Gaussian filtering and a contrast equalization.

2.2.2 Face verification algorithms

Once the face is cropped from the image or frame video and normalized, face verification can be performed. The objective is to decide if a test face corresponds to a target face. So, either accept or reject that the face corresponds to a specific target person.

Many approaches can be taken depending on the selection of the features. Earlier approaches use an eigenvalue decomposition of the covariance matrix of the pixels intensities (eigenfaces and fisher faces) [18] Then, Local Binary Patterns with histogram matching [19] [37] were preferred. More advanced techniques use deep neural networks (DNN) [21] [38] which outperforms previous algorithms, but several images are used for training (over $200M$).

For video recognition, sequential information [39] can be extracted using Hidden Markov Models (HMMs) [40] with PCA and probabilistic appearance models [41] to learn the movements of the face. Dynamic images [42] has also been studied with good performance in cross-matching problems with inclusion of audio features [43] with DNNs.

In this project two type of features are extracted, 2D-DCT and motion vectors of the lips, and implemented using GMMs and session variability techniques, that were first introduced in speaker recognition [13] [15] (explained in Section 2.1) and later applied to face recognition which aim to suppress within-class variation for face authentication [44] [2].

Two-dimensional Discrete Cosine Transform (2D-DCT) coefficients are extracted from a single snapshot which makes a compact representation of the image and are fast to calculate. First, the image is detected [31] and normalized [36], then, is divided into blocks (usually 12×12 pixels with one pixel shift) that are considered to be observations of the same signal (face image) and feature vectors are extracted for each of this blocks (with $N \times N$ pixels) in a zig-zag way where first $40 - 50$ coefficients $C(v, u)$ are used as explained in [45]:

$$C(v,u) = \alpha(v)\alpha(u) \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} f(y,x)\beta(y,x,v,u) \tag{2.29}$$

for $v, u = 0, 1, ..., N - 1$. Where:

$$\alpha(v) = \begin{cases} \sqrt{\frac{1}{N}} \text{ for } v = 0 \\ \sqrt{\frac{2}{N}} \text{ otherwise} \end{cases} \tag{2.30}$$

and,

$$\beta(y,x,v,u) = cos\left[\frac{(2y+1)v\pi}{2N}\right] cos\left[\frac{(2x+1)u\pi}{2N}\right] \tag{2.31}$$

An overview of this procedure is shown in Fig. 2.6.

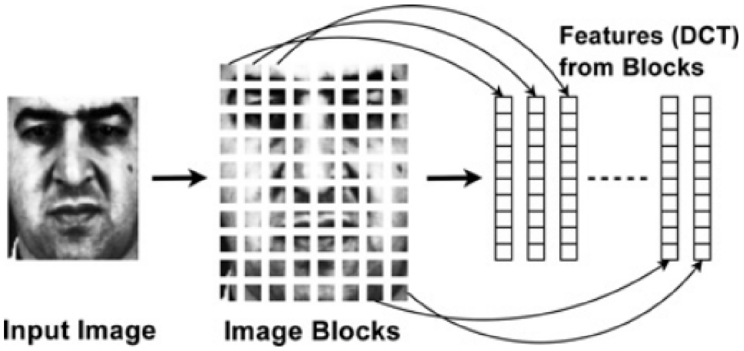

Figure 2.6: GMM parts-based approach. First 2D-DCT coefficients are extracted for each block using a zig-zag technique [2].

On the other hand, motion vectors from the mouth are extracted, which represent the apparent movement of the lips. Given that person verification is done from videos, motion vectors are a natural approach that takes advantage of the frame sequence. This motion is the velocity of the pixels in x and y direction obtained by a Taylor series approximation of the pixel variation between two consecutive frames:

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \tag{2.32}$$

$$\frac{\delta f}{\delta x}u + \frac{\delta f}{\delta y}p + \frac{\delta f}{\delta t} = 0 \tag{2.33}$$

Where,

$$u = \frac{\delta x}{\delta t} \qquad p = \frac{\delta y}{\delta t}$$

First, the face is detected in each frame using Viola and Jones [31] [23]. Then, the mouth is detected via face landmarks obtained by Lucas-Kanade algorithm [1] [46] to fix a bounding box around the lip area.

Motion vectors of the mouth are extracted using Farneback algorithm [47] based on polynomial expansion [48] where the velocity $u_i(x, y)$ and $p_i(x, y)$ for horizontal and vertical direction of each pixel (x, y) of frame i are calculated. A visual representation of motion vector is shown in Fig. 2.7 where the velocity is represented by an arrow on each pixel. Then, a feature vector w_i is constructed for each frame i by concatenating the absolute value of the velocities of each pixel, given by $\sqrt{u^2_{i(x,y)} + p^2_{i(x,y)}}$.

As seen in sections 2.1.2 to 2.1.3.2 for speaker verification same methods can be applied here (i.e., GMM, Total Variability, i-vectors and cosine distance). The

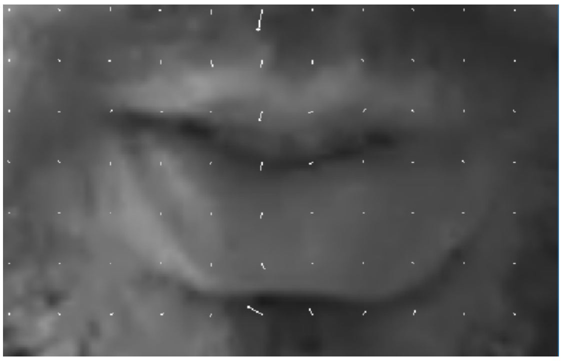

Figure 2.7: Pixel velocity bounding box while closing the mouth. 77 pixels are monitored based on a downsampled image of 11 × 7.

only difference is seen in the feature vectors where 2D-DCT coefficients and motion vectors of the mouth are extracted from the images instead of MFCCs in the audio approach.

2.3 Evaluation

The accuracy of the algorithms is commonly based on the Receiver Operating Characteristic (ROC) curve or Detection Error Trade-off (DET) curve that plots False Acceptance Rate (FAR) vs False Rejection Rate (FRR) [4] [12] [24] [15] based on different values of a threshold τ as described in equations 2.34 and 2.35. The final error is measured in terms of Equal Error Rate (EER) that finds a point on the curve (specific τ) with closest equal value for FAR and FRR.

$$FAR = \frac{\#\text{ False Acceptance}}{\#\text{ Impostors}} = \frac{\#\text{ impostor scores} > \tau}{\#\text{ of impostors}} \tag{2.34}$$

$$FRR = \frac{\#\text{ False Rejections}}{\#\text{ of Genuines}} = \frac{\#\text{ genuine scores} < \tau}{\#\text{ of genuines}} \tag{2.35}$$

Datasets are usually divided into three sets: training, development and test. In the first one, a model for each speaker is constructed based on distributions of the feature vectors. In the second one, scores and optimal threshold are found obtaining the EER:

$$\text{find } \tau \text{ development given } FAR(\tau) = FRR(\tau) = EER \tag{2.36}$$

Finally, using the threshold found in the development, Half Total Error Rate (HTER) is obtained in the test set that is calculated as the mean of FAR and FRR at the given threshold:

$$HTER_{test} = \frac{FAR(\tau) + FRR(\tau)}{2} \tag{2.37}$$

An example of DET curve is shown in Fig. 2.8 where development scores are between 10 and 20 for impostors and between 15 and 30 for genuine. EER is found at 18 with a value of 19.8% for EER. Test scores go from 0 to 26 for impostors and from 14 to 44 for genuine achieving FAR of 31% and FRR of 14% at threshold equal to 18.

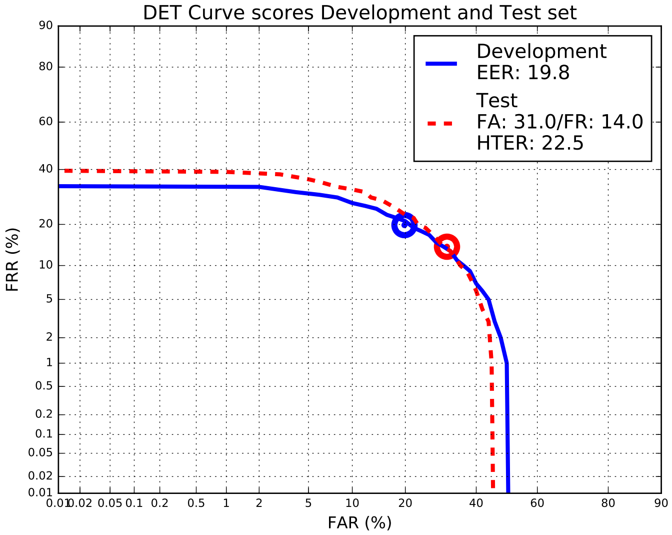

Figure 2.8: DET curve development and test. Increasing the threshold reduces false acceptance and increases false rejection.

Data 3

The availability of good data sets is very important when developing and testing classification algorithms. It is also important to have a large data-set in order to build a reliable UBM, as discussed in Chapter 2.

For the UBM it is important to have a large diversity of people to represent the global population. For the development of the algorithm, we need (ideally) a large number of people with a good balance between male and female and with many videos from each person recorded from different sessions to separate the enrolment phase (training) from development and test. More videos available for training result in better models obtained per person and more videos and sessions for testing can be traduced in better conclusions obtained from the results.

Ideally, we would like to have recordings that are of good quality so that degradations such as noise, reverberation and blur can be added artificially during development and also a "real scenario" data set with actual degradations for final evaluation.

Finding a good dataset for audio-visual verification is a hard task and constructing one takes a lot of time and resources due to the number of speakers needed and different sessions that must be booked for all speakers. Adding noise and reverberation might be done post recordings but clean conditions are a must.

In the next pages, a description of the main available datasets are shown, as well as a video dataset captured in the wild using Youtube as main source which will be used as UBM for an audio-visual verification.

3.1 Freely datasets

In this section the main datasets used in this project containing face images and audio are presented which are free for research purposes.

- **Voxforge**: This dataset offers a collection of transcribed speech for use with Free and Open Source Speech Recognition Engines. A small dataset selected from the English corpus is extracted [23] [49]. There are 30 speakers with 6561 audio recordings which correspond to nine hours. Recordings tend to be clean which is useful for this project as it can be corrupted with different noises and convolved with acoustic impulse responses for reverberation. The main problem with this dataset beside the size is that it lacks of visual information.

- **AT&T** [3]: This is an old face image dataset taken between 1992 and 1994 by AT&T Lab (Cambridge). It contains images of 40 people (ten each) with different lights, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). The size of each image is 92 × 112 with 256 grey levels per pixel.

- **Labeled Faces in the Wild (LFW)** [50]: As the name said, this dataset contains labelled faces collected around the web. It have more than 13000 face images and 1680 people has more than 1 photo.

- **YouTube Faces DB** [51]: This is a large database based on Youtube videos captured in the wild. It contains 3425 videos of 1595 different people obtained from the same people than LFW [50]. It contains images of different resolutions and different poses. The main problem is that the videos are broken into frames which makes audio not available. Also, the quality, as one can expect from Youtube videos, is not ideal. This dataset is the baseline to construct an audio-visual dataset "in the wild".

- **VoxCeleb 1 and 2** [16] [17]: These datasets contain the URLs of celebrity videos uploaded to Youtube. It provides text files with time stamps from the utterances. The first version has over 100.000 utterances of 1.251 celebrities and the second one has over a million utterances of 6.112 speakers. As mentioned for previous Youtube datasets the changes of poses, face obstruction and noise make it not ideal for audio-visual verification.

- **The VidTIMIT Audio-Video Dataset** [11]: This is one of the only video dataset that is free available containing audio and visual information. It consists of 43 speakers with ten videos per person reciting short statements of three to six seconds each. It is recorded over three sessions with a delay of one week between each one. Six videos correspond to the first session, two for the second and two for the last one. Recordings are performed in an office with fan noise which makes it not ideal for this project but the fact that contains both audio-visual information makes it one of the best candidates to test the proposed algorithms.

3.2 Proprietary Datasets

Besides free datasets, there exist a number of private databases which are not public available to everyone or not free. Usually, this databases contains more speakers, better quality and different scenarios. A few of them are presented next.

- **TIMIT** [52]: Is one of the oldest audio datasets for voice verification. It contains 630 speakers each reciting ten sentences. The price for non-members is 250 US dollars.

- **Mobio** [53]: Is an audio-visual dataset taken from 152 people with 100 male and 52 female from 2008 to 2010 in six different places. It contains native and non-native English speakers with 12 sessions. It was recorded using a mobile phone (Nokia N93i) and a laptop computer (2008 MacBook). It is free available (under permission) but is not allowed to use it for commercial purposes.
- **The Extended M2VTS Database** [54]: This is a multi-modal database from the University of Surrey that contains four recordings of 295 subjects recorded over a period of four months. It has audio files, video sequences and 3D-Models of the face. Fees for each set (audio, images and 3D-models) are up to 200 pounds for industry purposes.
- **The BANCA Database** [55]: This is a large dataset made for multi-modal verification. It consists of four European languages. High and low quality microphones are used and three different scenarios are presented (controlled, degraded and adverse). 208 people were captured over three months (1308 sequences), half men and half women. The English sequences for audio-visual signals cost 1000 pounds for academic purpose and 2000 for the industry.

3.3 Youtube video Dataset

Motivated by the lack of large suitable databases the construction of one is done to use it as UBM, where ideally we would require several hundred speakers. To do this 5,749 names present in the LFW database are used [50] to search for videos of this people in Youtube based on [51].

For each result in the query, first videos are downloaded and the aim is to find at most two videos of at least ten seconds each with only one face on it in all frames. If no faces are found in the video next result in the query is used. After five videos of the same subject without single faces, the search continue with the next person in the list.

For each video all the frames are extracted, and face detection is applied using Viola and Jones [31] [23] with LBP features [56]. Then, for each frame t for video V_{in}, a vector containing the corners coordinates of the face in x and y axis is constructed as $f_t(V_{in}) = [x_1, x_2, y_1, y_2]^T$. If similar properties of the face in next frames are founded, denoted by a distance vector ω, it is assumed that is the same person in the video. To save the video sequence a minimum time of τ frames are needed. The face search is presented in Algorithm 1.

Some Youtube videos tend to be too long and analysing each frame is slow and not optimal. To make the algorithm faster a step of n frames is used at the beginning of the search.

Once a face is found over $\tau = 10$ seconds, pixel variation is analysed over the frames to check if the face corresponds to the same person in one scene. This is

Algorithm 1: Face Search Youtube

```
Input N names in LFW
for Videos $V_{in}$=1...N do
    count=0
    $t_1 = 0$
    $t_2 = 0$
    for Frames t=1...F-1 do
        $h(t) = \|f_t(V_{in}) - f_{t-1}(V_{in})\|$
        if $h(t) < \omega$ then
            count++
            $t_2 = t$
        else
            count=0
            if $t_2 - t_1 \geq \tau$ then
                break
            else
                $t_1 = t$
    Output $V_{out} = \begin{cases} V_{in}(t_1 : t_2) \textbf{ if } (t_2 - t_1) \geq \tau \\ \text{NaN } \textbf{Otherwise} \end{cases}$
```

done by taking the difference between average pixel intensities of past and future frames:

$$Dif(t) = \left| \frac{1}{N_x + N_y} \sum_{i=0}^{N_x} \sum_{j=0}^{N_y} \frac{1}{\beta} \sum_{k=0}^{\beta} \Big[I_{t-k}(i,j) - I_{t+k}(i,j) \Big] \right| \tag{3.1}$$

Where $N_x N_y$ is the size of the image, $I_t(i,j)$ is the pixel intensity value for coordinates i and j for the frame t and β is the window size used for the past and future frames.

To decide that the face corresponds to the same scene a threshold is used where for each frame $Dif(t) \leq \theta$. If $Dif(t) > \theta$ for one or more frames it is assumed that the face is not from the same scene so it is probably not the same person and that range of frames are discarded from the database.

Finally, the last check is done going back to Algorithm 1 for all frames in the range to confirm that is, in fact, the same person in the same scene.

An example of the faces founded on Youtube is shown in Fig. 3.1, where different poses, sizes and illumination conditions are present.

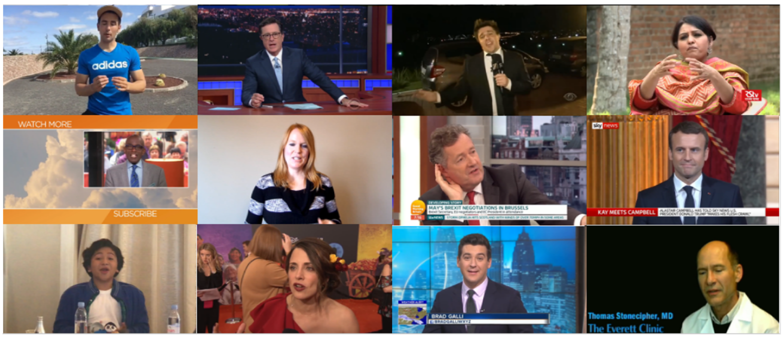

Figure 3.1: Examples frames from Youtube dataset

Audio-visual robust VAD 4

In this Chapter a Voice Activity Detector is constructed using audio and visual information via MFCC and motion vectors of the mouth which will also be used for person verification in Chapter 5.

VAD plays an important role in speaker verification. Frames that not contain the speaker voice may change the means and variances of the distribution of any feature used for the task, which will lead to a wrong model. This is caused by introducing noise signals like fan, babble or even other speakers in the room.

In section 2.1.1 a VAD was already introduced for the audio signal. In this Section, another method is presented where audio and visual signals are combined to create a robust VAD against noises. This method is based on [9] [10], where MFCCs are extracted from audio signals and motion vectors (MV) are used for the visual signal.

Previous works on audio-visual VAD rely on supervised training, using 2D-DCT coefficients [57] and HMMs (Hidden Markov Models) and colour skin to detect lips [58].

4.1 Feature extraction

One of the main problem of using frames from audio and visual information at the same time is the sampling mismatch. Usually audio frames are extracted every 10-30ms and video sequences are recorder at 24 or 30 frames per second. Having the same number of frames for both views means that one image frame must correspond to $1/fps$ second of the audio signal, where fps is the frames per second of the video sequence. For example, if a video is recorded at 30fps and audio frames are obtained with an overlap window of 50% means that the window size for audio must be $66ms$ which may be not ideal considering that audio signals are assumed stationary only for short frames of $20-30ms$. Nevertheless, using this technique to match both signals a VAD outperforms algorithms that use only visual or audio features.

MFCCs are obtained from audio signals using the same method as explained in Section 2.1.1 where F coefficients are extracted. A feature vector v_i is obtained for each frame i where $v_i \in \mathbb{R}^F$. To add some robustness J frames vectors are concatenated from previous and next frames forming the audio feature vector $v_i \in \mathbb{R}^{(2J+1)F}$.

For visual features, motion vectors are extracted as explained in Section 2.2.2.

Normalization of the face and mouth is applied, which was not considered in [9]. First, face rotation is performed to align the eyes and a bounding box is placed on the mouth with corner position (in x-axis) based on the eyes with width w_{bb}. The height of the bounding box is set to $\frac{7}{11} \times w_{bb}$ and the y-axis corner position is based on the distance between key points from eyes and nose. The bounding box is then downsampled to a size of 11×7 pixels. Fig. 4.1 shows an example of face detection and mouth bounding box.

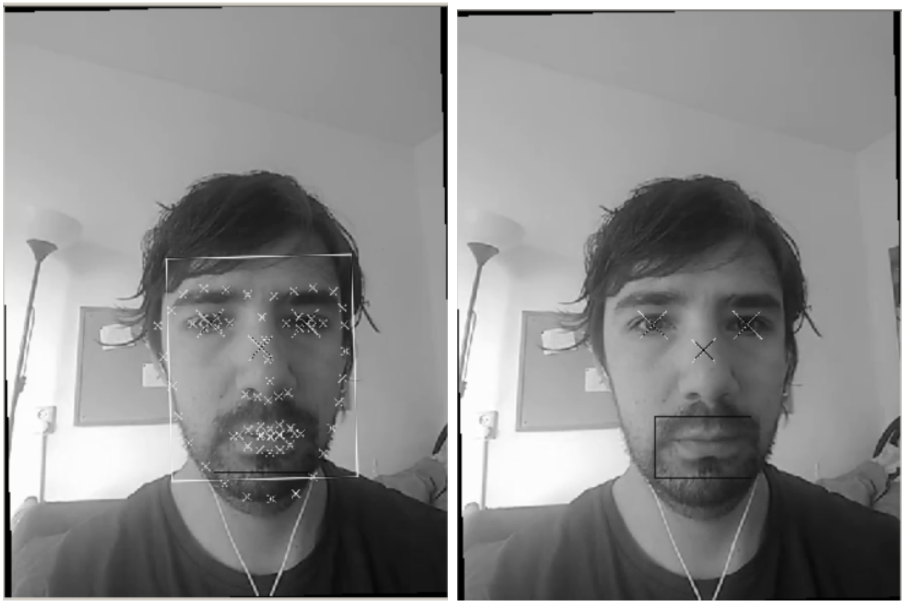

Figure 4.1: Landmark position and face correction for mouth tracking. (left) Bounding box face and landmark position. (right) Eyes and nose points used for mouth bounding box.

In this way and using J previous and next frames the visual feature vector for each frame is defined as $w_i \in \mathbb{R}^{(2J+1)WH}$ where WH is the product of the width and height of the mouth area (77 pixels in this case).

4.2 Diffusion Maps

Once the feature vectors v_i and w_i are extracted from each frame i diffusion maps are applied separately to each modality and combined in a later stage.

First, a similarity kernel $K_v \in \mathbb{R}^{NxN}$ is constructed using feature vectors v_i for $i = 1 \cdots N$ frames where each entry (n, m) of the matrix represent similarities between frames n and m:

$$K_v(n,m) = \exp(-\frac{||v_n - v_m||^2}{\epsilon_v}) \tag{4.1}$$

Where ϵ_v is the kernel bandwidth which controls the connectivity of the graph [10]. When $||v_n - v_m||^2 < \epsilon_v$ there are high similarities between frames n and m as seen in equation 4.1 and both nodes are considered connected. On the other hand, when $||v_n - v_m||^2 >> \epsilon_v$ similarities are negligible and nodes are considered disconnected. A common practice is to set the kernel bandwidth ϵ_v such that each point is connected to at least one other point:

$$\epsilon_v > \max_m [\min_{n \neq m}(||v_n - v_m||^2)] \tag{4.2}$$

This is a necessary condition for the graph to be connected such that there is a path between all pairs of frames. On the other hand the kernel bandwidth should be sufficiently small to prevent links between different contents, such as voice and no-voice [10].

The affinity kernel K_v defines a graph where each node represents a frame and edges between nodes are similarities given by equation 4.1. The same process can be applied for visual information using feature vector w_i to construct the similarity matrix K_w.

To combined both modes a unified matrix M is constructed given by the matrix product or the Hadamard product:

$$M = K_v \cdot K_w \tag{4.3}$$

or

$$M = K_v \circ K_w \tag{4.4}$$

This new matrix M attenuates the view specific interferences as explained by [10].

The graph of singles modes do not need to be connected, but only the combined graph M should be. This is explained by the fact that if two points are connected in the multiple view graph it must be connected in at least one of the single mode graph. The bandwidth ϵ_v is calculated as:

$$\epsilon_v = C \max_m [\min_{n \neq m}(||v_n - v_m||^2)] \tag{4.5}$$

where $C = 0.5$ gives good performance based on experimental results.

4.3 Audio-Visual VAD algorithm

After computing matrices K_v for audio and K_w for visual information and combining them into a unique matrix M an eigenvalue decomposition is applied and the first eigenvector corresponding to the largest eigenvalue is used to separate between speech and non-speech.

Finally, the whole algorithm is described in Algorithm 2.

Algorithm 2: VAD using similarities Kernels [10]

```
Calculate feature vectors $v_i$ and $w_i$ for each frame $i = \{1 \cdots N\}$
Calculate $K_v$ and $K_w$ from equation 4.1
Calculate matrix $M$ from equation 4.3 or 4.4
Obtain leading eigenvector $\nu_1$ related to $M$
for frame $i = 1 : N$ do
    if $\nu_1(i) > \tau$
    $i$ =Voice
    else
    $i$ =no-voice
```

4.4 VAD results

In [58] a dataset of six videos recorded at ten fps of different people is used to test their algorithm. Each sample has one minute of audio alternating between voice and no-voice (50%-50%). Each recording contains real-world noises, such as people talking in the background and natural movements of the face. Results in terms of accuracy are shown in Table 4.1 for each mode comparing their method and the proposed one here using cross validation for the threshold. Similar results are obtained by both methods (slightly worse in the proposed method) but in [58] cross validation is done over all frames, using same videos for train and testing while in the proposed approach cross validation is done over videos instead of frames.

Accuracy per view [%]	Difussion maps	HMM [58]
A-VAD	87.90 ± 0.1	90.09
V-VAD	78.45 ± 0.14	80.15
AV-VAD	89.48 ± 0.07	92.07

Table 4.1: Accuracy per mode for six videos of six different people. Threshold for Diffusion Maps is selected based on equal error rate (EER) for each folder in cross validation.

Example of VAD over time of one video sample and the ROC curve using all videos for the proposed method on each mode and the combination using the matrix product and the Hadamard product is shown in Fig. 4.2 where the combination of Hadamard product shows the best results achieving an Area Under the Curve (AUC) of 0.95.

Further experiments show the performance under adverse conditions testing on clean speech, adding babble noise and noise from another speaker talking at

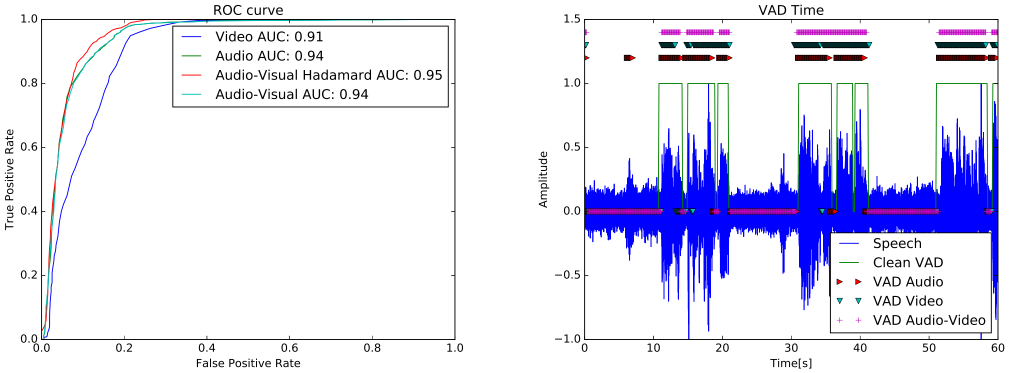

Figure 4.2: ROC curve VAD 6 people (1 minute each). Results for single views audio and video (green and blue) and audio-visual for Hadamard product (red) and matrix product(cyan).

the same time, which are examples of harmful noises given that are similar to the target speaker. Same visual information is used for three cases. Results are shown in terms of AUC of the ROC curve using different modes: only audio information, only visual and audio-visual using the matrix product and the Hadamard product in equation 4.3 and 4.4, respectively. Best performance is obtained using the Hadamard product between the two modes as seen in Fig. 4.3. Speech from another speaker gives the worst result for audio mode achieving 0.81 AUC which is improved to 0.96 for the multi-view approach. In this case, audio-visual VAD relies mostly in the visual information which achieves 0.94 AUC.

Finally, to show the importance of face tracking and alignment an experiment where the face is detected every half second and no face rotation is performed is shown in Fig. 4.4 where visual VAD obtain 0.72 AUC compared with 0.94 AUC shown in Fig. 4.3.

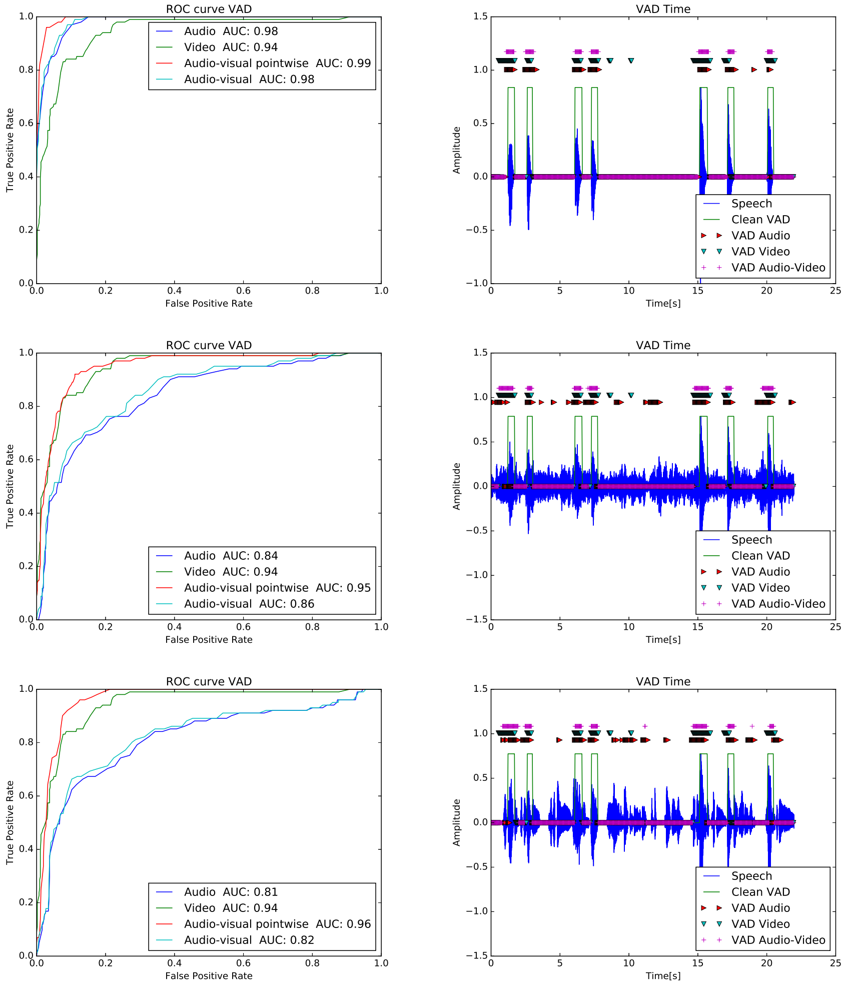

Figure 4.3: ROC Curve (left) and VAD in time (right) different methods: clean audio (top), Babble noise 0dB SNR (middle), noise from another speaker with 0dB SNR (bottom)

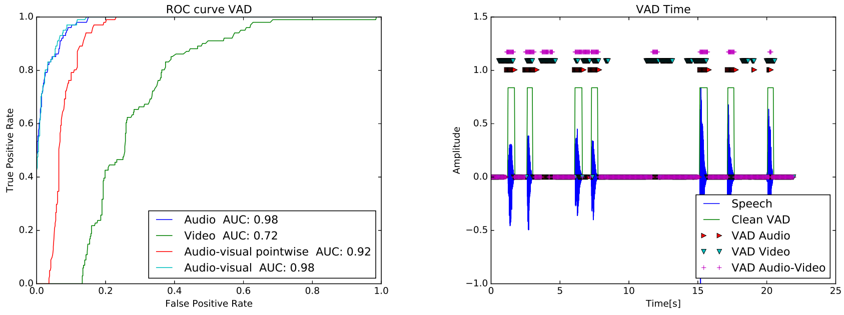

Figure 4.4: ROC Curve (left) and VAD in time (right) no face detection and rotation.

Three mode audio-visual authentication 5

In this Chapter experiments using VidTimit dataset are shown to compare person verification algorithms using a robust VAD presented in Chapter 4. Single mode verification achieves good performance with standard algorithms under ideal conditions, but when these conditions are not met due to noise and reverberation for audio mode and noise and changes in illumination for visual mode, multi modal authentication plays a crucial role to improve accuracy.

5.1 Audio-Visual verification

The proposed method combines MFCC, together with motion vectors of the mouth and 2D-DCT coefficients for a single face image. Features are explained by GMMs and authentication is implemented with end-factor analysis and likelihood ratio explained in Chapter 2. There are two main approaches that can be taken:

- Include audio-visual modes in the same feature space.
- Obtain scores from each mode independently of each other and train an additional classifier using scores as features for final verification.

Both approaches have been tested before for audio-visual verification [8] [4] combining different features (pixel intensities and LBP) for visual information than the proposed here.

5.1.1 Feature level fusion

Combining audio-visual information in the same space is not a trivial problem as seen in Chapter 4. Besides the problem of sampling mismatch, one has to deal with the *"curse of dimensionality"* [20] given that not much data is available and we want to minimize the number of videos for enrolment without losing accuracy. Using more than 100 features can damage the performance of the algorithm even when using a decent amount of videos (around 30 seconds per person).

One way to address this problem has been to use boosting classifiers [4] that search for features pairs which minimize the misclassification rate with quadratic discriminant analysis, but no detail on implementation and results for this approach is given.

A common approach to reduce the dimensions is to use Principal Component Analysis (PCA) or any other dimensionality reduction algorithm (ICA, LDA, etc.).

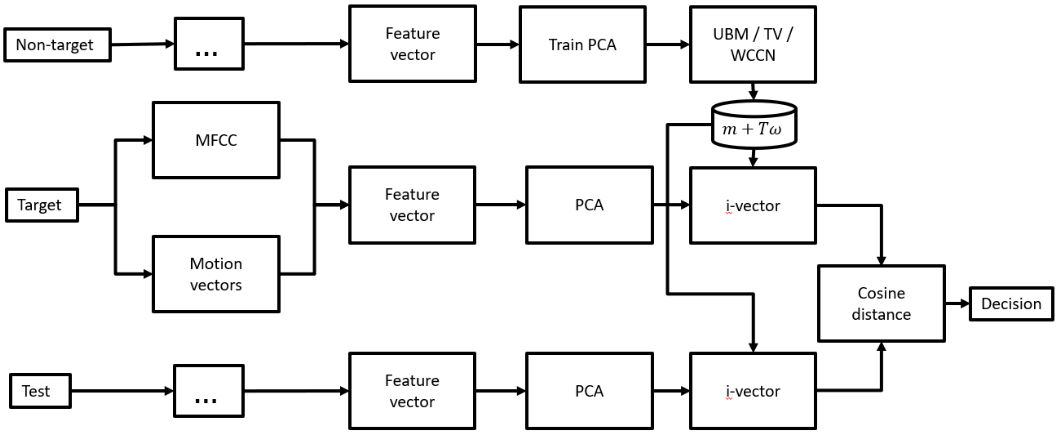

Figure 5.1: Feature level fusion flow diagram. Features are combined in an earlier stage for non-target, target and test samples. PCA is trained for non-target samples before estimating the UBM and applied to target and test samples.

PCA helps to reduce EER which use the covariance matrix of the features and project the variables into a new subspace based on an eigenvalue decomposition [20]. PCA was chosen for dimensionality reduction assuming correlation on the features given that they explain similar aspects of the voice.

Only MFCCs and MV of the lips are used for this method given that normalizing and extracting 2D-DCT coefficients for all frames is too slow and the feature vector will no correspond to the same frames if only one snapshot is used.

The final algorithm is explained in Fig. 5.1 where an audio-visual VAD is applied before combining the features to train one UBM and Total Variability (TV) matrix. End-factor analysis is used with cosine distance of i-vectors for final score.

5.1.2 Score level fusion

Score level fusion is the most common approach to combine different modes while doing authentication [4]. The idea is simple, classify each mode separately (audio and visual information) and then combine the results with an additional classifier. Doing this also addresses the *"curse of dimensionality"* and sample mismatch. First, a score is assigned to each test video sample based on single mode classification in the development set. This score can be obtained from the likelihood ratio test or from the cosine distance between each pair of target-test i-vector depending on the algorithm. Then, having scores for each mode the problem is to classify each sample based on the score features. This is an easy task that can be resolved using a linear classifier like LDA, SVM, Logistic, etc. which is trained in the development set. The proposed method uses end factor analysis for MFCCs and motion vectors and likelihood ratio for 2D-DCT coefficients explained

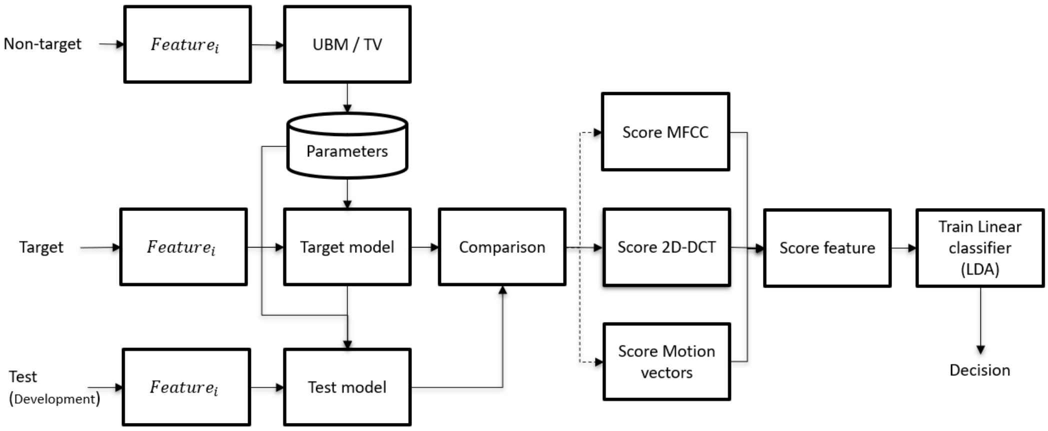

Figure 5.2: Flow diagram score fusion. Scores are obtained for each feature where end-factor analysis is used for MFCC and MV and GMM likelihood ratio is used for 2D-DCT coefficients.

in Section 2.1.2. The inclusion of 2D-DCT coefficients makes sense in this step because sample matching is not needed as algorithms for each feature are trained independently. Final scores for all methods are used to obtain a new feature vector to train an LDA classifier which performs systematically better than any other classifier based on our experiments.

Different combination of modes are analysed (MFCC+MV, MFCC+2D-DCT, MV+2D-DCT and MFCC+MV+2D-DCT). A description of the steps to obtain each score and the final decision is shown in Fig. 5.2 where $Feature_i$ correspond to each type of feature (MFCC, MV and 2D-DCT coefficients). For end-factor analysis (MFCC and MV) the *Parameters* explain the Total variability matrix $m + T\omega$. The *target* and *test model* correspond to the i-vectors and the *comparison* and *score* is the cosine distance. For likelihood ratio (2D-DCT coefficients) the *Parameters* correspond to the UBM, the *target model* is the adaptation of the UBM to each person and the *test model* is the probability of the features belonging to the target model and to the UBM ($p(x|\lambda_{target})$ and $p(x|\lambda_{UBM})$). Finally the *comparison* is done using the likelihood ratio test.

The likelihood ratio performs better than end-factor analysis for 2D-DCT coefficients because only one image is used for training and i-vectors needs a large number of samples to achieve good performance.

5.2 Experimental Results

The proposed method includes 77 motion vectors of the mouth together with 24 MFCC per frame for the audio signal as features and also used for VAD. For score fusion 47 2D-DCT coefficients extracted from a single image are included.

GMMs with 16 Gaussian are used for feature representation and verification is done using End Factor Analysis (with a subspace of 43 for the i-vectors) described in section 2.1.3.2 with WCCN and cosine distance for scoring.

Feature fusion and score fusion are tested in the VidTimit dataset [11] together with 200 Youtube videos for the UBM extracted using the algorithm presented in Chapter 3. The VidTimit dataset is recorded in three sessions, divided into six, two and two videos, each containing one sentence per speaker. The first session is used for training and the last two for development and test.

First, audio-visual results using two features (MFCCs and motion vectors) are compared with single mode algorithms (2D-DCT, MFCC and motion vectors) in Fig. 5.3 together with a learning curve of the half total error for the test set for different sentences size (number of videos) recorder per person in the training set.

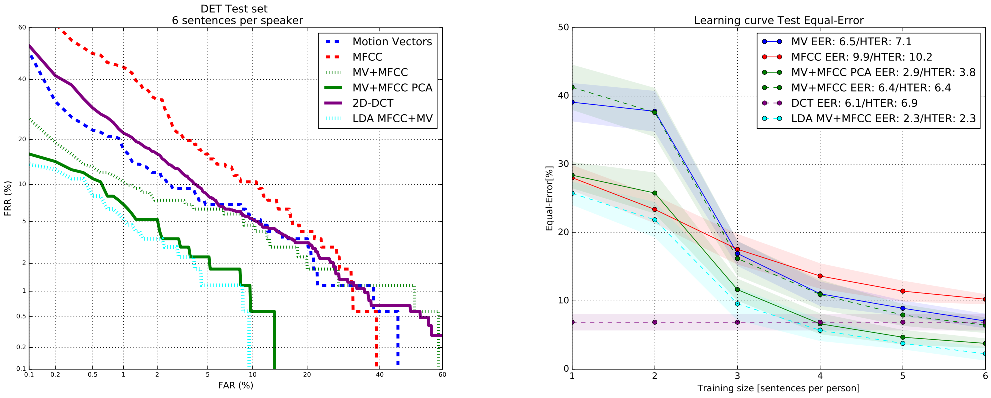

Figure 5.3: VidTimit results test set different algorithms. DET curve (left) and learning curves with best EER and HTER (right).

The best result achieves 2.3% of EER and HTER combining MFCCs and motion vectors scores with a LDA classifier to separate genuine from impostors in the development set using six sentences per speaker (around $24 - 30$ seconds) in the training phase. Combining MFCCs and MV into one feature vector and reducing the dimension to 63 (that explains 90% of the variance) achieves 2.9% EER and 3.8% of HTER. The worst result is obtained by using MFCCs as features which are explained by the noise present in this dataset.

From the results, combining modes in a later stage using a linear classifier outperforms other algorithms (single mode and feature fusion). Applying PCA to feature fusion helps in terms of EER but still performs worst than LDA third classifier, meaning that there is some information that is lost while doing the transformation.

Using score fusion of three algorithms from MFCC, motion vectors and 2D-DCT coefficients of a single face image, the algorithm outperforms all other classifiers.

Fig. 5.4 show the comparison where the later combination of three scores outperforms any other of two classifiers. Also, a scatter plot for the test set can be seen in Fig. 5.5 where genuine scores are separated from impostors with a linear classifier (LDA) trained in the development set.

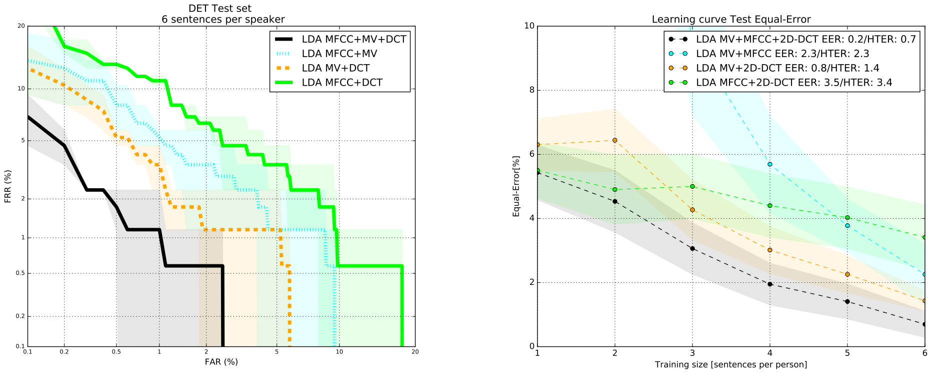

Figure 5.4: (left) DET curves switching between development and test set different combination of third classifier (LDA) using MFCC, Motion Vectors (MV) and 2D-DCT coefficient. (right) Learning curve test set best EER and HTER for score fusion.

The LDA classifier achieves an EER of 0.2% and HTER of 0.7% which outperform the best combination of two features: MV and 2D-DCT (0.8% EER and 1.4% of HTER). Combining three modes from audio (MFCC), the motion of the mouth (MV) and face image (2D-DCT coefficients) outperforms any other method studied in this project.

Further experiments with different noise types, levels and reverberation for audio signals together with blur images and different light conditions are analysed in Appendix B to show the robustness of multi mode authentication against adverse conditions.

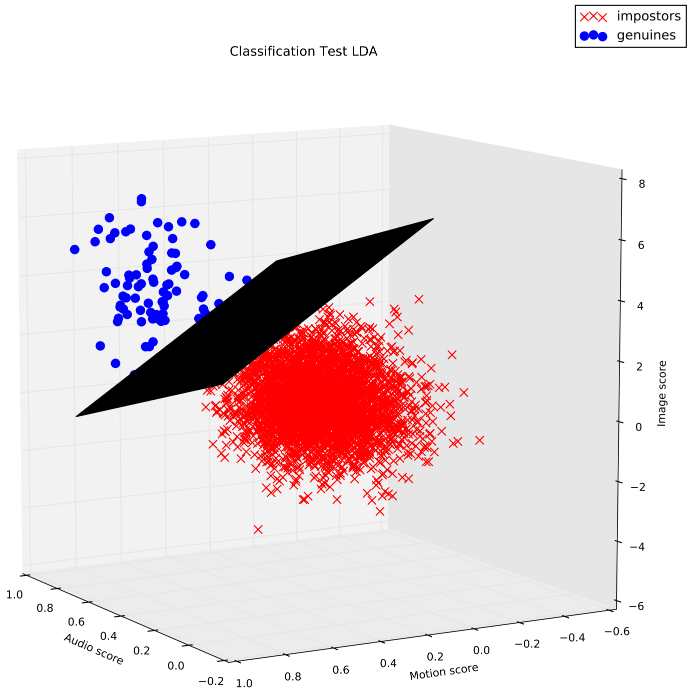

Figure 5.5: Scatter plot Test set. Decision boundary (black) trained in development set separates genuine (blue) from impostors (red) given audio (MFCC), Motion (motion vectors) and image (2D-DCT coefficient) scores.

Conclusions and Future Work 6

Multi-modal authentication has been studied and analysed in this project. Using multi modal authentication algorithms have shown to perform better than most single modes methods. The potential of end factor analysis has been shown not only for audio features but for the motion of the mouth. Results under ideal conditions show that single mode algorithms perform good but the performance decreases while testing under adverse environments as seen in Chapter 5 and Appendix B.

Speaker verification achieves good results with clean speech and no reverberation using end factor analysis but adding noise (fan or babble) below $5dB$ of SNR can degrade the results in more than 12% of EER. Reverberation also increase the error which can be coped by training with different room acoustic impulse responses.

For face authentication, several methods were studied. Gaussian Mixture Models with likelihood ratio achieves good performance when using a single snapshot of the video for training [2]. Results using 2D-DCT coefficients as features achieves EERs between $3-6\%$ and $0.5-1.7\%$ in the VidTimit and AT&T dataset, respectively where several images are used for training in the second case. Degradation of the image, such as Gaussian noise, blur and changes in illumination, increases the EER up to 10% in the AT&T dataset.

Motion vectors of the mouth together with MFCCs can be used for VAD using diffusion maps introduced in [9] and [10]. Importance of VAD for person verification when noise is present has been studied, Chapter 4 shows that detection and normalization of the mouth plays a critical role, affecting the performance for the visual mode where AUC of the ROC curve can be improved from 0.72 to 0.94 when the detection is done right. Comparison with supervised methods achieves similar results (around 90% of accuracy over six minutes of recordings). Also, using the same features for VAD and person verification makes a suitable solution for implementation.

Combining three modes: single image, audio and motion of the mouth outperform classical approaches. Half Total Error of 0.7% is achieved for the VidTimit dataset in the test set compared to 2.3% and 1.4% when using only two modes (MFCC plus motion vectors and 2D-DCT plus motion vectors). The combination in a later stage, known as score fusion, gives better results than feature fusion, where sample matching and the "curse of dimensionality" are not resolved completely.

Further work is needed to speed up computing time for motion vectors that rely on face detection and landmark positions on the face, which could be slow depending on the duration of video sequences. Also, a 3D model of the face can be used in the future to deal with face movements where the algorithms fail.

One of the key steps in any classification problem is the training phase that depends on the dataset selected. Few of them were used for experiments in this project. For speaker verification, Voxforge was preferred due to cleaner speech. An audio-visual approach was tested in VidTimit dataset, which was one of the only free datasets available with both audio and visual sequence information. Results show a high variance on some algorithms when changing the training size and switching between development and test sets. Experiments on different and larger data sets are needed to verify the robustness and generalization of the algorithms.

Youtube videos are a good source for Universal Background Model but hard to deal for person verification due to face movements and noise. Further experiments using separate UBM for male and female need to be done together with studying the ideal number of speakers which in this project where set to 200 (more than one hour) and we see no improvement when this number is increased, due to the small number of people to verify (43 for VidTimit dataset).

Having larger datasets will make it ideal to test a neural network approach which shows good results in speaker recognition [16] [17], face verification [38] [21] and some cross-modal authentication [43].

Appendix A

A.1 Expected value i-vector

In Chapter 2 the expected value for the i-vector is given by equation 2.24 for utterance u:

$$E(\omega(u)) = l^{-1}(u)T^t\Sigma^{-1}\tilde{F}(u) \tag{A.1}$$

with $l(u) = (I + T^t\Sigma^{-1}N(u)T)$.

To prove this we need to show that equation A.2 is true.

$$P_{T,\Sigma}(\omega|u) \propto exp(-\frac{1}{2}(\omega - E(u))^t l(u)(\omega - E(u))) \tag{A.2}$$

Defining the posterior probability $\gamma_t = P(i|x_t, \lambda_{UBM})$ for each Gaussian i and $\bar{\gamma}_t = [[\gamma_t(1)]_F, [\gamma_t(2)]_F, ..., [\gamma_t(C)]_F]^T \in C \times F$ together with the supervector $\bar{X}_t = [x_{11}, x_{12}, ..., x_{CF}]^T$ for the observation t for each feature dimension and each Gaussian.

Recalling that $M = m + T\omega$, using the same notation from the Baum-Welch statistic from Section 2.1.3.2 and applying Bayes rule [15] [14]:

$$\begin{aligned}
P_{T,\Sigma}(\omega|u) &\propto P_{T,\Sigma}(X|\omega)N(\omega|0,I)\\
&= \prod_{t=1}^{L} P_{T,\Sigma}(x_t|\omega)N(\omega|0,I)\\
&\propto exp(-\frac{1}{2}\sum_t \bar{\gamma}_t(\bar{X}_t - (m + T\omega))^t\Sigma^{-1}(\bar{X}_t - (m + T\omega)))exp(-\frac{1}{2}\omega^t\omega)\\
&= exp\Bigg(-\frac{1}{2}\sum_t \bar{\gamma}_t\Big((\bar{X}_t - (m))^t\Sigma^{-1}(\bar{X}_t - m) - 2\omega^tT^t\Sigma^{-1}(\bar{X}_t - m) + \omega^tT^t\Sigma^{-1}T\omega\Big) - \frac{1}{2}\omega^t\omega\Bigg)\\
&\propto exp\Bigg(\omega^tT^t\Sigma^{-1}\sum_t \bar{\gamma}_t\Big((\bar{X}_t - (m))^t - \frac{1}{2}\omega^tT^t\Sigma^{-1}T\omega\sum_t \gamma_t - \frac{1}{2}\omega^t\omega\Bigg)
\end{aligned}$$

$$= exp\left(\omega^t T^t \Sigma^{-1} \tilde{F}(u) - \frac{1}{2}\omega^t T^t \Sigma^{-1} N(u) T \omega - \frac{1}{2}\omega^t \omega\right))$$

$$= exp\left(\omega^t T^t \Sigma^{-1} \tilde{F}(u) - \frac{1}{2}\omega^t (T^t \Sigma^{-1} N(u) T + I)\omega\right)$$

$$= exp\left(-\frac{1}{2}(\omega^t l(u)\omega - 2\omega^t (l(u) l(u)^{-1}) T^t \Sigma^{-1} \tilde{F}(u))\right)$$

$$\propto exp\left(-\frac{1}{2}(\omega - l(u)^{-1} T^t \Sigma^{-1} \tilde{F}(u))^t l(u)(\omega - l(u)^{-1} T^t \Sigma^{-1} \tilde{F}(u))\right)$$

Where the solution of $E(w(u))$ in equation A.2 is $l(u)^{-1}T^t\Sigma^{-1}\tilde{F}(u)$ and the covariance $cov(w(u), w(u) = l(u)$.

A.2 Active Appearance Models (AAM)

AAMs are one of the most common techniques for modelling and segmenting deformable objects [46]. These are parametric models that fit the shape and appearance of a specific object. Different algorithms are used to fit the model to a specific face, where a minimization of a global error is computed.

AAMs are composed of three models as described in [46]: shape, appearance and motion model.

The shape of the object is defined as the location of a set of L points (x_i, y_i) $\forall i = \{1, ..., L\}$ on a face image. To fit the points on a particular face a training set of N images is used with the correct L landmarks positions. Shape model is obtained by a technique called Point Distribution Model (PDM) wich is obtained by appying Principal Component Analysis (PCA) to the training set of object's shape $S = \{s_1, s_2, ..., s_n\}$ and $s_i = [x_{i1}, y_{i1}, x_{i2}, y_{i2}..., x_{iL}, y_{iL}]$.

Then s can be written as:

$$s = \bar{s} + \sum_{i=1}^{n} p_i s_i = \bar{s} + Sp \tag{A.3}$$

Where $\bar{s} \in \mathbb{R}^{2L \text{ x } 1}$ is the mean shape, $S \in \mathbb{R}^{2L \text{ x } n}$ and $p \in \mathbb{R}^{n \text{ x } 1}$ are the shape basis and shape parameters, respectively.

The appearance model is obtained by warping the original images into a common reference frame and applying PCA to the warped images. This is done over a feature space defined for all training images. Then the appearance model could be written in terms of the matrix composed of basis vectors A and appearance parameters c together with the mean texture $\bar{a}$:

$$a = \bar{a} + Ac \tag{A.4}$$

The motion model is denoted by $W(x;p)$, that extrapolates the position of all pixels from the reference frame to a particular shape instance s and vice-versa [46]. This model relate the shape and appearance models with each other.

The main assumptions on AAMs are that the shape is approximated by the shape model given in equation A.3 and the object appearance is model after the image is warped by the motion model as:

$$i[p] \approx \bar{a} + Ac \tag{A.5}$$

Where $i[p] = vec(I(W(x;p)))$ and denotes the vectorized version of the warped image [46].

The warped function $W(x;p)$ can be computed in several ways, like PieceWise Affine (PWA) and thin plate splines (TPS) [59].

The goal of AAMs is to fit the Image $I(x)$ and the model $M(W(x;p)) = A(x)$. This could be seen as an optimization problem that minimizes the sum of the square of the difference given in equation A.6 with respect to the shape parameters p and appearance parameters c.

$$\sum_{x \in \bar{s}} [\bar{a} + Ac - I(W(x;p))]^2 \tag{A.6}$$

Several formulation and solution have been presented in the last years using gradient descent algorithms (Lucas-Kanade Optimization) described in [46] and [59].

Appendix B

In this Appendix, experiments under adverse conditions are shown using Voxforge dataset for voice verification and AT&T dataset for face verification [3]. For voice experiments degradation from noise and reverberation are implemented [60] and for visual experiments, noises and different light conditions are added to the original images.

B.1 Voice verification under adverse conditions

The following model is used for noisy speech:

$$x(n) = s(n) * h(n) + v(n) \tag{B.1}$$

where s(n) is clean speech, h(n) is the room impulse response and v(n) is additive background noise.

B.1.0.1 Noise and reverberation

One of the main problems of i-vectors is that is susceptible to noise and reverberation. In clean environments (clean channel) with almost no noise and no reverberation, this technique produces very good results with less than 1% of EER. Adding different kind of noises like fan or babble and reverberation from different rooms for the same test audio samples can degrade the results to $20-30\%$ of EER, which makes it not suitable for applications in these scenarios.

A full comparison of different scenarios is shown and explained in the next pages for voice verification using the Voxforge database [49]. The variables to consider are four: audio duration, noise type (fan, babble), noise level ($0dB - 30dB$) and reverberation for different rooms (lobby, lecture and meeting room). A detail explanation of how the noises and room impulse responses are obtained can be found in [60]. The Signal-to-Noise ratio is defined based on equation B.2 as:

$$SNR_{dB} = 10\log_{10}\left(\frac{\sum s(n)^2}{\sum v(n)^2}\right) \tag{B.2}$$

Table B.1 and B.2 show the parameters and sets used in the experiments for 20 speakers and 10 samples per person.

Number of Gaussians	Number of audio features	Subspace dim
128	60 (19 MFCC + Energy + Δ + $\Delta\Delta$)	20

Table B.1: Parameters used for audio test i-vector algorithm

Training	Development	Test
50[%]	30[%]	20[%]

Table B.2: Number of samples per speaker

Results are shown in terms of Equal Error Rate (EER) and Half Total Error Rate (HTER) for development and test set, respectively and plotted using DET curve.

B.1.1 Audio duration

As one can expect, increasing the audio duration in the training set leads to better performance. This can be seen in Fig. B.1 where the comparison between one minute (30 seconds for training) and 2.5 minutes (1.25 minutes for training) is shown. The performance increases from 3.4% to 1.4% EER. The same happens for the test set where a difference of 2.3% is seen in HTER.

As other studies suggested [61] [2], the i-vectors need an important number of training samples to have good accuracy when estimating the variability matrix T. Without enough data, variations within and between speakers may not be taken into account while estimating T. Here is shown that doubling the training time per speaker the EER can be reduced by half.

B.1.2 Noise type

The second parameter is noise types, where no noise, fan and babble were studied. Reverberation is not considered yet and 2.5 minutes of recording per speaker is used for the next experiments, which as seen before, gives better results. The model at this point is:

$$x(n) = s(n) + v(n) \tag{B.3}$$

Figure B.2 shows DET curves for different noise types used in train and test set with constant SNR of 10dB. The worst results from this graph is obtained when training with no noise and testing with babble with 5.3% of EER. Babble noise gives the worst results because MFCCs are estimated from the speaker speech and speech noise, which makes it a mixture of both signals. Fan noise has less impact, given that the noise is not from another speaker.

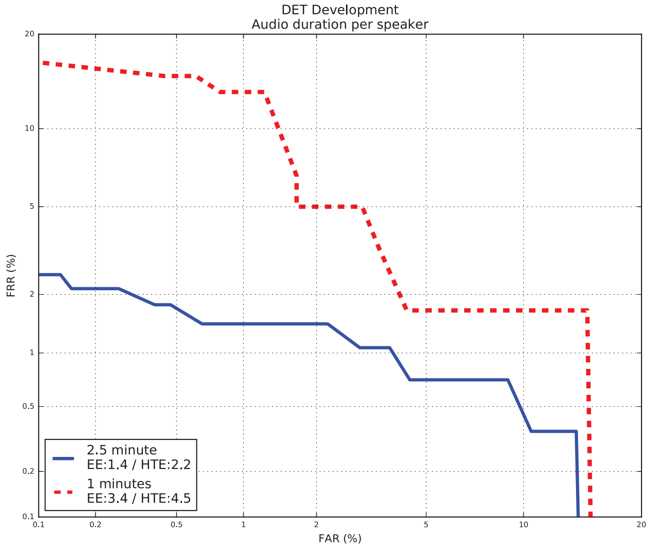

Figure B.1: DET curve development set for 1 minute of recording per speaker (red) and 2.5 minutes per speaker (blue). EER: Equal Error in development set [%]. HTER: Half total error for test set [%]

Another interesting result is that best performance is achieved when training and testing with the same type of noise, so for example if the test contains fan noise the best results are obtained while training with fan noise.

B.1.3 Noise level

The difference in noise level is one of the critical aspects for voice id verification. Figure B.3 shows that training without noise and testing at different levels of SNR can differ up to 12.5% in EER. Also, after reducing the noise to a certain level (SNR over 10dB) difference between experiments tend to decrease.

Interesting results can be seen when training with noise. Figure B.4 shows that training with fan noise achieves similar results for different SNR in the test sets. Training with fan noise at $10dB$ of SNR obtains HTER between 3% and 4.4% for SNR between 0 and 30 in the test set. Increasing the noise in training produce worse results, but the variance is small, with HTER between 3.8% and 5%.

In Table B.3 different level of noises are introduced together with noise types. Best results are obtained when training and testing with the same noise type. When testing with low SNR (for example fan at 0dB), best results are obtained when training with some noise (fan at 10dB). A particular case occurs when testing with babble at 10dB, where training with fan at 10dB outperforms training with babble, but the differences are negligible, less than 0.1% of HTER.

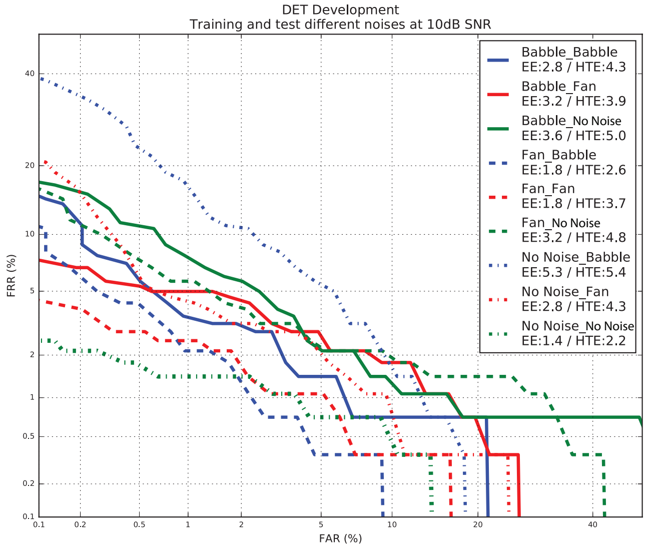

Figure B.2: DET curve development set for 2.5 minutes in meeting room. Different type of noises, fan, babble and no noise in train and test set are plotted together with the results in terms of EER and HTER. Same line style correspond to same noise type for training and same color correspond to same noise type for test.

Training/Test	Fan(0dB)	Babble(0dB)	Fan(10dB)	Babble(10dB)	Fan(20dB)	Babble(20dB)	No Noise
Fan (0dB)	5.0/7.5	9.2/9.7	4.6/6.1	3.3/4.4	10.3/9.2	6.0/4.8	17.0/16.8
Babble (0dB)	7.0/8.4	10.6/10.5	7.5/7.3	6.1/7.5	12.8/12.9	8.5/7.7	17.5/17.2
Fan(10dB)	1.8/4.4	10.4/9.9	1.8/3.7	1.8/2.6	2.2/4.0	2.1/2.8	3.2/4.8
Babble(10dB)	4.6/6.1	7.5/9.2	3.2/3.9	2.8/4.3	2.1/3.8	2.1/3.6	3.6/5.0
Fan(20dB)	7.1/8.0	13.9/13.6	1.8/3.3	3.2/3.9	1.4/2.7	1.8/1.9	1.8/2.0
Babble(20dB)	5.0/7.2	8.6/10.3	1.8/3.8	1.8/2.7	1.8/3.0	1.4/1.9	1.8/2.6
No Noise	14.3/15.0	15.6/14.3	2.8/4.3	5.3/5.4	1.8/2.5	2.1/2.4	1.4/2.2

Table B.3: Noise type and SNR. Results for development/test sets in terms of EER and HTER respectively. Train and test in meeting room (no reverberation), for different noise types.

Tables B.4 and B.5 show that worse performances are achieved when training with SNR of $0dB$ and testing with no noise or high SNR (over $20dB$). This is traduced in EERs of 17% and 17.5% when training at $0dB$ for fan and babble noise, respectively and testing with no noise. Also, testing with $0dB$ of SNR and training without noise get results over 14.3% of EER and HTER.

Training with noise between 10-20 dB of SNR gives a good trade-off for testing in different environments. This can also be seen in Table B.3 together with tables B.4 and B.5, where EER of around 5% can be achieved for different test scenarios.

Training/Test	0dB	5dB	10dB	15dB	20dB	30dB	No Noise
0dB	10.6/10.5	9.3/9.1	6.1/7.5	6.4/6.7	8.5/7.7	14.2/13.1	17.5/17.2
5dB	8.2/10.0	5.3/8.0	4.6/6.4	4.3/5.8	3.6/5.1	5.3/6.2	6.4/7.0
10dB	7.5/9.2	3.9/5.0	2.8/4.3	3.1/3.9	2.1/3.6	2.9/3.9	3.6/5.0
15dB	7.5/8.3	2.5/4.9	1.9/3.9	2.4/3.8	1.8/3.6	1.4/3.0	2.1/3.9
20dB	8.6/10.3	3.2/4.4	1.8/2.7	1.8/2.7	1.4/1.9	1.1/2.5	1.8/2.6
30dB	13.2/13.8	6.4/7.0	3.1/3.9	1.8/2.9	1.8/1.9	1.4/2.3	1.7/2.1
No Noise	15.6/14.3	8.8/8.0	5.3/5.4	3.2/3.6	2.1/2.4	1.4/2.6	1.4/2.2

Table B.5: Babble noise level. Results for development/test sets in terms of EER and HTER respectively. Train and test in meeting room (no reverberation), Babble noise at different noise levels.

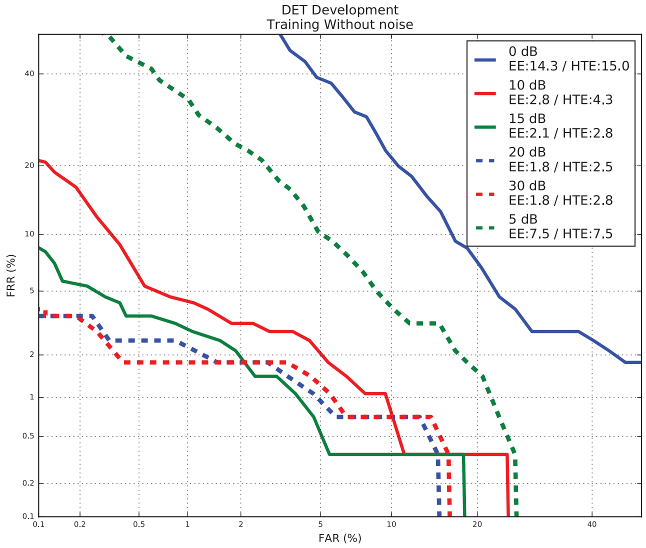

Figure B.3: DET curve development set for 2.5 minutes of audio in meeting room. Training without noise and testing with fan noise at different levels

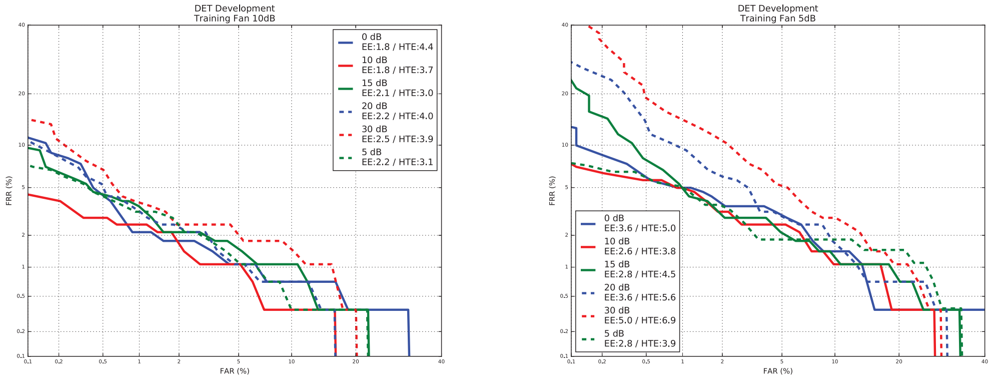

Figure B.4: DET curve development set for 2.5 minutes of audio in meeting room. Training with fan noise at 10dB(left) and 5dB(right) and testing with same noise type at different levels

Training/Test (SNR)	0dB	5dB	10dB	15dB	20dB	30dB	No noise
0dB	5.0/7.5	4.3/4.7	4.6/6.1	6.8/7.3	10.3/9.2	13.9/11.4	17.0/16.8
5dB	3.6/5.0	2.8/3.9	2.6/3.8	2.8/4.5	3.6/5.6	5.0/6.9	5.7/9.2
10dB	1.8/4.4	2.2/3.1	1.8/3.7	2.1/3.0	2.2/4.0	2.5/3.9	3.2/4.8
15dB	3.9/6.5	2.4/3.9	1.4/3.1	1.1/2.3	1.4/2.5	1.4/2.1	1.8/2.2
20dB	7.1/8.0	3.6/5.5	1.8/3.3	1.8/2.8	1.4/2.7	1.8/2.4	1.8/2.0
30dB	9.2/11.3	4.3/5.5	1.8/3.8	1.8/2.6	1.8/2.4	1.8/2.3	1.8/2.0
No noise	14.3/15.0	7.5/7.5	2.8/4.3	2.1/2.8	1.8/2.5	1.8/2.8	1.4/2.2

Table B.4: Fan Noise level. Results for development/test sets in terms of EER and HTER respectively. Train and test in meeting room (no reverberation), Fan noise at different noise levels.

B.1.4 Reverberation

Reverberation degrades the performance of voice id verification as well. Different acoustic impulse responses (AIR) are tested from a lecture room with a reverberation time (T_{60}) of 0.638 seconds, a meeting room ($T_{60} = 0.437s$) and a lobby ($T_{60} = 0.646s$) [60] where model of equation B.1 is used.

Figure B.5 shows that there is no big difference when training in one room and testing in a different one. Best result is achieved while training and testing in the lecture room, but the curves overlap each other showing no clear pattern.

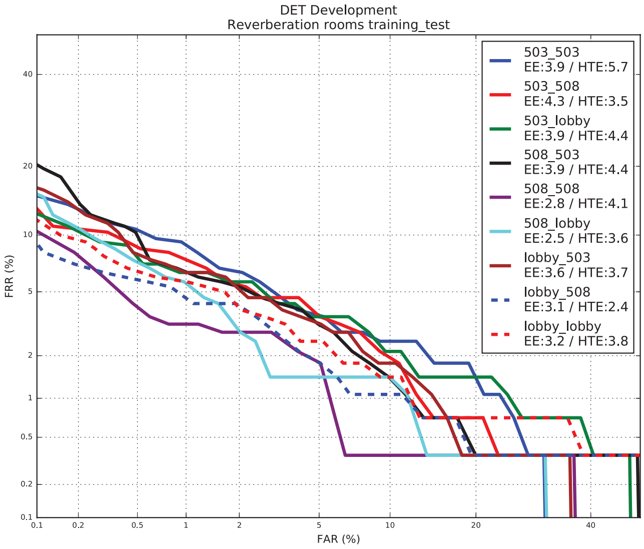

Figure B.5: Different room responses while training and testing without noise. Room 503 is a meeting room, room 508 is a lecture room and lobby

The real difference is observed when training and testing with and without reverberation. This is shown in Fig. B.6, where each subplot shows results for different rooms. As expected training and testing without reverberation gives better results. For reverberation in the test set is better to train with reverberation as well, where a difference between 0.4% and 1.1% is seen in EER when training with and without reverberation. Experiments with reverberation in the test set can degrade the results from 1.4% to 4.3% of EER.

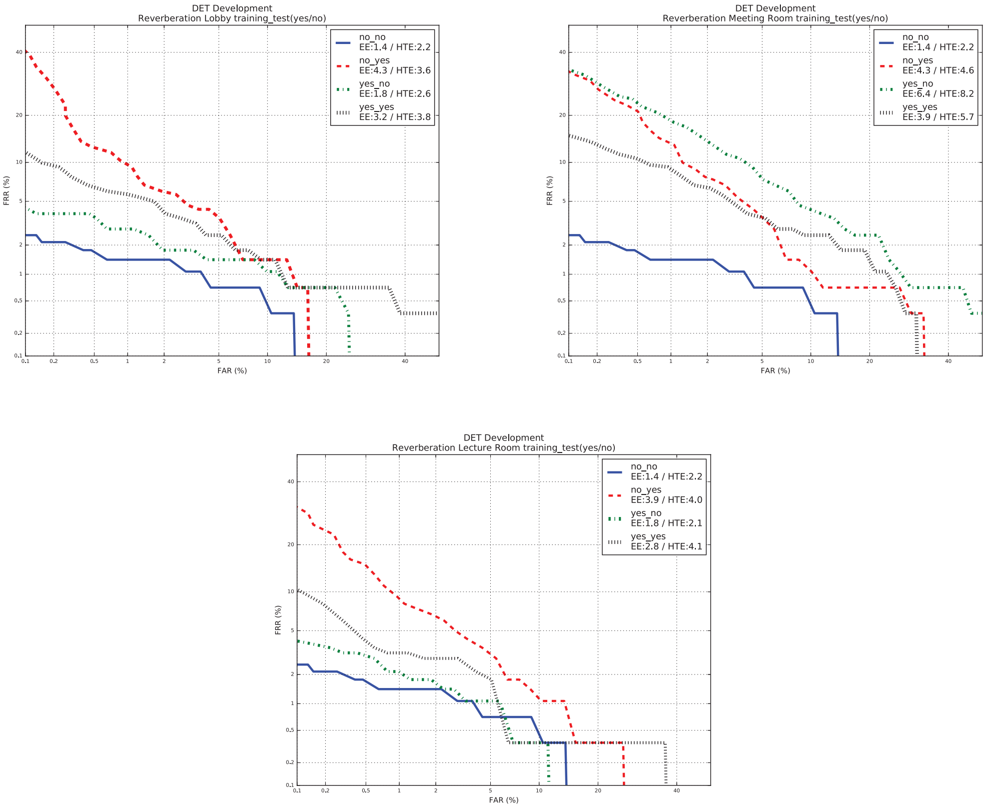

Figure B.6: Training and testing with and without reverberation for lobby(upper left), meeting room (upper right) and lecture room (bottom). Training and testing without reverberation (blue line) achieves 1.4% of EER. Worst results are obtained while training without reverberation and testing with (red line) which can varies from 3.9% to 4.3% depending on the room.

Difference between rooms are less harmful than difference between noise types for training and testing. A maximum difference of 1.8% is achieved between rooms (lecture and meeting) for training and testing in the same noise (babble) compared to a maximum difference of 2.9% when training and testing in the same room (lecture) for different noise type (fan and babble).

A full comparison of different AIR and noise types are shown in Table B.6 where EER and HTER are shown while training and testing at 10dB of SNR. This table shows similar results for testing the same kind of noise while training in different rooms.

Training/Test	Fan (from Lobby)	Babble (from lobby)	Lobby Fan	Lobby Babble	Lecture Fan	Lecture Babble	Meeting Fan	Meeting Babble
Fan(from Lobby)	1.4/2.7	2.1/4.2	5.0/5.4	8.9/9.8	3.9/4.2	4.6/7.3	3.2/5.9	7.5/9.1
Babble(from lobby)	3.2/3.5	2.8/5.1	5.3/8.9	6.4/7.6	7.9/9.2	4.5/7.5	5.0/7.3	5.0/9.3
Lobby-Fan	3.2/3.2	3.6/5.7	4.0/4.8	6.7/7.6	3.6/5.7	4.3/5.9	4.6/7.6	6.4/8.4
Lobby-Babble	6.4/7.1	5.4/5.8	6.4/7.1	6.8/7.6	8.5/9.9	5.4/6.4	6.7/9.0	6.4/7.4
Lecture-Fan	2.1/2.0	3.6/6.4	4.0/5.5	7.9/7.7	2.8/4.5	3.2/5.5	3.6/5.6	7.9/9.6
Lecture-Babble	4.6/5.1	4.6/6.2	5.0/5.1	5.3/6.8	7.1/8.2	4.2/6.1	5.4/7.8	6.0/8.1
Meeting-Fan	3.3/4.5	5.0/7.9	3.2/4.6	6.8/8.6	5.0/6.0	4.6/4.9	3.9/4.6	6.4/8.2
Meeting-Babble	7.8/8.6	5.3/6.7	6.0/6.6	7.1/7.3	8.2/6.3	6.8/7.5	6.4/7.4	7.5/8.9

Table B.6: Room and noise type. Results for development/test sets in terms of EER and HTER respectively. Train and test in different rooms and noise types at 10dB SNR.

Finally a comparison for all rooms and fan noise at different SNR levels is shown in Table B.7. This shows that differences in SNR level affects more the performance that difference in rooms. For the same SNR (0dB) in training and testing a maximum of 3% is achieved for different rooms (lobby and meeting room) while a maximum difference of 6.4% is obtained for training and testing in the same room(lecture) for different noise level (0-20dB).

Training/Test	Lobby 0dB	Lobby 10dB	Lobby 20dB	Lecture 0dB	Lecture 10dB	Lecture 20dB	Meeting 0dB	Meeting 10dB	Meeting 20dB
Lobby 0dB	6.7/7.9	7.1/8.0	8.2/8.6	8.5/8.5	8.2/7.8	10.0/10.2	6.4/8.2	6.1/7.1	8.1/9.2
Lobby 10dB	7.5/8.8	4.0/4.8	4.3/6.3	3.6/5.6	3.6/5.7	3.6/6.1	8.2/9.9	4.6/7.6	4.3/6.6
Lobby 20dB	13.2/13.6	5.0/4.9	3.9/3.7	4.7/5.3	2.8/3.7	2.5/3.6	14.2/14.0	3.9/6.0	3.8/4.1
Lecture room(0dB)	6.1/8.9	6.0/7.8	9.6/12.4	6.1/6.8	8.9/10.4	12.5/14.4	4.6/6.6	4.6/8.0	8.1/11.8
Lecture room(10dB)	11.4/12.2	4.0/5.5	3.8/5.9	3.2/4.2	2.8/4.5	4.6/5.9	12.1/13.8	3.6/5.6	3.2/4.3
Lecture room(20dB)	14.2/14.4	5.0/5.1	3.5/4.5	3.2/4.4	2.1/4.1	2.8/4.7	13.9/15.5	3.9/4.9	3.2/4.4
Meeting room(0dB)	10.7/10.8	11.4/10.8	13.1/13.7	11.1/9.5	13.9/13.3	17.4/15.5	8.2/9.5	8.5/9.7	12.5/12.5
Meeting room(10dB)	7.5/9.7	3.2/4.6	4.9/5.4	5.0/5.1	5.0/6.0	6.3/6.1	8.9/10.8	3.9/4.6	4.3/5.4
Meeting room(20dB)	12.5/13.6	4.3/5.8	3.9/5.4	5.7/5.1	4.7/4.1	3.9/3.9	16.0/16.8	6.0/6.8	3.9/5.7

Table B.7: Room and SNR. Results for development/test sets in terms of EER and HTER respectively. Train and test with fan noise for different room response and SNR levels.

B.2 Audio-visual verification under adverse conditions

Once voice id verification under adverse conditions has been studied, the improvement in performance while adding visual information will be crucial where bad results are obtained using only audio features.

To analysed visual adverse conditions the AT&T dataset [3] is used with 2D-DCT coefficients as features. Ideal and degraded images are used to analysed the possible improvement one can get when combining both modes. Three types of distortions are applied: Gaussian noise, blur and changes in illumination. Gaussian noise and blur are measured in terms of peak signal-to-noise ratio (PSNR) defined in equation B.4 where MAX_I is the maximum possible pixel value of the image and MSE is the mean squared error defined in equation B.5 using the original image I of size $m \times n$ and the noisy image K. For Gaussian noise a media of 0 is taken together with some pixel variance to achieve a PSNR of $8.5dB$. Blur is done by taking the average pixel of a kernel area and replace the central pixel with that value. Taking an area of 5×5 achieves a PSNR of $8dB$.

$$PSNR_{dB} = 10 \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \tag{B.4}$$

$$MSE = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} [I(i,j) - K(i,j)]^2 \tag{B.5}$$

Changes in illumination are defined with a gamma correction from equation B.6 and based on the exponent value g that is applied to the image I_{in}. Values of g are set between 0.2 and 2.0 which correspond to a PSNR of $4dB$ and $3dB$ respectively.

$$I_{output} = I_{in}^{1/g} \tag{B.6}$$

Figure B.7 shows examples for Gaussian noise, blur and changes in illumination for a sample face image from AT&T dataset [3]. Pairs of audio (Voxforge) and face image (AT&T) are used as input for the experiments where ten samples per person are divided into train (50%), development (30%) and test (20%) set. End factor analysis is applied for verification where 47 2D-DCT coefficients are extracted for each image and 19 MFCC plus energy, delta and delta-delta functions are used to compute 60 features per frame for audio signals. Motion vectors are not studied due to the lack of clean datasets with video signals.

DET curve of Fig. B.8 shows that combining clean audio and images with a third LDA classifier improves the performance of single mode methods as seen in Section 5.2. Results for the test set shows that scores from audio and images can be separated perfectly, where classes do not overlap, this is also due to the small set of audio/image pair (two per speaker) from the test set.

If noisy speech is used for training and testing the combination of both modes performs as good as image verification. Figure B.9 shows results for audio signals trained with fan and tested with babble noise at $5dB$ of SNR and clean images.

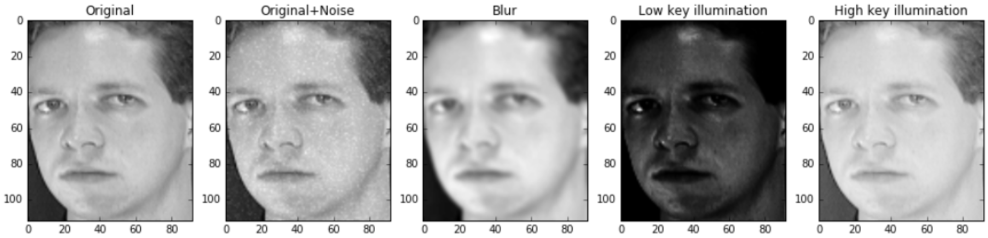

Figure B.7: Image sample AT&T dataset [3]. From left to right: Original image, Gaussian noise, blur and gamma correction ($\gamma = 0.2$ and $\gamma = 2.0$.)

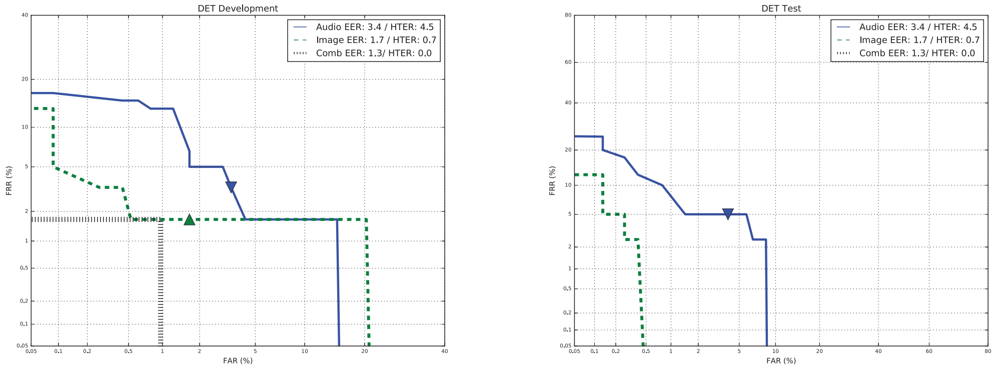

Figure B.8: Audio-Visual DET curve ideal conditions. Audio signal Voxforge dataset with 1 minute per speaker without noise and without reverberation. Visual signal AT&T dataset without any distortion.

Results for image scores and LDA third classifier overlap each other, showing that the audio score does not contribute to the final decision due to adverse conditions.

When similar performances are achieved by both modes, the combination of them outperforms each single result. This can be seen in Figure B.10 where third classifier achieves 2.4% of HTER compared to 8.4% and 9.7% of audio and image mode algorithms.

Finally, if adverse conditions for image signals are present the overall performance of the third classifier also get better results than each single mode methods as seen in Fig. B.11.

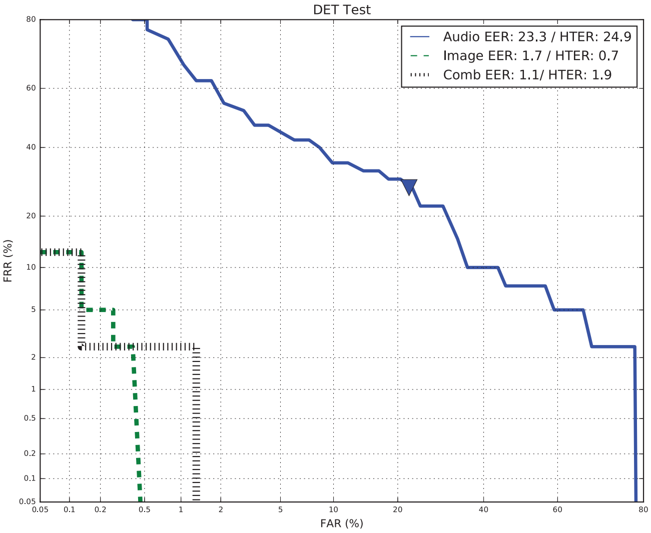

Figure B.9: Audio-visual DET curves low audio quality. Audio signal train with Fan noise at $20dB$ and tested with babble at $5dB$ in the meeting room. Image signal original samples for the AT&T dataset.

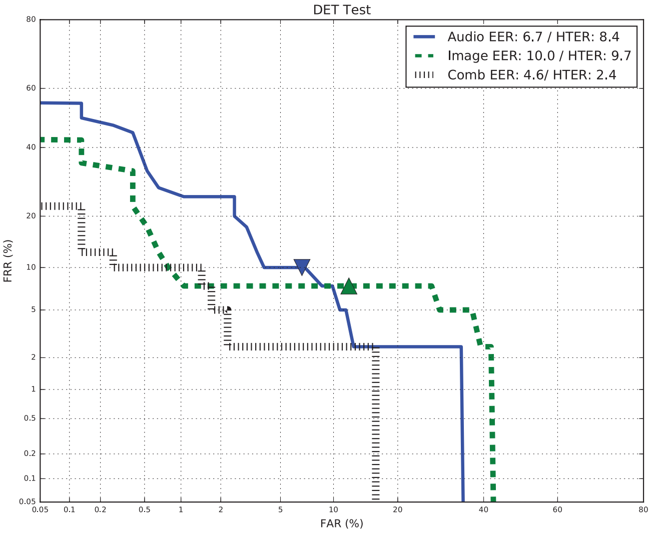

Figure B.10: Audio-visual DET curves similar conditions. Audio signal with fan noise at $30dB$ in meeting room. Visual signals trained with Gaussian noise with bright light and tested with blurred images in dark light.

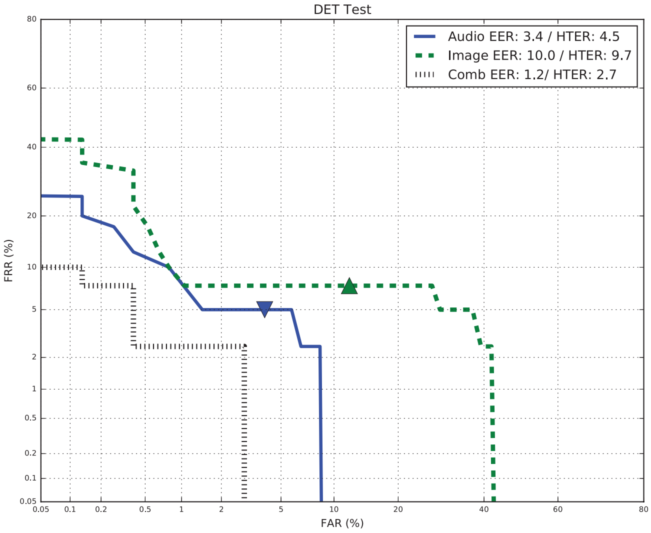

Figure B.11: Audio-visual DET curves low image quality. Audio signals original audio Voxforge dataset. Visual signals trained with Gaussian noise with bright light and tested with blurred images in dark light.

Bibliography

[1] J. Alabort-i-Medina, E. Antonakos, J. Booth, P. Snape, and S. Zafeiriou, "Menpo: A comprehensive platform for parametric image alignment and visual deformable models," in *Proceedings of the ACM International Conference on Multimedia*, ser. MM '14. New York, NY, USA: ACM, 2014, pp. 679–682. [Online]. Available: http://doi.acm.org/10.1145/2647868.2654890

[2] C. McCool, R. Wallace, M. McLaren, L. E. Shafey, and S. Marcel, "Session variability modelling for face authentication," *IET Biometrics*, vol. 2, no. 3, pp. 117– 129.

[3] "AT&T Database of Faces." [Online]. Available: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[4] P. Tresadern, T. F. Cootes, N. Poh, P. Matejika, A. Hadid, C. Levy, C. McCool, and S. Marcel, "Mobile biometrics: Combined face and voice verification for a mobile platform," *IEEE CS*.

[5] D. Evans, "The internet of things how the next evolution of the internet is changing everything," Cisco Internet Business Solutions Group, Tech. Rep. MSU-CSE-06-2, April 2011. [Online]. Available: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[6] S. Samee. (2017) Identity fraud soars to new levels. [Online]. Available: https://www.cifas.org.uk/newsroom/identity-fraud-soars-to-new-levels

[7] Pindrop. (2017) 2017 call center fraud report. [Online]. Available: https://www.pindrop.com/resources/download/report/2017-call-center-fraud-report/

[8] G. Chetty and M. Wagner, "Audio-visual multimodal fusion for biometric person authentication and liveness verification," in *Proceedings of the 2005 NICTA-HCSNet Multimodal User Interaction Workshop - Volume 57*, ser. MMUI '05. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2006, pp. 17–24. [Online]. Available: http://dl.acm.org/citation.cfm?id=1151804.1151808

[9] D. Dov, R. Talmon, and I. Cohen, "Audio-visual voice activity detection using diffusion maps," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 4, pp. 732–745, April 2015.

[10] ——, "Kernel-based sensor fusion with application to audio-visual voice activity detection," *CoRR*, vol. abs/1604.02946, 2016. [Online]. Available: http://arxiv.org/abs/1604.02946

[11] C. Sanderson and B. C. Lovell, "Multi-region probabilistic histograms for robust and scalable identity inference," in *Advances in Biometrics*, M. Tistarelli and M. S. Nixon, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 199–208.

[12] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.

[13] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, vol. 16, no. 5, 2008.

[14] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING*, vol. 13, no. 3, 2005.

[15] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING*, 2011.

[16] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *INTERSPEECH*, 2017.

[17] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *INTERSPEECH*, 2018.

[18] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun 1991, pp. 586–591.

[19] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *Computer Vision - ECCV 2004*, T. Pajdla and J. Matas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 469–481.

[20] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, Fourth Edition*, 4th ed. Orlando, FL, USA: Academic Press, Inc., 2008.

[21] O. M. Parkhi, A. Vedaldi, A. Zisserman *et al.*, "Deep face recognition."

[22] D. Jurafsky and J. H. Martin, *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Prentice Hall, 2009.

[23] A. Anjos, L. E. Shafey, R. Wallace, M. Günther, C. McCool, and S. Marcel, "Bob: a free signal processing and machine learning toolbox for researchers," in *20th ACM Conference on Multimedia Systems (ACMMM), Nara, Japan*, Oct. 2012. [Online]. Available: https://publications.idiap.ch/downloads/papers/2012/Anjos_Bob_ACMMM12.pdf

[24] T. Kinnunen and H. Li, “An overview of text-independent speaker recognition: From features to supervectors,” *Speech Communication*, vol. 52, pp. 12–40, 2010.

[25] Reynolds and Rose, “Robust text-independent speaker identification using gaussian mixture speaker models.” *IEEE Trans. Speech Audio Process.*, vol. 3, pp. 72–83, 1995.

[26] O. Glembek, L. Burget, N. Dehak, N. Brummer, and P. Kenny, “Comparison of scoring methods used in speaker recognition with joint factor analysis,” pp. 4057–4060, 04 2009.

[27] R. Vogt and S. Sridharan, “Explicit modelling of session variability for speaker verification,” *Comput. Speech Language*, vol. 22, no. 1, pp. 17–38.

[28] P. Kenny, “Joint factor analysis of speaker and session variability: Theory and algorithms,” *CRIM, Montreal,(Report) CRIM-06/08-13*, vol. 14, pp. 28–29, 2005.

[29] O. H. Andrew, S. Kajarekar, Sachin, and A. Stolcke, “Within-class covariance normalization for svm-based speaker recognition,” 01 2006.

[30] R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, Ed.

[31] P. Viola and M. J. Jones, “Robust real-time face detection,” *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004. [Online]. Available: https://doi.org/10.1023/B:VISI.0000013087.49260.fb

[32] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119 – 139, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002200009791504X

[33] M. P. T. Ojala and D. Harwood, “A comparative study of texture measures with classification based on featured distributions,” *Pattern Recognition*, vol. 29, no. 1.

[34] C. Atanasoaei, “Multivariate boosting with look-up tables for face processing,” Ph.D. dissertation, EPFL, 2012.

[35] G. Zhao and M. Pietikainen, “Dynamic texture recognition using local binary patterns with an application to facial expressions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 915–928, June 2007.

[36] X. Tan and B. Triggs, “Enhanced local texture feature sets for face recognition under difficult lighting conditions,” *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1635–1650, June 2010.

[37] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, "Local gabor binary pattern histogram sequence (lgbphs): a novel non-statistical model for face representation and recognition," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 1, Oct 2005, pp. 786–791 Vol. 1.

[38] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[39] J. R. Barr, K. W. Bowyer, P. J. Flynn, and S. Biswas, "Face recognition from video: a review," *IJPRAI*, vol. 26, 2012.

[40] X. Liu and T. Cheng, "Video-based face recognition using adaptive hidden markov models," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 1, June 2003, pp. I–I.

[41] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman, "Video-based face recognition using probabilistic appearance manifolds," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 1, June 2003, pp. I–I.

[42] H. Bilen, B. Fernando, E. Gavves, and A. Vedaldi, "Action recognition with dynamic image networks," *CoRR*, vol. abs/1612.00738, 2016. [Online]. Available: http://arxiv.org/abs/1612.00738

[43] A. Nagrani, S. Albanie, and A. Zisserman, "Seeing voices and hearing faces: Cross-modal biometric matching," *CoRR*, vol. abs/1804.00326, 2018. [Online]. Available: http://arxiv.org/abs/1804.00326

[44] R. Wallace and M. McLaren, "Total variability modelling for face verification," *IET Biometrics*, vol. 1, no. 4, pp. 188–199.

[45] C. Sanderson and K. K. Paliwal, "Polynomial features for robust face authentication," in *Proceedings. International Conference on Image Processing*, vol. 3, Sept 2002, pp. 997–1000 vol.3.

[46] J. Alabort-i-Medina and S. Zafeiriou, "A unified framework for compositional fitting of active appearance models," *CoRR*, vol. abs/1601.00199, 2016. [Online]. Available: http://arxiv.org/abs/1601.00199

[47] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[48] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis*, J. Bigun and T. Gustavsson, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 363–370.

[49] E. Khoury, L. El Shafey, and S. Marcel, "The idiap speaker recognition evaluation system at nist sre 2012," in *NIST Speaker Recognition Conference*. NIST, Dec. 2012.

[50] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments."

[51] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity." *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR).*

[52] J. S. Garofolo, W. M. F. Lori F. Lamel, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, *TIMIT Acoustic-Phonetic Continuous Speech Corpus.*, 1993, no. LDC93S1.

[53] C. McCool, S. Marcel, A. Hadid, M. Pietikainen, P. Matejka, J. Cernocky, N. Poh, J. Kittler, A. Larcher, C. Levy, D. Matrouf, J.-F. Bonastre, P. Tresadern, and T. Cootes, "Bi-modal person recognition on a mobile phone: using mobile phone data," in *IEEE ICME Workshop on Hot Topics in Mobile Multimedia*, Jul. 2012.

[54] K. Messer, J. Matas, J. Kittler, and K. Jonsson, "Xm2vtsdb: The extended m2vts database," in *In Second International Conference on Audio and Video-based Biometric Person Authentication*, 1999, pp. 72–77.

[55] E. Bailly-Baillière, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Mariéthoz, J. Matas, K. Messer, V. Popovici, F. Poree, and J.-P. Thiran, "The banca database and evaluation protocol," 06 2003.

[56] C. C. Atanasoaei, "Multivariate boosting with look-up tables for face processing," Ph.D. dissertation, ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE, 2012.

[57] I. Almajai and B. Milner, "Using audio-visual features for robust voice activity detection in clean and noisy speech," in *2008 16th European Signal Processing Conference*, Aug 2008, pp. 1–5.

[58] V. Peruffo Minotto, C. Lopes, J. Scharcanski, C. Jung, and B. Lee, "Audiovisual voice activity detection based on microphone arrays and color information," vol. 7, pp. 147–156, 02 2013.

[59] I. Matthews and S. Baker, "Active appearance models revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, Nov 2004. [Online]. Available: https://doi.org/10.1023/B:VISI.0000029666.37597.d3

[60] J. Eaton, N. D. Gaubitch, A. H. Moore, and P. A. Naylor, "Estimation of room acoustic parameters: The ace challenge," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 24, no. 10, pp. 1681–1693, Oct. 2016. [Online]. Available: https://doi.org/10.1109/TASLP.2016.2577502

[61] P. Verma and P. K. Das, "i-vectors in speech processing applications: a survey," *International Journal of Speech Technology*, vol. 18, no. 4, pp. 529–546, Dec 2015. [Online]. Available: https://doi.org/10.1007/s10772-015-9295-3