DEEP LEARNING FOR AUDIO VISUAL SPEAKER DIARIZATION

Argyrios S. VARTHOLOMAIOS

University of Thessaly
Department of Electrical and Computer Engineering

Volos, 2017

TITLE OF THESIS: DEEP LEARNING FOR AUDIO-VISUAL SPEAKER DIARIZATION

(ΒΑΘΙΑ ΜΑΘΗΣΗ ΓΙΑ ΟΠΤΙΚΟ-ΑΚΟΥΣΤΙΚΗ ΚΑΤΑΛΟΓΟΠΟΙΗΣΗ ΟΜΙΛΗΤΗ)

AUTHOR:

ARGYRIOS S. VARTHOLOMAIOS

Overseeing Professors:

Gerasimos Potamianos, Associate Professor
Antonios Argyriou, Assistant Professor

Submitted to
University of Thessaly in partial fulfillment
of the requirements for the degree in the department of

Electrical and Computer Engineering

Volos, 2017

ACKNOWLEDGMENTS

I would first like to thank my Thesis advisor Associate Professor Gerasimos Potamianos of the Department of Electrical and Computer Engineering at the University of Thessaly for his guidance and mentoring. With our frequent communication and cooperation I managed to overcome any hurdles that came in my way towards concluding my research. Additionally, I would like to thank my fellow student Alexandros Koumparoulis for his contribution and our helpful discussions.

Last but not least, I have to thank my family and friends for their inexhaustible and unconditional love and support throughout these years, during both the darkest and the fairest of days.

ΠΕΡΙΛΗΨΗ

Η καταλογοποίηση ομιλητή είναι η διαδικασία της αναγνώρισης του «ποιος μίλησε πότε» σε ένα κομμάτι ήχου ή βίντεο χωρίς να είναι γνωστά η διάρκεια της ομιλίας ή το πλήθος των ομιλητών. Για να επιτευχθεί, αυτό η διαδικασία της καταλογοποίησης χωρίζεται σε τρία μέρη, την εξαγωγή χαρακτηριστικών, την κατηγοριοποίηση, και τέλος την αξιολόγηση της καταλογοποίησης.

Κατά τη διάρκεια της εξαγωγής χαρακτηριστικών, συγκεκριμένα χαρακτηριστικά επιλέγονται από τα δεδομένα ήχου και βίντεο που στη συνέχεια θα χρησιμοποιηθούν ως είσοδος στο σύστημα. Αρχικά, όσον αφορά τα εικονικά χαρακτηριστικά, επιλέγεται ως περιοχή ενδιαφέροντος η ευρύτερη περιοχή γύρω από το στόμα του ομιλητή και καταγράφεται καρέ-καρέ. Δύο διαφορετικές μέθοδοι εντοπισμού της περιοχής ενδιαφέροντος αναπτύχθηκαν, ο ένας είναι στηριγμένος στην Επεξεργασία Εικόνας ενώ ο άλλος βασίζεται στη Βαθιά Μάθηση. Στη συνέχεια εφαρμόζεται διακριτός μετασχηματισμός συνημιτόνου (DCT) και επιλέγονται ως χαρακτηριστικά οι πρώτοι 30 συντελεστές του μετασχηματισμού. Αντίστοιχα, 39 συντελεστές MFCC επιλέγονται ως τα ηχητικά χαρακτηριστικά.

Για την καταλογοποίηση εφαρμόζεται η μέθοδος της ανάλυσης κανονικοποιημένης συσχέτισης (CCA) που χρησιμοποιείται για την εύρεση συσχέτισης μεταξύ των χαρακτηριστικών ήχου και εικόνας. Ως μέτρο της συσχέτισης επιλέχτηκε ο συντελεστής συσχέτισης του Pearson (PPMCC), και με τη βοήθεια μιας τιμής κατωφλίου γίνεται η κατηγοριοποίηση.

Στο τελικό στάδιο, αυτό της αξιολόγησης της καταλογοποίησης, εφαρμόζονται δύο διαφορετικές μέθοδοι, το δημοφιλές F1-score και ο Ρυθμός Σφάλματος Καταλογοποίησης (DER). Ο DER υπολογίζεται ως το ποσοστό του χρόνου που δεν αποδίδεται σωστά σε κάποιον ομιλητή.

Για τη διαδικασία της καταλογοποίησης αναπτύχθηκε μια εργαλειοθήκη που βασίστηκε στη βιβλιοθήκη υπολογιστικής όρασης OpenCV, στη γλώσσα προγραμματισμού Python και σε διάφορα στοιχεία αυτής όπως το numpy, το sklearn, το scipy, και το matplotlib.

Τέλος η βάση δεδομένων που χρησιμοποιήθηκε για τα πειράματα είναι η οπτικοακουστική βάση ψηφίων σε ιδανικό περιβάλλον της IBM, η οποία λειτούργησε ως βάση για την ανάπτυξη τριών συνόλων δεδομένων πολλαπλών ομιλητών που διαφοροποιούνται μεταξύ τους με βάση την κίνηση των ομιλητών.

ABSTRACT

Speaker diarization is the task of determining “who spoke when” in an audio or video recording that contains an unknown amount of speech and also an unknown number of speakers. In order to achieve this, the diarization process is divided to three parts: feature extraction, diarization, and evaluation.

During feature extraction, specific features are selected from the visual and audio data and fed into the diarization system. For the visual features, the selected region of interest (ROI) is the lower part of each speaker’s face where the mouth is detected and tracked frame by frame. Two different methods were implemented and are presented here for the ROI detection: one based on Image Processing (IP) and one on Deep Learning (DL). The discrete cosine transform (DCT) of this ROI is finally selected as the visual features. Likewise, 39 Mel-frequency cepstral coefficients (MFCCs) are selected as the audio feature.

For the diarization process, the audio-visual features are fused through a simple linear interpolation and Canonical Correlation Analysis (CCA) is applied through a windowed operation in order to find a meaningful correlation between the visual information from each speaker and the audio stream. As a measure of correlation the Pearson product-moment correlation coefficient (PPMCC) is used and a threshold value is utilized to classify each window/segment.

The last stage is the evaluation, where the system’s classification is compared to the dataset’s ground truth. Two methods were utilized for the diarization evaluation, the commonly used F1-score metric and the Diarization Error Rate (DER) as described by the National Institute of Standards and Technology in the NIST Rich Text (RT) Diarization evaluations. The DER is computed as the fraction of speaker time that is not correctly attributed to that specific speaker.

The diarization process was developed as a toolkit using the popular computer vision library OpenCV and the Python programming language accompanied with a collection of modules such as numpy, sklearn, scipy, and matplotlib.

Finally, the database used for the experiments is the IBM audio-visual databases of connected digits collected in a studio-like environment, which was the basis for three manually developed multiple speaker datasets differing in by the amount of speaker’s motion in the video.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

LIST OF ACRONYMS

CCA	Canonical Correlation Analysis
CDE	Click Detection and Elimination
CNN	Convolutional Neural Network
CoIA	Co-Inertia Analysis
DBN	Dynamic Bayesian Network
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DP	Deep Learning
FFT	Fast Fourier Transform
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
IP	Image Processing
MCM	Mutual Constant Motion
MCMS	Mutual Constant Motion with Silence
MFCC	Mel-Frequency Cepstrum Coefficients
NIST	National Institute of Standards and Technology
NN	Neural Network
PPMCC	Pearson Product-Moment Correlation Coefficient
PSD	Power Spectrum Density
SDE	Spectral Density Estimation
SSM	Single Speaker Motion
VAD	Voice Activity Detection
VGG	Visual Geometry Group

CHAPTER 1: Introduction

1.1 Speaker Diarization

Speaker diarization has become a fundamental quest in the realm of machine learning in recent years. Its main goal is to find the answer to the question of “who spoke when” in an audio or video. The main building blocks of a diarization system include the components for audio segmentation, speech detection, speaker clustering, and speaker identification in an audio-visual stream. In order to develop such system, a combination of machine learning methods is employed for the audio and video manipulation.

In any modality (video or audio) a very important factor in diarization is the choice of feature space so that it contains information that allows differentiating between speakers, and ideally no other kind of information. At first the audio stream has to be segmented between the speech and the non-speech parts. Afterwards, it is divided in to speech segments through a diarization window, and each segment is classified to the corresponding speaker. As far as manipulating the visual information, a subsystem is required to be developed in order to identify and track certain visual features in each speaker. Finally, the diarization system using the information acquired by the audio-visual processing makes an educated guess of who is speaking in each segment.

The boom of smart devices such as cameras, phones, and wearables has triggered an increase in the means of capturing audio and video and a consequent increase into vast amounts of data that need to be interpreted in a more cohesive way. That is why, as stated above, speaker diarization is a popular task and it has many commercial and forensic applications. In the audio-only domain it can be used in assisting speech recognition systems and facilitate the searching and indexing of audio archives, such as telephone communications, podcasts, and radio interviews. Likewise some of the most common applications, when multimodal information is available, are speaker identification in multi-speaker environments, such as broadcast news, multi-panel conferences, lectures, and meetings. Thus, speaker diarization is an extremely important area of audio-visual processing research.

Consequently, the different application domains of speaker diarization vary greatly and pose different problems that are dependent upon the recording environment, recording quality, lighting, number and positioning of speakers, noise, and overlapping speech. It becomes clear that different practices need to be applied to confront those issues and are very dependent on the means available for capturing audio-visual information. For instance special green-backgrounds rooms are developed to assist in speaker detection through background subtraction, and multiple cameras and microphones are used to capture multimodal information and tackle lighting issues and overlapping speech respectively.

1.2 Relevant Work

A great contribution to the field of speaker diarization has been made by the National Institute of Standards and Technology (NIST), through the NIST Rich Text (RT) Diarization Evaluations [2]. In this work, NIST organized evaluations of with transcribed data from conferences and lectures, and documented significant advancements in the domain of speech processing such as Speech-To-Text (STT) conversion, Speech Activity Detection (SAD) and Speaker Diarization. Most importantly, NIST established the standards for the evaluation of speaker diarization by introducing the Diarization Error Rate (DER), a metric that measures the fraction of speaker time that is not correctly attributed to the correct speaker. Naturally, DER has become the most common measure of diarization in the literature.

Furthermore, the necessity of commercial applications has led the development of on-line diarization systems as the ones presented in [3] and [4]. In [3], Hung and Friedland propose a Gaussian Mixture Model GMM-based, bimodal, multi-camera system with a single stationary microphone, applied on over 4.5 hours of non-scripted audio-visual meeting data. Initially, a GMM is trained on audio data, creating speaker models in an off-line speaker pool. During the on-line operation, each audio segment is matched with a speaker from the speaker model pool. Lastly, the multichannel visual stream is processed through a visual activity quantification system that detects the speaker's presence with a GMM that models the distribution of chrominance coefficients in the YUV color-space and finally displays the potential speaker. On a similar note in [4], Noulas et al. integrated audio-visual features for on-line speaker diarization using a Dynamic Bayesian network (DBN). Their model describes the causal relationship between the system state and observations extracted from the data, but tests were limited only to two-person camera views.

Another interesting approach was found in [5], where audio-visual synchrony was used to match speech with speaker. Initially, the authors of [5] applied a Hidden Markov Model (HMM) to achieve an agglomerative clustering of speakers. Next, for the visual features extraction of their system they used a combination of luminance variations of skin color, detected through the YUV histogram, and vertical pixel displacement, calculated with the Lucas-Kanade-Tomasi optical flow algorithm. Finally the audio-visual synchrony is detected and measured using two different methods: Mutual Information (MI) and Canonical Correlation Analysis (CCA). The later of is also employed in this Thesis.

1.3 Thesis Contribution

This Thesis investigates the idea of "audio-visual synchrony", which considers the strength of relationship between audio signals and video image sequences. Much like in [5], in this Thesis, speaker diarization is treated as a synchronization problem. The premise behind this approach is that once the number of speakers and their positioning has been detected in the visual data, the task of diarization is to find the speaker whose motion has the highest correlation with the audio stream. In other words, the intensity of someone's gestures, facial expressions, and lip motion should be consistent with the energy of the audio signal. Exploiting this relationship can allow us to identify the speaker.

1.4 Method Description

The task of audio-visual speaker diarization described in this Thesis is approached as a speaker-to-speech correlation problem. This logic is applied to a manually created dataset based on the IBM studio recordings database of connected digits. Unfortunately, this database contains only data. Therefore, to overcome this issue, three dual speaker datasets were developed from the original database consisting of 602 videos each. During every video both speakers follow one after the other to utter digits. Additionally, over the duration of one's speech, the other remains silent, and by the end of the video there is a five second segment of overlapping speech. The difference between each dataset is in the type of motion on the part of the silent speaker. Namely there are three types of motion:

- Constant motion –where both speakers appear to be talking.
- Silent motion –where the silent speaker moves but appears to be silent.
- No motion –where the silent speaker remains motionless.

The manual construction of these experimenting datasets allowed for the effortless establishment of the system's ground truth and thus a confident evaluation of the diarization. As for the diarization operation, it undergoes three stages in each dataset:

- audio feature extraction
- visual feature extraction
- audio-visual fusion and diarization

The whole speaker diarization process is explained below and illustrated in Figure 1.

For the audio feature extraction, the well established Mel-frequency cepstral coefficients (MFCCs) were used. In particular, during audio preprocessing, a 39–dimensional acoustic feature was extracted, consisting of 12 MFCCs, the normalized log-energy, and their first and second derivatives (delta and delta-delta).

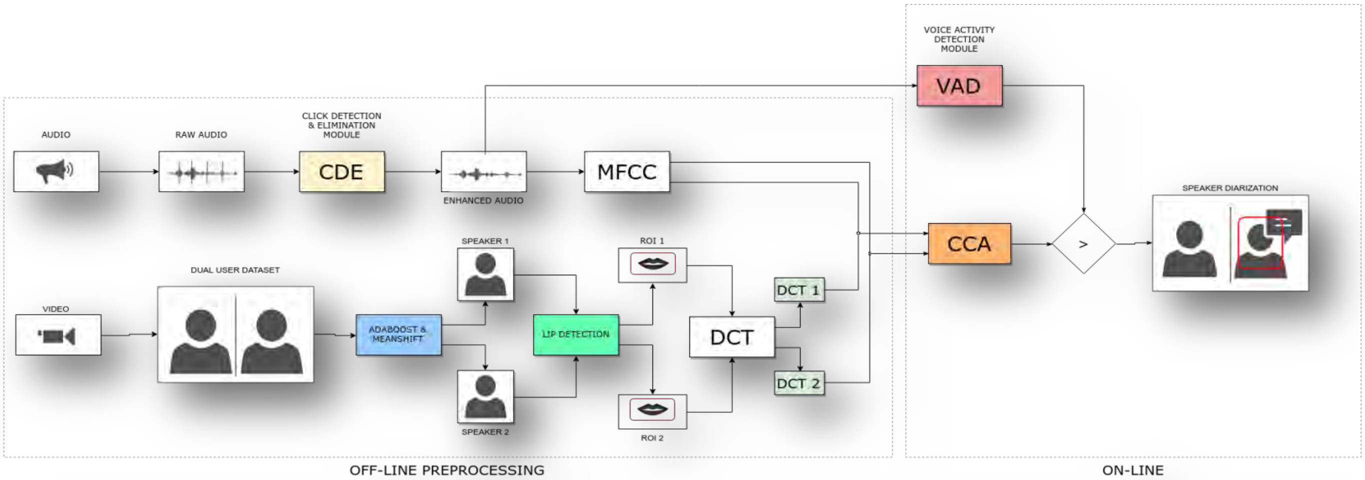


Fig. 1: *The proposed audio-visual diarization system consists of an off-line and an on-line stage. Acoustic and visual feature extraction constitute the off-line stage, while the diarization process occurs on-line.*

As far as the visual feature extraction is concerned, the area around the speaker's mouth was selected as a region of interest (ROI). In that direction two methods were implemented for capturing the ROI. The first is based on traditional Image Processing (IP), while the other involves Deep Learning (DL).

As far as the IP method is concerned, a series of computer vision algorithms was implemented in order to detect the face area and track the lip movement. Firstly, the popular Adaboost algorithm with Haar Cascades [6] is applied to detect the face. Next, the HSV color space is used to develop a skin mask to be used for eliminating false faces detected by the Adaboost algorithm and for background subtraction, by establishing a skin color region and removing with the pixels outside that region. Secondly, that initial detection is used to track the face with Meanshift algorithm [7] by moving the ROIs bounding box, following the face's motion reflected in the histogram of the back-projected image, to the area with maximum pixel density. Furthermore, a Sobel filter is applied in the vertical axis to detect potential lip lines, and the lip motion is tracked by employing a simple frame difference method [8]. Lastly, all the aforementioned methods were combined through a weighted sum in an aggregate mask while median and dilation filters were applied to reduce the noise and amplify the mask.

With regards to the second method, it was the product of the work of Alexandros Koumparoulis as a part of an audio-visual speech recognition system. The process is described in detail in his Thesis [28]. His ROI detection mechanism involves the implementation and training of various convolutional neural networks (CNNs) based on the manually annotated facial features of the original dataset.

In the final step of the visual feature extraction, in every video frame the broader mouth ROI is used as input for the two dimensional Discrete Cosine Transformation (2-D DCT). The first 30 DCT coefficients, selected in a zigzag fashion [9], are used as the visual feature vector.

For the purpose of the audio-visual fusion, the DCT coefficients of the visual features are interpolated to the audio feature frame rate. Ultimately, canonical correlation analysis (CCA), a common approach to detect the underlying relationship between two different signals, is used to measure whose speaker's visual features has the highest correlation with the audio features. The concept of CCA implies that high audio-visual correlation translates to audio-visual synchrony, and thus facilitates speaker detection and diarization.

1.5 Thesis Overview

This Thesis is divided into 7 chapters, each dedicated to describing the different stages leading to audio-visual speaker diarization. After this first introductory chapter, a breakdown of the content of the rest of the Thesis is presented:

Chapter 2: this chapter contains a detailed description of the original IBM studio dataset, as well as the process of creating the three dual-speaker datasets from it. Additionally the logic behind the system's ground truth is presented along with a description the voice activity detection system.

Chapter 3: here is attempted a more in-depth analysis of the feature extraction process. The chapter is divided between acoustic and the visual feature extraction, and the mathematic formulas of MFCC and DCT are explained in detail. In addition, an exhaustive overview of the ROI extraction process is presented along with figures relative to the different stages.

Chapter 4: a comprehensive and thorough explanation of the definition of CCA, along with its mathematic interpretation is offered in this chapter. Lastly, the reasoning behind using CCA in a speaker diarization system is revealed.

Chapter 5: this chapter introduces the two metrics used for the evaluation of the diarization system. The Diarization Error Rate and F-score are utilized to measure the performance of diarization on each of the datasets.

Chapter 6: the results of the speaker diarization experiments are presented in this chapter. The results are provided in tables and figures and explained with comments and examples.

Chapter 7: the final chapter contains a summary and discussion of the results of chapter 6, along with a meaningful discussion about future prospects and improvements of the speaker diarization pipeline.

CHAPTER 2: Database

2.1 Database Description

The experiments and the evaluation of the described speaker diarization process were conducted on the IBM audio-visual studio database of connected digits, which was also used in [10]. This database consists of a single speaker at a time, recorded in frontal head pose under a controlled environment with uniform background and lighting. In each video of the database the speaker pronounces short strings of connected digit from zero to nine. A total number of 50 unique speakers appear in the recordings creating a set of 6689 videos with a total duration of approximately 10 hours.

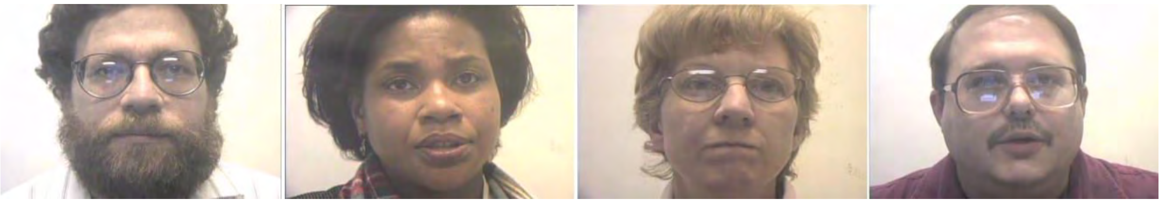


Fig. 2: *Example video frames of four subjects of the original database*

As far as the technical characteristics of the database, are connected each video is MPEG2-encoded, at 30 frames per second and in 704x480 pixel resolution. Regarding the audio, it is recorded in a single audio channel with a 16 kHz sampling rate.

2.2 Dataset Construction

Unfortunately, the original database contained single speaker recordings only. Consequently, a database appropriate for speaker diarization needed to be created. Working towards that direction meant combining the audio-visual data from the original database, creating a multi-speaker database.

The first step was to combine all the 6689 short, single speaker videos into longer, single speaker videos. Therefore, for every five short videos of every subject, one longer was generated, by concatenating multiple short ones, thus resulting in 1205 new videos, which would be the bedrock of the new database. Naturally, the same process was used for the audio data as well.

After establishing the foundations of the new database, the generated videos needed to be combined further in order to create the multi-speaker bimodal audio-visual data for the purpose of speaker diarization. With this intention, three unique dual-speaker datasets were developed. The basis of these three datasets was formed by running a script that randomly combined two unique subjects thus creating 602 new dual-speaker, frontal face videos. Additionally, each video was divided into three parts:

- A part where the left speaker is talking and the right remains silent for the duration of the left speaker's speech.
- A part where the right speaker is talking and the let remains silent for the duration of the right speaker's speech.
- A part where both speakers are talking, creating an overlapping speech segment of 5 seconds.

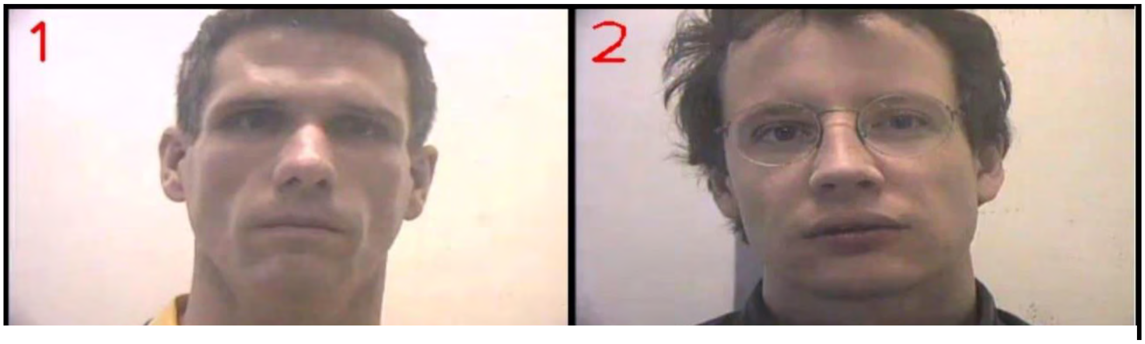


Fig. 3: *An example of a dual-speaker video. This video belongs to the SSM dataset.*

Despite the fact that the new datasets share some common traits in terms of structure and content, they vary greatly in terms of speaker motion. Following is a presentation of the three datasets that were developed in order to test the limits of the diarization system:

2.2.1. Single Speaker Motion Dataset (SSM): This is the most straightforward of the three datasets. In these videos only the speaker who is currently talking is allowed to be in motion, while the other one remains motionless, excluding the 5 seconds of overlapping speech segment where both subjects move and talk. This dataset is expected to yield the best results in terms of diarization since the speech is likely going to be correlated with the speaker in motion. Each video is divided in 3 segments as described in Table 1.

Currently Speaking	Left Speaker	Right Speaker	Duration
left speaker	in motion	motionless	left speaker's audio track
right speaker	motionless	in motion	right speaker's audio track
both	in motion	in motion	5 seconds

Table 1: *SSM dataset description*

2.2.2. Mutual Constant Motion Dataset (MCM): This dataset was developed with a little more novelty introducing motion to both subjects. In other words both speakers are in constant motion, regardless of who is actually talking, excluding again the 5 seconds of mutual speech and motion segment. This dataset is developed to test the extremes of the diarization system and naturally is expected to yield the worst results, since the diarization process could become problematic when both subjects are in constant motion. Each video is divided into 3 segments as described in Table 2.

Currently Speaking	Left Speaker	Right Speaker	Duration
left speaker	in motion	in motion	left speaker’s audio track
right speaker	in motion	in motion	right speaker’s audio track
both	in motion	in motion	5 seconds

Table 2: *MCM dataset description*

2.2.3. Mutual Constant Motion with Silence Dataset (MCMS): This dataset was developed with a more nuanced approach, taking into consideration the silence of the no-talking speaker. The idea behind this was to attempt and capture the motion of each subject while not talking, with the purpose of simulating the motion of a silent person. In order to achieve this, a voice activity detection (VAD) system was developed and employed to capture the silent parts in each speaker's audio stream. The VAD system is a simple thresholding algorithm that parses the audio stream and detects sequential silent segments and it is further below. Afterwards, the timestamps of these silent segments were used to capture clips of the original video where the subject remained silent. Finally these clips were sequentially concatenated and projected iteratively for the duration of the other subject’s speech. This is the most interesting dataset and is expected to test the quality of the diarization system. Each video is divided in 3 segments as described in Table 3.

Currently Speaking	Left Speaker	Right Speaker	Duration
left speaker	in motion	in “silent motion”	left speaker’s audio track
right speaker	in “silent motion”	in motion	right speaker’s audio track
both	in motion	in motion	5 seconds

Table 3: *MCMS dataset description*

2.3 Voice Activity Detection System

Voice activity detection (VAD), is a very important component in any speech/audio processing system including speech coding, speech recognition, speech enhancement, and audio indexing. The required characteristics of an ideal VAD system are: accuracy, robustness, simplicity, and resistance to noise. In this particular diarization system the voice activity detector enjoyed a twofold application. Firstly, it was used in order to determine the ground truth of the diarization system, a subject discussed later on. Additionally, in similar manner, it was utilized in the development of the MCMS dataset as described above in section 2.2.3.

The concept behind the development of the VAD system was to find a certain value in the energy of the audio signal that would function as a threshold and service in classifying the audio signal into speech and non-speech segments. Specifically, the VAD algorithm included two tasks: defining a silence threshold and detecting the silence.

In this direction, the silence threshold is established by selecting a 500 ms audio sample located at the beginning of each recording. Since there is a lack of speech in the beginning of every recording, the threshold is defined by calculating the average energy of said sample. Afterwards, the audio signal gets segmented and parsed through a 250 ms window, and if the average energy of the window exceeds the threshold, it is designated as a potential silent segment. Finally, if three or more consecutive audio segments are found to contain low energy values, they are classified as silence. During the VAD system implementation two problems were encountered.

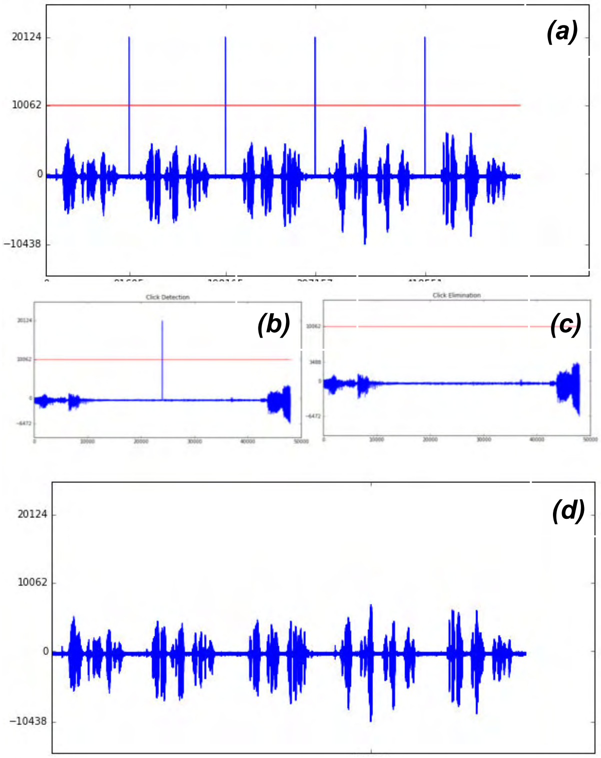


Fig. 4: *Click Detection and Elimination. (a) Click effect observed in audio signal, the horizontal red line indicates the threshold; (b) enhanced view of click effect before elimination; (c) enhanced view of click effect after elimination; (d) audio signal after click elimination*

The first and major problem was an increased noise spike observed at the beginning of each audio track, which had an effect in determining the VAD threshold value. That meant that there were 5 such spikes in every audio track, Figure 4(a), since each track consists of 5 separate clips, containing speech from the same speaker. These effects of short time impulsive disturbances in the signal are called impulse noise or click effect and are usually attributed to electromagnetic interference, poorly maintained recording equipment, or digital manipulation. In order to overcome this obstacle a click detection and elimination (CDE) algorithm was developed. The CDE algorithm defines a threshold value that labels signal values above that threshold as impulse noise. After the click detection, a median filter is applied to eliminate the effect. With regards to the details of the CDE implementation, initially the half point of the maximum impulse peak is set as the threshold value. In addition, the audio is parsed through a 100 ms window, and if the maximum signal value of the audio segment in the window exceeds the threshold value, the segment contains a potential click. If three or more consecutive audio segments are found to contain a potential click, then all those segments are labeled as an inconsistency. Finally in order to remove that inconsistency, a median filter of the windows before and after the designated click is calculated and applied over the conflicting area, thus removing the click. The whole process is fast and efficient, and no preprocessing is required.

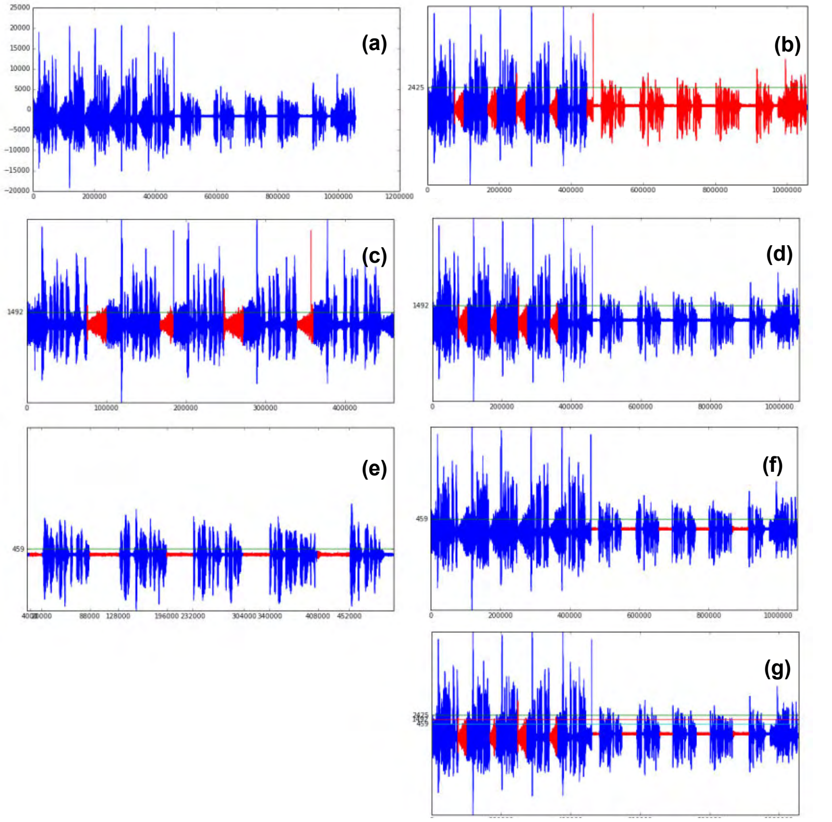


Fig. 5: *Problematic initial silence threshold: (a) audio signal with inconsistent frequency levels between the combined audio tracks: (b) failed VAD attempt to detect silence; (c) silence detection only on the first part; (d) the effect of processing the first part; (e) silence detection only on the second part; (f) the effect of processing the second part; (g) combined result of VAD*

The second problem in developing the VAD system had to do with calculating the silence threshold value. Specifically, each dual-speaker audio track consists of two different subjects being recorded at different conditions, thus creating varying noise effects between each speakers' speech, hindering establishing an efficient silence threshold value. As a result, in order to overcome this obstacle, the VAD algorithm had to be applied in both audio segments separately and combine the results at the end. The process is illustrated in Figure 5. The horizontal green lines in the figures indicate the potential silence threshold, and the red color indicates silent audio segments. Initially in Figure 5(b) the algorithm sets a silence threshold that depends on the speech frequency of the first part of the audio signal (speaker 1). That threshold is too high resulting in the second part of the signal (speaker 2) being identified as silence. In order to repair this behavior the VAD algorithm processes the two parts separately by setting a different threshold for each part, as it is observed in figures 5(c) and 5(e). Finally, the detected silence intervals from both parts are merged and projected on the original signal. The outcome can be seen in Figure 5(g).

The proposed VAD, as stated above, is applied in both the preprocessing stage (off-line) by creating the MCMS dataset and in the on-line stage by defining the ground truth. The advantages of the system are its simplicity, its efficiency and the fact that it offers real-time audio processing.

2.4 System’s Ground Truth

This section is important in order to understand the evaluation process of the diarization system. During the evaluation, the classification produced by the diarization system is compared with the system’s ground truth and updates the system’s metrics.

For the purpose of establishing the ground truth, the speech duration of each subject is necessary along with the duration of the overlapping speech part. This can be calculated by measuring the duration of each single speaker video from the original database. During the diarization process, the total duration of the video is divided into segments, but only the segments containing speech are taken into consideration. In other words, during the audio-visual fusion the audio is being processed by the VAD system in order to recognize the silent segments and ignore them. To summarize, the system’s ground truth consists of the speech-only segments of each speaker along with the overlapping speech part.

CHAPTER 3: Feature Extraction

Feature extraction is a crucial preprocessing step in pattern recognition, audio-visual processing, and machine learning problems. A feature can be described as “a prominent or distinctive part, an attribute or aspect of a greater body of information”. In essence, feature extraction is a medium for evoking meaningful, informative and non-redundant values (features) from an initial set of data that will later facilitate the learning mechanism.

When performing analysis of complex data, such as audio or video, one of the major problems derives from the number of variables involved. Analysis with a large set of variables generally requires a large amount of memory and computation power; also it may cause a classification algorithm to overfit to training samples and generalize poorly to new samples. Thus, feature extraction involves reducing the amount of resources required to describe a large set of data. It is a general term for methods of constructing combinations of the variables to get around these problems, while still describing the data with sufficient accuracy. In this regard, feature extraction can be described as a case of dimensionality reduction.

3.1 Audio Feature Extraction

Speech is based on a sequence of discrete sound segments that are linked in time, these segments are called phonemes. Phonemes are assumed to have unique articulatory and acoustic characteristics. In a speech detection system it is paramount to engulf these characteristics as a part of the audio feature extraction process. Therefore, Mel frequency cepstral coefficients (MFCCs) are employed in this particular diarization system, a feature widely utilized in many automatic speech and speaker recognition systems [13]. In order to illustrate the relationship between phonemes and Mel-frequency cepstrum, a brief introduction to phonology and signal processing is necessary.

While the human vocal mechanism can produce an almost infinite number of articulatory gestures (mouth movements), the number of phonemes produced is finite. English for instance, as spoken in the United States, contains 16 vowel and 24 consonant phonemes [11], while the Greek language consists of 25 phonemes in total [12]. Each phoneme has distinguishable acoustic characteristics, and in combination with other phonemes, forms larger units such as syllables and words. Knowledge about the acoustic differences among these sound units is essential to distinguish one word from another in a speech recognition system. When speech sounds are connected to form larger linguistic units, the acoustic characteristics of a given phoneme will change in correlation with the shape of its immediate phonetic environment, because of the interaction among various anatomical structures (such as the tongue, lips, and vocal chords). The shape of the vocal tract determines what sound is uttered and manifests itself in the envelope of the short time power spectrum [11].

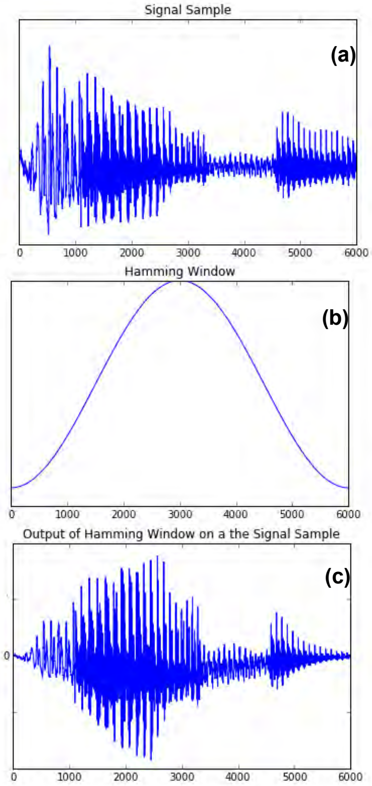


Fig. 6: *Example of Hamming windowing: (a) initial signal sample; (b) Hamming window; (c) Hamming window output*

MFCCs are dominant features for speech recognition since the Mel-frequency cepstrum (MFC) is the representation of the short term power spectrum of a sound in the Mel-scale. The Mel-scale relates the perceived frequency, or pitch, of a pure tone to its actual measured frequency. Humans are much better at discerning small changes in pitch at low frequencies than they are at high frequencies. Incorporating this scale makes the audio features match more closely what humans hear. The formula for converting from frequency to Mel scale is [13]:

$$M(f) = 1125\, ln(1 + \frac{f}{700}) \quad (3.1)$$

To transition from Mels to frequency is given by:

$$M^{-1}\,(m) = 700\, exp(\frac{m}{1125}) - 1 \quad (3.2)$$

where f is the frequency in Hertz (Hz) and m the frequency in Mels. The calculation of the MFCC features is a multistep process that is described below:

a) **Framing:** This is a signal quantization step. The speech signal is non-stationery but can exhibit quasi-stationary behavior in shorter intervals. This is why the signal is segmented in frames. Each frame is 25ms with a 10ms step. If the frame is much shorter, the included samples are not enough to get a reliable spectral estimate, if it is longer the signal changes too much throughout the frame. In a 16 kHz signal with a 25ms frame each frame contains $0.025 * 16000 = 400\, samples$, while each step is 160 samples.

b) **Windowing:** This step has to do with tapering the previous sampling process through a window function. For that purpose, the Hamming window is used. The effect of a Hamming window on a signal can be seen in Figure 6**,** while its effect on the whole audio signal is presented in Figure 7(b). The formula of the Hamming window is:

$$w[n] = \begin{cases} 0.54 - 0.46cos(\frac{2\pi n}{L}) & 0 \leq n \leq L - 1 \quad (3.3) \\ 0 & otherwise \end{cases}$$

c) **Discrete Fourier Transform (DFT):** This step is used to calculate the power spectrum density (PSD) of each frame by using DFT on the windowed signal. Direct computation of the DFT takes $O(N^2)$ complex multiplication so other algorithms are used, known as the Fast Fourier Transforms (FFT), that take $O(Nlog_2 N)$ complex multiplications and calculate DFT indirectly. This is motivated by the human cochlea (an organ in the ear) which vibrates at different locations depending on the frequency of the incoming sounds. Each location of the cochlea that vibrates, fires different nerves informing the brain that certain frequencies are present. A 512 point FFT is performed in the system but only the first 257 coefficients are kept. The process up until this step is displayed on figure 7.

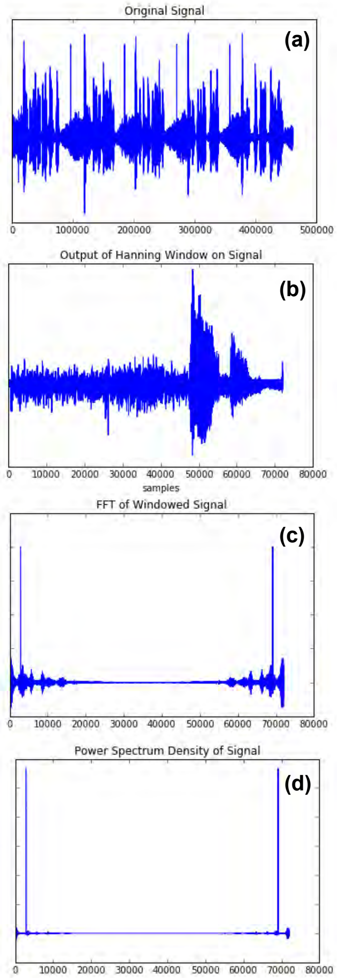


Fig. 7: *Calculating the PSD of a single speaker audio track; (a) original signal; (b) the signal through a 25ms Hamming window with 10ms window step; (c) FFT of windowed signal; (d) PSD of signal*

d) **Mel filtering:** This step applies the Mel filterbank to the power spectra. In signal processing, a filterbank is an array of band-pass filters that separates the input signal into multiple components, each one carrying a single frequency sub-band of the original signal. In this case a 26-filter Mel filterbank is used (Figure 8(a)); the first filter is very narrow and gives an indication of how much energy exists near 0 Hz. As the frequencies get higher the filters get wider as they become less susceptible to variations. The effect of the different filters is observed in Figures 8.(c) and (d).

e) **Logarithm of all filterbank energies:** This is a compression operation that makes the features match more closely to what humans actually hear. The logarithm allows the use of cepstral mean subtraction, which is a channel normalization technique.

f) **Discrete Cosine Transform (DCT):** A discrete cosine transform (DCT), expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. In essence DCT is a dimensionality reduction and compression process because only the first 12 coefficients are kept since they are of the most important. These are the MFC coefficients.

g) **Deltas calculation:** Deltas and delta-deltas are known as velocity and acceleration coefficients. MFCCs contain the power spectrum of a single frame but significant information can also be found in the trajectories of the MFCCs over time. This is why the first and second MFCC derivatives are also included in the audio feature vector. The formula for the calculation of deltas is:

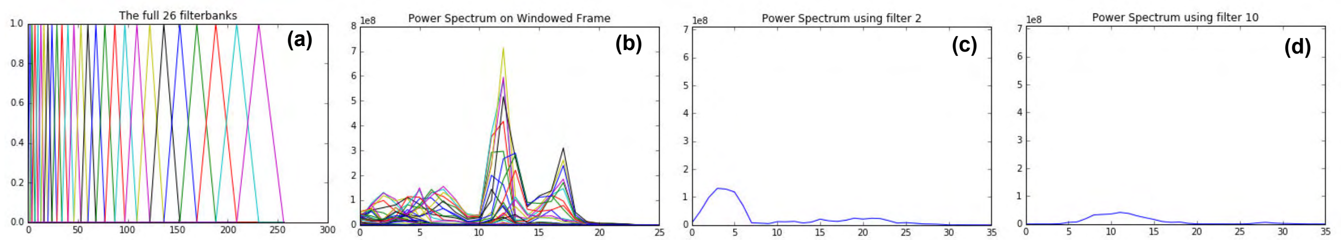


Fig. 8: *Demonstration of filterbanks on PSD: (a) The full 26 filterbanks; (b) power spectrum on windowed frame; (c) power spectrum using filter 2; (d) power spectrum using filter 10*

$$d_t = \frac{\sum_{n=1}^{2} n(c_{(t-n)} - c_{t+n})}{2\sum_{n=1}^{2} n^2} \quad (3.6)$$

where d_t is the first derivative coefficient of the 12 c_t coefficients produced in stage (f). The second derivative coefficients are calculated by the same formula by replacing c_t with the first derivative coefficients.

To summarize, the 12 first MFCCs along with the energy and their first and second derivatives are selected. In total, 39 are the selected features for the acoustic feature extraction.

	Energy index 0	Indices 1-12
MFCC	1	12
Delta	1	12
Delta-Deltas	1	12
39 acoustic features	**3**	**36**

Table 4: *Acoustic feature vector details*

3.2 Visual Feature Extraction

A crucial step towards speaker diarization is the detection of the subjects in a visual stream. The face of a subject contains the necessary information to determine whether the subject is talking. For this reason, the face of each subject in a video is the region of interest (ROI) for the system's visual feature extraction.

Much like [3] and [4], this diarization system has a preprocessing stage (off-line training), in which the ROI extraction occurs. In that direction, two different methods were used to test the robustness of the diarization system. The first applies the popular Adaboost and Meanshift algorithms along with a combination of image processing techniques for lip tracking. The second method involves deep learning and the usage of convolutional neural networks (CNNs). This method was developed and implemented by Alexandros Koumparoulis [28] as a part of his diploma Thesis.

Both ROI extraction methods are examined, tested, and compared in the diarization system, and the results are presented in chapter 6.

3.2.1 Image Processing Method

In machine learning, computer vision is the science field that tries to imitate the part of human cognition that understands and interprets visual stimuli and reproduce it to a machine or computer in order to gain high-level understanding from digital images or videos. In essence, it involves the development of a theoretical and algorithmic basis to achieve automatic visual understanding from a single image or a sequence of images.

A popular computer vision framework is OpenCV [18], a programming library developed for functions mainly aimed at real-time computer vision. OpenCV was originally developed by Intel's research center in Nizhny Novgorod (Russia), it was later supported by Willow Garage, and is now maintained by Itseez. The library is cross-platform and free for use. The library version for the Python programming language was used for implementing the ROI extraction along with a collection of Python modules such as numpy, sklearn, scipy, and matplotlib.

The ROI extraction task is divided into three parts. The detection of the subject’s face, the detection of the subject’s facial features of interest (i.e. mouth, eyes, etc.), and the tracking of these features. The process was executed on the individual single speaker data, and the results were later integrated to the three multi-speaker datasets. The different methods involved in the ROI extraction are illustrated next.

a) **Adaboost Algorithm:** Boosting is an approach to machine learning based on the idea of creating a highly accurate prediction rule by combining many relatively weak and inaccurate rules. The output of the other learning rules ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. The AdaBoost (Adaptive Boost) algorithm of Freund and Schapire [14] was the first practical boosting algorithm, and remains one of the most widely used and studied, with applications in numerous fields.

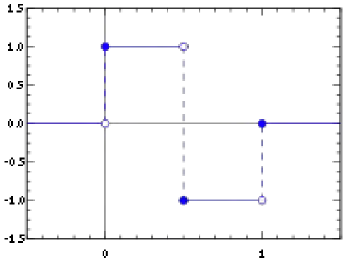


Fig. 9: *The Haar wavelet [16]*

One major application of Adaboost comes from the work of Viola and Jones [6] in the field of visual object detection using a Haar-like feature classifier. Haar-like features are conceptually based on the Haar wavelets, a step function taking values 1 and -1, on $[0,\frac{1}{2})$ and $[\frac{1}{2},1)$, respectively. The graph of the Haar wavelet is given in Figure 9. It is known that any continuous function can be approximated uniformly by Haar functions [15]. In the same sense any visual information can be described by Haar-like features. Instead of working with pixel intensities (RGB pixel values), which is computationally heavy, Viola and Jones proposed working at specific locations of an image through a detection window. By dividing the window in rectangular adjacent regions they sum up the pixel intensities in each region and calculate the difference of these sums. These differences are used to broadly classify image subregions. Two Haar classification examples can be seen in Figure 10. A single Haar-like feature, classifying a small region of an image, is a weak classifier and cannot by itself recognize an object. That is where the

AdaBoost algorithm comes into play by combining several Haar-like features into an organized body of classifiers called Haar cascade.

The OpenCV library proposes a nuance in its implementation of the Viola-Jones algorithm by adding a feature that allows object detection through a multi-scale image pyramid, such that the face detection can be scale-invariant. Finally, in the speaker diarization system the Viola and Jones method is used in two occasions; for an initial detection of the subject's face through a frontal face pre-trained Haar cascade, and for a detection of the subject's mouth in the region of interest of the lower half of the subject's face.

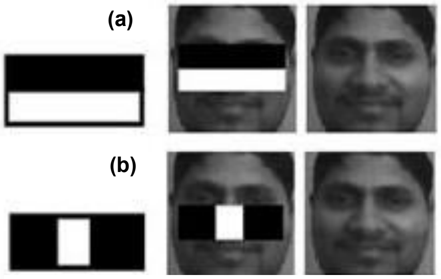


Fig.10: *Haar-like feature classification: (a) Haar feature that looks similar to the eye region, which is darker than the upper cheeks, is applied onto a face; (b) Haar feature that looks similar to the bridge of the nose is applied onto the face [16].*

b) **Mouth Detection with Computer Vision:** In the case of failing mouth detection by Adaboost, a series of image processing steps occurs on the lower part of the detected face.

Initially a vertical Sobel edge detection filter is applied on the lower part of the face. Sobel filter or Sobel operator is an image processing technique that convolves two 3x3 kernels with the original image. Of the two kernels one is responsible for horizontal edge detection and the other is for vertical. Supposing A is the array representing the original image the Sobel operation is described by the following formula:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A \quad and$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \quad (3.7)$$

The idea behind using the horizontal Sobel filter is to attempt a detection of the lips outline and create a mask around it. A hindering factor to that process can be facial hair presence. The effect of the horizontal Sobel filter is seen on Figure 11(c).

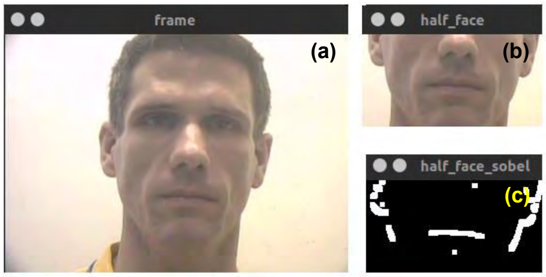


Fig. 11: *Effects of the Sobel filter: (a) original image; (b) lower half of detected face; (c) vertical Sobel edge detector.*

c) **Skin Detection:** Next, a skin detection algorithm is applied on the image of the lower half of the detected face, in order to amplify the confidence of the lip detection. This technique is based on the lip detection algorithm described on [17].

Initially, histogram equalization is applied on the lower part of the face. Histogram equalization is a technique used to enhance the contrast in an image by redistributing the image intensities across its histogram. The distribution of the histogram is divided in a range of values called bins, the aim is to redistribute those bins in a uniform distribution; the effect can be seen on Figure 13.

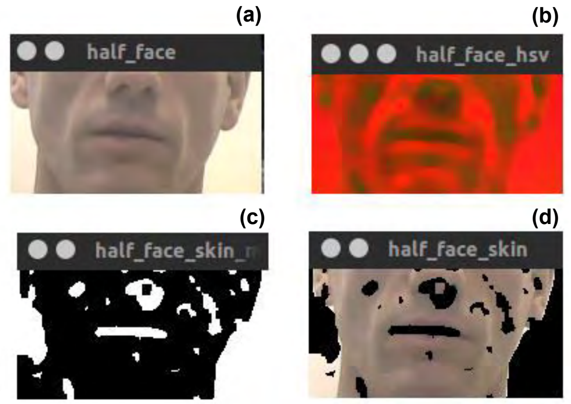


Fig. 12: Skin detection: (a) original lower half face; (b) HSV color space transition; (c) skin color mask; (d) mask application.

Afterwards, the selected image region is transformed to the HSV color space from the original RGB. The HSV color space modifies the image pixel from the red (R), green (G) and blue (B) channels to the hue (H), saturation (S) and value (lightness) (V) channels. The idea behind this, as mentioned in [19], is that in the RGB color space the human skin color information is divided between the red and blue channels in a color scale from yellow to brown. Whereas in the HSV color space, skin color information, regardless of the other channels, resides solely in a low range of hue channel values. These low hue values are thresholded and classified into binary values (0, 255), thus creating a skin mask, that is later applied on the original image. The procedure can be seen in Figure 12.

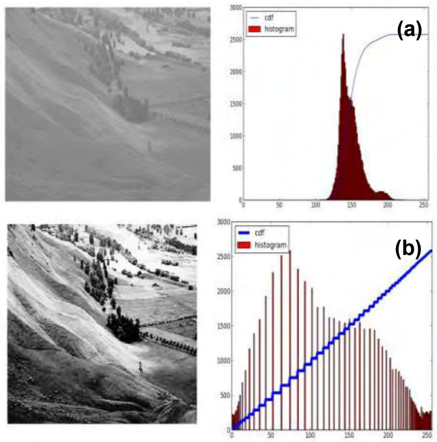


Fig. 13: Histogram equalization [18]: (a) original gray-scale image with low contrast and its narrow histogram; (b) image after equalizing the distribution of its histogram.

d) **Frame Difference**: The final visual manipulation on the lower part of the face is a frame difference technique used to detect pixel dispositions from frame to frame as described on [8]. The technique is straightforward. A gray-scale filter is applied to both the previous and the next frame and their absolute difference is calculated. Next, the subtraction result is thresholded in order to create a mask, and finally a median filter and a dilation effect is added to remove noise and amplify the result. The final frame difference mask can be seen in Figure 14.

At the final stage, all the aforementioned techniques are fused together in the aggregated weighted mask displayed on Figure 15(b). Afterwards, the aggregated mask gets thresholded (Figure 15(c)) to binary values (0, 255). Next a connected component algorithm is applied that allows blob detection (Figure 15(d)). It implements 8-point pixel connectivity [21], detects the different white blobs in the mask and calculates their centroids. The Manhattan distance between each centroid defines their affinity and whether the blobs should be connected. Finally the widest blob with the greater area is identified as the mouth and a rectangle is drawn around the general region of interest. The result of the detection can be seen on Figure 15(e).

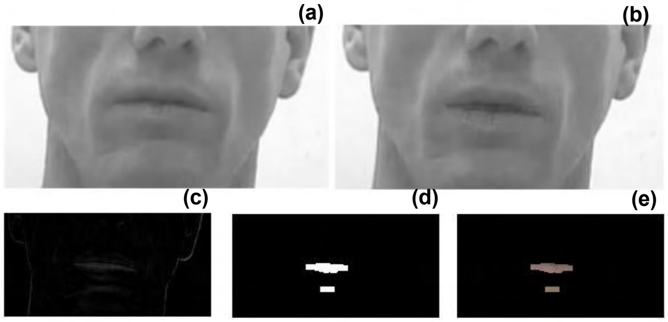


Fig. 14: *Frame difference: (a) current frame in gray scale; (b) old frame in gray scale; (c) frame subtraction; (d) frame mask after thresholding and dilation; (e) mask application.*

Having described the region of interest detection process, the subject has to shift to its tracking. Tracking the region of interest is a result of Meanshift Algorithm**,** a powerful and versatile non-parametric iterative algorithm that can be used in many applications like finding modes in data, clustering, color segmentation and image processing. Meanshift was introduced in 1975 by Fukunaga and Hostetler [22] as a clustering technique and has been extended to be applicable in other fields like Computer Vision, an extension that allowed its use in this speaker diarization system.

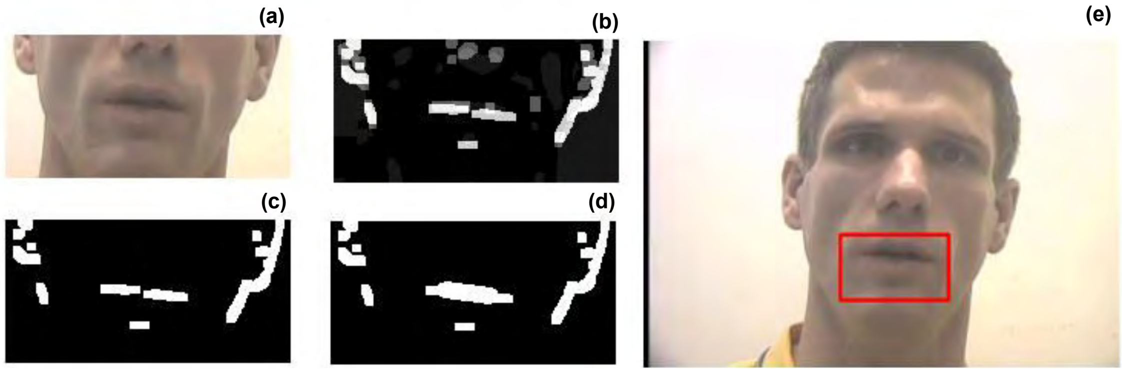


Fig. 15: *Mouth detection: (a) original half face frame; (b) aggregated mask; (c) frame mask after thresholding; (d) mask after blob connection; (e) final mouth ROI detection*

From the clustering point of view, meanshift builds upon the concept of kernel density estimation (KDE). KDE is a method to estimate the probability density function of a set of data. In order to achieve that meanshift uses a kernel on a set of points, as a weighted function and a window for the calculations. The sum of all the kernels constitutes the probability density function. A Gaussian kernel is most commonly used. Meanshift works by placing a circular window C on a set of data defined by a kernel; it exploits KDE by finding local maxima in the density function and applying a hill climbing algorithm iteratively on the window, thus shifting it towards the peak. The local maximum is considered as the area with the highest density. The hill climbing process is based on iteratively finding the mean (center of the circle) and centroid inside the window and shifting the center towards that centroid until convergence to the local peak. An iteration of the algorithm can be seen in Figure 16; $C1$ is the initial position of the circular window and $C1\_o$ its center. If the window's centroid -point $C1\_r$ - is different from the center, the window and its center moves to that centroid and so forth until the window engulfs the region with the highest density. Finally, meanshift is cluster-agnostic, in the sense that it does not require any assumptions about the number and size of clusters and depends only on the bandwidth parameter of the kernel which in the case of a Gaussian kernel is the standard deviation.

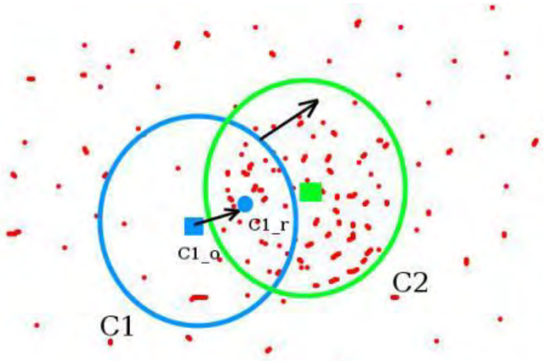


Fig. 16: *Meanshift algorithm [23]*

Apart from clustering, Meanshift can showcase its robustness in the realm of image processing and especially visual tracking objects in a video. The intuition behind the tracking is the same as in clustering. That is to move the region of interest (window) towards the peak of the probability density function. In a more detailed approach, Meanshift commences by selecting a ROI in an initial image in a visual sequence. In the next image the algorithm will try to construct a confidence map usually based on the color histogram of the initial image. That confidence map is in essence the probability density function of the color of the ROI. Afterwards, meanshift will try to move the ROI towards the area with the highest density; that is the area whose histogram matches the ROI histogram best.

In the implementation of the speaker diarization system, meanshift algorithm is used to track the subject's face ROI. The initial ROI detection is performed by the Viola and Jones Haar cascades method. The essential property that is attempted to get tracked is the subject's skin color. In order to achieve that, the ROI of the initial image is converted into the HSV color space which, as mentioned above, is capable of accommodating the skin color information in its hue channel. Afterwards, the histogram of the HSV ROI is extracted and normalized and a

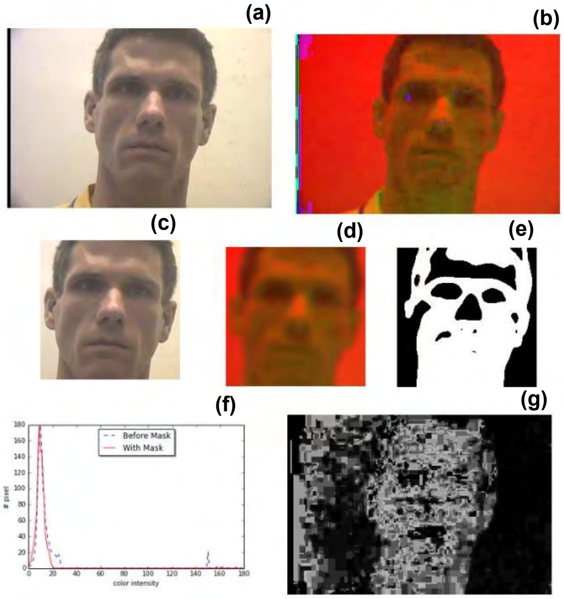


Fig. 17: *Meanshift tracking: (a) initial frame; (b) HSV color of frame; (c) face (ROI); (d) HSV color of ROI; (e) skin color mask; (f) color distribution before and after skin color mask; (g) the product of backprojection.*

skin mask is produced and applied on the HSV ROI. Subsequently, a process called histogram backprojection [24] is executed. Backprojection takes the HSV color space of the next frame and calculates the probability of whether a pixel's intensity value in the next frame belongs to the ROI histogram of the previous frame; thus creating a density map that represents the possibility of skin or skin-like features. Finally, the meanshift algorithm is applied in order to shift the ROI towards the area with the highest density of skin-like pixels. The process is displayed on Figure 17. The Figure 17(g) presents the product of backprojection, a skin or skin-like color density map on the next frame. The lighter colors in the image represent the highest color intensity which translates to greater the skin color density. When the subject moves, this movement is depicted on the backprojection map, and the meanshift algorithm will attempt to track that movement by shifting the ROI towards the new area with the greater density.

3.2.2 Deep Learning Method

Machine learning is a type of artificial intelligence (AI) that facilitates a computer's ability to learn and essentially teach itself to evolve as it becomes exposed to new and ever-changing data. Deep learning (DL) is one paradigm of performing machine learning, and the technology has become a hot topic due to the unparalleled results it has yielded in applications such as computer vision (object/face recognition), speech recognition, natural language understanding, and cyber threat detection. In addition, most of the top companies in the computer industry, including Google, Facebook, Baidu, and Microsoft have already developed commercial applications based on this technology, thus setting a trend for more to follow.

The method mentioned here exploits the great capabilities of DL in face recognition, and is based on convolutional neural networks (CCNs) an increasingly popular subclass of neural networks. The detection network presented in [28] was trained according to the widely held Oxford Visual Geometry Group (VGG) model. For more information on the implementation method and the VGG jargon, the reader is encouraged to study the original publication.

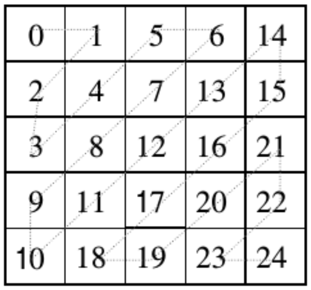
3.2.3 Feature Selection

The selection of the visual features is performed with the Discrete Cosine Transform (DCT), which as a mean of feature extraction is the most commonly used method in the examined bibliography [9, 25, 26, 27]. Apart from the realm of feature extraction for machine learning, DCT finds applications in a variety of technologies in the broader field of cognitive computing that encompasses artificial intelligence, signal and image processing. Amongst its numerous applications, the most popular are in lossy audio and image compression such as MP3 and JPEG. The use of cosine rather than sine functions is critical for compression, since it turns out that fewer cosine functions are needed to approximate a typical signal, whereas for differential equations the cosines express a particular choice of boundary conditions.

In an attempt to dive into the mechanics of DCT, it can be described as a Fourier-related transform similar to DFT and the Fourier series, with the difference that it only uses real numbers. The main reason for the widespread use of DCT as a feature extraction method is the high compaction of the energy of the input signal onto a few

DCT coefficients and the availability of fast implementations of the transform, similar to the Fast Fourier Transform (FFT).

The discrete cosine transform expresses a finite sequence of data in terms of a sum of cosine functions oscillating at different frequencies. There are 8 different types of DCT (DCT I -VIII), and by far the most commonly used in visual feature extraction is the 2D DCT-II. Assuming a UxV matrix of pixel intensities in an image the mathematic formula for 2D DCT is the following [9]:



0	1	5	6	14
2	4	7	13	15
3	8	12	16	21
9	11	17	20	22
10	18	19	23	24

Fig. 18: *Zigzag selection of DCT coefficients in a 5x5 image block [9].*

$$c_{m,n} = W_m \, W_n \sum_{u=0}^{U-1} \sum_{v=0}^{V-1} p_{u,v} \, cos\left(\frac{m\pi(2u+1)}{2U}\right) cos\left(\frac{n\pi(2v+1)}{2V}\right) \qquad (3.8)$$

$$0 \leq m \leq U-1, 0 \leq n \leq V-1$$

$$where \;\; W_n = \begin{cases} \sqrt{1/_V} & if \quad n = 0 \\ \sqrt{2/_V} & \quad\;\; otherwise \end{cases} \qquad (3.9)$$

$$and \;\; W_m = \begin{cases} \sqrt{1/_U} & if \quad n = 0 \\ \sqrt{2/_U} & \quad\;\; otherwise \end{cases}$$

In the later formula, $p_{u,v}$ refers to the intensity of the pixel in the u^{th} row and v^{th} column of the matrix P that represents the ROI of the subject's mouth. The DCT output is a 2-dimensional coefficient matrix with 2 important properties: Firstly, it is revealed that the most visually significant information is concentrated in the upper left corner of the matrix decreasing in a zigzag manner [25] (Figure 18). This fact means that the less significant information, residing in the lower right corner of the matrix, can be ignored, resulting in information compression, with the tradeoff of some loss in quality. This technique is used for audio and image compression in MP3 and JPEG. The process can be seen in Figure 19(a). Secondly, in [27] it is proposed that lateral image symmetry can be achieved by discarding the odd frequency components of the resulting matrix (Figure 19(b)). These properties brand DCT method of feature selecting in essence a dimensionality reduction algorithm.

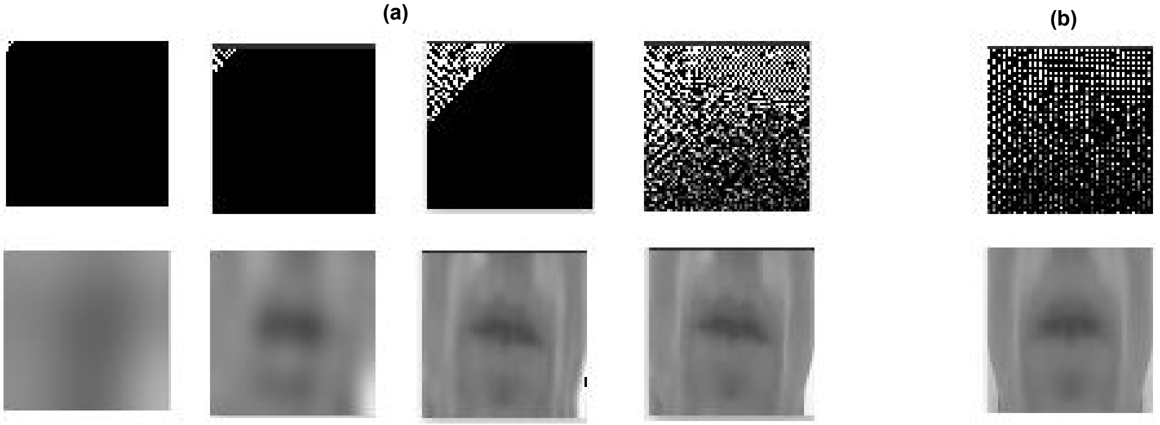


Fig. 19: *Zigzag scanning of the DCT matrix of a 64x64 image and the resulting visual information residing in the most significant coefficients. From left to right: (a) 10, 50, 500 coefficients and the whole DCT matrix (4096 coefficients); (b) Lateral symmetry example, the odd components are discarded resulting in a symmetric mouth.*

After the ROI's DCT according to [26], there are 3 different strategies of selecting the type of features:

- **Energy features:** the L features with the highest energy
- **Variance features:** the L features with the highest variance
- **Relative variance features:** the L features with the highest variance after normalization to their mean value.

For this particular diarization system, the energy features are selected. Initially the detected mouth is scaled to a 64x64 rectangle and is used as input to the DCT function of OpenCV. Finally the energies of the first 30 coefficients of the resulting matrix are calculated and selected as the visual feature vector in every video frame.

3.3 Visual Feature Upsampling

Having concluded the process of feature selection, the next stage in the speaker diarization system is the visual feature upsampling to match the acoustic sampling rate. This is achieved through a simple linear interpolation of the visual to the audio features. The acoustic features are extracted with a 10ms sample rate while the video's frame rate is 30 fps. For the interpolation, the visual features have to be fitted to acoustic sample rate at 100 fps.

CHAPTER 4: Canonical Correlation Analysis

4.1 Audio-visual Synchrony and Correlation

The next stage in the speaker diarization system is the actual diarization mechanism itself. This mechanism is the heart of the diarization system and is based on finding the correlation between the extracted features. In particular, the process of canonical correlation analysis (CCA) is used to determine the degree of correlation between the acoustic and visual features. Canonical correlation analysis determines a set of canonical coefficients, orthogonal linear combinations of the variables within each set that best explain the variability both within and between sets. CCA is a common approach to detect the correlation between two different signals [5, 29, 30, 31, 34]. As proposed in [5] and [30], CCA can be employed by a speaker diarization system in order to detect the audio-visual synchronization. The general idea behind the utilization of CCA is that, if measurements in two different sensory modalities are correlated, then they are likely to be generated by a single underlying common cause. For example, a high correlation can emerge between the movement of the pixels around the mouth ROI and the acoustic energy during a speaking segment caused by the speaker talking.

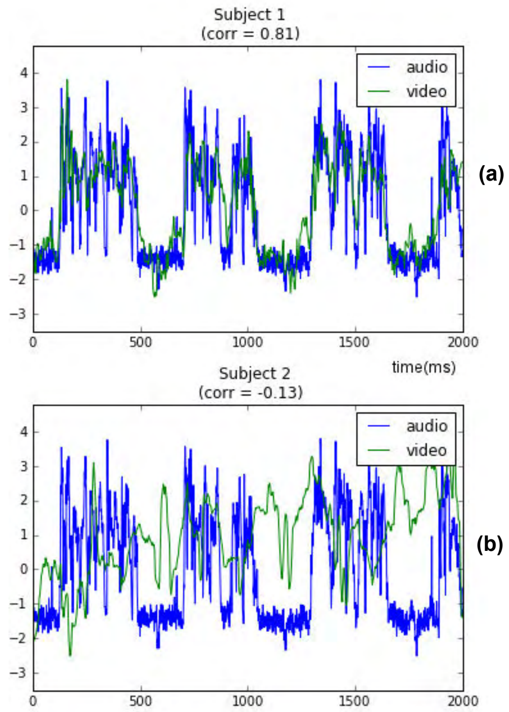


Fig. 20: *Example of audio-visual synchrony and correlation: (a) audio-visual correlation between source and audio; (b) uncorrelated audio and source*

Hershey et. al. [32] pioneered the use of audio-visual synchrony for speech detection and sound localization. Their work was inspired by the ventriloquism effect, the intuitive observation that psychophysical and physiological evidence suggest that audio-visual contingencies play an important role in the localization of sound sources. The ventriloquism effect suggests that there is important information about sound location encoded in the synchrony between the audio and video signals. In other words, sounds seem to emanate from visual stimuli that are synchronized with the sound. Ultimately synchrony is an effect of the causal relationship between visual and acoustic events.

An example of audio-visual synchrony using CCA can be seen in Figure 20. This test was conducted on the mutual constant motion (MCM) dataset, where the audio-visual features extracted from a 2 seconds segment were selected. Afterwards, The audio-visual correlation between the audio MFCC and the mouth ROI DCTs of the two subjects is examined by applying CCA. The speech segment belongs to 'Subject1' so its acoustic features (MFCCs) are expected to correlate better with the visual feature extraction (DCT) from ‘Subject1’. This expectation is realized in the figure, where ‘Subject1’, as the original acoustic signal source, correlates much better with the audio than ‘Subject2’. In this example the Pearson coefficient was used as a measure of correlation and it indicates that ‘Subject1’ scored 0.81, while ‘Subject2’ scored -0.13 in a scale from 1 to -1. This example is a testimony to how well audio correlates with visual stimuli.

Apart from canonical correlation analysis, other measures of detecting audio-visual synchrony are mutual information (MI) [5, 32, 34], multiple linear regression [25, 29] and co-inertia analysis (CoIA) [33, 34]. Irrelevant of the measure, the consensus is that speech is intrinsically bimodal in the sense both signals contain information the dynamics of the articulators

4.2 CCA Under the Hood

In statistics, canonical correlation analysis (CCA) is a way of making sense of cross-covariance matrices of two vectors. CCA offers a measurement of how much and in what direction two given multidimensional variables are correlated. The formula of CCA is provided in [31]. Given two column vectors $X = (x_{1\,,} \dots, x_n)^{\,T}$ and $Y = = (y_{1\,,} \dots, y_m)^{\,T}$ of random variables, their cross-covariance is defined as $C_{XY} = cov(X, Y)$, which is a nxm matrix whose entries are the covariances $cov(x_{i\,,}\, y_j\,)$.

CCA aims to calculate the projection vectors u_x and u_y, with dimensions n and m respectively, so that their linear combinations with of X and Y, $X' = u_x^T X$ and $Y' = u_y^T Y$ maximize the correlation ρ_{XY} as defined:

$$\rho_{XY} = \frac{u_x^T C_{XY} u_y}{\sqrt{u_x^T C_{XX} u_x}\,\sqrt{u_y^T C_{YY} u_y}} \qquad (4.1)$$

These X' and Y' linear combinations are called first pair of canonical variables. In order to find multiple pairs of canonical variables the process must be repeated with the constraint that they are to be uncorrelated with the first pair of canonical variables. The maximum number of canonical variables is $min\{m, n\}$. The correlation between each pair of canonical correlations creates the canonical component vector or correlation coefficients.

As a synchrony measure CCA involves computing and maximizing vectors u_x and u_y that solve Equation 4.1 and then computing time-windowed estimates of the correlation of the auditory and visual features projected on these vectors at each point in time. The final judgment to the quality of the correlation is passed according to threshold values of an overall measure based on the correlation coefficients.

4.3 CCA in the proposed Speaker Diarization System

In the proposed diarization system, CCA is used to evaluate the correlation between the audio and the visual features in each speaker's speech. After the audio and visual features are interpolated, the audio-visual sequence is parsed by a window, thus dividing the duration of the sequence in segments. As mentioned earlier (section 2.3), only the segments containing speech are taken into account. The silent segments are detected and discarded through the VAD system.

Classes	Meaning
Class 1	Subject 1 is speaking
Class 2	Subject 2 is speaking
Class 3	Overlapping Speech
Class 4	Silence

***Table 5:** Classification table.*

Assuming now that the audio and video are synchronized, the main intuition behind the implementation is that each segment consists of time series of acoustic and visual feature vectors, $X_a : (x_1^a, \dots, x_n^a)$ and $X_v : (x_1^v, \dots, x_n^v)$ with dimensions $n\,x\,39$ and $n\,x\,30$ respectively. CCA is employed to calculate the first 10 correlation coefficients of the audio and visual features between both subjects by maximizing the correlation between the linear combinations of the feature vectors and their projections.

$$X'^{(k)}_{a_i} = H^{(k)}_{a_i} X_a \quad and \quad X'^{(k)}_{v_i} = H^{(k)}_{v_i} X_v \qquad (4.2)$$

where $i = 1, \dots, 10$ indicates the computed component, $m = 1\ or\ 2$ is the current subject, $X'^{(k)}_{a_i}$ and $X'^{(k)}_{v_i}$ are the linear combinations or canonical variables and finally, H_{a_i} and H_{v_i} are the projection matrices with dimensions $n\,x\,39$ and $n\,x\,30$ respectively. The formula to maximize is:

$$\rho_i = max\ \left[\left(H^{(k)^T}_{a_i} X_a, H^{(k)^T}_{v_i} X_v \right)\right] \qquad (4.3)$$

The overall audio-visual correlation measure r_{av} is defined by the correlation, as described by the Pearson Correlation Coefficient (PCC) or Pearson product-moment correlation coefficient (PPMCC), between each pair of canonical variables for all subjects.

$$r^{(k)}_{av_i} = E\left(X'^{(k)}_{a_i} X'^{(k)}_{v_i}\right) \qquad (4.4)$$

This operation produces two sets of CCA coefficients, one for each subject. The overall correlation measure r_{av} between X'_a and X'_v varies depending the dataset, the diarization window and the number of CCA coefficients that are taken into account. The different variations used as diarization thresholds for each subject are:

$$r^{(k)}_{av} = \frac{1}{10} \sum_{m=1}^{10} [r^{(k)}_{av_i}]^2 \quad (4.5) \quad and \quad r^{(k)}_{av} = \frac{1}{10} \sum_{m=1}^{10} r^{(k)}_{av_i} \quad (4.6)$$

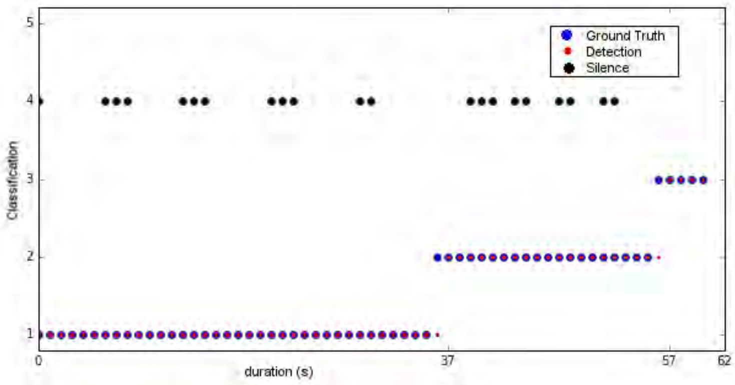


Fig. 21: *Illustration of speaker diarization on the SSM dataset.*

Finally an educated guess of who is currently speaking is attempted based on the overall correlation measure. Much like in the synchronization example presented in Figure 20**,** the two values are compared and the greater value defines the potential speaker, unless their absolute difference is less than a predefined and tested threshold. In that case it is decided that the segment belongs in the overlapping speech part. An illustration of the diarization process, applied to the SSM dataset, is available on Figure 21**.** The video in this example is 62 seconds long and is divided in segments of 1 second each (1000ms diarization window). Every segment is classified into ‘Subject 1’, ‘Subject 2’, both or silence, which correspond to classes 1, 2, 3, and 4 respectively. In this particular example a segment is misclassified as ‘Subject 1’, when it actually belongs to ‘Subject 2’, and a segment is attributed to ‘Subject 2’, when it belongs to the overlapping speech part.

For the evaluation of the diarization system, three window types of various lengths are tested namely 100ms, 500ms, and 1000ms. For every window, each segment is classified to either or both of the subjects or it is labeled as silent segment. The CCA process and the Pearson correlation coefficient are calculated with the popular Python modules scipy.sklearn, and numpy respectively.

CHAPTER 5: Speaker Diarization Evaluation

The performance of the speaker diarization system was evaluated using two popular metrics: the F1-score (also F-score or F-measure) and the diarization error rate (DER). It is important to state that while the classification of the segments was concluded in time intervals, according to the diarization window size (100ms, 500ms, 1000ms), the evaluation was executed into the frame level. This was done in order to produce an overall and comparable evaluation result for all windows. Finally, for reasons of consistency with the established bibliography the phrases *ground truth* and *reference* or *ground truth speaker* and *reference speaker* are going to be used interchangeably.

F-score is the main metric for evaluating binary classification; it provides a quantitative answer to how accurate a classification system is by comparing its decisions against the “gold standard” of the system ground truth. In this particular system, F-score evaluates whether each frame was classified correctly or not.

The diarization error rate is the main metric applied in speaker diarization experiments as described and used by NIST in the Rich Text (RT) evaluations [2]. The DER is measured as the fraction of time that is not attributed correctly to a speaker or to non-speech. In order to measure performance, an optimum one-to-one mapping of reference speaker IDs to system output speaker IDs needs to be computed. The measure of optimality will be the aggregation, over all reference speakers, of time that is jointly attributed to both the reference speaker and the (corresponding) system output speaker to which that reference speaker is mapped. This will always be computed over all speech, including regions of overlap but not regions of silence.

5.1 F-score

The calculation of F-score depends on the estimation of two variables, Precision and recall [20]. Precision (also called positive predictive value) is the fraction of retrieved instances that are relevant, while recall (also known as sensitivity) is the fraction of relevant instances that are retrieved.

For classification tasks, the terms *true positives*, *true negatives*, *false positives*, and *false negatives* compare the results of the classifier under test with the system reference. The terms *positive* and *negative* refer to the classifier prediction, and the terms *true* and *false* refer to whether that prediction corresponds to the system reference. With all this in mind, the mathematic formulas for calculating precision and recall are the following:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \qquad (5.1)$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \qquad (5.2)$$

As a measure, high precision means that the system classifying algorithm returns substantially more relevant results than irrelevant ones. High recall on the other hand means that an algorithm returned most of the relevant results. Since both measures are important, usually we measure our systems with F-score which is the harmonic mean of recall and precision. F-score has a parameter that sets the tradeoff between recall and precision. The standard F-measure is F1, which gives equal importance to recall and precision:

$$F1 = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision} \qquad (5.3)$$

The general formula for F-score is the following:

$$F_\beta = (1+\beta^2) \cdot \frac{Recall \cdot Precision}{(\beta^2 \cdot Precision) + Recall} \qquad (5.4)$$

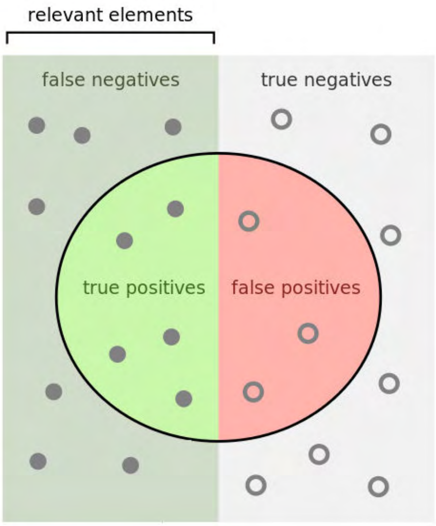


Fig. 22: *Precision and recall [20].*

In the speaker diarization system presented here, each video frame is classified by the acoustic and visual information it contains. This means that when a frame is labeled as *true positive,* the correct number of speakers and their ID were identified so the classification is *correct*. On the other hand, when a frame is labeled as *false positive,* it means that either the number of speakers or their identity is wrong, so it is classified as *false alarm.* In addition, when a frame is labeled as *false negative,* it means that no speaker was identified on that frame, so it is classified as *miss*. Finally, the prediction of *true negative* frames is of no use to the system, because those frames would be describing the absence of acoustic events in the specific frame. This information has no application, since only frames containing speech are processed.

5.2 Diarization Error Rate

As proposed in the NIST RT Diarization evaluations [2], to measure the performance of the proposed systems, the diarization error rate (DER) will be computed as the fraction of speaker time that is not correctly attributed to that specific speaker. This score will be computed over the entire audio-visual sequence to be processed in a frame by frame basis, including regions where more than one speaker is present (overlapping speech regions).

In the original NIST publication, the DER evaluation is calculated by dividing the file in segments, then mapping each reference speaker to a segment, and finally comparing this mapping to the system prediction. In the speaker diarization system presented here, each dataset file is divided using three (3) different segmentation windows of 100ms, 500ms, and 1000ms. The diarization process is enforced on each segment individually through CCA (section 4.3). After diarization, for reasons of consistency, in order for every experiment to have the same evaluation base, the calculation of DER is applied to the video file frame by frame. Given the dataset to evaluate, in each video file, both the reference and the hypothesis need to be established frame-wise. The diarization error time for each frame n is defined as:

$$E(n) = T(n)[max(N_{ref}(n), N_{sys}(n)) - N_{correct}(n)] \qquad (5.5)$$

where $T(n)$ is the amount of time between each frame (frame rate), $N_{ref}(n)$ is the number of reference speakers that are present in frame n, $N_{sys}(n)$ is the number of system (detected) speakers that are present in frame n, and $N_{correct}(n)$ is the number of reference speakers in frame n correctly assigned by the diarization system.

The mathematic formula for calculating DER in a given dataset Ω is:

$$DER = \frac{\sum_{n \epsilon \Omega} E(n)}{\sum_{n \epsilon \Omega}(T(n) \cdot N_{ref}(n))} \qquad (5.6)$$

The diarization error time includes the time that is assigned to the wrong speaker, missed speech time, and false alarm speech time:

- **Speaker Error Time:** the speaker error time is the amount of time that has been assigned to an incorrect speaker. This error can occur in frames where the number of detected speakers is greater than the number of reference speakers, but also in frames where the number of detected speakers is lower than the number of reference speakers, whenever the number of detected speakers and the number of reference speakers are greater than zero.
- **Missed Speech Time:** The missed speech time refers to the amount of time that speech is present but not labeled by the diarization system in frames where the number of detected speakers is lower than the number of reference speakers.
- **False Alarm Time:** The false alarm time is the amount of time that a speaker has been labeled by the diarization system but is not present in frames where the number of detected speakers is greater than the number of reference speakers.

CHAPTER 6: Evaluation Results

After having discussed the acoustic and the two visual feature extraction systems, the diarization process and the evaluation metrics, this chapter is dedicated to the presentation and illustration of the experiments conducted on the three manually developed datasets. The structure of the presentation involves displaying and comparing the performance of both *image processing* and *deep learning,* as visual feature extraction methods, and how they cope with the different diarization windows, across each dataset.

The experiments were conducted across a total of 1806 videos, with the same threshold values for both feature extraction methods. The results are the average values of precision, recall, F-score and DER for the 602 videos of each dataset.

6.1 SSM Dataset

Image Processing

Diarization Window (ms)	Precision	Recall	F-score	DER (%)
100	0.97	0.99	0.98	5.48
500	0.94	1.0	0.97	8.13
1000	0.97	1.0	0.98	3.46

Table 6: *Image processing SSM results.*

Deep Learning

Diarization Window (ms)	Precision	Recall	F-score	DER (%)
100	0.96	0.99	0.98	5.86
500	0.94	1.0	0.96	8.88
1000	0.97	1.0	0.98	4.38

Table 7: *Deep learning SSM results.*

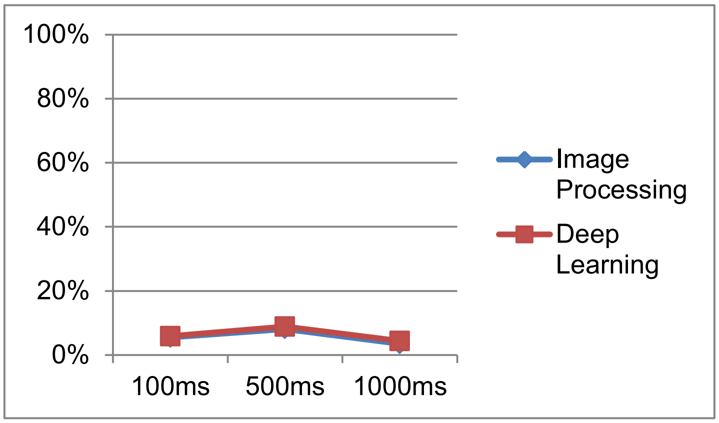


Fig. 23: *DER comparison of image processing vs deep learning for SSM dataset*

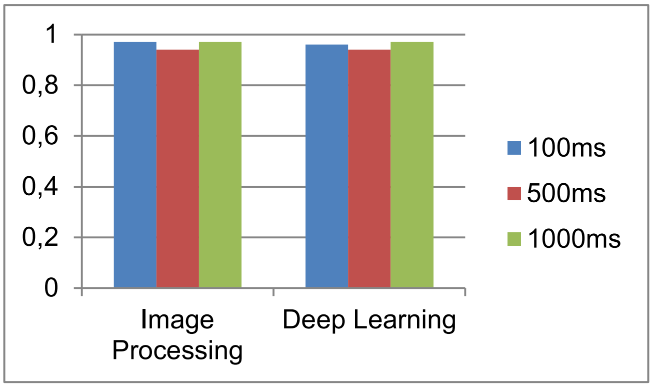


Fig. 24: *Precision comparison of image processing vs deep learning for SSM dataset*

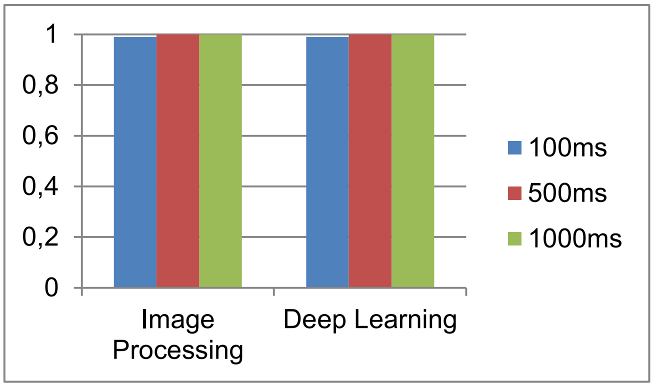


Fig. 25: *Recall comparison of image processing vs deep learning for SSM dataset*

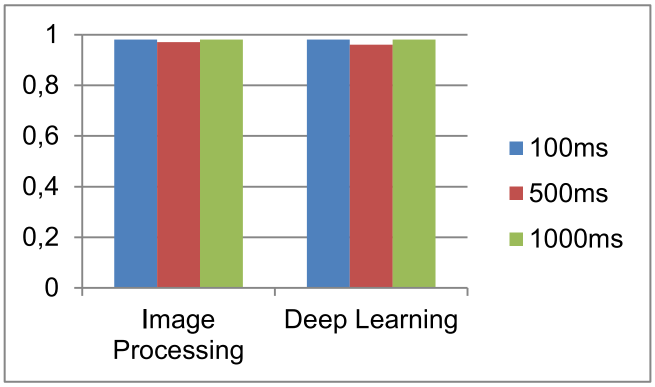


Fig. 26: *F-score comparison of image processing vs deep learning for SSM dataset*

As expected, the diarization system produces good results for the SSM dataset. The nature of the dataset assists the detection processes since only the talking subject is allowed to be in motion, while the silent subject remains motionless. This effect produces high correlation between the moving subject and the acoustic stimulus, which leads to its identification as the speaker. In terms of the feature extraction the results are comparable and no method has significant advantage over the other. Finally, as far as the diarization process is concerned, the 500ms window produces the worst DER for both methods, while the 1000ms window performs the best.

6.2 MCM Dataset

Image Processing

Diarization Window (ms)	Precision	Recall	F-score	DER (%)
100	0.59	1.0	0.74	52.26
500	0.57	0.99	0.73	54.07
1000	0.6	1.0	0.74	50.68

Table 8: *Image processing MCM results*

Deep Learning

Diarization Window (ms)	Precision	Recall	F-score	DER (%)
100	0.59	1.0	0.74	51.79
500	0.58	0.99	0.73	53.22
1000	0.59	1.0	0.74	51.3

Table 9: *Deep learning MCM results*

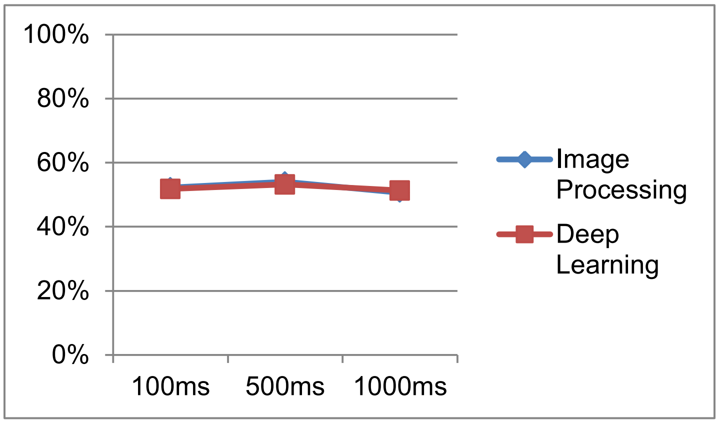


Fig. 27: *DER comparison of image processing vs deep learning for MCM dataset*

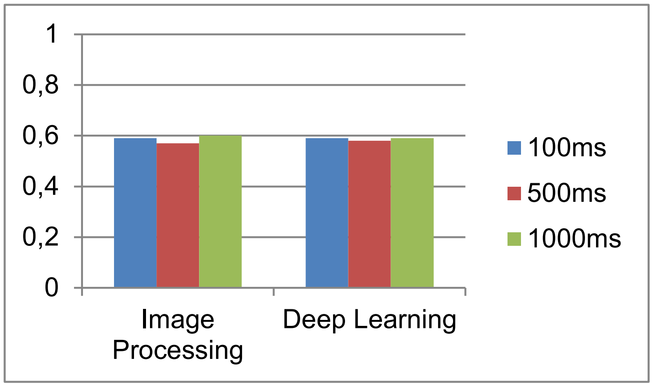


Fig. 28: *Precision comparison of image processing vs deep learning for MCM dataset*

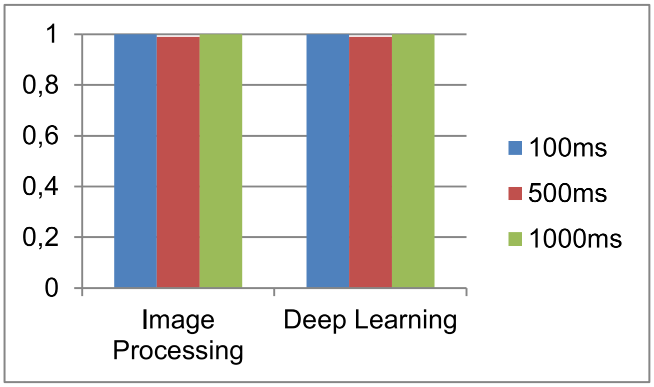


Fig. 29: *Recall comparison of image processing vs deep learning for MCM dataset*

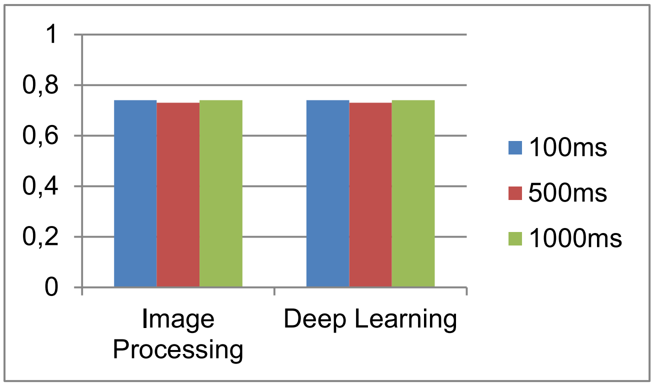


Fig. 30: *F-score comparison of image processing vs deep learning for MCM dataset*

As far as the MCM dataset is concerned, the diarization system performance is moderate in comparison to its performance for the SSM dataset. The nature of the dataset hinders the detection processes, since both subjects are in motion. This effect produces correlation between both subjects and the acoustic stimulus, so the system speaker identification thresholds need to be adjusted accordingly. In terms of the feature extraction, the results are again comparable, and no method has significant advantage over the other. With regards to the diarization process, the 500ms window produces again slightly worse DER for both methods than the other windows, while the 1000ms window performs the best.

Finally, as for the F-score, it is important to notice that while the precision of the diarization is not particularly inspiring, the system's recall remains good. This means that although the system produces a significant number of false alarms (misclassification-false positives), it succeeds in detecting almost all the segments containing speech, which results in low false negatives, thus high recall.

6.3 MCMS Dataset

Image Processing

Diarization Window (ms)	Precision	Recall	F-score	DER (%)
100	0.64	0.97	0.77	47.44
500	0.73	1.0	0.84	35.31
1000	0.58	1.0	0.73	52.58

Table 10: *Image processing MCMS results*

Deep Learning

Diarization Window (ms)	Precision	Recall	F-score	DER (%)
100	0.64	0.97	0.77	47.38
500	0.76	1.0	0.86	31.23
1000	0.58	1.0	0.73	52.52

Table 11: *Deep learning MCMS results*

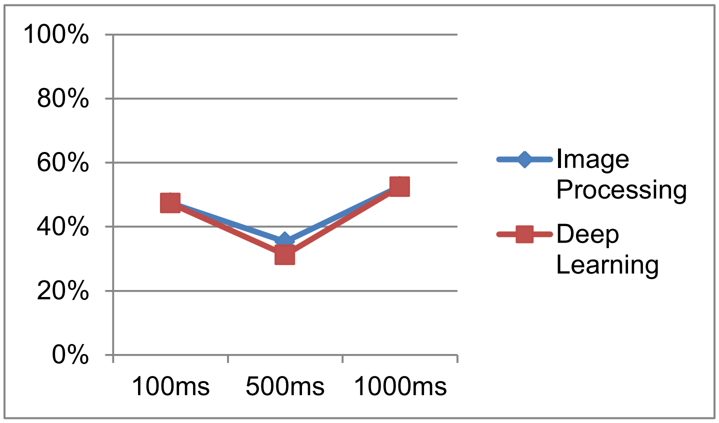


Fig. 31: *DER comparison of image processing vs deep learning for MCMS dataset*

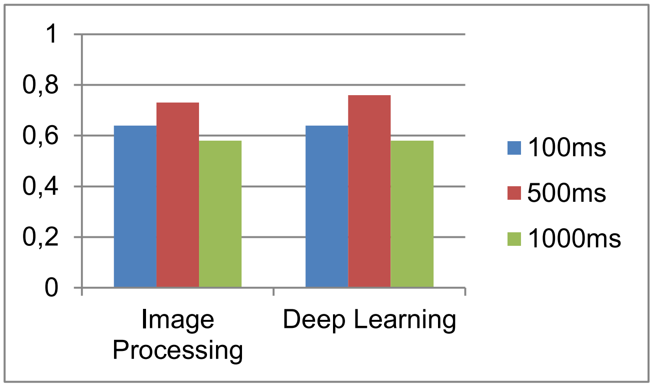


Fig. 32: *Precision comparison of image processing vs deep learning for MCMS dataset*

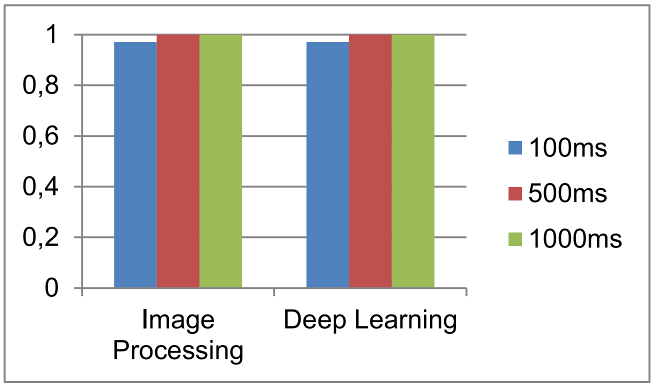


Fig. 33: *Recall comparison of image processing vs deep learning for MCMS dataset*

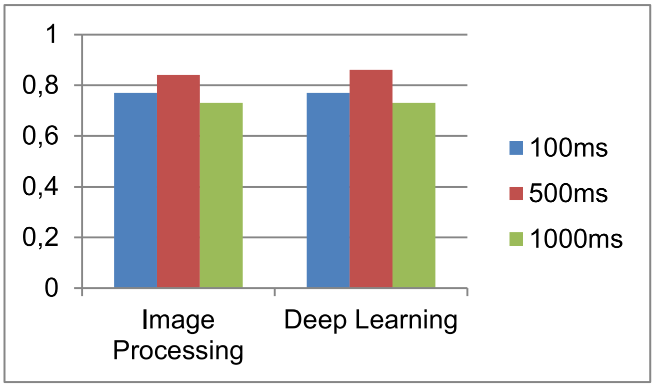


Fig. 34: *F-score comparison of image processing vs deep learning for MCMS dataset*

As mentioned in section 2.1 the MCMS dataset is the most intriguing one, since it was designed as a natural conversation simulation between the two subjects. In these experiments although the diarization system failed to reach a performance similar to the one against SSM dataset, it managed to surpass the DER values the MCM dataset. This result was expected, since the SSM dataset was developed as *the best case* and the MCM as the *worst case scenario.*

In terms of DER, the diarization system performs slightly worse than MCM for the 1000ms window, but achieved better results for the 100ms window. On the other hand its function on the 500ms window has dramatically improved by almost 20% for both the visual extraction methods.

As far as the extraction methods, it can be observed that both approaches are on par, with a slight advantage for deep learning. The two methods perform better with the 500ms window while their performance decreases for the 100ms and the 1000ms window.

Finally in terms of the F-score, the results are similar to the DER. Both deep learning and image processing have increased performance in this dataset over the MCM especially for the 500ms window. In addition, as far as the precision and recall of the system, the same pattern as in the MCM dataset is observed, where the precision of the diarization is not particularly satisfactory, but the system recall remains excellent.

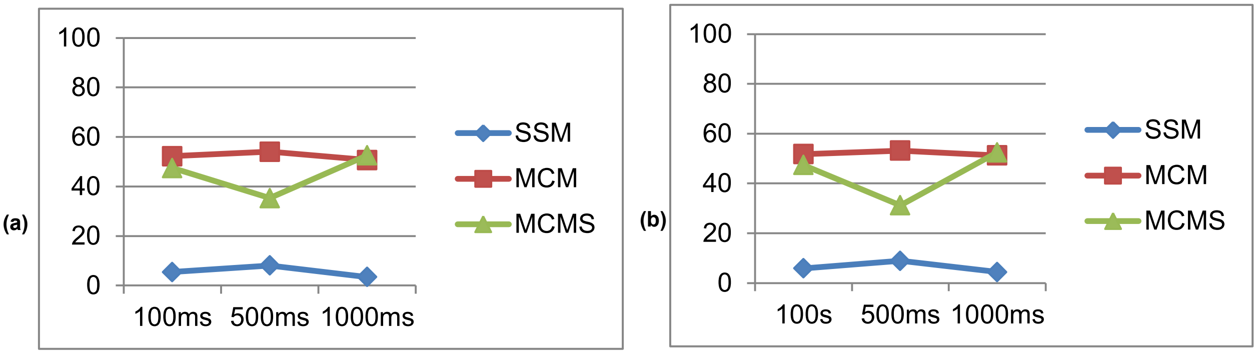


Fig. 35: *DER variation across different diarization windows: (a) image processing; (b) deep learning*

The aggregate results and performance of the diarization process across all datasets and for the different diarization windows can be seen in Figure 35.

CHAPTER 7: Conclusions

This Thesis has proposed a speaker diarization system based on acoustic and visual synchrony between two speakers, using canonical correlation analysis, and evaluated its robustness against a manually developed database of varying difficulty. The system was based on MFCCs for the acoustic and DCT for the visual feature extraction. The visual features were acquired using two different methods, one based on deep learning (DP) and one on image processing (IP). The performance of both methods was comparable producing highly accurate diarization for the less complicated dataset (SSM), achieving a DER as low as 3.46% for the IP and 4.38 for the DP method. In addition, the system produced satisfactory results -35.31% for the IP and 31.23% for the DP method - in the more challenging dataset (MCMS). Lastly, both methods performed subpar in the more challenging dataset (MCM) that contained constant motion from both subjects, with a 50.68% DER for the IP and 51.3 for the DL method.

Furthermore, the CCA mechanism used for detecting the audio-visual synchrony confirmed the existence of a strong correlation between acoustic and visual representations of speech that can be exploited for resolving the diarization task. Additionally the results, illustrated in Figure 35, showcase an increased performance for smaller diarization windows, up to 500ms rather than a larger one in the scale of 1000ms.

In conclusion, the experiment results suggest that both visual feature extraction methods are up to par and capable of providing similar levels of visual information when dealing with the diarization task. Moreover, CCA, although performed in a manually developed database can be a useful tool in detecting audio-visual synchrony and correlation.

REFERENCES

[1] En.wikipedia.org. (2017). Precision and recall. [online] Available at: https://en.wikipedia.org/wiki/Precision_and_recall [Accessed 30 Jan. 2017].

[2] The 2009 (RT-09) Rich Transcription Meeting Recognition Evaluation Plan. Online: http://www.itl.nist.gov/iad/mig//tests/rt/2006-spring/docs/rt06s-meeting-eval-plan-V2.pdf, accessed on December 21, 2016.

[3] H. Hung and G. Friedland. “Towards audio-visual on-line diarization of participants in group meetings”. Workshop on Multi-Camera and Multi-modal Sensor Fusion Algorithms and Applications - M2SFA2, Marseille, France. 2008.

[4] A. Noulas and B. Krose. On-line multi-modal speaker diarization. In: Proceedings of the 9th International Conference on Multimodal Interfaces. p.350-357, 2007

[5] G. Garau, A. Dielmann, and H. Bourlard. “Audio–visual synchronisation for speaker diarisation.” In: Proceedings of INTERSPEECH 2010

[6] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features", IEEE Conference on Computer Vision and Pattern Recognition, 2001

[7] G.R. Bradski. "Real time face and object tracking as a component of a perceptual user interface". In Proceedings of 4th IEEE Workshop, p.214,219, 1998

[8] N. Singla. “Motion detection based on frame difference method.” International Journal of Information & Computation Technology., 4(15), p.1559-1565, 2014

[9] B. Schwerin and K. Paliwal. “Local-DCT features for facial recognition.” In: 2nd International Conference on Signal Processing and Communication Systems (2008).

[10] P. Scanlon, G. Potamianos, V. Libal, and S. M. Chu. “Mutual information based visual feature selection for lipreading,” in Proc. ICSLP, p. 4–8. 2004

[11] X. Huang, A. Acero, and H. Hon. “Spoken language processing: a guide to theory, algorithm, and system development.” Prentice Hall, 2001.

[12] Σ. Χατζησαββίδης and Α. Χατζησαββίδου. “ΓΡΑΜΜΑΤΙΚΗ ΝΕΑΣ ΕΛΛΗΝΙΚΗΣ ΓΛΩΣΣΑΣ Α΄, Β΄, Γ΄ ΓΥΜΝΑΣΙΟΥ.” 1st ed. Οργανισμός Εκδόσεως Διδακτικών Βιβλίων (Ο.Ε.Δ.Β.), p.19., 2011.

[13] G. Vyas and B. Kumari. “Speaker Recognition System based on MFCC and DCT”, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-2, Issue-5, 2013.

[14] 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 55(1), p.119–139, 1997

[15] B. Vidakovic and P. Mueller. “Wavelets for kids a tutorial introduction. 1st ed.” Duke University, p.2-4, 1991

[16] En.wikipedia.org. (2017). Viola–Jones object detection framework. [online] Available at: https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework [Accessed 7 Feb. 2017].

[17] N. Sarafianos, T. Giannakopoulos, and S. Petridis. “Audio-visual speaker diarization using fisher linear semi-discriminant analysis.” Multimedia Tools and Applications, 75(1), p.115-130, 2014

[18] Docs.opencv.org. (2017). OpenCV documentation index. [online] Available at: http://docs.opencv.org/ [Accessed 7 Feb. 2017].

[19] G. P. Surampalli, J. Dayanand, and M. Dhananjay.” An Analysis of Skin Pixel Detection using Different Skin Color Extraction Techniques.” International Journal of Computer Applications, 54(17), p.1-5, 2012

[20] En.wikipedia.org. (2017). Precision and recall. [online] Available at: https://en.wikipedia.org/wiki/Precision_and_recall [Accessed 26 Feb. 2017].

[21] En.wikipedia.org. (2017). Pixel connectivity. [online] Available at: https://en.wikipedia.org/wiki/Pixel_connectivity [Accessed 8 Feb. 2017].

[22] K. Fukunaga and L. Hostetler. "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition", IEEE Transactions on Information Theory vol 21 , p.32-40 ,1975

[23] Docs.opencv.org. (2017). OpenCV: Meanshift and Camshift. [online] Available at: http://docs.opencv.org/3.1.0/db/df8/tutorial_py_meanshift.html [Accessed 12 Feb. 2017].

[24] M. J. Swain and D. H. Ballard. "Indexing via color histograms". Third international conference on computer vision,1990.

[25] I. Almajai and B. P. Milner. “Maximising audio-visual speech correlation.” In: Auditory-Visual Speech Processing, 2007

[26] M. Heckmann, K. Kroschel, and C. Savariaux. “DCT-based video features for audio-visual speech recognition” Conference: 7th International Conference on Spoken Language Processing, INTERSPEECH 2002

[27] G. Potamianos and P. Scanlon. "Exploiting lower face symmetry in appearance-based automatic speech reading", Int. Conf. Audio-visual Speech Processing, 2005

[28] A. Koumparoulis. “Deep learning for audio-visual speech recognition”, Diploma Thesis, p.7-12, 2017

[29] I. Almajai, B. P. Milner, and J. Darch. “Analysis of Correlation between Audio and Visual Speech Features for Clean Audio Feature Prediction in Noise”. In: Interspeech 2006 - ICSLP Ninth International Conference on Spoken Language Processing, 2006

[30] P. Ruvolo and J. Movellan. “An alternative to low-level-synchrony-based methods for speech detection.” In: Proceedings of the 23rd International Conference on Neural Information Processing Systems, p.2029-2037, 2010

[31] L. Xie, Y. Xu, L. Zheng, Q. Huang, and B. Li. “Speech Pattern Discovery using Audio-Visual Fusion and Canonical Correlation Analysis”. Interspeech, Portland, Oregon, USA, September 9-13, 2012.

[32] J. Hershey and J. Movellan. “Audio vision: Using audio-visual synchrony to locate sounds.” In: NIPS (1999)

[33] E. A. Rua, H. Bredin, G. G. Mateo, G. Chollet, and D. G. Jiménez. “Audio-visual speech asynchrony detection using co-inertia analysis and coupled hidden markov models”. In: Pattern Analysis and Applications, p.271-284, 2008

[34] H. Bredin and G. Chollet. “Audio-visual Speech Synchrony Measure: Application to Biometrics.” In: EURASIP Journal on Advances in Signal Processing, p.1-11, 2007