Imperial College London
Department of Computing

Discrete & Continuous Audio-Visual Recognition of Spontaneous Emotions

by
Michael A. Nicolaou

Submitted in partial fulfilment of the requirements for the MSc Degree in Advanced Computing of Imperial College London

September 2009

Abstract

Our work follows the recent shift in emotion recognition, from analysing posed expressions within controlled lab environments to the recognition of spontaneous emotion expressions obtained in real-world settings.

Firstly, we provide a novel segmentation method for extracting audiovisual segments from databases annotated in a dimensional emotion space, overcoming synchronisation and other issues. We perform discrete emotion recognition on the extracted segments, classifying them into positive or negative emotional states with Coupled Hidden Markov Models (CHMM). We show how the noise reduction and background speech removal from the audio cues can improve the classification accuracy by 10%. We demonstrate the first application of Support Vector Machines for classification in the likelihood space in emotion recognition, improving our initial accuracy of 79.1%, attained by fusing the facial expressions, audio and shoulder cues to 88.2%, accompanied by an analogous increase in all the combinations of cues.

For continuous emotion recognition, we present the first approach in the literature which fuses facial expressions, shoulder and audio cues. We experiment with state-of-the-art learning techniques such as the Long Short-Term Memory (LSTM) neural networks and Support Vector Machines for Regression (SVR), while we present the first applications of feature and decision-level fusion with LSTMs.

Referring to the correlation coefficient of the prediction with the ground truth, we show that LSTMs overperform SVRs by 7.3% on average. We examine the significance of the duration of sequences for LSTMs with respect to the performance of audio cues, while we show that the shoulder cues do not provide significant aid towards continuous emotion recognition. We demonstrate how the audio cues slightly overperform the HMM accuracy for discrete emotion recognition when experimenting with continuous to discrete classification.

We experiment with dimensionality reduction, while we present a novel approach for learning correlations and patterns within the valence/arousal values. This method provides an increase of the correlation up to 6.6%. We perform decision-level fusion with LSTMs, and in combination with our previous approach for learning valence/arousal correlations we present a further improvement to the correlation performance by a maximum of 32% with respect to feature-level fusion with SVRs. Comparing with feature level fusion with LSTMs the improvement is 17% for arousal and 19% for valence.

Our results approximate the average human coder correlation with respect to the valence ground truth and the maximum coder correlation with respect to arousal by 3%, while comparing to the averaged human coder correlation for arousal, our system performs better by 8%.

Acknowledgements

This project would not have been possible without the invaluable discussions and the advice that I was given by both my supervisors, Dr. Maja Pantic and Dr. Hatice Gunes; I thank them both. Furthermore, I thank the entire HCI^2 group and especially Stavros Petridis for his help with the extraction of audio features. Finally, I thank my dear, close friends and family for well, everything.

Dedicated to the tradeoff

"... among stars to wander forever,
weightless without a headline,
without thought,
without newspapers to read,
by the light of the Galaxies"

- Allen Ginsberg, from White Shroud

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The field of automatic emotion recognition is a highly active, multi-diverse research area, which combines knowledge from fields such as behavioural sciences, cognitive sciences and psychology , as well as fundamental knowledge and state-of-the art findings in the field of machine learning, computer vision and human-machine interaction [86, 174, 84].

While initially, research on human emotions focused on the recognition of posed emotional expressions captured in controlled, laboratory settings [86], research attention has recently shifted towards the analysis of spontaneous emotions in real life scenarios. This shift poses a set of new, interesting and deeply challenging issues in the field, ranging from difficulties in accessing and maintaining spontaneous emotion databases, to the relaxation of constraints and assumptions imposed in laboratory settings.

Furthermore, most of the research in the field [86, 203] has been focusing on recognition in terms of discrete emotional states, such as the basic emotion categories (happiness, sadness, surprise, fear, anger and disgust)[59, 80, 54]. Many researchers though, claim that the affective states encountered and experienced in everyday life do not typically fall into the basic categorisation proposed; instead, these emotional states are more subtle and complex [12], in way that the basic emotional categories can not fully capture. This observation has led researchers in affect recognition to explore the modelling and interpretation of such complex and subtle everyday life emotional states in terms of latent dimensions. An example of such an approach, is the valence/arousal emotion space [152], a 2D space representing how positive or negative an emotional state is with the valence dimension, and how active or passive with the arousal. It is noted, that besides the increase in dimensionality from the basic emotions approach (which can be considered 1D), the valence/arousal description approach presents us with a continuous space, with real numbers instead of labels for each of the dimension. This significant addition can allow the realisation of automatic recognition systems which perform continuous recognition on a sequence of cues, dealing with actual continuous emotions.

The work we present in this project lies at the centre of the state-of-the-art research in affect recognition. Our goal is firstly to extract audiovisual segments portraying spontaneous emotional expressions which can be considered to be either negative or positive, and subsequently to perform various experiments in both discrete and continuous audiovisual emotion recognition. Due to the nature of the data, we do not only have to deal with basic emotional states, but also with more subtle everyday life expressions. These affective states manifest in the SAL database, which

carries the audiovisual data we work on. The database presents a scenario where a virtual avatar interacts with a human subject, and during the conversation various emotional states are exhibited by the individual, including states such as boredom, tiredness and depression.

In order to extract such segments, the actual material of the database needs to be in someway assessed in terms of the desired characteristics. For example, some approaches rely on manual segmentation [29]. The SAL database though has been annotated by a set of coders (i.e. human experts) who watch and listen to the audiovisual material and provide a continuous annotation in the valence-arousal space. In order to extract the segments, we present an automatic method which combines the available annotations and attempts to extract the relevant data, while also attempting to overcome several issues that are presented. For example, the most obvious issues relate to the fact that more than one person is providing the annotations. Subsequently, not all the persons agree on the annotation at a certain time, while there are also issues of synchronisation across the coders.

While extracting the relevant audiovisual data, the segmentation process also presents us with a set of valence/arousal annotations, acquired during the analysis of the individual coder annotations. The produced set of annotations is what we consider in our continuous recognition experiments to represent the ground truth.

Having extracted the audiovisual segments, features that will be used for learning need to be extracted. We extract features from the audio signal that accompanies the data and carries essentially the speech of the individual, from the facial expressions by tracking 20 fiducial points on the face, while we also track 5 points relating to the shoulder movement of the individual, with a goal of capturing the upper body movement.

Then, we perform a set of experiments on discrete emotion recognition, classifying our data into the positive and negative emotional classes. In order to perform such a classification, we use Hidden Markov Models, which are generative probabilistic models. Two HMM models are trained, one for each class. When presented with a sequence (in our case, the sequence of features extracted from the audiovisual segment) they output a likelihood, describing how probable it is that the model produced this sequence. Therefore, we obtain two likelihoods, one from the negative and one from the positive class HMMs. This is where we experiment with increasing the accuracy of the HMMs, by using another popular classification technique, Support Vector Machines (SVMs) in order to classify in the likelihood space. We show that this method outperforms other methods that have been used for this scenario in emotion recognition. It is noted that when using multiple cues, the fusion in HMMs is done model-level, with Coupled Hidden Markov Models (CHMMs).

Moving to continuous recognition, the scenario changes: We now have a set of audiovisual data and features (as before) but for each frame of each sequence (or video), we have a 2D annotation in the valence/arousal space, the ground truth. Our goal now, is to train a set of learning techniques which would estimate the valence/arousal ground truth, thus performing actual continuous emotion recognition. For such a task, we take advantage of a state-of-the-art sequence learning technique, the Long Short-Term Memory recurrent neural networks (LSTM). LSTMs have been known to overcome issues that appear in traditional recurrent neural networks and have the ability to learn from long range temporal dependencies in the input data. Furthermore, we experiment with another technique, Support Vector Regression (SVR), a popular algorithm which is though static - it does not perform sequence learning but just considers the training data as a

set of frames, and not a set of videos (sequences of frames). We will show that in general, LSTMs outperform SVRs for the task at hand.

Finally, we will discuss the issue of evaluating the estimated ground truth from such a technique, while we will also explore methods which can increase the accuracy and performance of such systems, by learning patterns in the valence/arousal annotations and also correlations between the two dimensions. We experiment with both feature-level and decision-level fusion of cues: Feature-level refers to compiling multiple cues (in our case, the shoulder, audio and facial expression cues) in a feature vector which will be subsequently used as input for the learning technique, while with decision-level, a classifier is trained for each set of cues and the results of each classifier are combined.

1.1 Key Contributions

Having summarised the project itself, we will now present our key contributions:

- We present a novel method for the segmentation of audiovisual data which has been annotated in the valence/arousal dimensional space. These annotations are typically from more than one coders and achieving agreement across the coders is one of the great challenges in the field [84]. Our method attempts to overcome issues of disagreement, synchronisation and other inconsistencies that appear in such manual annotations. Furthermore, a ground truth sequence is composed from the analysis, corresponding to each of the segments extracted.

- We experiment with positive vs. negative discrete classification of spontaneous emotional states. Following our initial experimentation with HMMs, we experiment with classification in the likelihood space. We present the first experiments with using SVMs for likelihood classification in emotion recognition, where we show that they outperform other previously used techniques. We also present an assessment of the improvement in the recognition accuracy from the audio signal, by the removal of background noise and speech.

- Continuous emotion recognition has been attempted for speech [199, 117] and the fusion of facial expressions and speech [101]. We present the first approach which focuses on the fusion of three sets of cues: facial expressions, shoulder and audio. We also present the first application of LSTMs on cues related to the visual modality, such as facial expressions and shoulder movement, assessing for the first time the performance of LSTMs with both decision and feature level fusion. We compare the performance of LSTMs to SVR, while we provide comments and discussion on the metrics that may be used for ranking and comparing the performance of such systems.

- We provide experiments which indicate the significance of the length of training sequences for LSTMs, and demonstrate how the overall performance is affected when the features are extracted at a different rate. Furthermore, we describe an experiment for discrete emotion classification given the estimated continuous values from the continuous emotion recognition experiments.

- We provide experimental results with using dimensionality reduction techniques, while we experiment with the further improvement of the performance of our techniques, by providing the first attempts to incorporate correlations between valence and arousal into continuous emotion recognition systems. The related methodologies are then applied with decision-level fusion.

1.2 Structure

The work which we present in this report can be summarised as follows:

- In Chapter 2, we present a review of the theory behind emotion recognition systems. We discuss approaches in emotion description and refer to how emotions are perceived by humans, while also elaborating on issues and challenges that are the focus of research in the field, such as the optimal fusion of modalities, the discrimination between spontaneous and posed emotional states and the dimensional and continuous emotion recognition.
- The theoretical foundations of a set of learning techniques which have been used for continuous emotion recognition are presented in Chapter 3. We focus on Long Short-Term Memory recurrent neural networks and Support Vector Machines for Regression, since these are the basic methodologies used in our project. Furthermore, we briefly discuss Condition Random Fields (CRFs), a probabilistic framework for labelling sequential data. CRFs are described since they have been used for (quantized/labelled) continuous emotion recognition [199].
- Chapter 4 will refer to the process of actually segmenting the audiovisual material, producing the ground truth and extracting the relevant features. We also describe our annotation pre-processing experimentation.
- The essence of our experimental work is presented in Chapter 5. We describe various experiments in both continuous and discrete emotion recognition, ranging from SVM for classification in the likelihood space to fusion of cues/modalities with LSTMs and SVRs. Fusion with LSTMs includes experiments with decision level fusion, while we also apply dimensionality reduction techniques. Finally, we explore whether patterns and correlations in the valence/arousal annotations can further improve our recognition accuracy.
- Finally, in Chapter 6 we summarise and further discuss our findings and conclusions, while we denote topics for future work in the direction of this project.

Chapter 2

Background & Related Work

Any emotion, if it is sincere, is involuntary
- Mark Twain

The background chapter will introduce the reader to concepts relevant to the theme of our project. Our description will cover general related areas and topics within the scope of emotion research, focusing mostly on aspects that will be used in the project while also briefly covering other relevant work. In more detail, Section 2.1.2 will introduce various approaches that relate to the description of affect, discussing some of their advantages and disadvantages. Section 2.2 will discuss emotion perception in humans, focusing on the audio and visual modalities. This section will include a discussion on the temporal features exposed in human emotion expression (Section 2.2.5), a topic particularly interesting with relation to the temporal characteristics of the Long Short-Term Memory networks we later use. Furthermore, in Section 2.4 we will briefly discuss topics related to data annotation and available emotion databases, while in Section 2.4.1 we will discuss issues in discriminating in discriminating posed vs. spontaneous emotional expressions. Finally in Section 2.5 we will refer to continuous dimensional emotion recognition.

2.1 Description of Affect & Emotion

The description of affect has been a long standing problem in the area of psychology. The selection of a proper description is of vital importance to the design of systems relating to emotion recognition, since firstly they would have to be able to process and detect information which is relative to the description, while the type of the description could cause variations in the experimental results of the system. Other secondary issues could involve the complexity of the description or the difficulty in obtaining the cues which would lead to a description of the selected type (which could affect performance of real-time systems).

It is notable that the definition and classification of emotions has been a long standing problem for the human race. The Stoics (3rd century B.C.) [75] considered a categorisation of emotions into pleasure and delight, distress, appetite and fear, while the *Li Chi*, a Chinese encyclopedia compiled during the 1st century B.C., discriminates between seven feelings, which are biologically hard-wired [1]to humans [158] :

[1] In fact, one of the emotion theories that we will discuss in Section 2.1, which is one of the dominant approaches in

> *What are the feelings of men? They are joy, anger, sadness, fear, love, disliking and liking. These seven feelings belong to men without their learning them* [112]

Other similar attempts took place by philosophers and scientists such as Descartes [47], Spinoza [172] and Darwin [43], until the modern theories on emotion classification which appeared in the 1970s, which we will describe in Section 2.1.

Throughout this chapter, we will often refer to terms such as *affect*, *feeling* and *emotion*. To resolve any ambiguities, it would be useful if we provided some definitions [120]:

- **Affect:** is an innately structured, non-cognitive evaluative sensation that may or may not register in consciousness.
- **Feeling:** feeling is affect made conscious, possessing an evaluative capacity that is not only physiological based, but that is often also psychologically (and sometimes relationally) oriented.
- **Emotion:** is psychosocially constructed, dramatised feeling.

Of course, emotion, affect and feeling, although they are terms which are deeply routed in human behaviour they are quite subtle when it comes to defining them and in many occasions the terms are used interchangeably, to a certain extent.

In general, there are three well accepted approaches to describing affect in psychology, which we will now summarise.

2.1.1 Theory of Basic Emotions

The basic emotions theory (or the *categorical approach*) is one of the most dominant approaches, which claims that there is a set of basic emotions which are universally identified by humans, due to being biologically fixed (innate) in human nature and thus independent of factors such as culture and origin. As we have mentioned in the introduction to this chapter, this notion of a categorical approach has been supported throughout the centuries, with references from the 3rd (Stoics, [75]) and 1st (Li Chi, [112]) century B.C.. Modern work, such as that of Ekman ([59, 80, 54]) who after researching stills of posed facial behaviour[2] ended up in categorising 6 emotions as basic: happiness, sadness, surprise, fear, anger and disgust. This theory is the most widely used in automated affect recognition systems [86, 203]. The fact that this approach is intuitively mapped from the description to the actual emotion by humans (since it is self-descriptive) is considered one of its major advantages. On the down side, it is a very limited discrete categorisation and using it by itself disables the expression of many non-basic emotions which do not fall into these categories. Also, there is dispute on whether emotional states such as surprise express real emotion or if they are just (affect) neutral states [136], while there has been criticism on the inability of this model to capture complex emotional states.

Furthermore, it is argued that these so called basic emotions are usually a small fragment of the emotions that are expressed in human every day life, whereas more subtle emotional states

modern psychology, considers again a set of emotions biologically related to humans.

[2] Ekman had participants from 10 different countries which would rate expressions based on 7 emotions

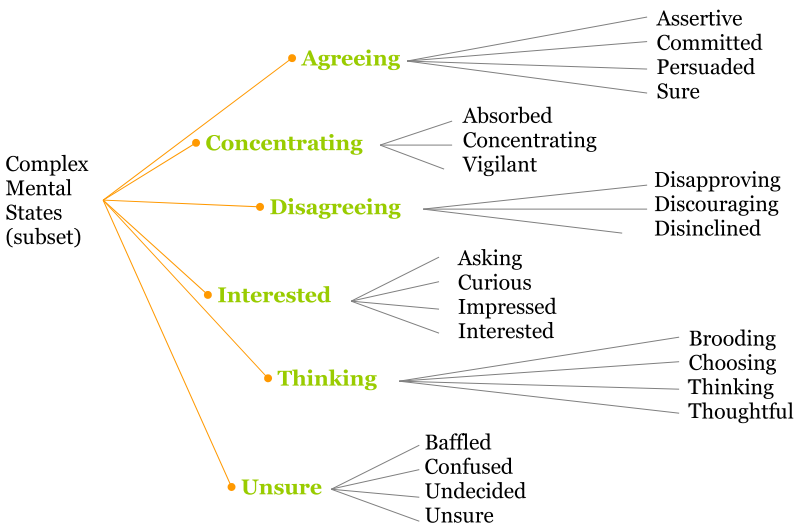


Figure 2.1: Complex Cognitive Mental States, image from [61]

are considered more common [12]. With respect to the latter statement, some researchers have focused on *cognitive mental states* (Fig.2.1), while claiming that these states do appear more often in everyday life [136]. Finally, we should note that there have been suggestions of maintaining a set of more than 6 basic emotions (ranging from 2 to 18 categories) [136, 196].

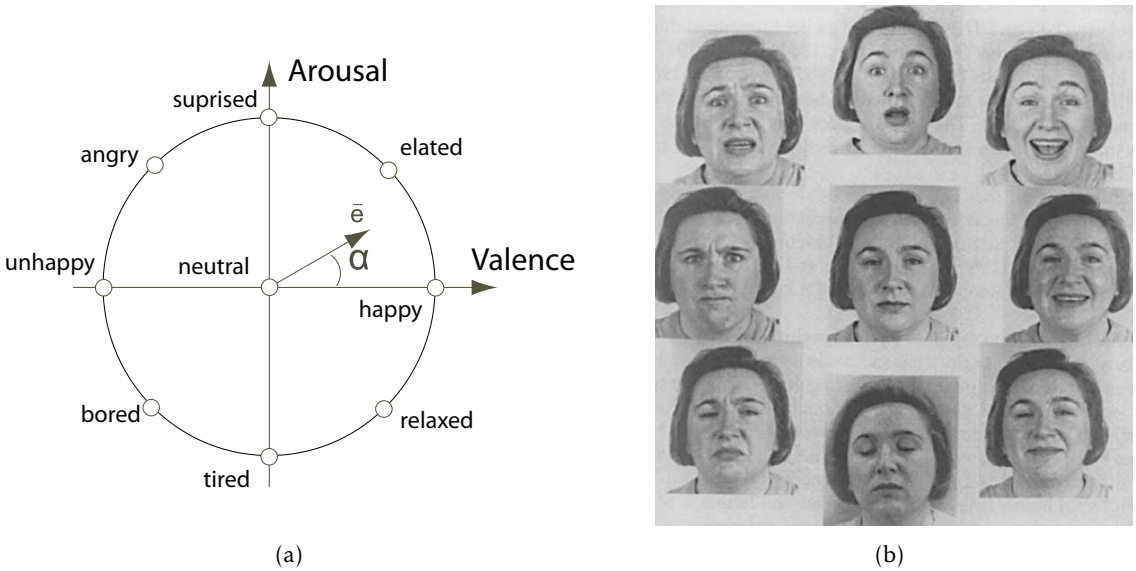
2.1.2 Dimensional Description

Another approach in describing affect is taking, instead of discrete emotions, a dimensional description, i.e. in a small number of latent dimensions (Fig. 2.3). Initially, the characterisation of emotions into such dimensions comes from Wilhelm Wundt, who in 1897 concluded that affect could be described by the dimensions of pleasure-displeasure, excitement-calm and strain-relaxation [23]. The modern relevant theory in psychology follows the model of Russel [157, 152], accepting two major dimensions, which are assumed to represent the major aspects of affect: arousal and valence (Fig. 2.2). Arousal is considered to designate the relaxed vs. aroused state, while the valence characterises the pleasantness (positive) or unpleasantness (negative) of the affective state. It should be noted that other work in dimensional emotion description also refers to a third dimension, *potency* or *power*, which refers to the degree of control that the individual feels with respect to the emotional state [44, 1, 137].

The negative aspect of this approach, is that certain emotions can become degenerately indistinguishable (i.e. by having the same value in each dimension), while certain emotions fall out of the two or three dimensional space. Also, in order to attain measurements of this type, special training is required, while there have been attempts to achieve measurements by measuring physiological parameters (e.g. EEG) [31].

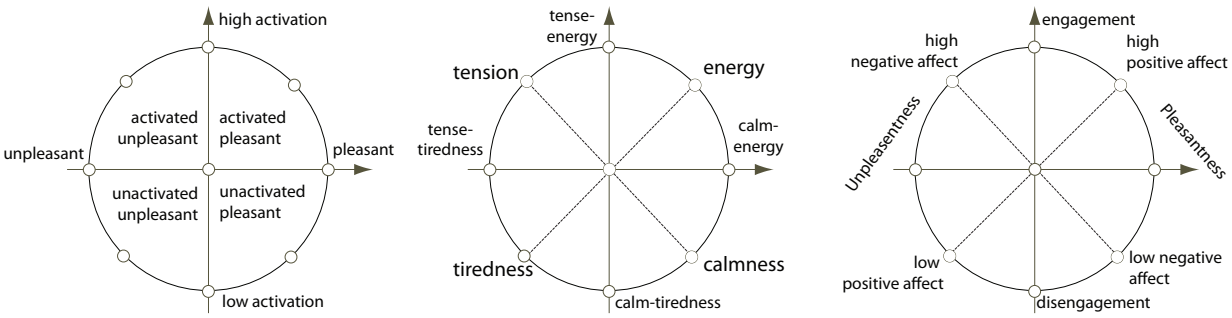
2.1.3 Appraisal Based Approach

An approach which has recently received attention in the psychological world is Appraisal Theory. Essentially, this approach supports that emotions are extracted from appraisals (or evaluations) of events, which are different for each person and context dependent. In order to approach this de-



(a) (b)

Figure 2.2: (a) Russel's valence-arousal space. The angle is represented by α while the vector $\bar{e}$ represents the emotion (point) as a parameter of valence and arousal . (b) Nine facial expressions arranged in the ordering of (a). Image from [150]



(a) (b) (c)

Figure 2.3: Other 2D emotion categorisation approaches: (a) Approach of Larsen and Diener [109] (b) Thayer [176], (c) Watson and Tellegen [175]

scription, some stimulus evaluation measurements are required, measuring e.g. novelty, intrinsic pleasantness, goal-based significance, coping potential, standard compatibility even legitimacy and importance [161]. Inherently, this method does have problems in translating to a computational model capable of emotion recognition, although there has been experimentation with e.g. neural networks [160].

Having described the three basic emotion description approaches, we denote that in the current project, we will base our work on the valence/arousal 2D emotion space, from which we will perform both discrete (using valence to indicate negative or positive emotions) and continuous recognition. Further details on emotion description are available in [162, 74].

2.2 Modalities & Emotion Perception

In this section, we will refer to the different modalities which carry affect related information, focusing on the body and facial expression cues (which essentially constitute the visual modality,

Section 2.2.1), as well as the audio modality (Section 2.2.2). Furthermore, we will discuss issues that relate to the fusion of the modalities, an uprising research challenge typically confronted when realising audiovisual recognition systems. Also, we will refer to the temporal characteristics of facial and body expressions (Section 2.2.5), which are considered highly significant in comparing spontaneous vs. posed affect recognition (Section 2.4.1).

2.2.1 Vision

2.2.1.1 Facial Expressions

In order to capture the complexity and provide a taxonomy of human facial expressions, Ekman and Friesen developed the Facial Action Coding System (FACS) [58] in 1978. This model is widely accepted as a standard to categorise the facial reflection of emotions, based on Carl-Herman Hjortsjö's book on the anatomy of facial features [30]. The FACS model consists of 32 *action units* (AUs) which in turn represent the contraction or the relaxation of one or more facial muscles (Fig. 2.4).

It is an important advantage of the model that AUs are objective representations of human expressions and are independent from any assigned interpretation, thus allowing further high level decisions and processing. It is also important to note that in 1984, the same authors developed an different version of the system (EMFACS) [68] which was entirely oriented towards emotion recognition, retaining only the AUs and AU combinations which were empirically or theoretically considered to be signals of emotions. This resulted in a reduction of coding time, while also providing some criteria for classification [102].

In general, the FACS model is widely used in combination with Ekman's theory of basic emotions in computerised systems, offering advantages such as culture independent detection, reduction of dimensionality and training data required [183]. Finally, it is important to mention that the mapping of facial expressions directly onto the 2D valence arousal space has been suggested [149, 156].

2.2.1.2 Body & Gestures

Psychologists and researchers in general have long attributed the expression of emotional states through body movement and bodily gestures (e.g. [88, 4, 130]), originating from Darwin and the description of animal and human emotion expression. There has been research work indicating the disambiguation of emotional states through analysing animal body expressions, while research shows that under specific circumstances, drawing information from the entire body leads to better appreciation of emotional states (e.g. in infants). Studies also show that results from analysing the body expressions are as significant as the voice modality, and in some cases as facial expression recognition [39]. Van der et al. [187] provide good recognition results for the basic emotions by body information and a possible enhancement/disambiguation of facial/vocal information. There has been research in combining posture and body information with kinematics [53, 82, 83, 118], while there were also attempts to relate emotions to kinematic[3] data (e.g., joint angle data for head tilt, rotation, neck flexion, shoulder abduction, elbow flexion and knee

[3]Kinematics is a branch of classical mechanics which relates to the description of motion

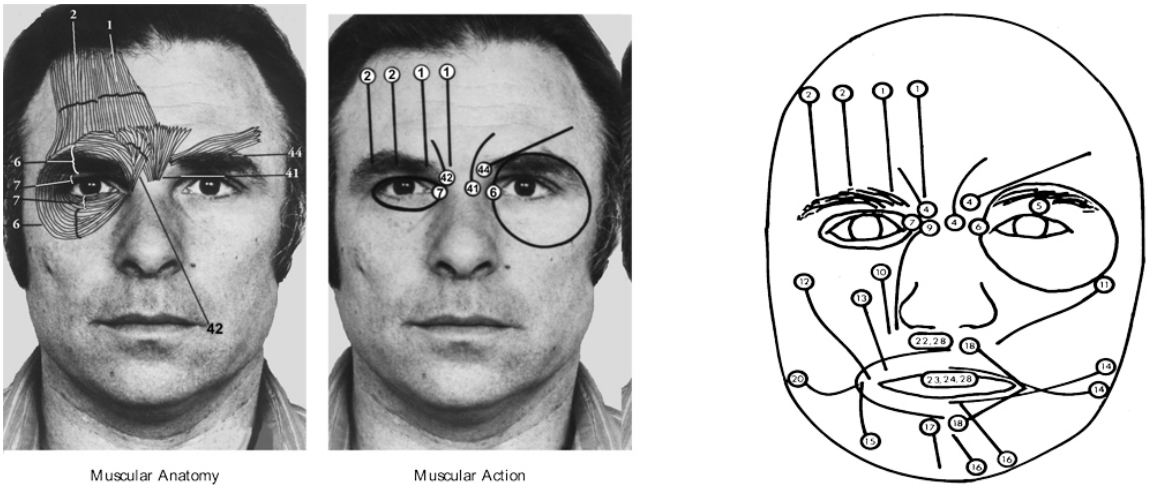


Figure 2.4: Left: Relation between muscular anatomy and muscular actions (Action Units). Right: The AUs of FACS. Circle represents fixed point towards which skin is pulled along the line during activation while number represents the AU. Both images come from [57].

flexion) and gait parameters [4] (velocity, cadence or steps per minute). Results for such attempts varied and demonstrated a difficulty in recognising emotions such as anger, while attaining best performance in recognising sadness. The most characteristic parameters expressing emotion were related to limb motion and general posture. It is important to notice that, in contrast to facial expressions, there is no standardised method in interpreting human postures and gestures (like FACS) and no equivalent to AUs, although there have been efforts in that direction (e.g. [107]).

2.2.2 Audio

Audio and speech are essential carriers of human affect, a fact confirmed in everyday human life. The acoustic behaviour of humans is separated into the transfer of linguistic, paralinguistic and extralinguistic information, although only linguistic and paralinguistic are communicative [110]. The *linguistic* part is refers to language itself, being precisely the explicit verbal part of the communication. The *paralinguistic* element refers to the non-verbal part of the communication, which is used as to modify the verbal meaning or convey emotion (e.g. falsetto in mocking), whether it is expressed unconsciously or consciously. Features such as volume, pitch and intonation are related to paralinguistics. The *extralinguistic* element refers to informative but not communicative information which might e.g. identify the speaker from overall pitch and loudness of speech. The extralinguistic part refers to information which has no conventional meaning, but is unintentional, for example pitch differentiation based on age and sex [42]. Usually in emotion [203, 86] and speech recognition [165], the discrimination is between verbal (linguistic) and non-verbal (paralinguistic, extralinguistic) elements of speech. Important information with respect to the expression of emotions is deemed to be conveyed in the paralinguistic part, while it has been reported that spoken messages are not reliable in expressing affective behaviour [133], as e.g. a different selection of words is used by different persons in order to express the same affective state, while other difficulties can be for instance, in cases where human speakers refer to emotional states which are irrelevant to their current emotional state. Despite the difficulties, there

[4] Gait analysis is related to the quantization of parameters in order to help athletes improve their performance or identify posture related problems

have been attempts to generate dictionaries of words and affective states, e.g. Whissell's dictionary of affect in language [195], which is essentially a list of 4000 words, with a 2D rating in the activation/evaluation space.

Implicit paralinguistic messages have, on the other hand, been proved to provide significant contribution towards emotion recognition appreciation, while parameters which have been identified as strong indicators of emotions are continuous acoustic measures, especially those who relate to the pitch (fundamental frequency) such as frequency range, the mean, median and variability values [86]. Further detailed surveys in this area include the work of Scher and Juslin [155] and Backorowski and Owren [114], while a survey of acoustic features is presented in [40]. It is important to note that while the identification of the optimal feature set is yet an open problem, human listeners are accurate in detecting basic emotions from prosody features (rhythm, stress, intonation) [155] and some nonbasic affective states from non linguistic vocalisations like laugh, cries, sighs and yawns [156].

2.2.3 Physiological Parameters & Heat

There have been other methods of attaining results and measurements of human affective states. We will refer to them briefly in this section, since they are not directly related to the methods that we will use in the project. Firstly, we will refer to measuring physiological parameters or bio-potential signals. The range of parameters ranges from measuring brain signals by functional Near Infrared Spectroscopy (fNIS), scalp signals by electroencephalogram (EEG), peripheral signals such as cardiovascular activity, electrodermal activity, Galvanic Skin Response (GSR) and the electromyograph (EMG). It is believed that these measurements can be translated to the valence-arousal emotion space (Section 2.1.2), but again the optimal feature set is yet to be discovered.

There have also been research attempts which suggest a correlation between emotional states and core body temperatures of mammals, e.g. the change in the facial temperature of monkeys when they are under stressful situations, or the body temperature of rats under similar fearful situations. It is also notable that a correlation has been found between measurements in blood flow and changes of affective states [179, 144], due to thermo-muscular activity. Thus, by obtaining objective measurements of the skin temperature change, there is a possibility of obtaining information for affect states of subjects. Again, a generic framework for these measurements is yet to be defined. For more details and research attempts, the reader can refer to [86].

2.2.4 Fusing Modalities

An uprising problem in the field of emotion recognition is the fusion of more than one modality/set of cues in order to enhance the performance and robustness of designed systems. It has been reported that in human-to-human communication, the combination of information from the speech, gestures and body posture is essential in many situations, such as when ambiguity of speech arises, when communication is noisy or weak (e.g. not good knowledge of language) [126, 125], while in everyday human communication these modalities are fused either consciously or subconsciously. McNeil emphasises what he calls the *conceptual expression* of gestures in combination with language, as he claims that the speaker is thinking in images and in words, expressing words by language and images by gestures. It is suggested that there is great influence from the

facial expressions to vocal characteristics (tone of voice, prosody) and vice-versa ([119, 45]), as there exist patterns of influence in affect information between the audio and visual modalities [173]. It has also been reported that body expressions disambiguate the classification of facial expressions as well as that body operation (which could be irrelevant to emotional expression) affects vocal features such as tone [187].

Summarising the findings in this area, we can deduce that it is of high importance to be able to optimally integrate information from audio and visual modalities, in order to construct more robust and efficient emotion recognition systems. The fusion of modalities though does introduce some further questions to the area. A first question would be which modality conveys the most significant information. In general, if we are fusing a set of modalities/cues, they can either agree (congruent) or disagree (incongruent). An example of the latter situation can be found in Fig. 2.5. Meeren et al. investigate the agreement and conflict of facial and body modalities, by presenting images of faces on body's to participants, with agreeing (e.g. happy face on happy persons body) or conflicting information (sad face on an exited persons body). The human participants opted towards the trusting the body expression where the information was conflicting, leading to an indication of the importance of bodily expression in the presence of ambiguous facial expressions. Another question, which is mostly dependent on the implementation part is whether the features extracted from the modalities used should be combined and fed as one input to the classifier (feature-level) or if a classifier should be responsible for each modality/cue separately and the final decision would depend on the decisions of each classifier (decision-level fusion). There is also the model-level fusion category, where the model itself takes care of fusing the separate data streams, e.g. Coupled Hidden Markov Models (CHMMs) ([203]). There have been difficulties in combining features from more than one modalities into one feature vector, and the majority of multimodal systems uses decision-level fusion [203]. Of course decision-level fusion poses another question. Since humans allow modalities to affect one another in human to human interaction, then there is some *loss* of information from considering modalities to be independent.

It is noted that psychological emotion research has not presented any similar results for other modality fusions, and this is still one of the major research questions in the area. Another issue that has not been explored in multi-modal emotion recognition is the distance of the person in question (with respect to the capturing devices) and the effect this has on the human perception of emotion, e.g. it has been suggested that close-ups of faces cause the human observer to focus on the eyes [187] while as the distance grows the whole body becomes more significant (of course at a certain distance the facial characteristics are not distinguishable). Also, it is important to note the issue of synchronised audio and video signals, since it has been suggested that humans find it difficult to perform recognition tasks under unsynchronized audiovisual stimuli [3].

2.2.5 The Significance of Temporal Features

Describing temporal characteristics is quite important in the area of affect recognition, since such information could provide further indications of the affective state of the person. As we will see in Section 2.4.1 the timing of smiles can demonstrate whether a person is posing a smile or is spontaneously smiling [37, 184]. The significance of these temporal characteristics has been shown in many studies, such as [5] where the importance of time slices against stills in personality judging

Figure 2.5: Demonstrating how ambiguities are resolved with information from multiple cues. In image (1), we have information only from the facial expression of the person. This information can be considered ambiguous, or even assumed to be an angry expression. In image (2), by attaining more information on the body expression of the person (here, the gesture) we can suspect that this is a celebration. In the complete image (3), more contextual information is added as another athlete in a similar celebrating pose appears. It is also noted that there is some contextual information in image (2), since from the persons uniform it can be deduced that he is an athlete.

is denoted and in [163], where discussion involves temporal features of "social" expressions such as smiles and other expression components such as yawning and eyebrow flashing.

The modern theory on describing temporal aspects of facial expressions is based on Ekman's work [55]. More specifically, the phases that constitute a temporal facial movement are the following:

- **Neutral.** The neutral phase is when there are no manifestations of muscle activation and the face is considered to be relaxed.
- **Onset.** Onset phase occurs at the beginning of an action, where the activity in the facial muscles begins, and gradually increases in intensity.
- **Apex.** Apex is the plateau when the intensity of the motion stabilises.
- **Offset.** The last phase is the offset phase, where the muscular action begins to relax.

Typically, human facial motions follow the pattern:

$$\text{neutral} \rightarrow \text{onset} \rightarrow \text{apex} \rightarrow \text{offset} \rightarrow \text{neutral}$$

as seen in Fig. 2.6. Moving to the temporal structure of body gestures, there have been similar results although body gestures are not as explored as facial expressions. Generally, such a gesture can take up to five phases [86, 127]: Firstly, the preparation phase, where the body parts move to the posture where the gesture stroke starts. At the end of the preparation, the pre-stroke hold state occurs where the body parts hold in position. Then, the stroke phase reaches the peak of intensity in the gesture, while the post-stroke hold is the phase were the final gesture position is reached. The retraction phase, is when the body parts return to the previous state. As argued in [127, 198] the only required part in this transitional process is the stroke, while all other phases are optional. To summarise:

$$\text{preparation} \rightarrow \text{pre-stroke hold} \rightarrow \text{stroke} \rightarrow \text{post-stroke hold} \rightarrow \text{retraction}$$

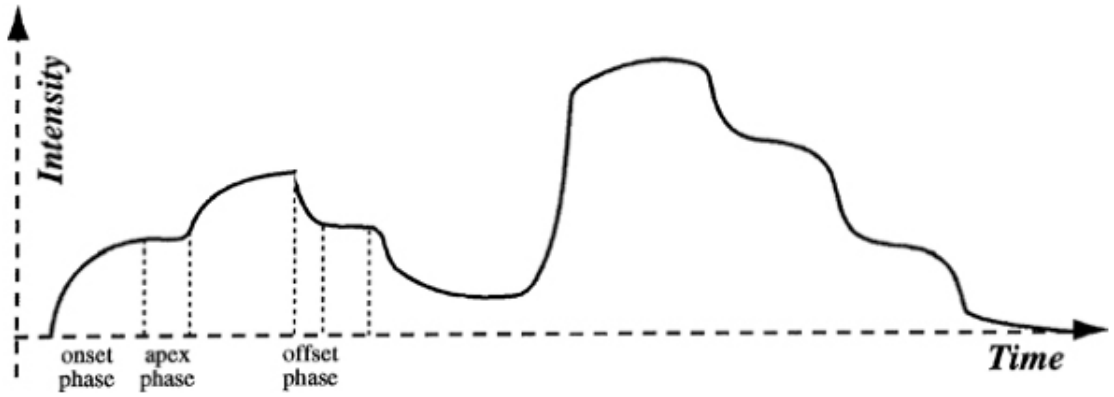


Figure 2.6: A hypothetical example from [60], where temporal facial phases are portrayed as functions of intensity. The neutral state is assumed to occur when intensity is around zero, e.g. observe the intensity when time is zero

Detailed surveys on body expressions are typically limited to observation of still images and thus, it is still an open research issue to extensively investigate temporal information and its effects, while more work is required for exploring the correlation of temporal information with physiological parameters [86].

2.3 Feature Extraction

This section will consist of some description in terms of methodologies used to extract features from either recordings of subjects or sensors, which will be then used for classification in machine learning affect recognition systems by using various methodologies such as Hidden Markov Models (HMM), Support Vector Machines (SVM), Baysian or neural networks. We will firstly describe the extraction of features from the facial expression domain (Section 2.3.1), where two separate steps need to be taken: The detection of the face and the subsequent extraction of features. Then, we will briefly refer to methodologies which target the extraction of information from body expressions (Section 2.3.2), while in Section 2.3.3 we will describe some related audio features, providing also their relationship to affective states. It should be noted, that in this project we will extract features from facial expressions and shoulder movements in a similar way to [184]. The details of feature extraction as used in the current project appear in Section 4.3.

2.3.1 Facial Expressions

Firstly, in order to extract features from faces, the actual location and face of the person must be detected (*Face Detection*). This process has been simplified by various assumptions, i.e. that there is only one face in the image [86] or by limitations in the posture of the person (front or profile view). The detection of the face continues with the training of classifiers with positive and negative examples of faces, while modern methods for face detection are based on the Viola-Jones algorithm [192], which has been extended and improved in [115, 66]. It is interesting to mention the open-source openCV library[5] which includes face detection based on the Viola-Jones algorithm.

[5] http://sourceforge.net/projects/opencvlibrary/

After the face is detected, the next step is to extract the desired features from the visual information. In general, there are two categories of approaches to feature extraction:

- **Feature-based approaches,** which make use of geometric features and detect specific facial features such as pupils, eye corners, mouth corners and so on. They make use of facial anatomy in order to evaluate these features, while comparing distances between the features. Some feature based approaches are [73, 34, 140, 139, 185]
- **Appearance-based approaches**, where certain regions are treated as wholes, and where image filters e.g. (Gabor or Haar filters) are applied to the entire face or to specific regions of the area. The filter techniques are similar to the ones used in face detection. Some relevant approaches are by Barlett et al. [13, 14, 15], Guo and Dyer [87] who use Gabor filters, Anderson and McOwner [7] who implement template matching and Valstar et. al [182] where temporal filters are exploited.

It is not clear whether appearance-based or feature-based extraction is best, since there have been surveys suggesting the better performance of either appearance-based [13] or feature based methods [178, 139, 185] in some situations. There have been attempts to produce hybrid systems (e.g. [178, 204]), and it has been suggested that methods which combine the two approaches could provide better results [138]. The interested reader can refer to [138] for more information.

2.3.2 Body and Gesture

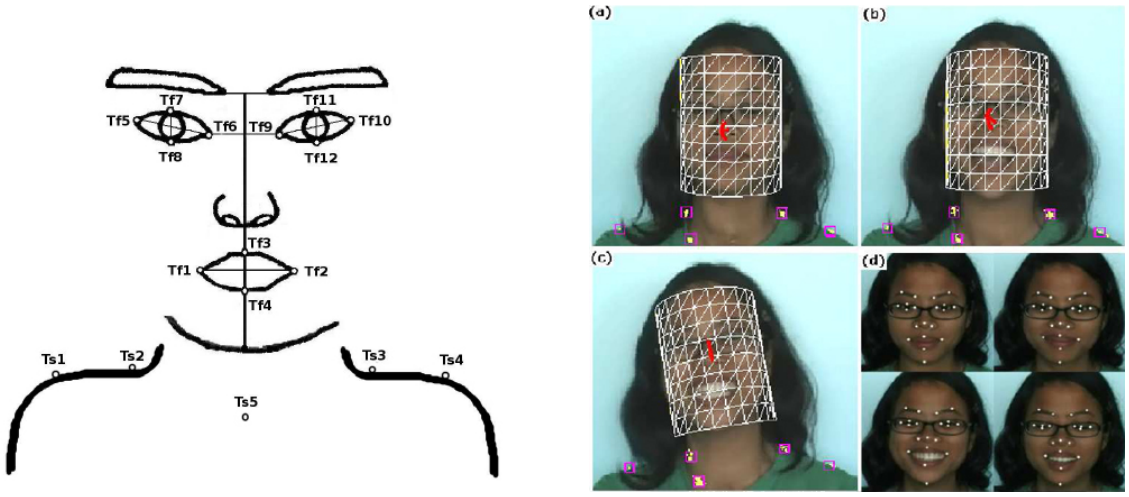
There have been many attempts in interpreting and capturing human gestures and body posture, combining techniques from fields such as computer vision and image processing, mostly targeting Human Machine Interaction (HMI) systems. Specific systems that make use of these capabilities are sign language recognition systems, computer control through gestures, alternative computer interfaces and systems which target emotion recognition.

According to [86], methodologies relating to gesture and body recognition are separated into three categories:

- **model-based**, which depend on the body or body parts by modelling them or recovering 3D configuration from vision processing
- **appearance-based**, which based the recognition process on 2D information, e.g. by tracing edges which could form body contours.
- **motion-based**, where the main characteristic tracked is related to motion

In general, gesture recognition is one of the most difficult tasks in computer vision and machine learning, due to the difficulties commonly appearing in such scenarios (illumination, background/foreground separation, edge tracing, background, occlusions). There is also the issue of separating out irrelevant body motions (which may occur during a proper gesture), determining when a gesture begins manifesting and when it terminates, while also another problem is when a gesture overlaps another.

There is quite a variety of techniques used for tracking, which is covered in [50], while an example of a system related to tracing specific features can be found in [134], where the system detects



(a) (b)

Figure 2.7: Tracking systems from [184], similar to what will be used in the project. (a) Points that are tracked for face and shoulder modalities (b) The tracking procedure: (a-c) tracking points for head and shoulder modalities, (d) for the face modality

shoulder positions by fitting a parabola to detected horizontal lines in the image and then using the weighted Hough Transform to detect the shape. In [184], head motion is detected with a cylindrical head tracker [200], while a 12 point tracker is used to capture facial features. In order to track shoulder motion, a particle filtering technique is used. The face and shoulder tracking points as well as an illustrative example is presented in Fig. 2.7.

Specifically, body gesture recognition requires the calculation of different features, such as the measuring the amount of motion compared to outline changes, hand velocity etc. It is again noted that these methods are optimised for very constraint environments and the development of generic body gesture systems is still an open issue.

Relevant extensive surveys on these areas include Yilmaz et al. [202] on general object recognition and specifically vision-based human motion analysis, Mitra and Acharya's [129], specific to hand gestures and facial expressions, and Poppe [151], which surveys modern approaches to vision-based human motion while also discussing theoretical issues of human motion in relation to modelling (e.g. kinematic models, silhouettes, contours). There is also a discussion on the issue of *estimation*, i.e. finding the set of pose parameters to minimise the observation error in relation to the model (or example set or projection function) used to estimate it. A survey on tracking and tracking methodologies can be found in [138].

2.3.3 Audio

Extracting features from audio is generally based on measurements which relate to the fundamental frequency or pitch, while the factors that are hailed as most significant are the pitch and energy [203]. In Table 2.1, we present a summary of such features in relation to emotion expression. A specific set of spectral features which has been identified to be suitable for speech recognition / speaker identification is the Mel-frequency cepstrum coefficients (MFCC), which is

Table 2.1: Sound features in relation to emotional states from [40]

	Anger	Happiness	Sadness	Fear	Disgust
Rate	Slightly faster	Faster or slower	Slightly slower	Much faster	Very much faster
Pitch Average	Very much higher	Much higher	Slightly lower	Very much higher	Very much lower
Pitch Range	Much wider	Much wider	Slightly narrower	Much wider	Slightly wider
Intensity	Higher	Higher	Lower	Normal	Lower
Voice Quality	Breathy, chest	Breathy, blaring tone	Resonant	Irregular voicing	Grumble chest tone
Pitch Changes	Abrupt on stressed	Smooth, upward in ections	Downward in ections	Normal	Wide, downward terminal in ects
Articulation	Tense	Normal	Slurring	Precise	Normal

often used in affect recognition systems[6] (e.g. in laughter detection [146]), while other attempts experiment with the quality of the voice [26] or measuring pauses/silence [48]. Following the shift towards spontaneous emotion detection, there were approaches which combined acoustic features and spoken words, while others used linguistic features to improve spontaneous emotion recognition. A detailed survey of such work is presented in [203], while it is important to note that deciding the optimal feature set for audio is still an open research problem.

[6]The number of coefficients used for emotion recognition are typically less than the ones used for speech recognition. Many systems, including our work in the current project, use 6.

2.4 Databases & Data Annotation

An important problem that researchers in this field are often confronted with is the proper acquisition and labelling of data. We have already referred to the problem of determining spontaneous vs. posed data (Section 2.4.1) and in general, the long-term goal of realising systems which perform automatic spontaneous emotion recognition. In fact, strictly speaking there are three types of affective data:

- **Posed,** where the participants are requested to produce the affective state on demand, usually in laboratory settings.
- **Induced,** where the experiment takes place in controlled environments which are designed in order to induce the affective states, e.g. by projecting movies to the participants or capturing human-to-human or human-to-machine interaction [11].
- **Spontaneous,** as in occurring in real-life settings, e.g. in naturalistic human to human communication.

Recording the subjects in such databases requires the use of cameras for facial and body expressions and microphones for recording the audio signals, while often motion capture systems are used to record 3D postures and gestures. Ideally, these sensors should be minimally intrusive to the actual recording process in order to avoid the disruption of the information to be captured. We discuss issues caused by the means used to capture the audiovisual data in SAL in Section 4.4.2. Such issues relate to variant noise levels in the audio signal, to the recording of speech signals coming from a virtual avatar interacting with the subject while in some cases the headset (microphone and headphones) partially occluded the face of the subjects.

Most existing affective databases contain posed data, where participants follow the neutral-onset-apex-offset-neutral transition we described in Section 2.2.5. This is because there are quite a few difficulties in capturing spontaneous manifestations, which are rare and filled with context-based changes (i.e. depending on the situation), while labelling such information is again tedious and error prone. Also, spontaneous manifestations are typically not in perfectly controlled laboratory situations (e.g. where the participant is asked to posed, look at the camera from specific angle etc.)

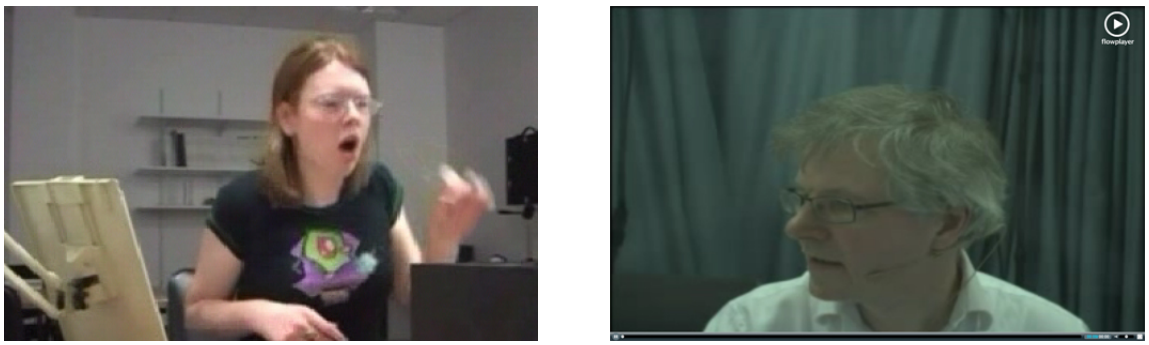
Nevertheless, due to the rising interest in detecting spontaneous emotions and rising up to real-life situations there have been attempts to generate databases of spontaneous emotions. Regarding the labelling of emotional states which appear in these databases, the methodology depends on the description of affect adopted (Section 2.1). Typically, the basic emotions are used for categorisation in posed databases, while spontaneous databases often use the valence-arousal emotion space.

In this project, we will deal with the recognition of continuous dimensional spontaneous emotional states. Thus, we will focus our description on such databases. It should be noted that recording subjects without their permission rises ethical issues (although it has been attempted [205]), thus typical recordings of spontaneous emotions are done in interview settings [138].

A database which contains spontaneous and posed data is the MMI Database [141]. It is considered to be the most comprehensive set of facial behaviour recordings [203], while images and

Table 2.2: MMI Database Feature Overview, adapted from [141]

Feature	Availability
static images	740
videos	848
emotion expression	√
single AU expression	√
multiple AU expression	√
AU-coded	√
no. of participants	19
participant age range	19-62
no. of ethnicities	3
gender	♀♂
lighting	universal
facial hair	√
glasses	√
profile view	√
downloadable & searchable	√



(a) (b)

Figure 2.8: Sample data (still) from the HUMAINE SAL (a) and SEMAINE (b) databases

videos are both available, from frontal and profile views. It includes more than 1500 samples, while the samples are encoded in the FACS system (Section 2.2.1.1). The posed data comes from experts or individuals who are taught how to activate the specific action units on their face - similarly to when expressing emotions. Some of the characteristics of the database are presented in Table 2.2.

In this project we will focus on the SAL (Sensitive Artificial Listener) [49] database, which is part of the HUMAINE project. The database contains spontaneous (induced) data[7], which have been collected during a human-computer conversation, which is conducted between a human participant and one of the four personalities (or virtual avatars), which have a specific pattern of behaviour (Poppy - happy, Obadiah - gloomy, Spike - angry and Prudence - pragmatic). The concept of the system is based on the assumption that the users are drawn into the own emotional states of their avatars. The database contains two female and two male subjects, while there are approximately 10 hours of footage available in the SAL database.

[7]Strictly speaking, the database contains induced data; Nevertheless, we will use the term spontaneous data throughout the project to describe the data.

A database with a similar scenario is the SEMAINE [168] database, which incorporates a SAL agent, with a goal of sustaining a conversation with a user. The goal is that the system will detect the user's facial expression, gaze and voice while engaging in conversation with a conversational agent who will be embodied, will have a face and a voice, will provide feedback and will have different scenarios and strategies, adaptive to the user's behaviour. Data from the SEMAINE project is currently being labelled in terms of the valence-arousal emotion space.

Another database which contains posed data which are appropriate for continuous affect recognition is the Montreal affective voices database [17], while a database which contains spontaneous audiovisual data is the Vera am Mittag speech database [81]. The characteristics of these databases are summarised in [84], while for more details on affect databases, the reader is referred to [85, 203].

As far as data annotation is concerned, a typical tool used is the Feeltrace tool [67], where the affective states of individuals are evaluated on the 2D dimensional valence arousal-space (Section 2.1). In the case of audio-visual recordings, the coder which is responsible for the annotation observes and listens to the recording; The coder moves a cursor within a circle, which represents the four quadrants of the valence/arousal (or activation/evaluation) space (Fig. 2.2). This coder annotation is then normalised and continuous values are then produced, within a normalised unit circle (i.e. in the range of -1 to 1). Other databases such as the Montreal Affective Voices Database [17] include ratings for each sample (specifically for the database in question, there are 10 ratings for each sample covering the range of arousal and intensity). While FACS encoding is the typical encoding for facial expressions, there have been attempts of labelling facial expressions by asking human observers to rate them in dimensional spaces (using a scale) [170].

It is important to state, that as we have also observed with our experience with this project, it is a great challenge to achieve the agreement of coders or observers in mapping the observed emotional stimulus into a dimensional emotion space. The Feeltrace tool has been criticised for the lack of intuitive operation and the requirement that coders who annotate data should be specially trained [203]. Since Feeltrace is the tool used for annotation in the SAL database, we will discuss some of the issues that arise from our experience with the processing of annotations generated with the aid of this tool in Section 4.4. Finally, it is noted that no certain coding scheme which can handle all possible communicative cues has been established and agreed upon [84]. For more details and review of the work on translating physiological signals and body postures to the dimensional space, the reader is referred to [84].

2.4.1 Posed vs. Spontaneous Emotional States

The criticism that affect recognition systems often receive is related to the problems that manifest when they are tested under real life situations. This is not due to many constraints/assumptions that are usually imposed in lab-like environments not only for evaluating the system, but also for the displayed affective behaviour which was used as a basis for the system training, and which was posed, i.e. the participants were asked to pose as e.g. happy. That is why recently, there have been studies that research the analysis of spontaneously manifesting affective states, e.g. by exploiting facial expressions [14, 37, 185, 8] and audio features [16, 111]. Of course, the shift from the controlled laboratory environment to real-life situations bears by itself many challenges and research questions which need to be addressed. Uncontrolled environments differ from lab

situations in almost every single factor: lighting is not controlled, the freedom of movement that the subject has is not constrained, there can be multiple persons in the range of the perception devices (e.g. cameras, microphones) while it is possible that the subject will be out of the range of such devices or that the signals will be quite noisy.

While it is suggested by studies [86] that the accuracy in detecting deliberately posed and naturally occurring or spontaneous expressions is in fact equally accurate, interesting findings relate to the differences between spontaneous and posed expressions. There has been a lot of work in detecting differences between spontaneous and posed behaviour by the Affect Analysis Group [6], while the temporal characteristics of phases as described in Section 2.2.5 have been found important in detecting spontaneous vs. posed smiles [37, 184]. It is also significant to denote the importance of modality fusion in discriminating between posed and spontaneous emotions. It is typical for spontaneous body expressions to be manifested along with an agreeing facial expression. There are different views on whether body motions or facial expressions are most expressive of the spontaneity of the emotional expression. The two main factors that contribute to this is the difficulty of control and the conscious censoring that humans can impose. Darwin's views were supporting the facial expressions, since as he claimed they body expressions are more easy to control, but looking at this problem from a different angle, Ekman [56] supports that humans usually try to censor their face - or they are more aware of the expression on their face - thus the body expressions would be more prone to expressing uncensored information. There has been work that also suggests that truthful and deceptive behaviour differs on the number of head movements [25, 24], or the lack of accompanying gestures [46].

To give some examples of systems which discriminate between spontaneous and posed affective states, we will firstly refer to the system of Valstar et. al [184], which discriminates spontaneous from posed smiles by geometric features and fusion using head, face and shoulder cues. Based on the data, the temporal facial states are detected, along with the activated AUs, while GentleBoost and Support Vector Machines (SVM) (Section 3.2) are used for the classification. Experimentation also occurs with modifying the abstraction level of fusion (early, mid-level and late), while the authors conclude that from the specific results, the head seems to be the most important modality. Another system is that of Littlewort et al. [116], which has a goal of discriminating real vs. fake pain. The system uses the face modality (FACS) to encode facial expressions, using 20 AU classifiers with input data images of posed and spontaneous facial expressions. The authors presented better accuracy compared to human FACS experts (72% to 52%), while they argue that such a method could be also used for other spontaneous expressions. It is important to note that in general, research on spontaneous vs. posed expressions, whether it is from psychology or in developing affect recognition systems agrees that the temporal dynamics appear to be highly significant in determining one from another [203].

The problem of determining spontaneous vs. posed affect and the problem of detecting spontaneous real-life affective states is highly gaining interest in research. Every day human to human communication is indeed multimodal, thus to correctly approach this problem, such multimodal approaches need to be used in experimental and automatic systems, in order to better understand human communication, and by exploiting this knowledge to build robust HCI systems.

2.5 Dimensional and Continuous Emotion Recognition

Systems which target in automatic dimensional emotion recognition, considering that the emotions are represented along a continuum, generally tend to quantize the continuous range into to certain levels. In [104], Kleinsmith & Bianchi-Berthouze discriminate between high-low, high-neutral and low-neutral affective dimensions, while Wollmer et. al [199] work with the SAL database and quantize the valence and arousal into 4 or 7 levels in various experiments. They then use Conditional Random Fields (CRFs) (Which we will describe in Section 3.3) to predict the quantised labellings. Attempts for discriminating between more coarse categorisations, such as quantisation into low, medium and high [106], excited-negative, excited-positive and calm neutral [32], or positive vs. negative and active vs. passive [28] have also been attempted. Karpouzis et al. [28] make use of Simple Recurrent Networks (Section 3.1) for the mapping, while Caridakis et. al [27] which use feedforward back-propagation networks for mapping into neutral and Valence-Arousal quadrants respectively. Both [28] and [27] use the SAL database and combine facial expressions, body gestures and/or audio.

As far as actual continuous dimensional affect recognition is concerned, there have been three attempts so far, two that deal exclusively with speech [199] and [117] and one approach fusing facial expression and audio cues [101]. Wollmer et. al [199] make use of learning techniques such as Long Short-Term Memory neural networks, a type of recurrent neural networks which are generally considered to be able to learn long range temporal dependencies, and Support Vector Machines for Regression (SVR). Grimm and Kroschel [117] again make use of SVRs, and compare to methods such as the distance based Fuzzy k-Nearest Neighbour and rule-based fuzzy logic estimator. In [101], Kanluan et. al present an approach which fuses facial expression and audio cues, experimenting again with SVRs and also late fusion using a weighted linear combination.

In the current project, we will experiment with methods such as Long Short-Term Memory neural networks (LSTM) (Section 3.1.4) and Support Vector Machines for Regression (Section 3.2). Our work will build on top of the current work in the speech area, and will attempt to incorporate facial expression, shoulder and audio cues into dimensional and continuous emotion recognition. For a detailed survey, comparison and review of dimensional affect recognition systems the reader is referred to [84]

At this point it is important to refer to some issues that relate to the realisation of such emotion recognition systems. Firstly, we have mentioned the challenge of achieving agreement amongst the coders or observers which provide annotations in a dimensional space. Most researches take the mean of the observers ratings or assess the annotations manually [84]. In Chapter 4, we will describe our procedure for producing the ground-truth with respect to the coders, which performs various procedures in order to maximise the number of coders in agreement and the agreement level itself. The baseline problem, to which we will again refer to in Chapter 4, refers to the concept of having "a condition to compare against" [84] in order to be able to successfully learn. Such a baseline allows the learning technique to capture the characteristics which define the state which is to be learnt. By referring to the temporal features of facial expressions (Fig. 2.6), a video which contains all the temporal transitions provides the classifier with a sequence which begins and ends in a neutral expression: a baseline. The generalisation of the system refers to whether it should be considered person-dependent or person-independent, that is, whether it should be personalised for the subjects available or should it generalise across them. Typically, dimensional

emotion recognition is reported in a subject dependant manner, since there are typically few subjects (e.g. 4 subjects in [199]). Finally, we denote that the learning techniques which are available may not be the most suitable techniques to capture the idiosyncrasies of continuous affect recognition. In general, techniques which are specific for affect recognition systems are a very important research issue in the field, and they should be able to accommodate multiple, unsynchronized and correlated in a non-linear fashion data.

2.6 Discussion

This chapter has touched upon some of the major topics that relate to the realisation of modern, state-of-the art affect recognition systems. We described the psychological background which supports the research work, by referring to approaches to emotion description and the perception of emotions in humans. We have discussed issues such as the fusion of cues/modalities and the temporal features of facial expressions and body gestures. We have discussed the extraction of features, mostly focusing on the visual and audio modalities. We have referred to the shift towards generating affect recognition systems for spontaneous emotional states, while finally we referred to the problem of dimensional emotion recognition and some of the databases that are available for researchers. We highlight the issues which we referred to in the last section, since many of them do appear in our attempts to segment and extract features from the audiovisual provided by the SAL database, which are described in the Chapter 4.

Chapter 3

Learning Techniques

This chapter will be dedicated to the description of state-of-the-art machine learning techniques suitable for continuous emotion recognition. We will focus our description on the two learning techniques which are used in this project, namely Long Short-Term Memory recurrent neural networks (LSTMs, Section 3.1.4) and Support Vector Machines for Regression (SVR). LSTMs are considered to be able to bridge long temporal intervals and thus learn dependencies in the input data which occur with a long temporal distance [91]. In order to properly introduce LSTMs, we will begin by describing the weaknesses manifested in traditional recurrent networks, and then proceed to the description of the solution proposed and implemented in LSTMS, while we will also briefly refer to improvements and additions to the original networks.

Furthermore, in Section 3.2.2 we will refer to Support Vector Machines (SVM) and Support Vector Machines for Regression (SVR). We will introduce the concepts behind SVM classification, refer to the non-linear mapping that is introduced by kernels, the parameterization of the algorithm determining the permitted amount of errors in the training data and more. We will then refer to SVR (Section 3.2.2), where we will provide an analogous description mostly focusing on the set of changes in the algorithm in order to perform regression. It is noted that we will apply the SVM technique in discrete emotion recognition (Section 5.1).

Finally, in Section 3.3 we provide a description of a probabilistic framework for labelling sequential data, Conditional Random Fields (CRF). Our description will not be as detailed as for the previous techniques, since this technique has not been used in our project. Nevertheless, we discuss essential characteristics of the method, while we provide references for further details.

3.1 Recurrent Neural Networks

In this section, we will provide some description of recurrent neural networks, in order to introduce the Long Short-Term Memory (LSTM) networks (Section 3.1.4). Recurrent neural networks differ from traditional feedforward neural networks (Chapter 4, [128]) topologically, in one basic rule: feedforward networks are allowed to have only forward connections (i.e. in the direction of input to output), while recurrent networks also allow feedback connections, thus permitting the formation of cycles and loops. The previous modification allows recurrent networks to be adapted to past inputs during training, as we will describe in the rest of this section.

3.1.1 From Feedforward to Recurrent Networks

Assuming that we have a regular feedforward network, given an input x at time t, the network performs the following mapping:

$$y(t) = \mathcal{F}(x(t)) \tag{3.1}$$

That is, the network, which has an internal configuration which consists of weights on connections between neurons[1] and the family of activation functions used, will map the input $x(t)$ at any time t to the output $y(t)$. It is important to stress that the output depends only on the current configuration and input. On the other hand, a recurrent network can operate on an internal state space, which ideally contains all relevant information from the past behaviour of the system. This expands the network capabilities by allowing it to capture temporal information and manifest learning abilities such as predicting the next output in sequences or forecasting. Thus, the recurrent network's output at time t, $y(t)$ would be a function of the current state of the network $s(t)$, which in turn depends on the previous state $s(t-1)$ and the current input $x(t)$:

$$y(t) = \mathcal{F}'(s(t)) \tag{3.2}$$

$$s(t) = \mathcal{G}'(s(t-1), x(t)) \tag{3.3}$$

To contrast the computational power that this extension presents us with in contrast to regular feedforward networks, it is enough to say the following: while a feedforward network, given enough hidden nodes can approximate any spatially finite function, recurrent neural networks (again assuming any number of hidden nodes) can represent any Turing Machine [159], while if real weights are used, the network can function as a super-Turing Machine [171], notions which are much more powerful than approximating finite functions.

In this section, we will refer to a neural network with one hidden layer, the input layer and the output layer. For referring to a node in the hidden or output layer, the subscripts h and o will be used respectively. We consider the input to have a size of n, while we consider m nodes in the hidden layer and m nodes in the output layer. The activation of a neuron belonging to the hidden layer of such a feedforward network will have an activation value $y_h(t)$:

$$y_h(t) = \sigma(net_h(t)) \tag{3.4}$$

$$net_h(t) = \sum_{i}^{n} x_i(t) w_{hi} + \beta_h \tag{3.5}$$

That is, the output is the *net* input to the neuron applied to the activation function σ (typically a non-linear such as the logistic function). The *net* input to the hidden node is the sum of the weights coming to node h from each input i (the input vector $\mathbf{x}$ has a size of n), while β is the bias of node h.

Assume a simple recurrent network, where besides the feedforward connections, the nodes of the hidden layers have one step delay feedback connections, that is the previous activation of the

[1] We will use the term neuron and node interchangeably

nodes in that layer is taken into account. Since there are more connections, a new set of weights v_{ij} is required. Again looking at the activation of a node in the hidden layer, $y_h(t)$, Equation 3.4 remains the same. What changes is the $net_h(t)$:

$$net_h(t) = \sum_{i}^{n} x_i(t)w_{hi} + \sum_{j}^{m} y_j(t-1)v_{hj} + \beta_h \tag{3.6}$$

where m is the number of nodes which have the feedback connection to node h and $y_j(t-1)$ is the previous activation of each of them. In the example presented in the section, we stated that feedback loops occur only in the hidden layer, so the equations for the output nodes of the network are the same as the feedforward networks:

$$y_o(t) = \sigma(net_o(t)) \tag{3.7}$$

$$net_o = \sum_{j}^{m} y_j(t)u_{oj} + \beta_o \tag{3.8}$$

Where u_{oj} are weights from the hidden nodes j to the output node o and again, σ is the activation function and β_o the bias of the output node.

3.1.2 Architectures of Recurrent Neural Networks

There have been a lot of architectures of recurrent networks over the past decades. In this section, we will provide a description of some of the most common architectures. It should be noted that fully recurrent networks are defined as neural networks where every neuron receives input from all other neurons in the existing network, while the order of the network is typically the number of neurons whose outputs are fed back with feedback connections.

State-Space Model: The state-space model is a recurrent network where the hidden neurons are the ones which decide the state of the network. That is, the output of the hidden layer is delayed for one time step before it is fed back to the input layer - and thus the previous activation of the hidden layer is considered to be part of the next input. It is generally considered that the hidden layer is nonlinear, while the output layer is linear. Simple Recurrent Networks (SRN) or Elman networks [63] are a considered a special case of the state-space model, where essentially a context (or copy) layer is used to store the activations of the hidden layer nodes for one time step and then feed them back to the input layer. The output layer in Elman networks can be nonlinear. Originally Elman used this type of networks to guess the next phoneme in a sequence of phonemes given as input (Fig. 3.1).

Input-Output Recurrent Model: Networks in this category are composed as multi-layer perceptrons, and have the characteristic of their actual output activation being fed back to the input. In more detail, this is done by using a tapped-delay-line memory of a certain size h, at both the input and output (the sizes can differ). If at any time t, we denote the current output of the network as $y(t+1)$ and the present input $x(t)$, then the data that will be applied to the input layer would be

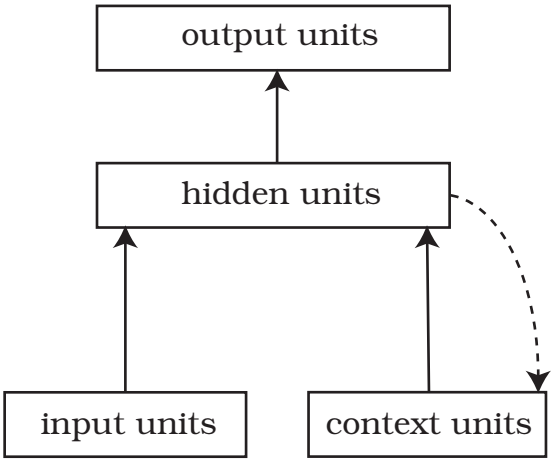


Figure 3.1: Elman network, where the non-dashed lines represent trainable connections

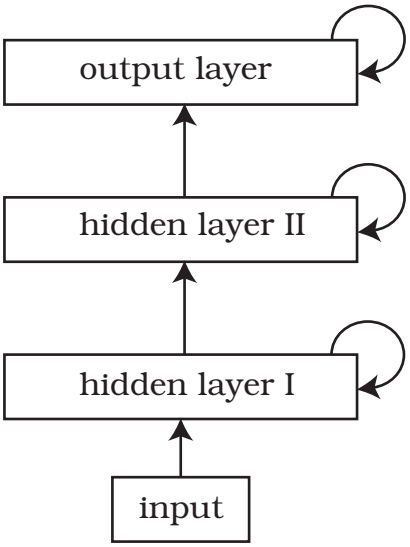


Figure 3.2: The multi-layer perceptron. It is noted that feedback connections have one time step delays, similarly to how the context nodes operate in Elman Networks

the h latest input values, that is $x(t), x(t-1), \ldots, x(t-h+1)$ (named as exogenous inputs) and the delayed output values of the model, $y(t), y(t-1), \ldots, y(t-h+1)$. It should be noted that another form of networks, Jordan networks, are similar to Elman networks with the main difference being that the output is fed to the input layer instead of the activation of the hidden layer. Jordan networks are also considered Simple Recurrent Networks.

Recurrent Multilayer Perceptron: Such networks have one or more hidden layers, with each computation layer (hidden and output) having a feedback around it (Fig. 3.2). The activation functions are not constrained.

Second-Order Network: Order here refers to how the *net* input to a neuron is calculated. When the *net* input of a node j is the sum of the products of weights and inputs from layers connecting to j, it is a first order network, just like the ones we have discussed up until now. Assuming node j receives inputs from two sets of nodes s_1 and s_2 with corresponding weights w and v, the *net*

input would be calculated as:

$$net_j = \sum_{i \in s_1} w_{ji} y_i + \sum_{k \in s_1} w_{jk} y_k \tag{3.9}$$

This is a first-order neuron. A second neuron would combine the inputs by multiplying them, and would have only one weight as follows:

$$net_j = \sum_{i \in s_1} \sum_{k \in s_1} w_{jik} y_i y_k \tag{3.10}$$

Second-order networks, which consist of second-order neurons, have the unique characteristic that they can represent transitions of states due to their multiplicative nature. E.g., in Equation 3.10, the weight w_{jik}, if positive denotes the presence of the transition $\{y_i, y_k\} \rightarrow \{\text{next state}\}$, while if negative the absence of it, thus providing a natural way to represent Deterministic Finite Automata (DFA).

Other Architectures

Other recurrent networks we have not discussed specifically in this section, is the Hopfield network [92], where all connections are symmetric and there is a convergence guarantee and the Echo State Networks [95], a type of networks which has a very sparsely connected hidden layer with a random assignment of weights to its neurons. Variations of architectures exist where some hierarchy is imposed. More information on recurrent architectures can be found in Chapter 15 of Haykin [105] and in Jain & Medsker [96].

3.1.3 Learning Algorithms

Training recurrent networks requires specific learning algorithms, adjusted to the characteristics and the idiosyncrasy of the technique. It is not surprising that recurrent network training algorithms adopt a lot of characteristics from feedforward networks. More specifically, two of the most well known algorithms for training recurrent networks, back-propagation-through time (BBTT) and real-time recurrent learning (RTRL) are both based on the method of gradient descent, similarly to the training of feedforward networks. In fact, BBTT is essentially an extension of the back-propagation principle, as we will discuss in the following section.

3.1.3.1 Back-Propagation & Back-Propagation Through Time

To introduce the BPTT algorithm, we will provide comparisons with the original back-propagation principle found in feedforward networks. The cost function frequently used is Summed Squared Error (SSE):

$$E = \frac{1}{2} \sum_{j}^{m} (t_j - y_j)^2 \tag{3.11}$$

where t_j stands for the target value that we desire the output node to attain, while y_j is the actual output of the node. The previous equation is summed over the entire training set. The gradient descent rule states that a weight change during the network training should be in the direction of

the negative gradient of the cost function, with respect to the current weight under examination. This is better comprehended when considering the gradient graphically. It points to the direction of greatest increase of the cost function, and thus moving in the opposite direction will decrease the actual cost. The weight change Δw would then be:

$$\Delta w = -\eta \frac{\partial E}{\partial w} \tag{3.12}$$

By using the chain rule for deriving the partial derivative of the error function w.r.t. a weight w_{ij}:

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}} = \frac{\partial E}{\partial net_j} x_{ji} \tag{3.13}$$

Here, we used x_{ji} to represent the input to node j from node i, which is the result of the partial derivative of net_j with respect to the weight w_{ij}, since all other factors are zeroed out. It is noted that $\frac{\partial E}{\partial net_j}$ does not change for different weights, and it is the factor which is represented by δ_j for a node j. Firstly, for output nodes:

$$\delta_o = \frac{\partial E}{\partial net_o} = \frac{\partial}{\partial net_o} \left(\frac{1}{2} \sum_i^m (t_i - y_i)^2 \right) = -(t_i - y_i)\sigma'(net_o) \tag{3.14}$$

where the derivative of the activation function is:

$$\sigma'(x) = (1 - x)x \tag{3.15}$$

when the logistic function is used:

$$\sigma(x) = \frac{1}{1 + e^{-net}} \tag{3.16}$$

Thus the rule for updating the weight of a connection from node h to an output node o is :

$$\Delta u_{oh} = -\eta \frac{\partial E}{\partial u_{oh}} = \eta \delta_o x_{oh} \tag{3.17}$$

The work is similar for nodes in the hidden layer. In this case, we consider how the activation of a node in a hidden layer influences the actual error function through the propagation of the value to the nodes in the output layer. Assuming the hidden node is node h, using the chain rule, we essentially express that the error depends on the activation of every output node o, which activation depends on the net input that arrives at the output node net_o which in turn is dependent on the output of the hidden node h and thus the input to the node and the relevant weight. Here we assume that any node in the hidden layer has a connection to every output node.

$$\frac{\partial E}{\partial w_{hi}} = \sum_o^m \left(\frac{\partial E}{\partial net_o} \frac{\partial net_o}{\partial y_h} \frac{\partial y_h}{\partial net_h} \frac{\partial net_h}{\partial w_{hi}} \right) \tag{3.18}$$

By taking into account that:

$$\frac{\partial E}{\partial net_o} = \delta_o, \frac{\partial net_o}{\partial y_h} = u_{oh}, \frac{\partial y_h}{\partial net_h} = \sigma'(net_h), \frac{\partial net_h}{\partial w_{hi}} = x_{hi} \tag{3.19}$$

we can define:

$$\delta_h = \sum_o^m \left(\frac{\partial E}{\partial net_o} \frac{\partial net_o}{\partial y_h} \frac{\partial y_h}{\partial net_h} \right) = \sum_o^m (\delta_o u_{oh} \sigma'(net_h)) \tag{3.20}$$

and finally, the weight update rule for an input weight to node h in the hidden layer is:

$$\Delta w_{hi} = \eta \delta_o u_{oh} \sigma'(net_h) x_{hi} \tag{3.21}$$

In back-propagation through time, we have the set of feedback connections from and to the hidden layer (in our example), with the accompanying weights, v_{ij}. The weight updates would then be as follows:

$$\Delta v_{hj}(t) = \eta \delta_h(t) y_j(t-1) \tag{3.22}$$

We have essentially added the time subscript, in order to emphasise that the previous activation ($y_j(t-1)$) at time $t-1$ is used in the weight update. It should also be stressed that the δ's for the calculating the recurrent weight updates are now again calculated with time steps in mind. Specifically, for *truncated back-propagation through time (BPTT(h))* [197] d steps are kept as a history of the network time steps. The authors claim that essentially this is required for the training to be practical with controllable memory requirements. Thus, the δ's, very similarly to Equations 3.14 and 3.1.3.1 are now calculated as follows, for the time window $[n-d, n]$:

$$\delta_h(t) = \begin{cases} \sigma'(net_h(t))(t_h(t) - y_h(t)) & t = n \\ \sigma'(net_h(t)) \sum_o^m (\delta_o(t+1) u_{oh}(t)) & n-d < t < n \end{cases} \tag{3.23}$$

Unfolding

The concept of unfolding is an integral part of training recurrent neural networks with back-propagation through time. Essentially, the temporal operation of the network is *unfolded* through time and the network is transformed in to a layered feedforward network. Assuming we have the temporal information for the training of the network, for the time interval $[n-h, n]$. Assuming the original network is G and the unfolded network is G_u. Then, the following are true:

- For each time step $t \in [t-h, t]$, the unfolded network has a layer $l(t)$ which corresponds to t and has the same number of neurons as the entire original network G. The neurons of the layer $l(t)$ are copies of the neurons in the original network G.
- For each layer $l(t)$ in the network G_u, the synaptic connection between two nodes i and j is a copy of the connection between nodes i and j in network G.

An example of unfolding can be seen in Fig. 3.3

3.1.3.2 Exponentially Decaying Error

Short-term memory in recurrent networks stands for stands for the storing of recent input events in the form of activations using the feedback connections, as discussed in the previous sections. When we refer to *Long-term memory* in such networks, we refer to the weight change of the networks during training. With conventional training algorithms for recurrent networks (such as

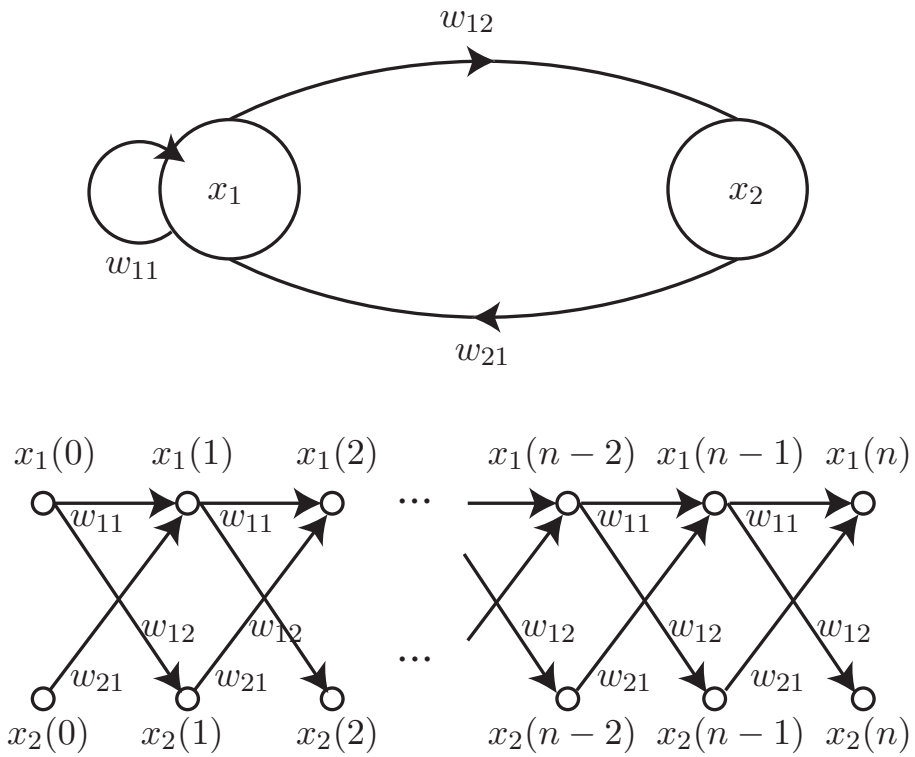


Figure 3.3: Unfolding a recurrent neural network. Top: the original network, Bottom: the network unfolded for n time steps

BBTT) error signals flowing backwards in time either vanish (the vanishing gradient problem) or exponentially increase. Thus, the training is considered unable to learn long term dependencies in the data [89, 19, 90]. According to Hochreiter's analysis [89], assuming a fully connected network with nodes 1 to n, excluding the input nodes, the (local) error propagated from a node u at time t to a node v for q time steps will be scaled as follows:

$$\frac{\partial \delta_u(t-q)}{\partial \delta_u(t)} = \begin{cases} \sigma'(net_u(t-1))w_{uv} & q = 1 \\ \sigma'(net_u(t-q))\sum_{l=1}^{n} \frac{\partial \delta_l(t-q+1)}{\partial \delta_u t} w_{lu} & q > 1 \end{cases} \tag{3.24}$$

The $q = 1$ part of the equation is derived by looking at Equation 3.23:

$$\delta_v(t-1) = \sigma'(net_v(t-1))\delta_u(t)w_{uv} \tag{3.25}$$

thus:

$$\frac{\partial \delta_v(t-1)}{\partial \delta_u(t)} = \sigma'(net_v(t-1))w_{uv} \tag{3.26}$$

The calculations are similar for the case of $q > 1$. By setting $l_q = v$ and $l_o = u$, Hochreiter shows by induction that:

$$\frac{\partial \delta_u(t-q)}{\partial \delta_u(t)} = \sum_{l_1=1}^{n} \cdots \sum_{l_{q-1}=1}^{n} \prod_{m=1}^{q} \sigma'_{l_m}(net_{l_m}(t-m))w_{l_m l_{m-1}} \tag{3.27}$$

which determines the error back flow. As stated in [91] this means that if:

$$|\sigma'_{l_m}(net_{l_m}(t-m))w_{l_m l_{m-1}}| > 1.0$$

for all m, then we have an exponential blow up of the error with q, while if

$$|\sigma'_{l_m}(net_{l_m}(t-m))w_{l_m l_{m-1}}| < 1.0$$

the error decreases exponentially with q (the vanishing gradient problem). The previous generalise for global error flow. Long-short term memory neural networks, which we will discuss in the next section, are not affected by this problem and thus demonstrate the ability to learn long range dependencies in the data.

3.1.4 Long Short-Term Memory Recurrent Neural Networks

In this section, we will describe the theory behind Long Short-Term Memory networks. In more detail, in Section 3.1.4.1 we will refer to a simplistic idea introducing the principle behind LSTMs. The fundamental architecture of LSTMs will be described in Section 3.1.4.2, while in what follows we will present three extensions to the basic idea: Forget Gates, Peephole Connections and the bidirectional LSTMs.

3.1.4.1 Constant Error Carousel

The Constant Error Carousel (CEC) is the main component of LSTM networks and is by itself a naive solution to the problem of the exponentially decaying error in recurrent networks. The idea is, that in order to prevent the error from changing exponentially, we can impose a constant error flow, initially for a unit j which has a feedback connection to itself, thus forming a loop. The local error backflow, according to Equation 3.23, is now

$$\delta_j(t) = \sigma'(net_j(t))\delta_j(t+1)w_{jj}$$

and thus, to enforce a constant error flow, $(\delta_j(t) = \delta_j(t+1))$, we would require that:

$$\sigma'(net_j(t))w_{jj} = 1.0 \tag{3.28}$$

By integrating the above equation we get:

$$\int \sigma'(net_j(t))w_{jj}\, dnet_j(t) = \int 1.0\, dnet_j(t)$$

$$\Rightarrow \sigma(net_j(t))w_{jj} = net_j(t) \tag{3.29}$$

That is, we arrive at the conclusion that the activation function should be linear and the activation of the node would remain constant:

$$y_j(t+1) = \sigma_j(net_j(t+1)) = \sigma_j(w_{jj}y_j(t)) = y_j(t) \tag{3.30}$$

Experimentally, this was ensured by the authors by using the identity function as the activation function, $\sigma_j : \sigma_j(x) = x, \forall x$, and by defining the weight $w_{jj} = 1.0$. This is essentially the component defined as the Constant Error Carousel (CEC) in LSTM.

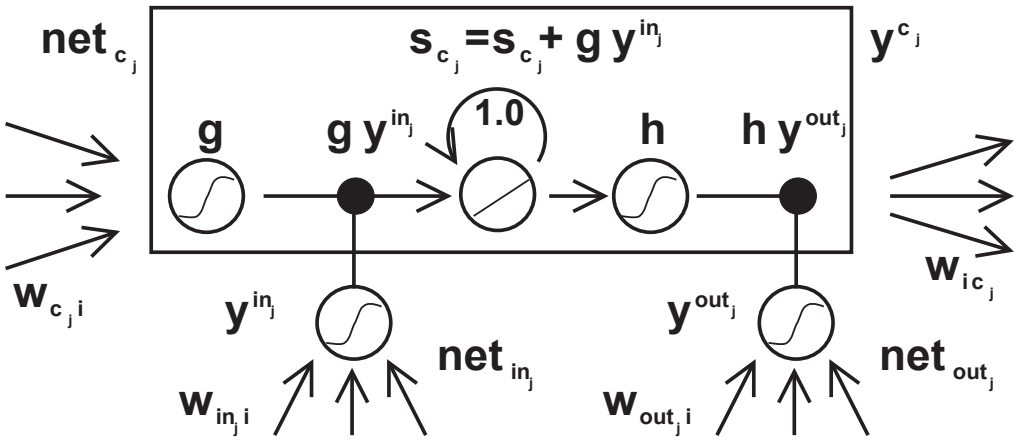


Figure 3.4: Architecture of an LSTM memory cell c_j, containing the gate units in_j and out_j. The basis of the CEC is the unit with the feedback connection with weight 1.0 (and a one time step delay). Figure from [91]

Hochreiter and Schmidhuber [91] comment on how this naive idea would present problems when other nodes and weights are involved in the network. Having such a linear node would present problems in non-trivial problems, having other incoming and outgoing weights to and from the CEC node to receive conflicting updates during training and thus making the learning process difficult.

3.1.4.2 Memory Cells and Gates

Memory cells and gate units are the proposed extension to the CEC carousel, in order to overcome the disadvantages that the CEC approach presents. A *memory cell* c_j, is defined by the following components:

- A CEC self-connected linear unit j , as described in Section 3.1.4.1
- A multiplicative input gate, introduced to protect the contents of j
- A multiplicative output gate, which protects other units from perturbation by unit j

A complete memory cell unit can be seen in Fig. 3.4. A memory cell c_j receives input from net_{c_j} (incoming synapses to the cell), the multiplicative input unit (gate) in_j and the multiplicative output unit (gate) out_j, whereas the two gates have an activation $y_{in_j}(t)$ and $y_{out_j}(t)$ respectively and are defined as follows:

$$y_{out_j}(t) = \sigma_{out_j}(net_{out_j}(t)), y_{in_j}(t) = \sigma_{in_j}(net_{in_j}(t)) \tag{3.31}$$

where the *net* inputs to the input/output gates and the memory cell are defined as:

$$net_{in_j}(t) = \sum_u w_{in_j u} y_u(t-1) \tag{3.32}$$

$$net_{out_j}(t) = \sum_u w_{out_j u} y_u(t-1) \tag{3.33}$$

$$net_{c_j}(t) = \sum_u w_{c_j u} y_u(t-1) \tag{3.34}$$

with summing over the range of u, which includes any other units connected with a synapse to the unit for which we examine the *net* input. These other units can be gate units, another memory cell or even a standard hidden node. The output of the cell c_j, $y_{c_j}(t)$ is dependent on an internal state s_{c_j}, and is defined as follows:

$$y_{c_j}(t) = y_{out_j}(t) h(s_{c_j}(t)) \tag{3.35}$$

$$s_{c_j}(0) = 0, s_{c_j}(t) = s_{c_j}(t-1) + y_{in_j}(t) g(net_{c_j}(t)), t > 0 \tag{3.36}$$

where here we have used h and g for the functions used. Function h stands for a function that scales the memory cell outputs and g for the function that squashes net_{c_j}. They are both differentiable.

As we can see in Equations 3.35 and 3.36, the activation of the input and output gates is multiplicatively involved in the calculation of the activation of the cell, $y_{c_j}(t)$ and the state $s_{c_j}(t)$. In that way, the input gate can decide when information in the cell should be changed and when not, while the output gate can decide if the information in the cell should affect nodes in outgoing paths or not. This is how this configuration avoids weight conflicts in outgoing and incoming weights to the cell. Regarding the error flow, the output gate scaling functions learn when to superimpose the error signal which is flowing backwards through the output gate. The corresponding input gate learns when to release this error flow. To clarify, when we say "protect" or "allow", we basically mean that a gate is turned "off" by having a very low activation - around zero and turned "on" when the activation is high - this in combination with the multiplicative involvement of the gates in the equations. E.g., in Equation 3.36, if the activation of the input gate is approximately $y_{in_j}(t) \approx 0$ then:

$$s_{c_j}(t) = s_{c_j}(t-1) + y_{in_j}(t) g(net_{c_j}(t)) \approx s_{c_j}(t-1)$$

It is also important for clarification, to state that a single time step refers to an entire forward (calculating activations) and backward pass (computing error signals for weights).

Memory blocks can be organised into *memory cell blocks*. A memory cell block of size S is essentially defined as a group of S cell blocks which have a shared input and output gate. A memory cell block of size 1 is a simple cell. It should also be noted that during learning, the errors arriving at cells do not get propagated further back in time although they change the incoming weights eventually. They are scaled when arriving at the input of the cell, then they remain within the cell, affecting the internal states s_{c_j}. When the error leaves the memory cell through the input get, it gets rescaled and then affects the incoming weights. This method conserves the local error flow of the CEC without presenting the vanishing gradient problem. The details of the learning process[2] appear in the appendix of [91], accompanied with a detailed description of the properties of LSTM networks. It should be noted that the learning algorithm of LSTM has the same update

[2]The learning algorithm of LSTM is largely based on the truncated BBTT which we have described in Section 3.1.3.1, while it also uses some elements from Real Time Recurrent Learning (RTRL), another training algorithm for RNNs which we have not discussed here. For more details, please see Chapter 5 of [105]

complexity as the BBTT algorithm ($O(W)$, where W is the number of weights in the network), while it is local in space and time[3]. Experiments found in the original paper focus almost exclusively on problems which contain long time lags include learning the embedded Reber grammar (a string generator typically used as a benchmark for recurrent networks) [154] and learning from noisy sequences, in order to demonstrate how such networks have the ability to store information extracted from events which are temporally distant.

3.1.4.3 Forget Gates

Gers et. al [72] point out an issue with the original architecture of LSTM networks that we have described until now (Hochreiter and Schmidhuber [91]). The characteristic of memory cells which allows information to be stored for an arbitrary duration in the CEC, can sometimes cause the states s_c to grow linearly when a time series is presented. When a continuous input stream is presented, there are cases where the cell states will grow unboundedly, saturating the output[4] of the squashing function h used for the output (Equation 3.35). This saturation can cause problems such as making the derivative of h disappear (due to the congregation of output values) and thus block incoming errors, while it will also make the output of the cell to equal the activation of the output gate, degenerating the cell to a BPTT unit. It is claimed that such problems did not appear in the original paper [91], since the cell states were reset manually after training for each sequence.

To overcome this issue, Gers et. a; [72] propose the use of what they call "forget gates" in order to learn to reset the memory blocks when it is decided that their contents are out of date and irrelevant. Forget gates replace the CEC 1.0 weight by the multiplicative activation of the forget gate y_{ϕ_j}, which is similar to the activation of the input/output gates (Equations 3.31):

$$y_{\phi_j}(t) = f_{\phi_j}\left(\sum_m w_{\phi_j m} y_m(t-1)\right) \tag{3.37}$$

where f_{ϕ_j} is a logistic sigmoid. The activation of the forget gate is used as the weight of the recurrent connection for the state calculation inside the memory cell (Equation 3.36). The new equation is:

$$s_{c_j}(0) = 0, s_{c_j}(t) = y_{\phi_j}(t) s_{c_j}(t-1) + y_{in_j}(t) g(net_{c_j}(t)), t > 0 \tag{3.38}$$

A memory cell with a forget gate is depicted in Fig. 3.5. Experiments using this modification to the LSTM model can be found in [72]. The experiments essentially use an extension of the Reber grammar, to demonstrate how forget gates can overcome failures of LSTM networks in cases where the state can not be reset manually (in continuous input streams), thus avoiding the saturation of activation functions.

[3] Local in space: the update complexity per step and weight does not depend on network size
Local in time: Storage requirements do not depend on input sequence length

[4] The output of the squashing function is considered to be saturated when many transformed values congregate at the upper/lower bound

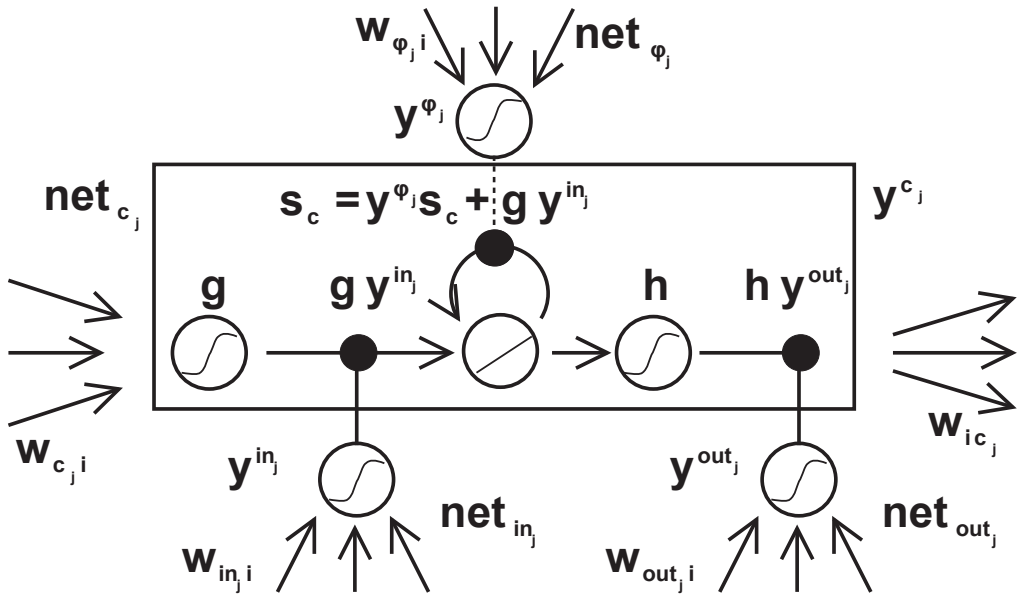


Figure 3.5: The LSTM memory cell which appears in Fig. 3.4, modified with a forget gate, the connection for which appears with a dashed line

3.1.4.4 Peephole LSTM

Gers [71], comments on how humans learn to recognise sequences of rhythmic patterns which are defined by the intervals between sub-patterns. Gers studies such patterns by extending the LSTM model with peephole connections. These connections are aimed at one potential weakness of the LSTM model. In an LSTM, each gate would receive connections from the input units and output units from other cells, but will not receive input from the CEC that it controls, due to the cell-architecture where the gate can observe only the output of the cell. The remedy proposed is the extension of the memory cell by the addition of weighted connections (peepholes) from the CEC to the gates which belong to the memory block. Thus, the gates should learn to shield the CEC from unwanted inputs during the forward pass, and irrelevant error signals during the backward pass. During learning, the peephole connections do not propagate error signals back from the gates to the CEC in order to keep the previous shielding of the CEC intact. The peephole memory cell appears in Fig. 3.6, while a detailed description of the modifications in the learning algorithm appears in [71]. Here, we will provide a description that is related to the forward pass through the network. We will follow the simplification presented in [71], where the output squashing function h is omitted (See Equation 3.35 and Section 3.1.4.2 for h).

Since there are recurrent connections now from the gates, there are two phases in updating peephole LSTM cells: The first phase, consists of calculating the activations of the input gate (y_{in}) and the forget gate (y_{ϕ}), as well as the input to the cell and the current cell state, s_c. The second phase, consists of calculating the activation of the output gate (y_{out}) and the output of the cell, y_c. The output gate update occurs after the cell state updates and is affected by the current state ($s_c(t)$) as it is itself affected by the forget gate and recent/current input (peephole connection). The *net* input for the input and forget gates (Equations 3.32 and 3.38) changes as follows:

$$net_{in_j}(t) = \sum_u w_{in_j u} y_u(t-1) + \sum_v^{S_j} w_{in_j c_j^v} y_{c_j}^v(t-1) \tag{3.39}$$

$$net\phi_j(t) = \sum_m w_{\phi_j m} y_m(t-1) + \sum_v^{S_j} w_{\phi_j c_j^v} y_{c_j}^v(t-1) \tag{3.40}$$

Essentially, another term has been added in both equations, summing over all the memory cells v in the memory block S_j (Section 3.1.4.2 for a description of memory blocks). The superscript c_j^u indicates that the parameter in question refers to the u-th memory cell of memory block j. The equations for the net input to the cell and the state calculation remain the same as in LSTM with forget gates (3.38), while the *net* input of the output gate (Equation 3.33) now has a view to the CEC by the peephole connection:

$$net_{out_j}(t) = \sum_u w_{out_j u} y_u(t-1) + \sum_v^{S_j} w_{out_j c_j^v} y_{c_j}^v(t) \tag{3.41}$$

The cell output equation also remains the same (Equation 3.36), generalised for memory blocks and by omitting the h function.

In Gers thesis [71], experiments are described which demonstrate differences in the performance of conventional vs peephole LSTM networks, namely in the following:

- Measuring spike delays (MSD), as in classifying spikes in input sequences, with the classification depending on the interval between the spikes. Here, peephole LSTM outperform LSTM.
- Generating timed spikes (GTS), as the "inverse" problem, which is obtained by switching the inputs and target outputs of MSD. In this case, the GTS setting could not be learnt by a network with no peephole connections, and
- The generation of other periodic functions such as nonlinear, triangular and rectangular waveforms, where peephole LSTM found "perfect and stable" solutions for all functions, LSTM with forget gates learnt only one function, while conventional LSTM never predicted the waveforms for more than two periods.

3.1.4.5 Bidirectional LSTM

Bidirectional LSTM (BLSTM), introduced by Graves and Schmidhuber [78] extend the bidirectional notion originally introduced for recurrent neural networks (BRNNs). As we have already discussed, LSTM networks attempt to overcome the issue observed with traditional RNNs, namely being unable to learn time dependencies longer than a few timesteps. Another weakness of traditional RNNs, is that since inputs are processed in a temporal order, they learn characteristics of the input by relating to the previous context and in general ignore future context.

Bidirectional RNNs [167, 9], present a modification to the learning procedure in order to overcome the latter issue of past and future context: They present each of the training sequences in a forward and a backward order, to two different recurrent networks respectively, which are themselves connected to a common output layer. Thus, the BRNN is aware of both future and past events in relation to the current timestep. BRNNs present improved results compared to RNNs in applications such as protein structure prediction [10, 35] and speech processing [69, 166, 78]. The

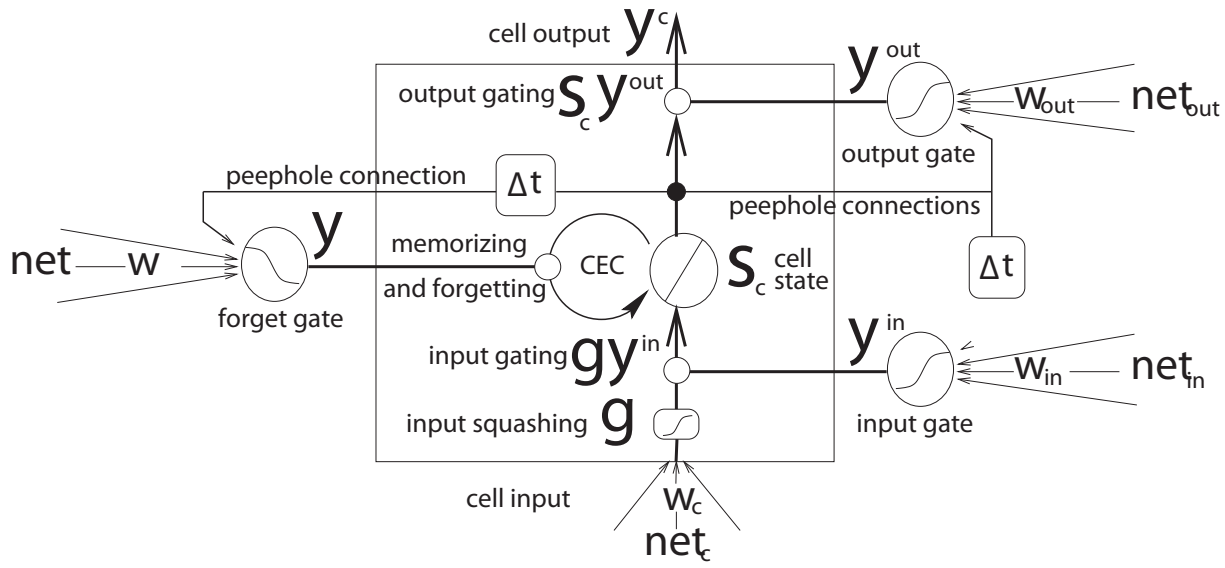


Figure 3.6: An LSTM memory cell with forget gates and peephole connections, with the latter connecting the cell state s_c to the gates of the same memory cell. Figure from [71].

idea is directly expanded for LSTM, and results presented for speech processing in [78], support that Biderctional Long Short-Term Memory (BLSTM) networks outperform unidirectional LSTM. The pseudocode for training LSTM [78] which is essentially quoted from the the bidirectional training procedure for RNNS with BPTT [166] is presented in Algorithm 1. It is assumed that the input sequence for training contains events from time t_0 to t_1.

Algorithm 1: Training a Bidirectional (Recurrent) Network (Outline)

```
1  begin
2      Forward Pass: feed input, determine output;
3          Do forward pass for forward states from t_0 to t_1;
4          Do forward pass for backward states from t_1 to t_0;
5          Do forward pass for output layer;
6      Backward Pass: Calculate error derivatives;
7          Do backward pass for output layer;
8          Do backward pass for forward states from t_1 to t_0;
9          Do backward pass for backward states from t_0 to t_1;
10     Update Weights;
11 end
```

3.1.5 Discussion

We will focus the discussion around recurrent neural networks on the last model that we have referred to, the LSTM neural networks, as they are the refinement used in this project. Specifically, the LSTM seem to be very promising in tasks which are composed of sequential and temporal inputs which do expose long range temporal dependencies. This is particularly interesting for our task, such temporal patterns and dynamics emerge in human emotional expressions (Section 2.2.5). Furthermore, for continuous emotion recognition the "emotional history" [199, 84] kept by such networks in terms of the long range dependencies learnt can be beneficial towards the recognition of ambiguous and subtle emotional states. Also, the non-linear squashing functions enable

the network to learn from non-linearly correlated data, while on the down side inherent neural network problems such as a tendency to overfit the data do manifest in LSTMs. We denote that LSTMs perform dynamic, sequence learning and are thus suitable for learning from sequences of features, such as our audiovisual segments.

We should state that a common use for LSTMs is for speech [76] and handwriting recognition [77, 79], while they have also been used for predicting protein localisation [177] and robotic surgery [123]. For emotion recognition though, they have up been only used in relation to speech data in [199] up until now.

3.2 Support Vector Machines

Support Vector Machines (SVM), introduced in 1995 by Vapnik [188] are a set of learning techniques used for both classification and regression. Given a set of input data a a set of points in an n-dimensional space (e.g. for binary classification, the set of points that belongs to class 1, same for class 2), a SVM will construct a hyperplane which separates the input sets in the dimensional space. In 2D, this is similar to how a perceptron unit is able to learn linearly separable problems. A SVM though, using quadratic optimisation, returns the optimal such hyperplane that maximises the margin between the two data sets. Also, by mapping the input space to another, typically higher in dimensionality, space (called the feature space) non-linearly solvable problems can be solved. The idea has been expanded for regression by Vapnik [190], as we will discuss in the following sections. Furthermore, it is interesting to note that SVMs generate convex optimisation functions, which by definition have only one optimum, thus escaping the problem of being stuck in local optima during learning (e.g. in neural networks training when using gradient descent). Finally, SVMs use the inductive principle of *Structural Risk Minimisation* [191] in order to avoid overfitting to the training data, by balancing the structural complexity of the model against the accuracy of correctly classifying the training data. It should be noted that although we use regression in the continuous emotion recognition part of the project, SVMs for classification are described in considerable detail since firstly most of the concepts generalise directly for regression and secondly, we use SVMs for discrete emotion recognition.

3.2.1 Support Vector Classification

In this section, we will refer to a binary classification problem in order to introduce Support Vector Machines. Consider a binary classification, where training examples are of the format $(\mathbf{x_i}, y_i)_i$ for the i-th pattern pair, where $\mathbf{x}_i$ are the input vectors and $y_i = \pm 1$ the target or label classification of the corresponding $\mathbf{x}_i$. Here, we assume that there are m training examples:

$$\mathcal{D} = \{(\mathbf{x_1}, y_1)_1, (\mathbf{x_2}, y_2)_2 \dots (\mathbf{x_m}, y_m)_m\}, \mathbf{x_i} \in \mathbb{R}^n, i \in 1 \dots m, y_i \in \{-1, 1\}$$

The $\mathbf{x}_i$ vectors define what we call the *input space*. As we have mentioned in the introduction, a SVM will find a hyperplane which separates the x_i vectors which have a corresponding $y_i = 1$ from those who have a corresponding $y_i = -1$

According to Statistical Learning Theory [189, 188], two observations that motivate the use of

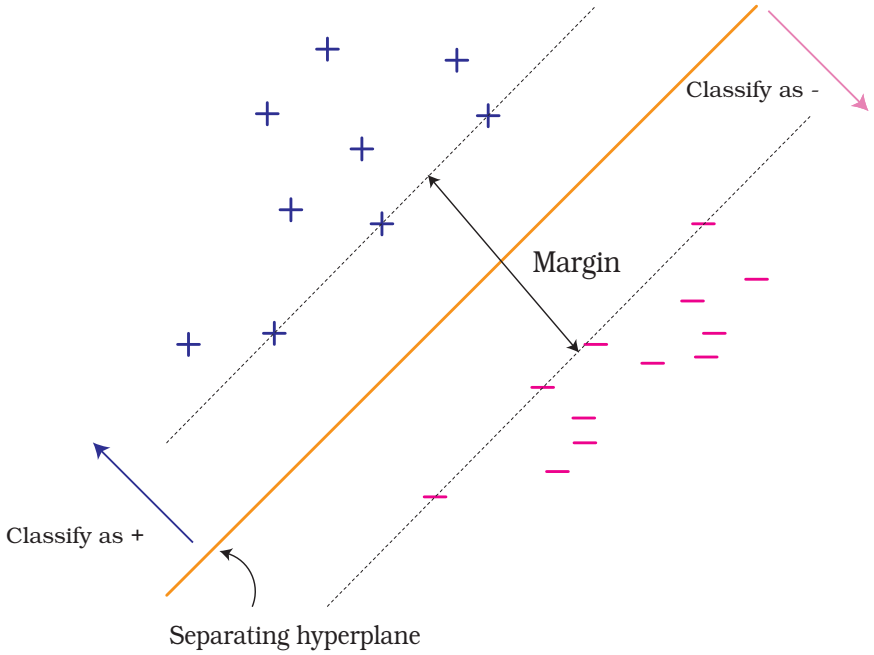


Figure 3.7: A binary classification problem and the optimal separating hyperplane

SVMs relate to the theoretical bounds on the generalisation error (i.e. the theoretical error that the model's predictions will have on unknown data sets) of a model:

Observation 1 *The generalisation error upper bound does not depend on the dimensionality of the space*

Observation 2 *The bound is minimised by maximising the margin, which referring to the binary classification problem, is the minimal distance between the separating hyperplane and the closest points (x_i) for each of the two input sets (± 1).*

The second point refers to the maximisation of the margin. In Fig. 3.7, we can see an illustration of the margin in the binary classification problem. Any other hyperplane which correctly separates the two data sets has a smaller margin from the hyperplane seen in the figure, thus theoretically the generalisation error bound would be larger had the learning technique selected any other hyperplane. It should be noted, that the name of the method is related to the name which is given to the points which fall directly on the dashed, parallel to the hyperplane, lines in Fig. 3.7, that is the points that are closer to the hyperplane than any other point in each set. These points are called *support vectors*.

In any dimensional space, the equation of a hyperplane can be written as:

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

where vectors in the equation are in the supposed dimensional space. Essentially $\mathbf{w}$ represents the normal vector to the hyperplane, and $\mathbf{b}$ the bias or the shift of the plane.

The decision function that relates to the classification of an input point $\mathbf{x}$ would be:

$$D(x) = sign(\mathbf{w} \cdot \mathbf{x} + b)$$

Since the sign function is invariant when scaling both **w** and **b** by a positive constant, the support vectors are fixed to be at a distance of 1 from the hyperplane:

$$\mathbf{w} \cdot \mathbf{x}_1 + b = +1$$

$$\mathbf{w} \cdot \mathbf{x}_2 + b = -1$$

where $\mathbf{x}_1$ is a support vector for the first class and $\mathbf{x}_2$ a support vector for the second. By subtracting the equations we get:

$$\mathbf{w} \cdot (\mathbf{x}_1 - \mathbf{x}_2) = 2 \tag{3.42}$$

Considering that we want to calculate the distance between the two data sets in relation to the hyperplane, what we actually want is the projection of the vector $(\mathbf{x}_1 - \mathbf{x}_2)$ on the unit normal vector for the hyperplane (which is perpendicular to the plane), i.e. $\frac{\mathbf{w}}{\|\mathbf{w}\|}$. This projection is given by the vector:

$$cos\theta(\mathbf{x}_1 - \mathbf{x}_2)$$

where θ is the angle between the two vectors. Since:

$$cos\theta = \frac{(\mathbf{x}_1 - \mathbf{x}_2) \cdot \mathbf{w}}{|(\mathbf{x}_1 - \mathbf{x}_2)\| \cdot \mathbf{w}|}$$

and from Equation 3.42, we get that the margin, γ is:

$$\gamma = \frac{2}{\|w\|}$$

Thus, maximising the margin is equivalent to minimising $\Phi(\mathbf{w})$[5]:

$$\Phi(\mathbf{w}) = \frac{1}{2}(\mathbf{w} \cdot \mathbf{w}) \tag{3.43}$$

subject to the constraints:

$$y_i(\mathbf{w} \cdot \mathbf{x} + b) \geq 1 \tag{3.44}$$

which essentially state that no data point should fall into the margin. This could have been expressed by the following two equations:

$$(\mathbf{w} \cdot \mathbf{x_i} + b) \geq 1, x_i \text{ in first class}$$

$$(\mathbf{w} \cdot \mathbf{x_i} + b) \leq 1, x_i \text{ in second class}$$

By combining Equations 3.43 and3.43, we get the following objective function by using the Lagrange multipliers (α_i, See [20], Chapter 3):

$$L(\mathbf{w}, b) = \frac{1}{2}(\mathbf{w} \cdots \mathbf{w}) - \sum_{i=1}^{m} \alpha_i [y_i((\mathbf{w} \cdot \mathbf{x}_i) + b) - 1]$$

By finding the stationary points where the partial derivatives of L are equal to zero, we get:

[5] where we used the inverse of γ and also squared the norm in order to remove the square root and simplify the problem.

- For $\frac{\partial L}{\partial b} = 0$, we get

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

- while for $\frac{\partial L}{\partial w} = 0$:

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

by substituting back to L and changing to the dual of the problem (which should be maximised in contrast to the primal form of the problem which was a minimisation problem):

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdots \mathbf{x}_j)$$

subject to:

$$\alpha_i \geq 0$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

This reveals that essentially the maximum margin hyperplane is a function of only the support vectors (something that is intuitively reasonable).

The new decision function for an input $\mathbf{z}$ is:

$$D(\mathbf{z}) = sign(\sum_{j=1}^{m} \alpha_j y_j (\mathbf{x}_j \cdot \mathbf{z} + b))$$

3.2.1.1 Dealing with non-separable datasets

In many situations, the data are non-separable. For the 2D situation, the separating hyperplane perfectly separates the two classes in Fig. 3.7 because the data are linearly separable. We can though switch an element of the one class with an element of the other in a way that no line would exist to perfectly separate the data. We will present an example of such a problem in the next paragraphs.

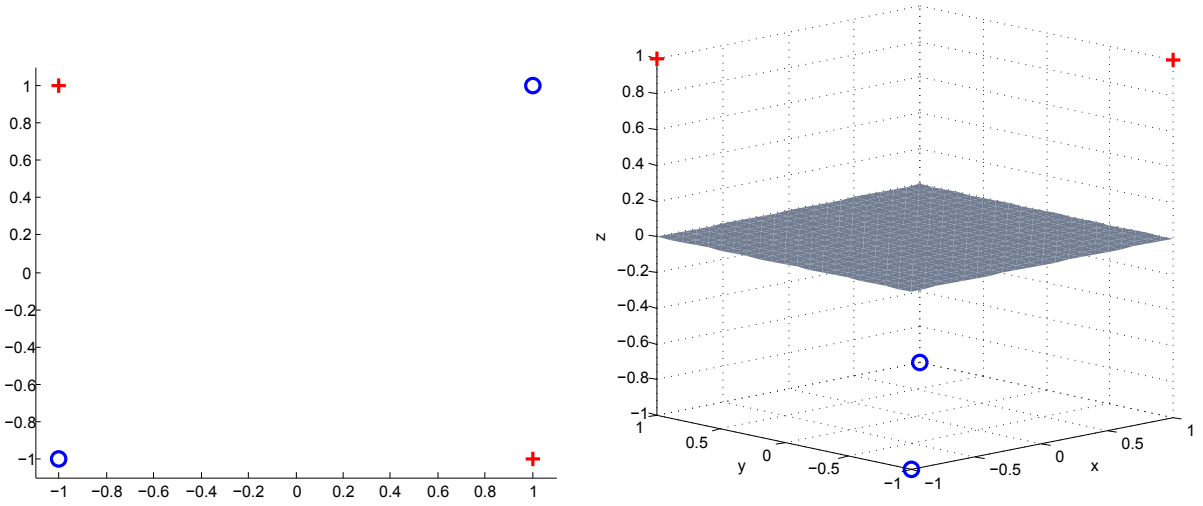
In order to overcome this issue, we will refer to the first observation made when discussing statistical learning theory (observation 1): *The theoretical generalisation error bounds do not depend on the dimensionality of the space*. Thus, the idea is to project the input space to another higher dimensional space called the feature space, by changing the inner product in which the x_i points appear, to another operation called the *kernel*:

$$\mathbf{x}_i \cdot \mathbf{x}_j \rightarrow K(x_i, x_j)$$

implicitly, a mapping function ϕ has been used to map the input space to the feature space:

$$\mathbf{x}_i \rightarrow \phi(\mathbf{x}_i), \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) = K(x_i, x_j)$$

thus, the kernel is the inner product between the mapped points in the feature space. It should be noted that the feature space should be a *Hilbert space* (or *inner product* space). Also, that the idea



(a) (b)

Figure 3.8: (a) The XOR problem: no hyperplane (line in 2D) exists that can perfectly separate the one class set from the other (b) Mapping the 2D XOR problem in 3D allows the perfect separation of the classes

of using a kernel function to enable procedures to be performed in the input space (following the method of Aizerman et al. [2]) rather than in the highly dimensional feature space (by computing the ϕ mapping functions and then the inner product) is a way to deal with the curse of dimensionality problem [18] (which here refers to an exponential increase in size when increasing the dimensionality).

To provide a trivial but very intuitive example on how mapping to a higher dimensional space can deem a problem separable and solvable in the feature space, we can consider learning the classic XOR problem, which is well known not to be linearly separable (and was one of the disappointments of original perceptron units as they were unable to learn it). By adding a 3rd dimension and thus mapping the 2D input space to a 3D feature space, the problem becomes perfectly solvable and the data sets are perfectly separable. This is essentially like pushing the one class away and pulling the other one closer, in the Z axis (Fig. 3.8).

Examples of kernels K are radial basis functions[6] (RBF) (where the value depends on the distance between $\mathbf{x}, \mathbf{x}'$):

$$K(\mathbf{x}, \mathbf{x}') = e^{-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}}$$

(here the parameter σ refers to the width of the function) and polynomial kernels:

$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x_i} \cdot \mathbf{x_j} + 1)^d$$

while specific kernels for strings or graphs/networks have been developed. The kernel should satisfy Mercer's condition. It is though enough to ensure that the kernel is positive semi-definite:

$$\sum_{i,j} K(x_i, x_j) c_i c_j \geq 0$$

[6] Note that the RBF implies the lift to a space of infinite dimension

where c_i and $_j$'s are real numbers.

It should be noted that a suggestion for dealing with multi-class problems, where more than two classes exist, is to generate a directed acyclic graph (DAG) where each node will decide a binary classification problem.

3.2.1.2 Soft Margins and Slack Variables

Typically, real life datasets contain noise. A SVM classifier might not generalise well to unknown data, if it fits the noise in the training dataset. That is why a *soft margin* can be introduced, in order to relax the *hard margin* constraint [38]. In other words, some examples are allowed to be misclassified. The objective function is balancing the trade-off between the maximising the margin and minimising the training error. In order to account for the degree of misclassification, slack variables (ξ_i) are introduced. Thus, Equation 3.44 now becomes:

$$y_i(\mathbf{w} \cdot \mathbf{x} + b) \geq 1 - \xi_i \tag{3.45}$$

The objective function is increased by attaching a term which penalises non-zero slack variables. If we want to minimise the sum of errors $\sum_{i=1}^{m} \xi_i$ then the objective function (was Equation 3.43) for minimisation becomes:

$$\Phi(\mathbf{w}, \xi) = \left(\frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{m} \xi_i \right) \tag{3.46}$$

where C is a constant, subject to the constraint proposed in Equation 3.45. It is noted that C controls the trade-off between our tolerance to training errors and the generalisation of the model. As C increases, the error term increases and thus less errors are tolerated. There is an inherent risk of overfitting to the training data though, in this way. Again, the Lagrange multipliers are used, and in the resulting dual problem, the slack variables vanish. The constant C (the soft margin parameter) remains, and appears as an additional constraint on the Lagrange multipliers. It should be noted that the optimal value of C should be found by experimentation, since there is no method to determine it theoretically. Additionally, it should be noted that the distance of a point in the feature space to the separating hyperplane can be normalised in a way that probabilities for a classification decision can be produced by the model, with algorithms which take into account the distance of a point from the hyperplane. Non-linear functions should be used with care since they could turn the optimisation problem into a non-convex problem, generating local optima.

3.2.2 Support Vector Regression

The concept of Support Vector Machines for classification has been also expanded for regression tasks [38]. Essentially, the approach maintains all the basic elements that characterise Support Vector Machines for classification purposes. A non-linear function is learnt by the model in a mapped feature space, induced by the kernel used. The advantage of having a convex function to optimise is preserved in the regression task for SVM. The goal is to optimise the generalisation bounds for regression. To do so, a loss function is defined, which is essentially used to weight the actual error of the point with respect to the distance from the correct prediction. Some commonly used loss functions appear in Fig. 3.9. The quadratic loss function (Fig. 3.9(c)) corresponds to the

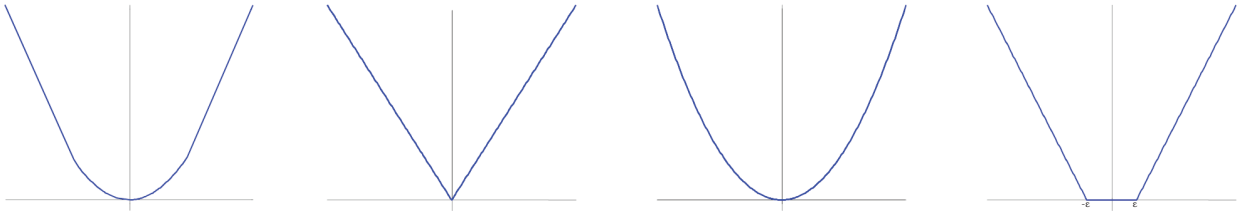


Figure 3.9: Commonly used loss functions: (a) Huber (b) Laplace (c) Quadratic (d) ϵ-insensitive

least squares error. The Laplacian loss function 3.9(b) is less sensitive to outliers, while the Huber loss function 3.9(a) is claimed to have optimal properties when the underlying distribution of data is not known [70]. The ϵ-insensitive loss function, introduced by Vapnik, is an approximation to the Huber loss function 3.9(a) and enables a more reliable generalisation bound [41]. This is due to the fact that unlike the Huber and quadratic loss functions, where all the data will be support vectors, the support vectors can be sparse with the ϵ-insensitive loss function, and (reasonably) sparse data representations have been shown to reduce the generalisation error [193]. More on loss function can be found in Chapter 3.3 of [164]

3.2.2.1 ϵ-insensitive Regression

We will refer to ϵ-insensitive regression, which uses the ϵ-insensitive loss function. In essence, the idea is that all points that fall within the ϵ-band have a zero cost. The ones outside the band have a cost assigned which is relative to their distance which is measured by the variables ξ. An example of non-linear ϵ-regression appears in Fig. 3.10.

Linear

For linear ϵ-insensitive Regression, we assume the following dataset:

$$\mathcal{D} = \{(\mathbf{x_1}, y_1)_1, (\mathbf{x_2}, y_2)_2 \dots (\mathbf{x_m}, y_m)_m\}, \mathbf{x_i} \in \mathbb{R}^n, i \in 1 \dots m, y_i \in \mathbb{R}$$

and now, the functional (Equation 3.46) which is to be minimised becomes:

$$\Phi(\mathbf{w}, \xi) = \left(\frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{m} \xi_i + \hat{\xi}_i \right) \tag{3.47}$$

, where now, ξ_i and $\hat{\xi}_i$ (positive slack variables) represent upper and lower constraints for the outputs of the model:

$$y_i - (\mathbf{w} \cdot \mathbf{x_i} + b) \leq \epsilon + \xi_i$$

$$(\mathbf{w} \cdot \mathbf{x_i} + b) - y_i \leq \epsilon + \hat{\xi}_i$$

Given that the ϵ-insensitive loss function is defined as:

$$L_\epsilon(y) = \begin{cases} 0 & |f(x) - y| < \epsilon \\ |f(x) - y| - \epsilon & otherwise \end{cases} \tag{3.48}$$

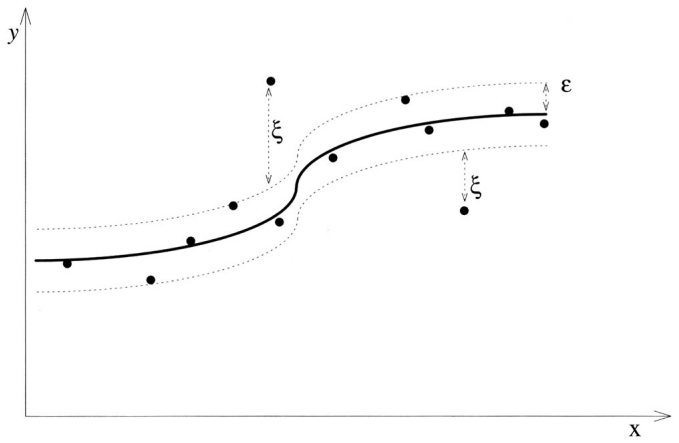


Figure 3.10: The ϵ-insensitive band for a non-linear regression function. There is no cost for the points within the band . Figure from [41]

the Lagrange multipliers and then the dual objective problem is formulated:

$$W(\alpha, \hat{\alpha}) = \sum_{i=1}^{m} y_i(\alpha_i - \hat{\alpha}_i) - \epsilon \sum_{i=1}^{m}((\alpha_i + \hat{\alpha}_i)) - \frac{1}{2}\sum_{i,j=1}^{m}(\alpha_i - \hat{\alpha}_i)(\alpha_j - \hat{\alpha}_j)K(x_i, x_j)$$

subject to:

$$\sum_{i=1}^{m}\hat{\alpha_i} = \sum_{i=1}^{m}\alpha_i$$

$$0 \leq \alpha_i \leq C, 0 \leq \hat{\alpha_i} \leq C$$

and the function modelling the data is now:

$$f(\mathbf{z}) = \sum_{i=1}^{m}(\alpha_i - \hat{\alpha_i})\mathbf{x_i} \cdot \mathbf{z} + b$$

for an input $\mathbf{z}$. The procedure is similar for other loss functions.

Non-Linear

Similarly to the non-linear kernel mapping described for Support Vector Machines for Classification (Section 3.2.1.1), it is common to adopt a non-linear model to fit certain datasets. Following the same fashion as SVM for Classification, a kernel can be used to induce a non-linear mapping to a high dimensional feature space where regression is to be performed. The approach, again by using the ϵ-insensitive loss functions, is inferred by the typical procedure, using the Lagrange multipliers and referring to the dual formulation of the problem. Finally, the regression function that models the data is:

$$f(\mathbf{z}) = \sum_{i=1}^{m}(\alpha_i - \hat{\alpha_i})K(\mathbf{x}_i, \mathbf{z}) + b$$

for an input $\mathbf{z}$. Again, function that have been already discussed in Section 3.2.1.1 can be used to replace the inner product operations (kernel).

3.2.2.2 Discussion

From Support Vector Machines, it is important to note that the convex optimisation function guarantees the optimal solution with respect to the parameters. Furthermore, the minimisation of the structural risk by the maximisation of the margin implies a further improvement on the generalisation of the system, as it avoids overfitting by not relying only on the training error to stop the learning. Furthermore, there is an inherent flexibility in the dimensional space used, while avoiding the curse of dimensionality and being able to conform to non-linearly correlated features and targets due to the non linear mapping. Also important, SVMs can generate probabilities for their classification decisions, while finally it is significant to note that all research work in continuous emotion recognition applied SVRs [199, 117, 101].

3.3 Log-linear Models & Conditional Random Fields

The task of assigning labels to sequences of observations is a very common learning task in fields such as bioinformatics and computational linguistics and speech recognition tasks [52, 153, 124]. A simple example is when a sequence of words is considered as the input, and a trained model would output the sequence of tags which indicate the appropriate part-of-speech (POS) tag (or label) for the corresponding word. In emotion recognition, the typical problem relates to outputting the label that corresponds to the sequence of observations, typically the features per frame of the segment to be tagged. Obviously, in continuous recognition one tag per observation should be generated while in discrete one tag per sequence.

Hidden Markov Models (HMMs) are one of the typically used methods for such labelling tasks. HMMs are generative probabilistic models that define the joined probability distribution, $p(X, Y)$ where X and Y are random variables which range over the observation and corresponding label sequences respectively. Without any independency assumptions, the task is typically intractable since all possible observation sequences should be enumerated. Thus, some independence assumptions are adopted by HMMs: The observation element at each time t only depends on the current state (label in this case) at that time t. That is, the states or values for previous time steps in the sequence do not have any influence. It is a fact though, that many real-life observation sequences manifest long range dependencies in their observation sequences [194], a fact that has been a motivation for the development of techniques such as the long short-term memory recurrent networks we described in Section 3.1.4.

Conditional models are frameworks which overcome some of the issues that HMMs manifest. Ideally, inference should still be tractable as in HMMs but unwanted independency assumptions should be limited or non-existent. These models manage the latter by selecting the proper label sequence y which maximises the conditional probability $p(y|x)$ where x is a novel observation sequence. The conditional nature of the model drops any independency assumptions regarding the sequences x.

Conditional Random Fields [108] (CRFs) are a probabilistic framework for labeling (and segmenting) sequential data, which is based on Conditional Models. A CRF is a form of undirected graphical model [194] [7] which defines a log-linear distribution over the label sequences given an obser-

[7] We adopt the characterisation of CRFs as undirected graphical models, as described in [194]. CRFs have been also

vation sequence. As we have mentioned, the main advantage against HMMs is the relaxation of independence assumptions, while also overcoming the label bias problem [108] (i.e. biasing towards states which have fewer successor states). It has been shown that CRFs outperform HMMs and other models such as maximum entropy Markov Models (MEMMs) on many sequence labelling tasks [108, 147, 169].

3.3.1 Conditional Random Fields as Undirected Graphical Models

In order to define conditional random fields as undirected models, we firstly present the formal definition of random fields, along with the formal definition of the Markov properties:

Definition 1 *Assuming that we have an undirected graph $G = (V, E)$ and a set of random variables $V = \{x_1, \ldots, x_n\}$, a Markov Random Field [36] is formed if the following (equivalent) Markov properties are satisfied:*

- **Pairwise Markov Properties***: Any two variables which are non-adjacent are considered conditionally independent, given all the other variables:*

$$X_u \perp\!\!\!\perp X_v | X_{V \setminus \{u,v\}} \quad \text{if } \{u, v\} \notin E$$

- **Local Markov Property***: Given the neighbours (directly adjacent variables) of a variable X_u, the variable is conditionally independent of all other variables:*

$$X_v \perp\!\!\!\perp X_{V \setminus (\text{neighbours}(v) \cup v)} | X_{\text{neighbours}(v)}$$

- **Global Markov Property***: Any two subsets $A \subseteq V$, $B \subseteq V$ are conditionally independent given a separating subset S (every path from a node in A to a node in B passes through a node in S):*

$$X_A \perp\!\!\!\perp X_B | X_S$$

Building on the definition of random fields, a CRF can be viewed as an undirected graphical model of a Markov random field which is globally conditioned on the random variable representing observation sequences, X. Given a graph $G = (V, E)$, with each node $u \in V$ corresponding to a random variable Y_u, with each Y_u obeying the Markov property w.r.t. G, then (Y, X) is a conditional random field. While the theory does not limit the structure of the graph, when modelling sequences the first-order chain is typically used (Fig. 3.11).

3.3.2 Maximum Likelihood and Conditional Likelihood

The entire set of log-linear models are based on the principle of maximising the likelihood, which we will briefly refer to in this section. Consider a random sample drawn from a distribution which belongs in a fixed set (or family) of probability distributions f, which dependent on the parameter set θ. The sample is considered to be a training set of n examples, $\mathbf{x} = \{x_1, x_2 \ldots x_n\}$. Assuming that

been described as undirected factor graphs.

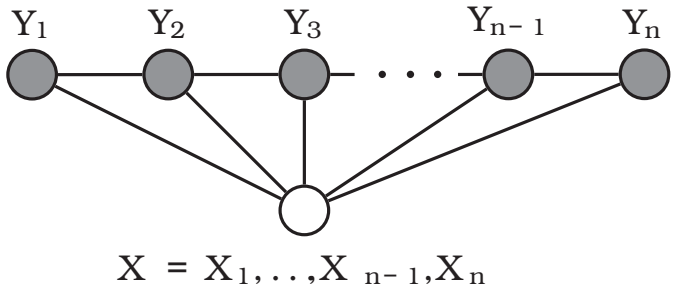


Figure 3.11: Graph of a linear chain-structured CRF. The shaded variables Y are generated by the model. Each Y_i is dependent on the entire sequence of observations, X.

the examples are independent, the probability of the set is the product of the probabilities of each example belonging in the training set[8]:

$$f(x_1,\ldots,x_n;\theta) = \prod f_\theta(x_j;\theta)$$

The goal is then, to find the proper distribution $\hat{\theta}$ from the family of distributions such that:

$$\hat{\theta} = \arg\max_\theta \prod_{i=1}^{n} f_\theta(x_i;\theta)$$

with the likelihood function defined as:

$$L(\theta;x_1,\ldots,x_n) = f(x_1,\ldots,x_n;\theta)$$

and restating the goal equation:

$$\hat{\theta} = \arg\max_\theta L(\theta;x_1,\ldots,x_n)$$

with $\hat{\theta}$ being called the maximum likelihood estimator of θ. It is noted that by taking the *log* of the product, the product becomes a sum. That is why models depending on this principle are called log-linear models, since they become linear when the logarithm of the model is taken, even if the model itself is non-linear. For Conditional Likelihood, we assume that the distribution of y is conditional on x, and also defined by the parameter θ. Given training pairs (x_i, y_i), the inference principle of maximum conditional likelihood is:

$$\hat{\theta} = \arg\max_\theta \prod_{i=1}^{n} P(y_i|x_i;\theta)$$

where now we do not need to assume that the x_i are independent in order to justify the product but we need to assume that y_i are independent conditional on x_i. We also assume that we do not want to predict the x_i.

[8]The semicolon in expressions such as $p(a|b;\theta)$ denotes that the item on the right of the semi colon, here θ is a parameter and not a conditional random variable, unlike a and b which are random variables. In a Bayesian framework where Bayesian analysis is conducted, parameters are viewed as random variables and thus expressions such as $p(\theta|a)$ or $p(a|\theta)$ are allowed. This is the notation used in [62] while in [194] a comma is used instead.

3.3.3 Logistic Regression

Logistic Regression is the simplest form of a log-linear model. The model is defined as:

$$p = p(y|x;\alpha,\beta) = \frac{1}{1 + exp - [a + \sum_{j=1}^{d} \beta_j x_j]}$$

where $j \in \{1,\ldots,d\}$ iterates over the features of a single example with a dimensionality d, while i is used to index over training examples, $i \in \{1,\ldots,n\}$. The log of the model is taken. We will not discuss this model further than just mentioning some of its advantages:

- In contrast to Naive Bayes classifiers, correlated features don't lead to problems.
- Can give well-calibrated probabilities (meaningful conditional probabilities).
- Can handle unbalanced training data.

It should be noted that the main difference with SVM (Section 3.2.2) is that while SVMs use objective functions for optimisation when training, logistic regression and log-linear models in general are based on the maximum likelihood principle.

3.3.4 Feature Functions

A feature function is considered as a function which takes into account the input space X and the label space Y. Formally, it is a mapping:

$$F_j : X \times Y \rightarrow \mathbb{R}$$

which is considered to express some characteristic of the data set. A weight accompanies this function in order to capture whether the characteristic expressed does occur in the data set frequently or not. Often, the feature functions are binary functions (presence/absence indicators). In part-of-speech tagging, such feature functions might be, e.g. "If the input starts with a capital letter and the tag for the previous output is not a verb then current tag is noun".

For linear-chain CRFs, according to the definition of conditional independence and considering the structure seen in Fig. 3.11, each feature function can depend only on pairs of adjacent y_is and the set of x_is (for more on this discussion see Section 2 of [194]).

3.3.5 The Conditional Random Fields Model

Considering x as a training example and y one of the possible labels, any log-linear model generalises the logistic regression model and assumes that:

$$p(y|x;w) = \frac{exp[\sum_j w_j F_j(x,y)]}{Z(x,w)}$$

where Z is a normalisation function (called the partition function):

$$Z(x,w) = \sum_{y'} exp[\sum_j w_j F_j(x,y')]$$

for all possible labels y'. The label predicted by the model is now:

$$\hat{y} = argmax_y p(y|x;w) = argmax_y \sum_j w_j F_j(x,y)$$

with each $F_j(x,y)$ being the feature function we have described above.

For each of these feature functions a single weight exists (w_j), and the combination $\sum_j w_j F_j(x,y)$ is able to take any positive or negative real value to capture the occurrence of the characteristic described by the feature function in the dataset. By getting the exponent of such a combination and dividing by the normalising factor we are attaining a value between 0 and 1 (which makes them valid probabilities).

A Conditional Random Field (CRF), applies the concept of log-linear models to a task which requires dealing with sequential data. The notation $\bar{x}$ and $\bar{y}$ will correspond to the input and output sequences respectively. Typically, there are n input words and n output tags. A CRF is defined as:

$$p(\bar{y}|\bar{x};w) = \frac{exp[\sum_j w_j F_j(\bar{x},\bar{y})]}{Z(\bar{x},w)}$$

While each feature function F_j is now defined as the sum across the sequence from $i = \{1,\ldots,n\}$, where the length of the sequences is n:

$$F_j(\bar{x},\bar{y}) = \sum_i f_j(y_{i-1}, y_i, \bar{x}, i)$$

As we have previously mentioned, feature functions with linear-chain CRFs depend on adjacent outputs (y_{i-1}, y_i) and can range over the entire input sequence $(\bar{x})$. Examples of such feature functions for POS tagging can be "if previous output tag was NOUN", "if current output tag is NOUN and previous input word is John". For continuous emotion recognition specifically, the input sequence would consist of frames, each frame containing features of a certain dimensionality and the output sequence would map to a (quantised) valence or arousal ground truth value. Feature functions could consider the whole sequence of frames (our input which would consist of a set of features for each frame) and the current and previous quantised ground truth value.

The goal of training a CRF is the search for the weight w that gives the best prediction:

$$\hat{y}^* = argmax_{\hat{y}} p(\hat{y}|\hat{x};w)$$

There are many training techniques for CRFs, as for example training by stochastic gradient descent (a technique commonly used with many classifiers such as neural networks as we have seen in Section 3.1.4), the Collins perceptron or Gibs sampling. For more on these techniques, please refer to [62].

3.3.6 Discussion

Our discussion on CRFs and the fundamental principles behind the theory has been brief. CRFs have been used in continuous emotion recognition [199], where the corresponding valence/arousal values are quantized into levels, and the corresponding CRF models are trained based on the assignment of a label to each level. The advantages over HMMs seem important, and the lack of any

independency assumptions for the input can again generate something analogous to the "emotional history" we mentioned when discussing LSTMs. It is important to note that CRFs deal with sequence learning, fitting the continuous emotional recognition scenario. It is part of our future work to experiment with using CRFs in multi-modal continuous emotion recognition. On the other hand, CRFs do seem to especially fit speech processing and since they operate with labels instead of real numbers, the emotion dimensions need to be quantized.

We have already referred to some interesting tutorials on CRFs. The one is by Charles Elkan given at CIKM'08 [62] which presents a general overview of log-linear models and CRFs, providing proofs and derivations of principles and training techniques, while also intuitive and clear examples. An introductory tutorial is Hanna M. Wallach's [194], while other tutorials and selected papers are presented in Chapter 6 of [62].

3.4 Discussion

This chapter has provided with a description of some of the most important techniques with respect to the field of continuous emotion recognition. Comparing across the techniques, we have two dynamic methods which deal with sequence learning, LSTMs and CRFs, and one static technique: SVM (both for classification and regression). As far as continuous emotion recognition is concerned, sequence learning has been considered an important characteristic [199, 84] due to the "emotional history" kept, allowing a learner to predict the next emotional state based not only on the current observation but also on the preceding observations, thus modelling the temporal dynamics of human emotion expression. It is noted that due to the bidirectional LSTMs and the ability of CRF feature functions to range over the entire input sequence, not only *past context* can be learnt (i.e. the emotional history) but dependencies on future events can be learnt as well. Furthermore, continuous emotion recognition deals with real values, and thus regression is required for proper prediction. The disadvantage of CRFs here is that they operate on labels. Thus, discrete labels need to be assigned to a set of quantized levels in a continuous dimensional emotional space. LSTMs and SVR both deal with regression on real values and they can be applied directly to continuous emotion recognition.

An important aspect is the ability to capture non-linear correlations which manifest in multiple cues for emotion recognition. The non-linear functions provided in LSTMs and the kernel mapping into feature space with SVRs allow non-linear patterns to be learnt by the technique. CRFs are also able to model such non-linear dependencies [64]. Training with SVM does not only rely on the training data error in order to stop training; the maximisation of the margin and thus the minimisation of the structural risk can be beneficial to problems of overfitting, while the convex optimisation function used presents no local optima problems. On the other hand, training algorithms for neural networks in general do manifest problems of overfitting and getting stuck in local optima. For CRFs, this depends on the algorithm used for training, e.g. a gradient descent algorithm can be used and may expose the aforementioned problems for neural networks. Overall, the issue of developing classification and regression methods specifically designed for emotion recognition, dealing with non-linear correlations and unsynchronised streams of data is still an open problem.

Chapter 4

Segmentation & Feature Extraction

We will now refer to the process of segmenting the audiovisual material and thus producing the final ground truth along with the corresponding audiovisual segments. In order to achieve this, we firstly perform a first stage of pre-processing for the coder annotations. Particularly, the main stages presented in this chapter, which in total consist of the entire segmentation and feature extraction stage, are as follows:

- **Annotation Pre-processing:** In this stage, presented in Section 4.1, the annotations for the audiovisual sessions will be pre-processed. The goal is to transform the data in order to deem them appropriate for the segmentation of videos based on the processed annotations. Our work here will relate to determining normalisation procedures and extracting statistics from the data in order to evaluate the outcome of the segmentation.
- **Segmentation:** In this stage (Section 4.2) the actual audiovisual material will be segmented according to the pre-processed annotations. The extraction regards the generation of both negative and positive segments of audiovisual material, with a goal of extracting a temporal window which covers an offset before and after the expression of the desired emotional state.
- **Feature Extraction**: After the sessions are segmented according to the previous stage, the features which will be used in order to train the classifiers are extracted (Section 2.3). This includes the following:
  - Tracking of facial features, resulting in 20 2D points which translates to a feature vector of 40 features.
  - Shoulder and upper body tracking, with 5 2D points which result in 10 features.
  - Audio Feature Extraction, where extract MFCC and prosody features, resulting in a set of 15 features. We also describe our attempt to improve the quality of the audio as to improve the recognition accuracy.

Furthermore, we denote that during the segmentation, we identify and attempt to overcome some issues that are manifested when working with the SAL database, issues which other relevant work with SAL did not deal with in detail (e.g. [199]). Finally, we should note that the pre-processing and segmentation stages are interlinked, and the *iterative* experimentation procedure we followed in order to achieve the best possible results included the following steps:

(a) Determine a normalisation procedure

(b) Determine the algorithm/parameters to produce the ground truth from individual coder annotations

(c) Segment the audiovisual sessions

(d) Generate the (combined) ground truth

(e) Evaluate resulting segments and ground truth by metrics and other tests

(f) Inspect the produced video segments

4.1 Annotation Pre-processing

4.1.1 Binning

The time-based annotation of the files (Section 4.4.1) presents us with an issue, which we will clarify with the following example, where coders *cc* and *dr* both annotate the file edA01. Let us look at the first three timestamps (in seconds) of each of the coders after the 27th second:

cc : 27.0062514, 27.0112514, 27.0152514

dr : 27.0129868, 27.0519158, 27.0919158

It is clear that there is no one-to-one correspondence in the timestamps across the coders. In fact considering that an audio frame is 0.02 seconds and that for audio feature extraction we would require one value per frame, it is obvious that some processing is required in order to deem the values more useful. The first processing step we performed is to bin the measurements of each coder. Since we aim at segmenting the video files, we generate bins which are equivalent to one video frame. This translates to a bin of 0.04 seconds, since we are dealing with 25 fps videos[1]. The convention that we follow is:

- The i–th bin corresponds to the $i-1$ frame, e.g. the first bin (bin 1) corresponds to the first frame (frame 0).
- The last bin (nth) corresponds to the time range of $[(n-1)*0.04, n*0.04]$, i.e. $t \in n$-th bin iff $(n-1)*0.04 \leq t \wedge t \leq n*0.04$.
- Any other bin i corresponds to the time range of $[(i-1)*0.04, i*0.04)$, i.e. $t \in i$-th bin iff $(i-1)*0.04 \leq t \wedge t < i*0.04$.

The basic binning algorithm is presented in Algorithm 2. It is noted that for frames where the coder provides no annotations, the valence and arousal fields are assigned a "not a number" (NaN) identifier.

[1] The video frame for 25 frames-per-second video is 0.04 seconds, and the corresponding audio frame is 0.02 seconds but for segmenting, we need both streams to agree on the segmenting point. Thus binning on the video frame is more appropriate

4.1.2 Normalising

It is expected that the valence and arousal measurements for each of the coders would not be in total agreement, mostly due to the fuzziness of the perception of such emotional states in humans. Thus, in order to deem the annotations comparable, we need to normalise the data. We can not model the coders themselves since no helpful information is available in that direction. Due to the latter, we used standard normalisation techniques. We experimented with:

(1) Normalising each coder file for each session to have an average value of zero for valence and arousal.

(2) Normalising each coder file for each session to have an average value of zero and a standard deviation of 1 for valence and arousal.

(3) Both of the above cases globally: the average and standard deviation normalisation taking into account all the sessions of a coder.

(4) Set the mean of each coder for a specific session to the mean of the combined measurements of all the coders for that session.

After extracting videos and inspecting the superimposed ground truth plots for these cases, we opted for local normalisation (as in cases a and b) in order to avoid propagating noise in cases where e.g. one of the coders is in large disagreement with the rest. An example of where such a situation arises is in session $rodB01$, where coder cc has a very low (0.06) correlation with respect to the rest of the coders (Table 7.2 in the Appendix).

The ground truth values for the first case and second above have been produced, while we also kept the individual coder values. The reason behind this is that since we have no information to model the users, we want to keep the possibility of training one classifier and the ground truth as recorded by each coder would be necessary to do so. In Fig. 4.1, we can see the valence plot for a specific session for the previous two normalisation attempts. We believe that just normalising to zero average will produce better results, since changing the standard deviation results in values which are firstly outside the range of [-1,1] where the valence and arousal values have been normalised, and secondly, by examining the overall ground truth plots, it also seems to generate more disagreement.

In addition to Fig. 4.1, and in order to assess the selection of our normalisation procedure and justify our previous comments, in Table 4.1 we present the mean squared error (MSE) which is calculated for each coder, with respect to the annotations provided by the rest of the coders participating in a session and then averaged. With no preprocessing, the averaged MSE amongst coder annotations is 0.72. When normalising to a standard deviation of one and a mean of zero , the MSE increases to 0.93. Our selected method of normalisation, just locally normalising to zero mean, produces the least error of 0.046.

It is noted that we analysed the cross-correlation of coder agreement by shifting the measurements in a range of 5 seconds, in order to identify if time shifting the annotations would provide better inter-coder correlation. No significant improvement to the correlation was observed (best improvement was around 0.036, average improvement 0.0014)

Table 4.1: Comparing the averaged mean squared error (MSE) of each coder with the respect to the rest of the coders for the two normalisation procedures, GD: normalising to a standard deviation of one and a zero mean. ZA: just normalising to zero mean. NP: No pre-processing

	$\mathbf{ZA}_{MSE}$	$\mathbf{GD}_{MSE}$	$\mathbf{NP}_{MSE}$
Valence	0.046	0.93	0.072
Arousal	0.0551	0.9873	0.0829

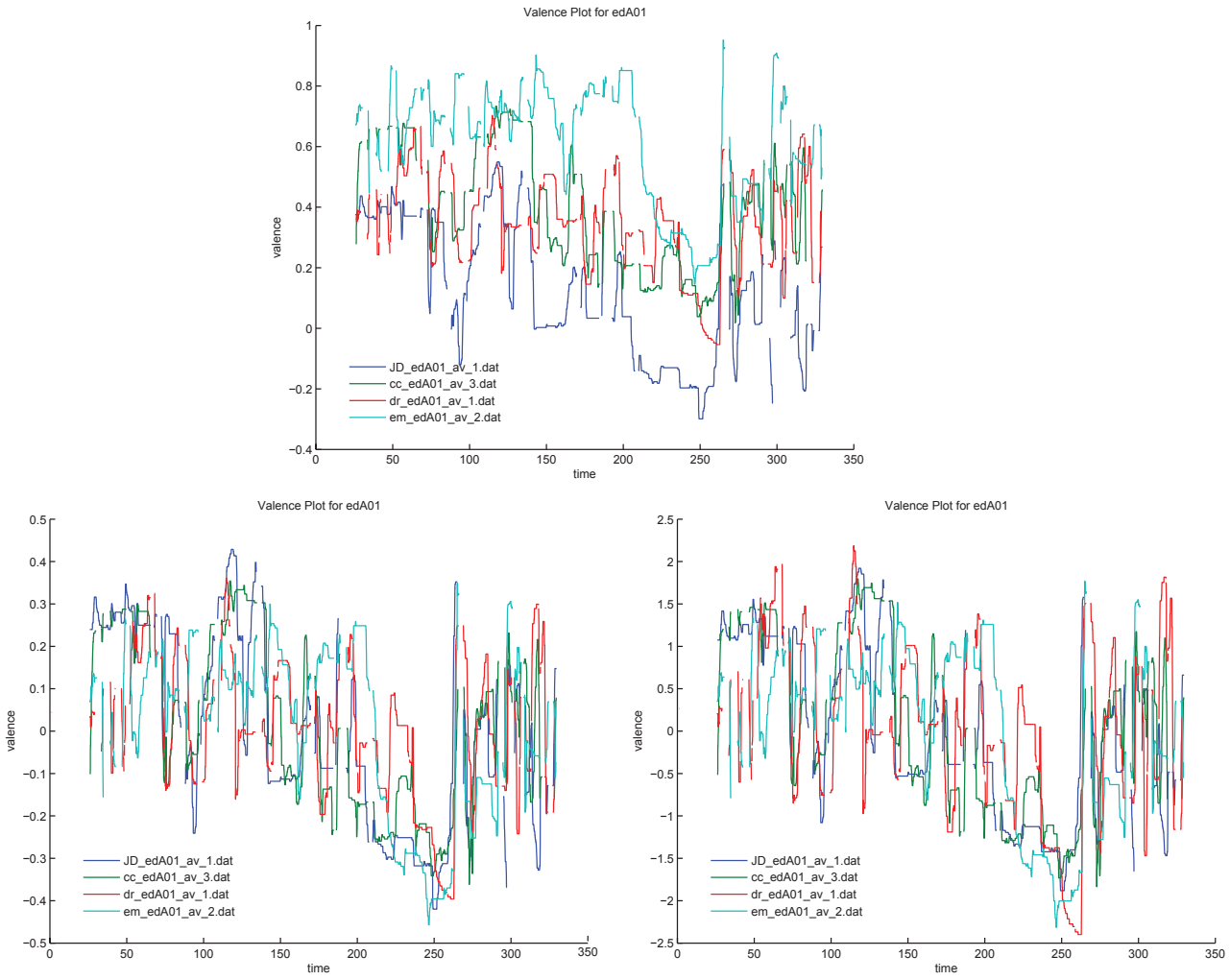


Figure 4.1: Top: The original coder annotations, before any pre-processing was applied. Bottom left: data with mean=0, bottom right: data with mean=0 and std=1

4.1.3 Statistics and Metrics

A set of useful statistics that we extract from the annotations regard the correlation and the emotion categorisation agreement percentages. One application of these statistics is to rank how much "trust" we should have towards a certain annotation, always in relation to the other coders. This is because, as we will see in Section 4.2, we choose certain subsets of coders to combine and produce annotations for a video segment. It is thus important to have a way to weight the subset at hand, and also propagate information from other coders which do not participate in the subset, in order to achieve measurements which are as best and representative as possible.

To proceed with this analysis, we firstly construct all pairs of coders which correspond to each video session. E.g., when we have a video session where four coders have provided annotations, this gives rise to 6 pairs. For each of these pairs we extract the correlation coefficient between the valence values of each pair, as well as the level of agreement in emotion classification in terms of positive or negative. We can formally state the previous metric as follows:

$$Agreement = \frac{\sum_{frame=0}^{n} e(c_i(frame).valence, c_i(frame).valence)}{|frames|} \tag{4.1}$$

where $c_i(frame).valence$ stands for the valence value annotated by coder c_i at frame $frame$. Function e is defined as:

$$e(i,j) = \begin{cases} 1 & if(sign(i) = sign(j)) \\ 0 & else \end{cases}$$

It should be noted that in these calculations we did not consider the NaN values as not to negatively affect the results. In other words, when there were missing values they were ignored.

After these metrics are calculated for each of the pairs, each coder is assigned the average of the results of all the pairs that the coder was a participant. E.g., if we have 4 coders for one video session, each of them will participate in 3 pairs, in each with one of the other coders thus providing three values, of which the average is assigned to the coder for that session. Finally, we attain what we call the $trust$[2] value, which is a measure of how confident we are towards the coder for that specific session. It is expected that there will not be total agreement between the coders, and the only use for such a metric is to *rank* the *relative* significance of the coders:

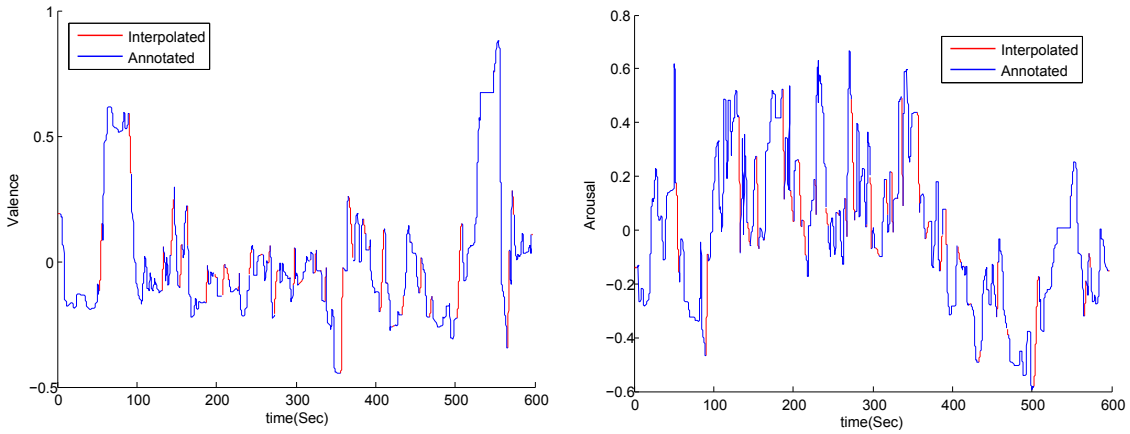
$$trust(coder) = \frac{a * correlation(coder) + b * agreement(coder)}{a+b}, 0 \leq a, b \leq 1 \tag{4.2}$$

We experimented with values of a, b such that $a + b = 1$. Examples we experimented with were using just the correlation ($a = 1, b = 0$), just the agreement ($a = 0, b = 1$) or balancing the values in between ($0 < a < 1, b = 1 - a$). Finally, we chose the correlation as the metric to be used in the segmentation process. It should be noted that correlation appears to be much more strict than agreement, thus providing us with better comparison amongst the coders. Also, during our experimentation we noticed that the use of the correlation evened out variances of coders with respect to the ground truth. We evaluated this by examining the ground truth for cases where different sets of coders where combined for the same frame of the same session video.

[2]The results for each of the coders and each session are presented in Table 7.2 in the Appendix along with the correlation

Algorithm 2: Binning the annotations of the coders
{set of bins, b} $\leftarrow$ *Binning*()

```
1  begin
2      //Initially, all members of any structures are considered to be zero
3      for each coder file c in the annotation files set do
4          for each annotation a in file c with a timestamp of t in coder do
5              Determine bin b where t ∈ b
6              b.valence ← b.valence + a.valence
7              b.arousal ← b.arousal + a.arousal
8              b.annotCount ← b.annotCount + 1
9          end
10         for all bins b in the set of bins do
11             Average b.valence and b.arousal by dividing with b.annotCount
12         end
13     end
14 end
```



(a) (b)

Figure 4.2: Plots of interpolated values for (a) Valence, and (b) Arousal. Values produced by interpolation appear in red.

4.1.4 Interpolation

In order to deal with the issue of missing values, we attempted to interpolate the actual annotations at hand. It should be noted that the interpolated annotations accompanied the original annotations in our segmentation process, which will be described below. In other words, we kept both the pre-interpolation and post-interpolation annotations (following the principle of least commitment) in order to maintain the option of experimenting with both.

There are many interpolation techniques available. We used piecewise cubic interpolation, given that this method does preserve the monotonicity and the shape of the data since we wanted the interpolation to be as less obtrusive as possible. The method is described in the MATLAB documentation [122, 121] and an example of the application can be found in Fig. 4.2.

4.2 Segmentation

After the annotations have been prepared, the next stage regards the actual segmentation of the audiovisual sessions. The procedure that we follow is presented in Algorithm 5, also making use of procedures presented in Algorithms 3 and 4. We should note that although we generated interpolated annotations in addition to the original annotations, the segmentation process is solely conducted on the original, not-interpolated data.

Firstly, we will describe the actual time window that we want the segment to capture. E.g. in capturing negative emotional states, if we assume that the transition from non-negative to a negative emotional state happens at t seconds, we would have the following window:

$$[t-1, t, t', t'+1]$$

where t' seconds is when the emotional state of the individual turns non-negative again. The procedure is completely analogous for positive emotional states. In other words, our goal is to capture the entire transition to and out of an emotional state. This is relevant to what is called the baseline problem: Generally, it is the problem of finding a condition to compare against and so to detect changes which lead machine learning techniques to recognise and classify automatically. Most such techniques *depend* on the existence of such a state [84]. Since we are performing audiovisual segmentation based on the video sessions, we are looking for "a frame in which the subject is expressionless and against which changes in subject's motion, pose and appearance can be compared" [84]. In posed emotion detection, typically the subjects are instructed to express a certain emotional state, which (when referring to facial expressions) corresponds to the temporal phases of a facial expression (neutral, onset, apex, offset - See Section 2.2.5). Since such neutral emotional states are typically not present in spontaneous data [84, 113], by capturing the transition *to* and *from* an emotional state, we capture the frames where the emotional state is changed and we provide a condition to compare against.

4.2.1 Detect and Match Crossovers

In Algorithm 3, for an input coder c we detect the crossing over from one emotional state to the other. That is detected by examining the valence values, and locating points where the sign changes. It is important to note that we use a modified version of the sign function, which returns 1 for values ≥ 0 (a value of 0 valence is never encountered in the annotations), -1 for negative and 0 for NaN. To demonstrate the effects of this choice with an example, in the following sequence of frames (with respective valence measurements) we detect a crossover from a non-negative emotional state to a negative one:

$$\text{frame 1: } NaN \text{ frame 2: } NaN \text{ frame 3: } -0.2 \text{ frame 4: } -0.3...$$

Here, the crossover is saved at frame 3. The reason behind this data handling, is that we have experimentally observed that NaN values are being recorded in many cases where the avatar is speaking. We have also noticed that the specific frames where NaN values co-occur with avatar speech, are before or after a person responds to the avatar, and this is again typically accompanied by an emotion expression. Thus, by separating based on the previous choice of valence values, we

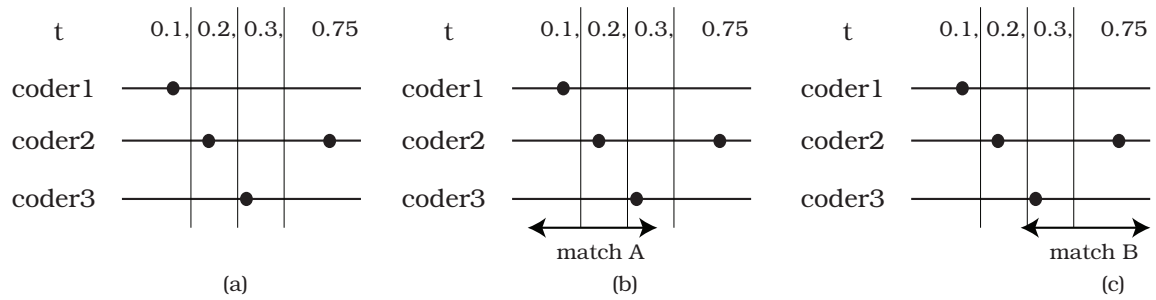


Figure 4.3: How a set of crossovers can be matched with more than one combinations within a temporal offset of 0.5 seconds. The black circles represent crossover points from a specific emotional state to the other.

essentially capture the time window that we want: before, during and after the emotion expression. The procedure returns the crossovers to negative and positive emotional states. Algorithm 3 returns the set of crossovers to a positive ($PosCrossOver$) and to a negative ($NegCrossOver$) emotional state.

Algorithm 4 is responsible for matching the previously attained crossovers (as described in the previous paragraph) across all the coders which participate for each audiovisual session. In more detail the procedure described in Algorithm 3 is executed for each coder, thus accumulating all the crossover points for each of the coders. The output is then passed to Algorithm 4.

The goal is to match crossovers across coders: E.g. if a session has 4 coders which provide annotations, due to synchronisation issues as discussed in Section 4.4.1, typically the frame where each of the coders detect the crossover from one emotional state to the other is not common amongst the set of coders for the file. Thus, we have to allow an offset for the matching process. This procedure performs precisely the process we described, it searches the crossovers detected by the coders and then accepts the matches where there is less than 0.5 seconds (which is the offset) time difference between the detections. The result of this procedure will be fed to another procedure which will segment the videos and produce the ground truth based on the resulting matchings.

When a matching is discovered, we remove the matched crossovers and continue with the rest. This of course poses an issue: There are different combinations of crossovers which may match using the offset. An example of such a scenario can be seen in Fig. 4.3.

In the figure, we can see four crossovers from a specific emotional state to the other. These crossovers can be matched in two ways, by remaining within the offset of 0.5 seconds: By matching the first three crossover points, from coders 1,2 and 3 or matching the last two points, from coders 3 and 2. If we were to pose this problem as an optimisation function, we would want to maximise the number of participating coders:

$$max[\sum_{i \in C} inMatch(i, mc)]$$

where C is the set of coders, mc is the matched crossover and $inMatch(i, mc)$ is a function which returns 1 if the coder i is in the matched crossover mc, 0 otherwise. Furthermore, we would like to minimise the temporal distances between the participating coders:

$$min[\sum_{i \in mc} \sum_{j \in mc, j \neq i} d(i, j)]$$

where i and j iterate over the set of crossovers in the matched crossover mc and $d(i,j)$ returns the time distance between the crossovers i and j. Therefore, by combining the two specific objective functions, we obtain an objective function for maximisation:

$$max[\sum_{i \in C} inMatch(i, mc) - S(\sum_{i \in mc} \sum_{j \in mc, j \neq i} d(i,j))]$$

where $S \geq 0$ is a constant that determines the tradeoff between having more coders in one match and having a shorter time distance between the matched crossovers. If we were to pose a constraint for the optimisation problem, it would be the predefined offset of 0.5 seconds between the crossovers that belong into a matched set. By examining the available dataset, in our case we decided to set the $S = 0$, i.e. just maximise the number of coders which participate in a matched crossover set. We based this decision on the following:

- The reliability of the ground truth produced would be higher if more coders agree on the crossover.
- By experimenting with just considering the number of coders, the offset amongst the resulting matchings was on average quite smaller than 0.5 seconds.

To just facilitate the selected objective function (maximise the number of participating coders) we do not need to solve an optimisation problem. It can be delt with by iterating over the entire set of crossovers, by firstly allowing only matches where all four coders agree, then where three of them agree and so on. This is expressed by the loop beginning in line 2 of Algorithm 4. We completely disregard cases where only one coder detects a crossover. To give an actual example, assume the following scenario:

$$\text{Coder 1: frame 1:0.5, frame 2: } -0.2, \text{ frame 3: } -0.25, ...$$

$$\text{Coder 2: frame 1: 0.5, frame 2: 0.4, frame 3: 0.3, frame 4: } -0.3, ...$$

Coder 1 detects a crossover to a negative emotional state at frame 2, while coder 2 detects it at frame 4. Since we have 25 frames per second, 2 frames difference between the detections is 0.08 seconds, where 0.08 is less than 0.5 and thus we have a matched crossover of size 2.

4.2.2 Segmentation Driven by Matched Crossovers

Finally, in Algorithm 5, the actual segmentation occurs. The procedure actually continues with the output of the algorithm described in the previous paragraph. After attaining the sets of matched crossovers from Algorithm 4 (lines 6,7) we show an iteration for all the sets of matched crossovers for the "to-Negative" transition, starting from line 8.

We call the variable that corresponds to the current matched crossover $mcos$, for which fields $mcos(i).frame$ corresponds to the $frame$ where the i-th crossover of the matched crossover $mcos$ occurred, $mcos(i).coder$ represents the coder which detected the i-th crossover of $mcos$, while we also assume that $mcos(i).valence$ is the vector holding the valence measurements for coder i participating in $mcos$.

Firstly, in lines 11:18, the frame where the crossover is inferred to occur for each member of the set returned by the previous algorithm is attained. To explain this we can look at the example given in the previous paragraph. A coder detects a crossover at frame 2, the other at frame 4. Average the frames to the nearest integer, and you can assume that the crossover happens at frame 3. In fact, in case we have only 2 coders agreeing, we use the *correlation* metric in order to weight the averaging, which as described in Section 4.1.3 provides a measurement of the relative importance of the annotations for each coder and propagates information from the other two coders not participating in the match.

After the crossover frame decision is taken, the start frame of the video segment has been decided. We subtract 25 frames before that frame in order to capture 1 second before the transition window. Then (lines 23 to 34), we retrieve the ground truth values for valence by incrementing the initial frames where each crossover was detected by the coders. Again, following the previous example, this means that we consider frame 2 of coder 1 and frame 4 of coder 2 to provide ground truth values for frame 3 (the average of 2 and 4). This gives us an averaged valence value, as described in the algorithm. Then, the frame 4 valence value (ground truth) would be the combination of frame 3 of coder 1 and frame 5 of coder 2 and so on [3]. The previous procedure of determining combined average values continues until the valence value crosses again to a non-negative valence value (following the previous example). The endpoint of the video is set 25 frames after crossing back to a non-negative valence value. Finally, we have determined the starting and ending point of the video segment, which is then extracted. For the extracted video, the ground truth are the valence and arousal values which are combined as in lines 28 and 32 of the algorithm. An example of how time-shifting is performed during segmentation is presented in Fig. 4.4.

Some other points regarding the segmentation are the following: Firstly, it should be noted that we experimented with trimming the endpoints in the 1 second offsets before and after the emotion expression in case there are other crossovers there. After the output videos were examined, we decided that it was best to keep these offsets without any trimming and decide manually whether to adjust them. Also, we have experimented with thresholding the arousal (or intensity) of the emotion in order to filter the output videos. The threshold has been imposed as detecting the average intensity over the expression and making sure that it had an absolute value of over 0.2 in each of the coders in order to be accepted. We also experimented with thresholding the duration of the emotion expression, from the endpoints of the window. It is noted though that the results of these attempts have had an advisory only part in the final video selection.

Finally, in Fig. 4.5 we present two segments which have been extracted by following the technique described, for a transition to a negative emotional state. We remind that the window we aimed to capture is a transition to and from an emotional state (considering NaN values in the annotations to be such separating points). The first dashed line presents the transition to that state, and the second out of that state. In the plots, we present the arousal and valence values after the interpolation, as described in Section 4.1.4. Thus, where no switch is observed in the valence values (i.e. frame 25 of plot (c)), it is because the segmentation was produced by using the original annotations, and before frame 25 NaN values were previously placed before interpolating. Nevertheless, we manually selected a subset of the segmentation results which we considered to best respond to the time window we desired, in order to capture the desired baseline. Typically, the segments

[3] As we have stated, in the case there are two coders the correlation is used to weight this average, we omitted this from the trivial example

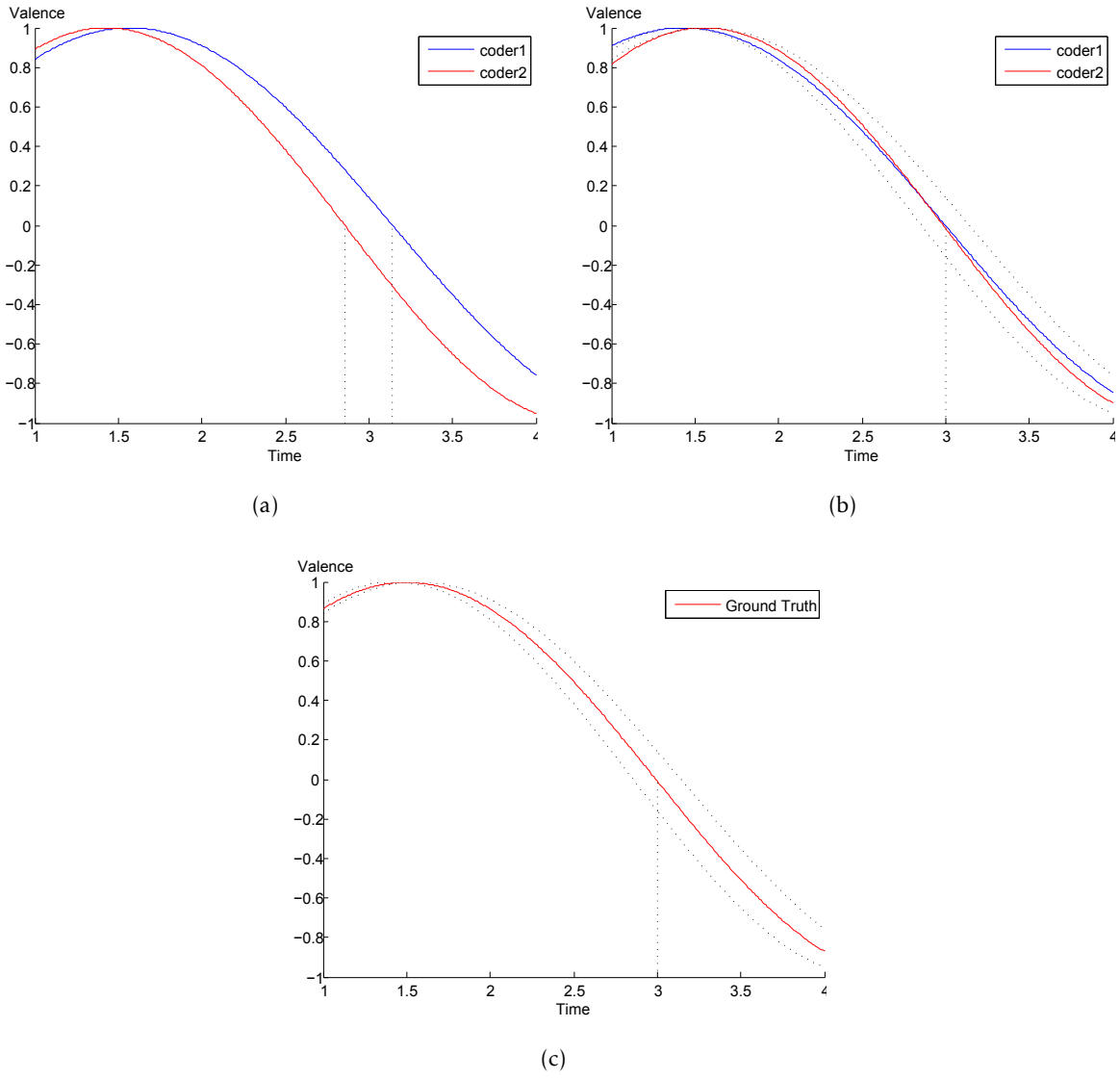


Figure 4.4: Illustration of time shifting during segmentation: (a) The valence values from two coders plotted against time for one video segment. The first coder observes a transition to negative valence at time 3.14, while the second observes the transition at 2.856. (b) The values of each coder are time shifted so that the transition to a negative emotional state is observed at $t = \frac{3.14+2.856}{2} = 2.999$ (the average). (c) The valence values of both the coders are averaged to attain the final ground truth for valence. In (b) and (c), the original locations of the values of each coder before shifting are shown in dashed black. It is noted that the process for attaining arousal values is analogous and follows the respective shifts for valence. Also, the correlation is used to weight the averaging when only two coders are available - we omitted this from the example for simplicity.

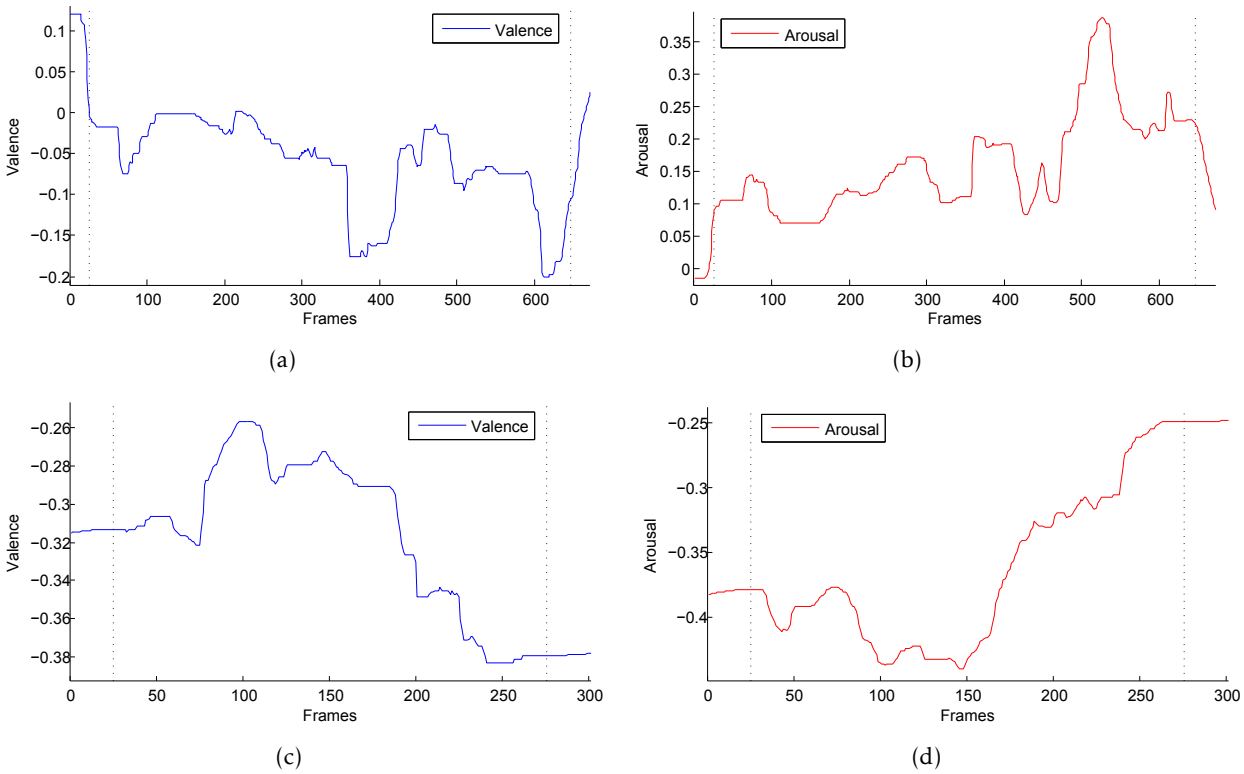


Figure 4.5: Two examples of interpolated valence ((a),(c)) and arousal ((b),(d)) plots from two individual segments produced by the segmentation procedure. The dashed lines denote the positions where the temporal window switches from one emotional state to the other, or where NaN values were observed in the valence annotations before interpolating. Please refer to the text for a discussion.

consist of a single interaction of the subject with the avatar, starting with the final seconds of the avatar speaking, continuing with the person responding (and thus reacting and expressing an emotional state) and concluding with the first seconds where the avatar responds, either due to the actual valence annotations or to the lack off annotations (see the missing values issue 4.4.1) which co-occurred with this phases. Given that in naturalistic data, emotion expressions are not generally preceded by neutral emotional states [113, 84], we considered this window to provide the best baseline we could achieve. Also, it is interesting to compare the plots in Fig. 4.5, with the plot of the temporal phases of facial expressions (Fig. 2.6).

It should be noted that we have selected a total of 22 minutes and 20 seconds of audiovisual data (Table 4.2). After the segmentation proceeded, we selected a subset of the videos produced by giving preference to videos where 3 and 4 coders agree. Also, we note that the entire segmentation procedure, is illustrated along with pre-processing and feature extraction (next section) in Fig. 4.9 and summarised in Section 4.5.

4.3 Feature Extraction

After the audiovisual sessions were segmented, the feature extraction process followed. In this section, we will provide some description of the extraction of audio features, as well as facial and shoulder tracking.

Table 4.2: Audiovisual data and annotations. All videos have a frame rate of 25 fps

Subject	Total Videos	Negative	Negative Duration	Positive	Positive Duration
ed	36	21	00:03:36	15	00:02:38
ell	35	19	00:03:53	16	00:02:40
ian	30	15	00:01:57	15	00:01:48
rod	33	18	00:03:34	15	00:02:12
total	**134**	**73**	**00:13:01**	**61**	**00:09:19**

Algorithm 3: Detecting crossovers in coder annotations: $\{PosCrossOver, NegCrossOver\} \leftarrow$ DetectCrossovers(coder c)

```
// bstr is the binned structure where every member is an annotation
// of the valence and arousal values at that frame by the specific coder
for each frame in bstr do
    if sign(bstr(frame).valence) ≠ sign(bstr(frame − 1).valence) then
        if sign(bstr(frame).valence) > 0 then
            Add frame to PosCrossOver structure
        end
        else
            sign(bstr(frame).valence) < 0
        end
        Add frame to NegCrossOver structure
    end
end
```

Algorithm 4: Match crossovers across coders for each session, maximising the number of coders participating: $\{MatchedCO\} \leftarrow$ MatchCrossOvers($CrossOvers$)

```
for Each session s do
    for i=4 to 2 do
        //Try and get as much as coders as possible to agree. 4 is maximum and 2 is
        minimum
        for Each crossover co in CrossOvers belonging to s do
            currentlyMatched ← co
            Find all crossovers co2 in CrossOvers which:
            - Belong to s
            - Are from different coders
            - co2 ≠ co ∧ abs(co2.time − co.time) ≤ 0.5 seconds //Threshold is 0.5
            Add the co2 to currentlyMatched
            if length(currentlyMatched) = i then
                mark all crossovers in currentlyMatched as seen
                add currentlyMatched to MatchedCO
                remove currentlyMatched from CrossOvers belonging to s
            end
        end
    end
end
```

Algorithm 5: Segment and produce ground truth: Segmentation()

1 **for** *each coder annotation file c* **do**
2 //Goal is to capture a transition to and from a negative emotional state to a
3 // non-negative, with a 1 second offset before and after (resp. for positive)
4 //We use the correlation for weighting when match has 2 coders
5 $\{PosCrossOver, NegCrossOver\} \leftarrow DetectCrossovers(c)$
6 $MatchedPos \leftarrow MatchCrossOvers(PosCrossOver)$
7 $MatchedNeg \leftarrow MatchCrossOvers(NegCrossOver)$
8 **for** *each matched set of crossovers mcos in* $MatchedNeg$ **do**
9 //Average time (frame) of crossing over to negative valence
10 //Remember that a 0.5 second offset has been used
11 **if** *length(mcos)* ≥ 3 **then**
12 //agreement in 3 or 4 coders
13 avgFrame = $int\left(\frac{\sum_{i=0}^{|mcos|} mcos(i).frame}{length(mcos)}\right)$
14 **end**
15 **else**
16 //2 coders agree, weight using correlation ($corr$)
17 avgFrame = $int\left(\frac{\sum_{i=0}^{|mcos|}(mcos(i).frame * corr(mcos(i).coder))}{\sum_{i=0}^{|mcos|} corr(mcos(i).coder)}\right)$
18 **end**
19 //1 second (25 frames) offset before
20 $startFrame = avgFrame - 25$
21 //Need to find where we cross from the negative emotional
22 //state back to a non-negative (or NaN)
23 incFrame $\leftarrow 0$
24 **repeat**
25 incFrame $\leftarrow$ incFrame + 1
26 **if** *length(mcos)* ≥ 3 **then**
27 //agreement in 3 or 4 coders
28 avgValence = $\frac{\sum_{i=0}^{|mcos|} mcos(i).valence(mcos(i).frame+incFrame)}{length(mcos)}$
29 **end**
30 **else**
31 //2 coders agree, weight using $corr$
32 avgValence = $\frac{\sum_{i=0}^{|mcos|}(mcos(i).valence(mcos(i).frame+incFrame) * corr(mcos(i).coder))}{\sum_{i=0}^{|mcos|} corr(mcos(i).coder)}$
33 **end**
34 **until** *sign(avgValence)=1 or avgValence is* NaN ;
35 //Add 25 frames after crossing back to non-negative (or NaN)
36 endFrame = (avgFrame + incFrame) + 25
37 //Video is segmented in the range [startFrame,endFrame]
38 //Ground truth (valence/arousal) is averaged as in lines 11-17
39 **end**
40 //The process is repeated in an analogous fashion for the "to-Positive" crossovers ($MatchedPos$) - See Line 8
41 **end**

4.3.1 Audio Features

4.3.1.1 Sound Pre-processing

As we have mentioned, the scenario of the SAL database incorporates the interaction of a virtual avatar with a subject. As far as audio processing is concerned, we are presented with two main issues:

(1) The avatar speech being part of the audio recording.

(2) The ambient noise present in the recordings.

To deal with the first issue, we firstly isolate the noise profile of the clip: We isolate the region where neither the avatar nor the subject are talking, while there are as less as possible interfering sounds. Based on the observation that the avatar voice was recorded at a lower volume than the speaker, since the individual was provided with a microphone and earphones during the recording, and since the speech of the avatar and the speech of the individual were typically non-overlapping, we proceeded by mixing the noise profile of the clip on top of the intervals where the avatar was speaking. The result is the removal of the speech of the avatar, while maintaining the same noise level for the clip. This process had to be manually performed, individually for each clip. An example of the procedure is found in Fig. 4.7. We deemed that the removal of the avatar speech was required since it would provide inconsistent indications for the emotional state of the subject.

Secondly, we have the issue of the ambient noise present in the recordings. We decided to remove the noise based on two observations:

- The ambient noise present at each session differs from session to session, even when the session is of the same subject. This would differentiate the features extracted compared to the audio, even when the actual characteristics of the speech of the person remain the same.
- The features we use (Mel-frequency Cepstrum Coefficients (MFCC)) have been reported to be sensitive to noise [180].

The procedure we followed takes advantage of the noise profile we captured in the avatar speech removal. More specifically, the noise profile captures the spectrum of the noise (Fig. 4.7). This spectrum is utilised in order to filter out (subtract) noise from the rest of the clip (reducing it by 40 dB in our case). The Fast Fourier Transformed (FFT) is used for producing the spectrum, for which we chose a size of 4096 elements. In Fig. 4.8, an example of how noise reduction is performed is illustrated.

4.3.1.2 Audio Feature Extraction

In this section we will describe the audio features that have been extracted after the actual audio has been pre-processed. Our audio features include the Mel-frequency Cepstrum Coefficients (MFCC) [99], as well as prosody features, such as pitch and energy. In more detail, the mel-frequency cepstrum (MFC) is actually a representation of the spectrum of an audio sample, which

is mapped onto the nonlinear mel-scale of frequency. The MFCC coefficients collectively make up the MFC for the specific audio segment, and by combining common signal processing transformations such as the Fourier transform and the discrete cosine transform, use a log based representation in order to better approximate the human auditory system's response. They have been typically used in speaker recognition, e.g. [97] and also emotion recognition, e.g. [146, 135]. The MFCC coefficients derivation typically includes the following steps: [65, 201]:

- Transform the windowed segment into the frequency spectrum by a Fourier Transform. The spectrum attained is called P.
- Map the powers of the spectrum P onto the mel-scale by transforming to the mel-frequency axis, in order to better approximate the human ear perception.
- Apply a convolution with triangular filters
- Take the log of the powers of the spectrum obtained by the previous step
- Apply the Discrete Cosine Transform (DCT) to the result of the previous step
- The MFCC are the amplitudes of the output of the previous step

For our specific application, we used 6 cepstrum coefficients, while also extracting prosody features, such as pitch (fundamental frequency) by using a Praat pitch estimator [143], and energy. These are essentially the typically used set of audio features [203]. In total, we have a set of 15 audio features. We note that we used a 0.04 second frame with a 50% overlap (i.e. first frame 0-0.04, second from 0.02-0.06 and so on), thus providing us with a double frame rate from the video frames so that the streams would be easily synchronised.

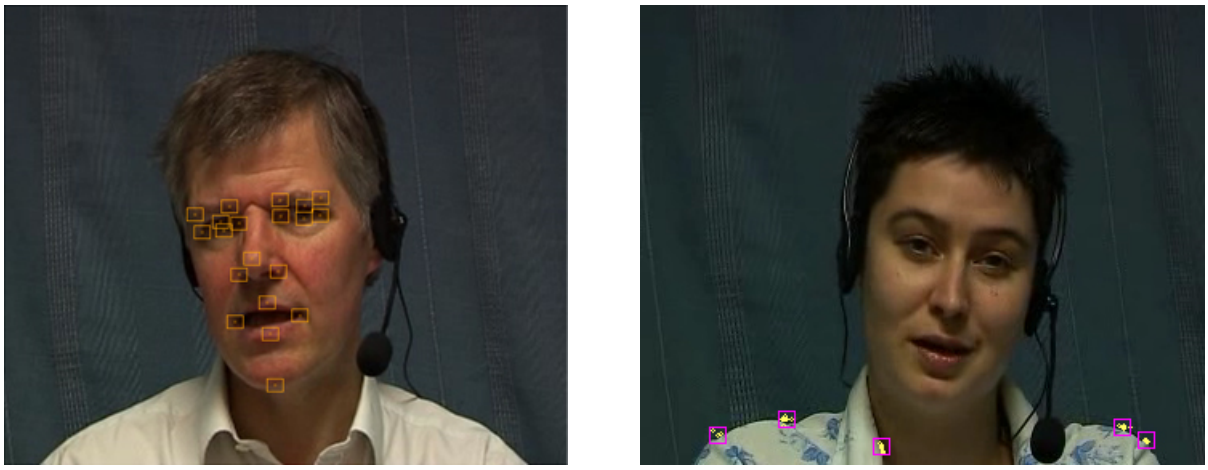
4.3.2 Face and Shoulder Tracking

In relation to the visual modality, we extract features by both tracking facial expressions and shoulder movement (in order to capture upper body movement).

The tracker used for extracting facial features of the audiovisual segments implements particle filtering with factorise likelihoods [142]. We extract 20 fiducial points, as seen in Fig. 4.6(a). As far as shoulder tracking is concerned, the tracker is based on auxiliary particle filtering [148]. Five points are tracked as seen in Fig. 4.6(b). A further description of the tracking methods is outside the scope of this project, detailed descriptions can be found in the relevant references given.

4.4 Issues

In this section, we will discuss some of the issues we faced during the stages of the project that have already been executed. We will provide more discussion on issues faced with the annotations of the database (Section 4.4.1), since these are the major issues that completely determine the segmentation and the emotional classification of the audiovisual material, while we will briefly mention some other issues in Section 4.4.2.



(a) (b)

Figure 4.6: (a) Snapshot from the facial feature tracker (b) Shoulder tracking

4.4.1 Timestamps, Synchronisation and Missing values

The annotations of the audiovisual sessions of the SAL database were made possible by using the FeelTrace [67] application. Some of the issues that the time-based operation of the application presents us with are the following:

(1) No one-to-one correspondence between the timestamps of each coder for a session: The annotations bear timestamps in seconds which differ from each coder and session to another. In other words, there is no consistent scale for annotations.

(2) Annotations are not available for the entire session: Throughout the annotation files, there are time intervals were annotations are not available.

(3) Synchronisation issues and timing issues, where it appears that annotations are not synchronised with the audiovisual data stream.

As far as the first issue is concerned, it has been dealt with by binning the annotations (Section 4.1.1). Things get somewhat more complicated on the second and third points. The second point refers to missing annotations for some sets of frames. We have identified that this could be due to a set of reasons, some of which are:

- The coder might not be certain on which annotation is appropriate.
- The coder might release the mouse button for some other reason.
- It has been observed that on a lot of occasions the coders stop the encoding when the avatar they are interacting with is speaking.
- The clock speed of the CPU has an effect on the frequency of measurements being recorded.

For the issue of synchronisation, some explanations are:

- The response time of each individual is dependent on the individual itself. Every person has a different reaction time for audiovisual stimulation.

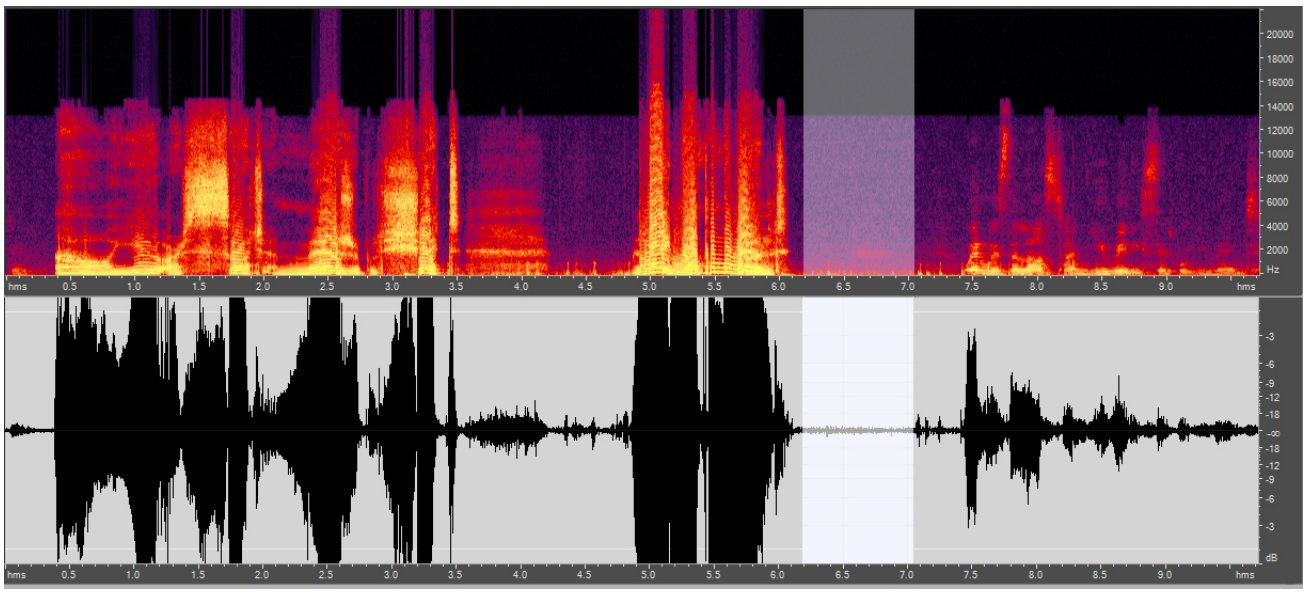


Figure 4.7: Capturing the noise profile of one of the sessions. Top: the frequency spectrum. Bottom: the dB waveform plot. In both plots the isolated noise profile is highlighted (6.2-7.1 sec. approximately)

- Again, the clock speed of the CPU has an effect on the synchronisation between the actual video and the recording of measurements.

Due to the fact that there are more than one possible explanations for the observations of the previous issues and due to the lack of information on the coders in order to be able to model them, it is really not possible to reverse engineer these observations and reinterpret the data. Approaches taken to improve these issues are the time-shift operations as seen in the segmentation algorithms (Section 4.2) and the interpolation done on the resulting annotations. These problems can be solved in a much more elegant and accurate way, only if during the annotation phase frame-by-frame ground truth annotations are provided.

4.4.2 Other Issues

- **Coder Modelling:** Information regarding the coders themselves would be highly useful in actually attempting to model the behaviour of the coders in order to have some more comparable annotations. This could be achieved by providing the annotations of each of the coder on predefined emotional states.

- **Subject Dependency:** From observing the subjects, we can note that different persons react in a different way during the interaction. One of the individuals (Ian) seems to have typically subtle and mild reactions while other persons such as Roddy have intense head movement and intense emotion expression. The task of generating a subject independent system with only four coders is extremely difficult (if at all possible), and these subject dependent reactions are another obstacle.

- **Audio Issues:** There are different *noise intensity levels* across the sessions of the clips, even for the same individual. We have already mentioned that the avatar with which the person is

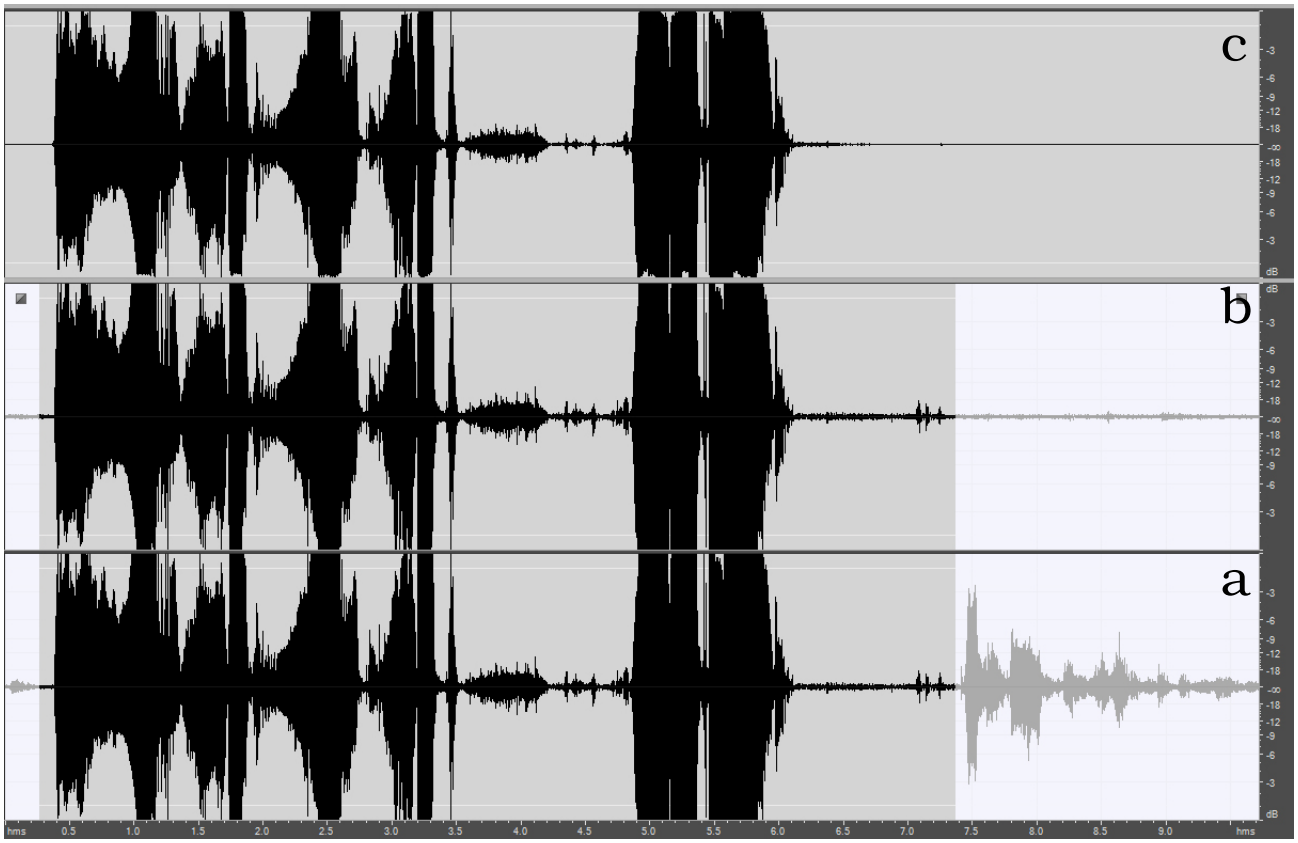


Figure 4.8: Waveform plots: (a) Before avatar speech removal. (b) After the noise profile of the clip has been mixed over the avatar speech, essentially replacing it. The processed time intervals appear highlighted (on a white background) (c) After noise reduction has been applied to (b)

interacting is also recorded and how we approached this issue along with the noise reduction in Section 4.3.1.1.

- **Tracking:** Sometimes during tracking the shoulders of some individuals go almost completely off camera, something which makes shoulder tracking quite difficult for the specific clips. Also, there are times where the face of the individual goes partially off camera. Since the data is spontaneous, we have occlusions of body features which are being tracked (e.g. occluding the face with hands). Our goal was to maintain the dataset as consistent as possible, but due to the intrinsic characteristics of spontaneous data this was not entirely possible. Another issue is that there are different lightning conditions across sessions, some individuals wear clothes which are quite similar to the background colour and that on certain occasions the audiovisual equipment used occludes tracked body parts.

Issues relating to the last two points, are essentially consequences of the fact that the data is spontaneous, and these difficulties are eventually expected to be dealt with if the automatic recognition systems are to be employed in real-life situations [86, 84].

4.5 Discussion

The contents of the chapter, namely pre-processing, segmentation and feature extraction, are summarised in Fig. 4.9. The first stage, that of pre-processing, starts with the set of annota-

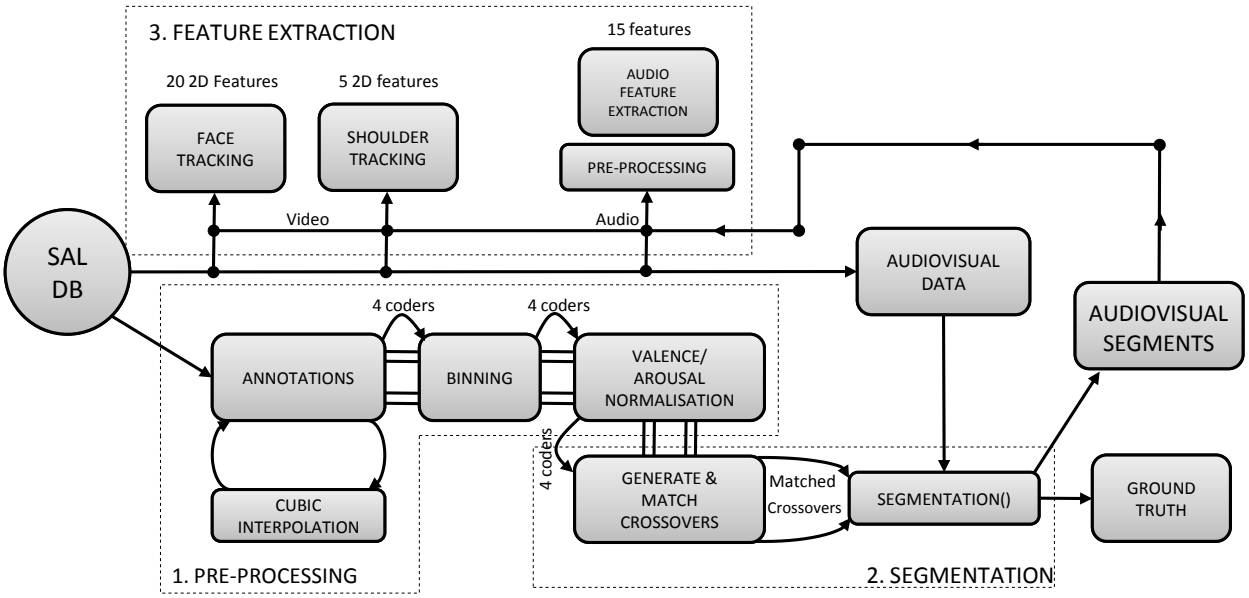


Figure 4.9: Pre-processing, segmentation and feature extraction

tions provided by the SAL database. We generate the interpolated valence/arousal annotations and maintain them along the pipeline. The annotations of the four coders are binned for each of the sessions, and then the normalisation procedures follow. Passing on to the segmentation stage, the crossovers (i.e. transitions to a negative (positive) emotional state) are detected (Algorithm 3), and then they are matched across the coders for each of the sessions (Algorithm 4). The matched crossovers for to-Positive and to-Negative transitions are then passed on to the segmentation procedure (Algorithm 5), which also has access to the audiovisual data from the SAL database. The audiovisual segments to be used, along with the respective processed ground truth are produced after the segmentation procedure terminates. Finally, in the feature extraction stage, the relevant audiovisual segments from the SAL database are tracked (for extracting facial expression and shoulder features), while the audio stream is pre-processed before the extraction of relevant features.

The pre-processing, segmentation and feature extraction stages were, in combination, a very time-consuming stage of the project, due to the iterated experimentation with normalisation, video extraction and ground truth evaluation as well as the tracking and audio pre-processing stages. The experimentation has led us to the final methodology used which was described, and our segmented material along with the produced ground truth will be forwarded to our experiments, described in Chapter 5.

Chapter 5

Experimental Results

The segmented data which we attained by the Segmentation procedure described in Chapter 4 will now be used for both discrete and continuous emotion classification. Starting with discrete, we remind firstly that we deal with two emotional states: Positive and negative. We remind that the window we attempted to capture during the segmentation procedure corresponded to a transition from an emotional state to another, and then the return to the first one, e.g. [positive, negative, positive], along with a small offset in the beginning and end of the segment. This transition is now labelled correspondingly, thus generating two sets of data which essentially form a classification problem of discriminating whether a sequence of data belongs to the negative or positive set. This is how we approach the problem in Section 5.1, by using Coupled Hidden Markov Models (CHMMs) and then Support Vector Machines (SVM).

The segmentation procedure also generated a set of ground truth 2D values in the valence/arousal space. These values will be used as the ground truth for continuous emotion recognition, discussed in Section 5.2. For this part we used learning techniques such as Long Short-Term Memory (LSTM) neural networks (Section 3.1.4) and Support Vector Regression (SVR) (Section 3.2.2). In the relevant section, we will explore many possibilities for experimentation. In more detail, we will explore feature-level fusion for both LSTMs and SVRs, provide a discussion on the proper evaluation of the estimated values while we will show results which indicate the effects of the length of sequences for dynamic learning. Our experiments will include the application of dimensionality reduction, while we will also experiment with decision-level fusion and LSTMs. Also, we will experiment with capturing patterns and correlations in the valence/arousal predictions. Finally, we note that all our experiments are evaluated over 10-fold cross validation. On the notation, we should state that in the result tables for our experiments the initials of the cues involved appear in the related column or row, e.g. F for facial expression cues, SA for fused shoulder and audio features and so on.

5.1 Positive vs Negative Discrete Emotion Recognition

Firstly, we should state that for our discrete experiments for emotion recognition, we used Coupled Hidden Markov Models (CHMMs) [22]. It is outside of the scope of this report to describe the operation and the theory behind Hidden Markov Models (HMMs), but we will briefly provide an intuitive-level description, resuming the brief reference we provided when discussing CRFs

(Section 3.3): HMMs are statistical models which are able to describe the operation of a system, having hidden states and corresponding output variables. HMMs run on time frames: They begin at a certain start state, similarly to a finite state machine (FSM). There are transitions leading out of the current state s, leading into some possible next states S. These transitions are labelled by a probability, which essentially states that if at time t the model is at state s, it can transfer to state $k \in S$ with a probability of p where p is the label of the edge from s to k. The output variable is again defined by a set of probabilities at each state and is produced at each time frame.

In our case, we can consider the output variables as the observations of the model. Then the problem that an HMM is responsible for solving, is "given a sequence of observations, how likely is it that the given model produced it?". In other words, HMMs return the maximum likelihood estimate of a model given the sequence of observations. One significant criticism of HMMs is the assumption that the next state is determined only by the current state (this is known as the Markov assumption), while there is also an independency assumption for the observations: The current output (observation) is statistically independent of the previous outputs. As we have discussed in 3.3 this is a point where Conditional Random Fields (CRFs) are considered superior to HMMs, as they drop the independency assumption.

The HMMs used for the audio and shoulder sequences have three and two states respectively, while they are ergodic models, i.e. a transition from any state to any other state including itself is permitted. The HMM for the facial expressions is a four state machine, with each state representing one of the temporal phases of facial expressions: neutral, onset, offset and apex. The fusion of each of the modalities/cues is model-level: That is where two-coupled and three-coupled HMMs are introduced, in order to accommodate two or three data streams. In such models, the next state does not only depend on the current state of the HMM of the current stream, but on the current states of all HMMs which participate in the model. For more details, we refer the reader to [145], since the configuration we used is essentially the same with the paper in question.

For our experiments with discrete emotion recognition, as we mentioned in the introduction, we separated our results into two classes: The class of positive and the class of negative emotion expressions. There are two common methods of evaluating learning techniques in such scenarios:

- Subject Independent: Data regarding one of the subjects is omitted during training, and is used only for evaluation.
- Subject Dependent: In subject dependent mode the data for evaluation is picked as regular n-fold cross validation proceeds. A percentage of data from each of the subjects is taken at each fold.

All previous work regarding the SAL database has reported with subject dependent results (e.g. [199]). Achieving a subject independent test with just four subjects is essentially very difficult (in [199] it is characterised as infeasible). Nevertheless, we performed some subject independent experiments, for which we present the results in Table 5.1 just for reference. The results confirm that subject independency with four subjects is very difficult. We will not comment on the results since any conclusions would be risky. The confusion matrices returned from training were substantially different and not comparable.

Proceeding to subject dependent recognition, we used 10-fold cross validation, with which we attained the results presented in table 5.2. From the single cues, we deduce that the face and

Table 5.1: Subject independent recognition performs very low with just 4 subjects over 10-fold cross validation. The labels for each column stand for the initials of each cue/fusion of cues: face (F), shoulders (S), audio (A) and fusion of: face/shoulders (FS), shoulder/audio (SA), face/audio (FA) and face/shoulder/audio (FSA) cues.

F	**S**	**A**	**FS**	**SA**	**FA**	**FSA**
59.7%	48.5%	56%	49.25%	48.5%	51.49%	54.47%

Table 5.2: Subject dependent recognition with CHMMs averaged over 10-fold cross validation

F	**S**	**A**	**FS**	**SA**	**FA**	**FSA**
73.13%	73.88%	61.19%	78.36%	68.66%	70.9%	79.1%

the shoulder cues are more distinctive for positive vs. negative classification than the audio cues. The model-level fusion which is used, shows that the fusion of the audio cues with either the facial expression or shoulder cues, slightly decreases their performance when used as single cues. On the other hand, the incorporation of both facial expression and shoulder cues increases their single performance. Overall, the fusion of all three cues provides us with the best results, something which agrees with the theoretical expectations [84]. We should denote here that we also experimented with dimensionality reduction, but no further improvement was observed.

5.1.1 Audio Pre-processing Evaluation

At this point, we will refer to the pre-processing procedure we applied to the audio signal of the segments (Section 4.3.1.1). We remind the reader that we firstly remove the avatar speech from the recordings, while afterwards we performed a noise reduction technique. The results show that there is an increase of approximately 6% when the avatar speech is removed, while an added $\approx$ 4% is gained by denoising the clip. In general, the noise pre-processing procedure appears to improve the performance by 10%; a quite significant improvement. This confirms the literature on MFCC, which mentions that the coefficients are sensitive to noise [180], while also showing that the interfering avatar speech provides incorrect indications regarding the emotional state of the subject. It should be noted that the background speech removed was generally non-overlapping with the actual subject speech (Section 4.3.1.1).

Table 5.3: Evaluating the effects of the audio pre-processing method with 10-fold cross validation. A^{**} represents the noisy audio signal, A^{*} the noisy audio signal after the removal of the avatar speech and A the audio signal from which the avatar speech has been removed, and noise reduction techniques have been applied.

$\mathbf{A^{**}}$	$\mathbf{A^{*}}$	**A**
51.49	57.46	61.19

Table 5.4: Classification in the Likelihood space by finding the best separating line with a gradient descent algorithm. Rows are accuracy (ACC) and increase or decrease in performance comparing with maximum likelihood classification (COMP).

	F	**S**	**A**	**FS**	**SA**	**FA**	**FSA**
ACC	85.07%	74.63%	63.43%	78.36%	67.16%	79.10%	79.85%
COMP	11.94%	0.75%	2.24%	0.00%	-1.50%	8.20%	0.75%

5.1.2 Classification in the Likelihood Space

As we have mentioned, HMMs are generative models. For our two classes (positive and negative emotional states), two HMMs are generated with each one modelling the respective class. Given an observation (in our case, a sequence of frames which can be considered as a vector containing the features) each of the models will output the likelihood of itself with respect to the observation. Let us consider the two likelihoods, which are the output of the models, as points in a 2D space with each dimension corresponding to the negative/positive class trained model. Then, the maximum likelihood[1] (ML) approach with which the results mentioned in Table 5.2 were attained, bisects the 2D plane with a line (consider the line y=x), and classifies everything above the line as class 1 and below as class 2. This line is the decision boundary used.

The approach we will describe was used in [145] with HMMs (which are Dynamic Baysian Networks) and in [51, 21] with Static Bayesian Classifiers. The goal is to shift the line we described in the previous paragraph in order to achieve maximum separability between the 2D points in the space. This is justified by the assumption that the learnt distribution by the CHMMs is not a perfectly accurate description of the true distribution, and thus this classification method can be improved.

In Table 5.4, we present the new classification results after the line was shifted by using a gradient descent algorithm. The algorithm essentially optimises the percentage of correctly classified examples by searching the line parameter space [100].

It is important to note that while the overall performance increases by far, there is some strange behaviour observed in the results. Firstly, the performance of shoulder and audio cues fusion together decreases. Secondly, the fusion of all the cues and the fusion of the face and shoulder cues, which in the original results performed better than any single cues, perform worse than the facial expression cues when used alone. This behaviour led us to conclude that perhaps some more complex function or a different classification technique is required to, as optimally as possible, separate this projection into the likelihood space, especially since our data also deal with subtle emotion expressions which are in many cases quite difficult to discriminate. By observing the data, we can hypothesise that the fusion of certain sets of cues generates a more complex distribution of points in the likelihood space.

5.1.3 Likelihood Space Classification with Support Vector Machines

In order to separate using a variety of functions and a selection of parameters for the optimisation, we resort to Support Vector Machines (SVM). We have described SVMs in Section 3.2, but we recall

[1] Which we approached theoretically in Section 3.3

one of the most important advantages of SVMs: They guarantee to find the optimal separating hyperplane in the feature space (mapped from the input space by a kernel function) given of course the defined parameters, minimising the structural risk of the model (i.e. maximising the margin, the distance of the separating hyperplane from any input points). This is due to the convex optimisation function that relates to the SVMs optimisation procedure, which in turn avoids issues of local optima which manifest in algorithms such as gradient descent. Firstly, we should note that the likelihoods were normalised by dividing with 10^3, in order to be scaled for the experiments, since it is generally considered good practise to scale when using SVMs. The range after this scaling was [-35.5, 14.4]. We denote that scaling the results further, or not scaling at all provided worse results, possibly due to over-crowding the distribution when scaling too much or due to the sparsity of the points when not scaled. Again, we use 10-fold cross validation in order to properly evaluate the performance of the generated models.

Firstly, we present some of the results achieved with the use of a linear kernel. That is, the kernel maps the inner products of the optimisation problem as follows:

$$\mathbf{x}_i \cdot \mathbf{x}_j \rightarrow K(x_i, x_j)$$

Some representative results that we attained with the linear kernel are presented in Table 5.5. Firstly, the parameters that we modify are the C parameter, which as described is the "penalty" term for the errors. Recalling our discussion for SVR (3.2.2) and specifically Equation 3.47, we pointed out that as C increases, then the error term of the minimisation problem increases and thus less errors during training are allowed. As we have mentioned, there is a risk of overfitting to the training data as we decrease the C parameter. The second parameter is the parameter E which controls the tolerance of the termination condition. This function can be changed in order to stop the training when a certain performance has been obtained on the training data.

Results (a) and (b) manifest a similar pathology to the one that has appeared in the gradient descent algorithm for finding the best separating line. The facial expression cues perform better than any other combination of cues/modalities. Face and shoulder as well as the fusion of all three cues come ins second and third respectively, although their accuracy is not far off. In experiment (c), the E parameter (which stands for the error tolerance for termination) was increased to 1. This is another way to avoid overfitting: Stopping the training when a certain error performance has been reached. In this case, this parameter produced some more reasonable results: The facial expression cues still do well (while there is a 2% drop with the addition of the early stopping parameter) but now the fusion of all three cues is higher than any single cue. The face and shoulders combined do worse than the face cues but better than the shoulder cues. We haven't discussed much about the audio stream by itself since it is generally low, and the combination with shoulder cues provides no great change. The combination with the facial expression cues though provides a significant increase in all cases.

We can provide some hypotheses for discussion, on the previous observations. Firstly, there is no significant change in either the face or shoulder cues from the ML classification. This can be thought of as a justification for the fact that the fused shoulder and audio cues do not change either. The face cues alone do increase more than 10% in experiments (a) and (b). We could hypothesise that this is the reason that any other set of cues which is combined with the facial expression cues increases with these parameters. In experiment (c), the increase in the facial

Table 5.5: Experiments with classification in the likelihood space, using a linear kernel. Parameters: (a): C = 1, E = 0.001 (b) C = 1.3, E =0.001, (c) C=1.3, E=1. Rows: Accuracy (ACC), Comparison to Maximum Likelihood (COMP)

		F	**S**	**A**	**FS**	**SA**	**FA**	**FSA**
(a)	**ACC**	**84.29%**	73.19%	61.70%	**83.68%**	68.79%	**80.66%**	82.75%
	COMP	*11.16%*	*-0.69%*	*0.51%*	*5.32%*	*0.13%*	*9.76%*	*3.65%*
(b)	**ACC**	**84.34%**	74.62%	61.65%	**83.68%**	68.79%	79.95%	83.52%
	COMP	*11.21%*	*0.74%*	*0.46%*	*5.32%*	*0.13%*	*9.05%*	*4.42%*
(c)	**ACC**	82.03%	**74.67%**	**62.53%**	80.77%	**69.51%**	79.23%	**86.54%**
	COMP	*8.90%*	*0.79%*	*1.34%*	*2.41%*	*0.85%*	*8.33%*	*7.44%*

expression cues performance is less than in the first two experiments, but still high (8.9%). The improvement of the audio cues is now is higher and this seems to apply for the fusion of all three cues as well. One assumption that we may state here is that because the audio cues have the worst performance, any small improvement on it would positively affect the fusion of other cues with it. This is also observed when fusing the audio and shoulder cues, but it is balanced off by the facial expression performance decrease when fusing the facial expression cues and the audio cues.

Of course, all these assumptions are risky and can not be further justified. More evaluation data are required in order to evaluate whether these results generalise to other data sets. We should make it clear that the best line chosen is different for each combination of cues. It does not necessarily mean that when an improvement has been presented for a set of cues (e.g. the Audio cues) this improvement will generalise when the cues are fused with another set of cues. This fusion will add a new set of 2D points to the set of likelihood points produced by the set of cues in question, and the new line will depend on this augmented set. Also, we should denote how the early stopping parameter in experiment (c), which was increased, helped to balance the increase across the fusion of cues. We can say that the face cues increase more when the training data are fit well: this means that the facial expression cues have a well defined distribution with respect to linear separation and can classify unknown data accurately when fitted to the training data. Unavoidably, the fusion of all cues produces a more complex distribution, thus the early stopping parameter causes a further increase.

We then experimented with a polynomial kernel function of degrees 2 and 3. Reminding from Section 3.2.2, the polynomial kernel function is defined as:

$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x_i} \cdot \mathbf{x_j} + 1)^d$$

We experimented with various parameters. We present two representative results in Table 5.6 for reference, although in general the polynomial kernels did not provide any significant improvement over the linear one. It is important to note that the CPU time for classifying any classification on cues which included the audio cues was extremely longer than the rest of the combinations, especially the third degree polynomials. This is an indication that our audio data are quite difficult to fit by 2nd and 3rd degree polynomials. The results though were not much different. We will not discuss polynomial kernels any further, and we will move on to classification with the Radial Basis Functions (RBF).

Table 5.6: Experiments with a polynomial kernel for classification in likelihood space. (a) A polynomial of degree 2 (C=1, E=0.001) and of degree 3 (b) (C=1.3, E=0.1) Rows: Accuracy (ACC), Comparison to Maximum Likelihood (COMP)

		F	S	A	FS	SA	FA	FSA
(a)	**ACC**	70.22%	**74.67%**	61.92%	55.93%	**68.74%**	**81.54%**	**83.57%**
	COMP	*-2.91%*	*0.79%*	*0.73%*	*-22.43%*	*0.08%*	*10.64%*	*4.47%*
(b)	**ACC**	**82.20%**	71.04%	**62.58%**	**82.20%**	67.25%	75.44%	**83.57%**
	COMP	*9.07%*	*-2.84%*	*1.39%*	*3.84%*	*-1.41%*	*4.54%*	*4.47%*

The radial basis function is defined as:

$$K(\mathbf{x}, \mathbf{x}') = e^{-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}} = e^{(-\gamma\|\mathbf{x}-\mathbf{x}'\|^2)}$$

The radial basis function is a function that essentially depends on the parameter γ which essentially regulates the width of the RBF function. We should point out here, that a RBF is considered to be able to produce results *at least as good* as a linear kernel [93], given though a certain combination of parameters. This is due to the high parameter space mapping that RBFs provide, thus providing them with the capability to learn highly non-linearly correlated relationships.

For RBF, we firstly performed a coarse grid search [93], for:

$$p \in 2^i, i \in \{-5, 5\}$$

where p is the parameter in question (both C and γ) [2]. These experiments were combined with 3 early stopping options:

$$E = \{0.001, 0.01, 1\}$$

After attaining the best results, we performed a more dense and detailed grid search in the same fashion, in order to attain the optimal results.

Our goal here is to find the best mapping for each of the combination of cues. It is obvious that each different combination of cues produces a different distribution of points in the 2D likelihood space. Thus, we assume that we need to tailor the parameters of a learner specifically for the distribution of points that characterises the specific combination of cues. Again, we evaluate our results using 10-fold cross validation.

In Table 5.7, we present the results of the best RBF for each combination of cues along with the parameters associated with the SVM. In each row, we present the best RBF found for the respective single/fused cues, which label the first column (i.e. the first row maximises the facial expression cues (F), the second row the shoulder cues (S) and so on). In Table 5.8, the best performing RBF kernel for each combination of cues (i.e. the diagonal of the first 8 columns of Table 5.7) is compared with maximum likelihood classification and the best line attained from the gradient descent procedure.

As we have previously commented, each function is tailored for the distribution produced by the specific combination of cues, ignoring the performance on the other combinations while testing

[2] Typically for SVR experiments, it is advised that the parameter C should be found by experimentation and then the rest of the parameters can be fine-tuned

Table 5.7: Applying RBF kernels for classification in the likelihood space. In each row, one RBF kernel is presented for which the parameters are specified in the last three columns (γ for the RBF, C the error penalty and E the early stopping parameter). The RBF kernel presented in each row targets to optimise the performance on a single combination of cues (specified in the first column for each row). The diagonal (of the first seven columns) lists the best results for each combination of cues.

OPT	F	S	A	FS	SA	FA	FSA
F	**87.53%**	75.44%	55.71%	80.66%	67.20%	79.95%	82.86%
S	83.02%	**79.07%**	55.00%	80.66%	65.00%	79.12%	82.14%
A	78.41%	72.47%	**66.92%**	73.08%	60.66%	74.67%	66.48%
FS	81.37%	73.96%	52.91%	**85.05%**	65.71%	82.20%	82.91%
SA	83.74%	71.54%	49.78%	82.86%	**74.07%**	79.95%	83.57%
FA	82.20%	65.44%	58.02%	77.69%	68.02%	**86.65%**	82.80%
FSA	79.07%	71.76%	48.46%	81.48%	60.77%	79.23%	**88.19%**

OPT	AVG	γ	C	E
F	**85.19%**	0.16	31	1
S	82.58%	0.28	5	1.5
A	72.45%	**8.65**	28	1
FS	82.14%	0.07	3	1.1
SA	83.65%	0.16	5	1.3
FA	82.50%	0.05	25	1.5
FSA	83.63%	0.03	4	1.05

Table 5.8: The performance of the best RBF kernel for each combination of cues using SVMs, compared to the maximum likelihood results (ML) and the best line found by the gradient descent algorithm (BLGD). All recognition rates increase for RBF

	F	S	A	FS	SA	FA	FSA	AVG
RBF	**87.53%**	**79.07%**	**66.92%**	**85.05%**	**74.07%**	**86.65%**	**88.19%**	**87.86%**
ML	73.13%	73.88%	61.19%	78.36%	68.66%	70.90%	79.10%	76.12%
BLGD	85.07%	74.63%	63.43%	78.36%	67.16%	79.10%	79.85%	82.46%

and training. Nevertheless, we will provide some brief comments based on Table 5.7. Firstly, it is interesting to note that the RBF which maximises the fusion of all cues/modalities, also maximises the averaged accuracy of the classification across all cues. Secondly, we denote that the average (AVG) accuracy for each RBF is better than the average accuracy of maximum likelihood classification (76.12%), except for the RBF which optimises the audio cues (3rd row). This is an indication that the distribution produced by the audio cues across the folds is quite different and difficult to classify than the rest (this can be also noted by the low recognition accuracy of the audio cues).

The difference in the γ parameter (which is always greater than zero) of the RBF kernel function for the maximisation of the audio cues recognition rate is easily observable. For the rest of the 6 combinations, the parameter takes no value above 0.3, whereas for the audio cues the parameter is 8.65. We mentioned before that the γ parameter functions as a regulator to the width of the RBF function. The bigger the gamma, the more prone to overfitting our method becomes as it will follow the distribution of points more closely. Decreasing the gamma causes a smoother surface for the function. In conclusion, our explanation here is that the other combinations of cues were easier to separate: Thus, a big γ caused the training to overfit to the training data, causing more misclassifications on the testing data. The audio cues were in any way difficult to classify and probably had a distribution for which the respective correct classifications for the testing data were difficult to be predicted. Increasing how closely the function follows this distribution by increasing the γ parameter provided a separation which covered the peculiarities and specific characteristics of the already ill-behaving points.

Furthermore, we denote that the early stopping parameter E which was selected for each set of cues was equal or above 1. Compared to the default 0.001 which is typically the default option for libraries implementing SVM such as svmlib [33], shows that it is easy for the classifier to overfit to the training data, demonstrating the noise present in the data. Of course, the characteristics of our data which also contain spontaneous subtle emotion expressions do provide a difficult problem for any learning technique. Finally, commenting on Table 5.8, we denote the superiority of the cue-specific RBF functions against the other approaches. Classification with RBF functions achieves results as high as 88.19% for the fusion of all modalities, whereas line separation by gradient descent and maximum likelihood alone provide results of 79%-80% for the corresponding cue combination.

In Table 5.9, we present the performance of SVR-RBF against maximum likelihood (ML) and best line found by gradient descent (BLGD) per fold, for the fusion of the facial expression/shoulder/audio cues. For all folds except the first one the increase provided by SVR-RBF overperforms or is equal to the other methods, reaching accuracy over 90% in 6 out of 10 folds and 100% accuracy in 2. It is also interesting to observe that in 4 folds the RBF and BLGD accuracy is the same.

Finally, we should denote that the maximum results attained for RBF do appear to be explainable by the theory related to emotion perception: The facial expression cues provide the best single cue recognition, followed by the shoulder and audio cues. Fusing the facial expression cues with any of the other single cues provides a small drop compared to the accuracy of the facial expressions, but provides a much bigger recognition rate than the other cues as single cues. Finally, the fusion of all cues and modalities provides us with the best possible results. We should note here that one

Table 5.9: SVR-RBF compared to maximum likelihood (ML) and best line found by gradient descent (BLGD) for the face/audio/shoulder cues fusion for each of the 10 folds.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
RBF	71.4%	92.9%	92.9%	78.6%	84.6%	92.3%	100.0%	92.3%	76.9%	100.0%
ML	85.7%	92.9%	92.9%	57.1%	69.2%	76.9%	61.5%	92.3%	76.9%	84.6%
GD	92.9%	92.9%	92.9%	64.3%	53.8%	92.3%	84.6%	61.5%	69.2%	92.3%

explanation for the slight drop when fusing the facial expression cues with another set of cues should be due to each cue providing conflicting classifications for a specific sequence. This does not come as a surprise from our data, where we have observed persons smiling while being ironic (thus negative), persons smiling while shouting in anger and so on.

5.2 Continuous Emotion Recognition

Beginning our description of the continuous emotion recognition experiments, it should be noted that in this chapter, by the term *ground truth*, we are referring to the resulting ground truth produced by the segmentation procedure (Section 4.2) after the annotations were interpolated (Section 4.1.4). Although as we have mentioned we kept results which were normalised by different procedures and not interpolated, finally we decided that our experiments would be conducted with the ground truth which was:

- Normalised locally to have mean equal to zero
- Interpolated using cubic interpolation
- Generated by the segmentation procedure described in Chapter 4

As far as the averaged metrics we present in this section are concerned, the error is calculated on average for 10 folds, for all training sequences (audiovisual segments) available (i.e. per sequence) or per fold (for all frames in fold).

For per sequence, assuming that we have $n = 10$ folds, that the function $t(f)$ returns the testing sequences for the f-th fold (with $|t(f)|$ begin the number of such sequences) and that $fr(k)$ returns the number of frames for sequence k, then the calculation of an averaged metric per sequence is defined as:

$$\text{Metric}_{ps} = \frac{1}{10}\sum_{f=1}^{10}\frac{1}{|t(f)|}\sum_{s\in t(f)}\frac{1}{fr(s)}\sum_{j=1}^{fr(s)}e(s,j)$$

with $e(s,j)$ being the specific error metric we are using, f ranges through all 10 folds, s over the sequences and j is an index to the frame number of the specific sequence s. The mean squared error (MSE) is a typical metric used for regression problems and also has been used for continuous emotion recognition [199]. This metric is attained by replacing the e function above with:

$$e_{MSE}(s,j) = (T(s,j) - E(s,j))^2$$

where $T(s,j)$ and $E(s,j)$ stand for the ground truth and estimated values for sequence s at frame j respectively. We refer to the above as the averaged mean squared error. The metrics that will be

Table 5.10: Mean Squared Error (per sequence), Correlation Coefficient (COR) and agreement (AGR) (Equation 4.1) of each coder (COD) for valence and arousal with respect to the ground truth.

	Valence			Arousal		
COD	**MSE**	**COR**	**AGR**	**MSE**	**COR**	**AGR**
JD	0.018	0.464	0.867	0.020	0.544	0.787
cc	0.023	0.440	0.798	0.020	0.410	0.823
dr	0.021	0.489	0.835	0.025	0.616	0.834
em	0.017	0.418	0.875	0.019	0.448	0.852
AVG	**0.020**	**0.453**	**0.843**	**0.021**	**0.505**	**0.824**

noted "per fold", essentially do not average per sequence, as described in the above equation but rather average over all the frames in the given fold:

$$\text{Metric}_{pf} = \frac{1}{10}\sum_{f=1}^{10}\frac{1}{fr(f)}\sum_{j=1}^{fr(f)} e(s,j)$$

where now the fr function is assumed to return the number of frames per fold.

It is noted that throughout this chapter, the result tables will typically include the following metrics, assessing the estimated values with respect to the ground truth: the mean squared error metric (MSE), the correlation coefficient (COR) and the agreement (AGRE, Equation 4.1). When a metric has a subscript of $_{pf}$, then the metric is calculated per fold. Otherwise, it is calculated per sequence.

Firstly, we provide some statistics which relate the individual coder annotations to the actual produced ground truth. In Table 5.10 we present some metrics which relate the produced ground truth to the coder values, after they have been processed. We can see that the MSE is 0.02 for both valence and arousal, while wee also denote the correlation average between the coder values and the ground truth, as well as the standard deviation of the coder values. The correlation values for valence can be found along with the correlations of all the coders as evaluated in pairs in Table 7.2 which is found again to be a little over 0.5. Of course the table statistics refer to the entire video session, by ignoring NaN values. Finally, we denote that we will use the average values for the metrics (Table 5.10) to refer to the human error in providing such annotations, values which we will finally use as a baseline to compare with our final results.

5.2.1 Configuration of Support Vector Machines & Long short-term memory neural networks

We will now refer to our experiments by specifying the configuration under which they were performed. Firstly, with using Support Vector Machines for Regression, as described in 3.2.2. In our experiments, we focused on using polynomial and radial-basis functions (RBF) kernels, as described in 3.2.1.1. We performed various experiments with optimising the parameters used, namely C, the penalty term for errors and γ for the RBF, while also the degree of the polynomial used for learning with a polynomial kernel. By the outcomes of the experiments, we found that the best results for polynomial kernels were given by polynomials of degree one, and for RBF

with a parameter $\gamma = 0.04$. The values which gave rise to the best results for the cost C were 0.1 and 0.15 for almost all the experiments, with some exceptions such as the arousal experiments with the audio cues, which provided better results with a C of 1. This is can be justified by considering that the audio cues are generally considered to perform good recognition for arousal, and in conclusion allowing less outliers by increasing the cost gives better results.

We also experimented with the termination conditions for the training error E, with the best results attained by the $E = 0.01$ and 0.2. So far, these are parameters that we have mentioned again in our previous experiments with SVMs (Section 5.1.3). We remind that since we are using ϵ-insensitive regression (Section 3.2.2), which is based on the ϵ-insensitive loss function (Equation 3.48), we essentially need to define the area in which outliers won't be charged with the penalty term. That is, the ϵ stands for the distance in which our estimation can be from the actual ground truth and yet not be charged. We experimented with various values of ϵ, and by assessing the results we finally attained better results with $\epsilon = 0.2$ for the RBF, while for the polynomial kernels $\epsilon = 0.1$ gave better results in some experiments (e.g. arousal for the feature-level fused face and shoulder cues).

As far as LSTM networks are concerned, we experimented with various configurations for each of the combination of cues. Networks we experimented on were bidirectional LSTM with forget gates and peephole connections (Section 3.1.4), typically contained from one to three hidden layers and a selection of nodes from 1 to 150. Our best results though were usually attained by using small networks of 1-10 nodes, while at some cases we achieved good results with networks of up to 50 nodes. Regarding the number of hidden layers, our best results were usually attained with networks of one hidden layer, but in other cases 2 and 3 layers gave good results. We experimented with both using a validation set and other criteria for early stopping, such as the performance on the training set.

5.2.2 Mean Squared Error Evaluation & Feature-Level Fusion

As the title reveals, this section will describe a set of experiments with feature-level fusion: That is by combining all the features from multiple cues into one feature vector which is then fed into a network. Notice that the audio stream has a double frame rate with respect to the actual video. Thus, for feature level fusion, when fusing the audio features with features from either the shoulder or the facial expression cues, the vector which is fed to the classifier is as follows,:

$$(a_1, v_1)_1, (a_2, v_1)_2$$

where each $(a_i, vj)_k$ is the k-th feature vector, a_i the features for audio frame i and v_j the features for video frame j. In other words, each vector of features from a video frame is repeated twice. The ground truth for the audio cues is then used for training and evaluating.

We present the best results with respect to the MSE for both polynomial and RBF kernels in Table 5.11. It should be noted that for a lot of combinations of cues for both valence and arousal, the polynomial kernel performs slightly better in most of the cases. On the previous matter, we should further note that in most cases the difference between the MSE of each kernel is very small (0.001 or 0.002) and perhaps insignificant. In conclusion, based on the MSE we can not draw

any significant conclusions on which function outperforms another. We will though explore other metrics in the following sections.

In Table 5.12, the results of our LSTM experiments along with the best results of the SVR technique are presented. In general, the LSTM technique outperforms the SVR methods in terms of the MSE. This was an expected outcome, since SVRs are static methods which do not perform sequence learning, while LSTMs are a dynamic technique specific for sequence learning and especially in learning temporal dependencies, which are obviously related to the expression of emotions in humans (e.g. temporal features, Section 2.2.5). Specifically in valence values, LSTMs are only outperformed in one case, that of the shoulder and audio cues fusion. But in general, our conclusions should again be careful since the MSE difference of 0.01 is very small. As far as arousal is concerned, the LSTMs provide results which are quite better in general, with a maximum improvement of performance from 0.87 to 0.77 for the shoulder and audio fused cues.

The audio cues are theoretically expected to do better [84] than the visual cues for regression on arousal values. In our experiments, always with respect to the MSE the audio cues do perform significantly better for arousal. It is also important to note that the fusion of the audio cues with the facial expression cues improve the 0.09 MSE attained with just the facial expression cues to the minimum MSE, previously only attained by the audio cues (0.084), while when fused with the shoulder cues they again better their single performance, by reducing the MSE from 0.093 to 0.092. For the fusion of all cues/modalities, the results are not better than the audio cues alone or the audio/face fusion but are better than the single face and single shoulder cues by themselves. For both the RBF and polynomial kernels, we can notice that the audio cues perform better or as good as any other set of cues, fused or not, with respect to the MSE.

Shifting the arousal discussion to the LSTM results, the audio clearly outperforms all other single and fused cues, while again improves the single cue performance of shoulder/facial expressions when fused. For the fusion of all cues/modalities though, this is not the case, as the error increases. In general, we have not observed the fusion of all cues/modalities to perform better than all the single or couple-fused cues in none of the cases. This could be related to topics such as the fusion method (as mentioned, the optimal fusion method is an open research issue) or the larger dimensionality presented when the cues are fused.

Regarding the valence values, the visual cues are expected to perform better than the audio cues [84]. We can not say that this is always the case for our experiments. For the polynomial kernel, the shoulder cues seem to perform slightly better than all the rest fused/single cues, while for the RBF kernel the audio cues perform quite better than the rest, evened only by the fusion of the facial expressions with the audio cues. The facial expression attain a slightly worse results and the shoulder cues perform worst as single cues. In the RBF kernel results, the fusion of modalities worsens the single-cue results for the fusion of shoulder and audio cues, while for the fusion of all cues the results are essentially the average of the results of each of the fused cues separately. Regarding LSTM, the facial expression cues and the audio cues perform the best, with the shoulder cues performing slightly worst. The fusion does not provide any improvement on the total minimum MSE error achieved.

At this point we should state that one possible factor that attributes to the at-least-as-good performance of the audio cues with respect to the visual cues is that the ground truth values and the training sequence lengths for the audio cues are double than the ones for the visual data. This es-

Table 5.11: SVR best results for polynomial (P) and radial-basis (RBF) kernels for both valence (V) and arousal (V)

$\mathbf{MSE}_{ps}$	F	S	A	FS	SA	FA	FSA	
V	**0.054**	**0.053**	0.054	**0.054**	**0.054**	0.054	**0.054**	**P**
A	**0.088**	**0.087**	0.087	**0.088**	**0.087**	0.088	0.088	
V	0.054	0.057	**0.053**	0.056	0.062	**0.053**	0.055	**RBF**
A	0.090	0.093	**0.084**	0.091	0.092	**0.084**	**0.087**	

Table 5.12: Support vector regression compared with LSTM networks for valence (V) and arousal (A)

$\mathbf{MSE}_{ps}$		F	S	A	FS	SA	FA	FSA
SVR	**V**	0.054	**0.053**	0.053	0.054	**0.054**	**0.053**	**0.054**
	A	0.088	0.087	0.084	0.088	0.087	0.084	0.087
LSTM	**V**	**0.052**	0.054	**0.052**	**0.053**	**0.054**	**0.053**	**0.054**
	A	**0.080**	**0.084**	**0.072**	**0.084**	**0.077**	**0.081**	**0.086**

sentially translates to having double training data while learning. The amount of training data is quite significant for every learning task and especially when we are performing sequence learning with LSTM, which are considered to be able to learn long range dependencies. Since the sequences are longer (double the size), then the LSTMs can pick up more dependencies in the input features. We will further explore this hypothesis of ours in Section 5.2.5, where we will show that extracting audio features for a frame rate equivalent to the one for which video frames where extracted, causes a great worsening in the performance of the audio cues. This is a powerful indication that the length of the training sequences for sequence learning is crucial and has a tremendous impact on the results.

Before proceeding into further discussion on the topic, we should consider whether the MSE, typically used for evaluating regression problems, is the metric that should alone be used in order to evaluate the performance of a learning technique on continuous emotion recognition. We should emphasise that our regression problem has a set of domain specific characteristics. For example, even if the MSE is higher in a certain case, perhaps metrics such as the correlation of the ground truth with the estimated values as well as other factors, such as the agreement (Equation 4.1, Section 4.1.3), which as we remind is a function that returns the percentage of cases where two values agree in their sign - and in our case amounts to the percentage where the estimated value agrees with the ground truth value on classifying the emotional state of the subject on a particular frame as positive or negative, seem to be very important to be overlooked for the sake of the MSE.

5.2.3 Beyond the Mean Squared Error

In this section, we will provide some discussion on certain pathologies that have manifested in our experimental results. Firstly, to describe problems that arise when ranking our results only by the MSE, we will to Fig. 5.1. In the figure, we can see the estimated and ground truth arousal values of one fold over n-fold cross validation for SVR with a polynomial (a) and RBF (b) kernel, as well as the corresponding LSTM network (c). What we observe in the plots, is that the best

performing technique in terms of the MSE is the polynomial SVR kernel. Intuitively, one could say that it is the worst though. A polynomial kernel does not seem to be able to generalise enough to capture the temporal behaviour of the arousal values across the frames, and is limited to a narrow band located at the mean of the entire ground truth distribution, not corresponding to peaks of the actual distribution since it does not really predict any peaks (and thus avoids making large valued errors).

To better describe the distributions, we will make use other metrics. Firstly, besides the correlation and the agreement[3] previously described, another metric that could be of interest is the derivative of the valence and arousal values against the frames. This metric essentially refers to the smoothness of the results and the mean squared error between the estimated and ground truth values can be obtained as an indication of the how different the curves that are defined are. This is similar to how the gradient is used to provide smoothness criteria in computer vision (e.g. active contours [103]). For example, for valence we would have:

$$DER(\mathbf{V}) = \frac{\Delta V}{\Delta f}$$

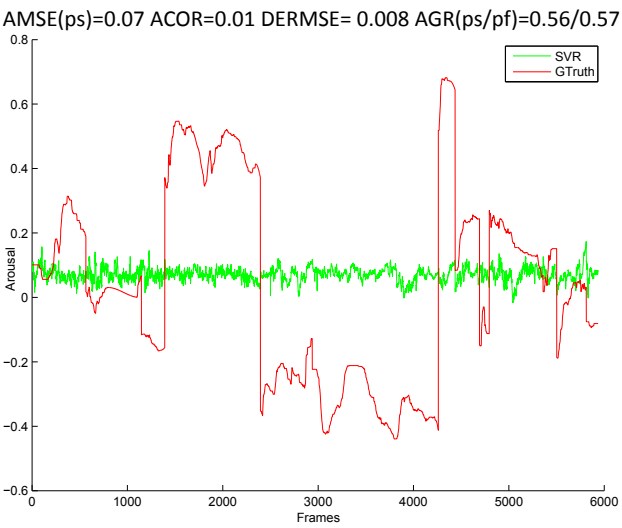
where V are the estimated/ground truth valence values and f the frames that belong to the sequence. The results for each sequence frame can be summed (by taking the absolute values to compensate for different signs) and then the mean squared error between the acquired absolute sums of the derivatives can be obtained.

Notice how this introduction of other metrics helps us discriminate the estimated values. Firstly, the correlation: The plot with the minimum MSE has the less correlation of all (0.01). The RBF kernel SVR has a much greater correlation of 0.3 with the ground truth, while the LSTM has the best correlation of 0.6. Although not as clear, the same ranking can be considered for the agreement values, where the RBF is better than the polynomial kernel and the LSTM better than SVR methods in general. We should note though that through our experience, a rise in the correlation does not necessarily imply a rise in the agreement values, as situations where the contrary occurred have been observed. Furthermore, the mean squared error derivative gives us an estimation of the error in terms of smoothness of the produced prediction curve with respect to the smoothness of the ground truth. It can be observed that the polynomial kernel and the LSTM have this error metric quite small, while the very spiky distribution produced by the RBF kernel with respect to the ground truth produces a large derivative error.

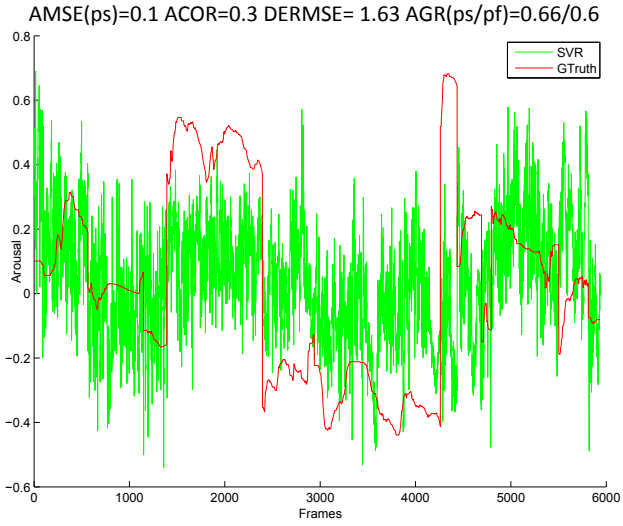
In conclusion, polynomial kernels (with a degree of one)[4] have been deemed as better performing in our experiments with respect to RBF kernels in certain occasions, but that they essentially produced distributions such as the one portrayed in Fig. 5.1, close to the mean with a narrow band. Despite the dominance of the LSTMs in the previous scenario, there is a corresponding pathology manifesting in these networks as well. In some cases, the MSE obtained when the LSTMs produced estimations with a distribution similar to the one produced by the polynomial kernel in Fig. 5.1 were better than the performance obtain at later stages, while the network fitted the training examples. A visualisation of such a case is depicted in Fig. 5.2, along with another network which performs well for the same fold. Notice that the MSE is quite small in both folds.

[3] We clarify that although we described the agreement metric in terms of valence, we also apply the metric in terms of the arousal, as in the agreement on whether the emotional state of the subject is considered passive or active

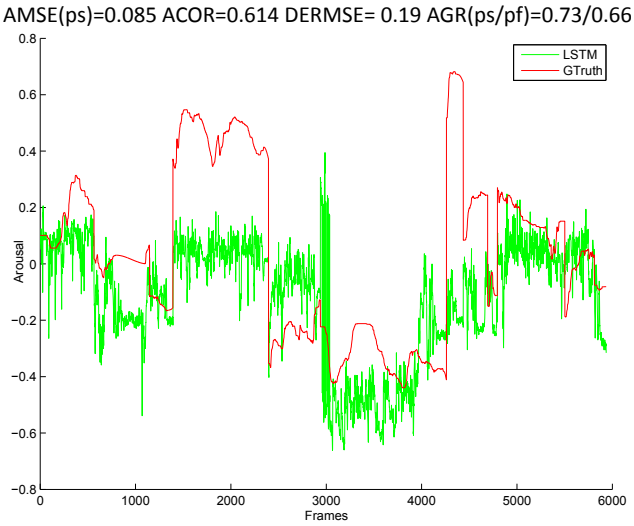
[4] First degree polynomials were also reported to perform better than other degree polynomials in [117]



(a)

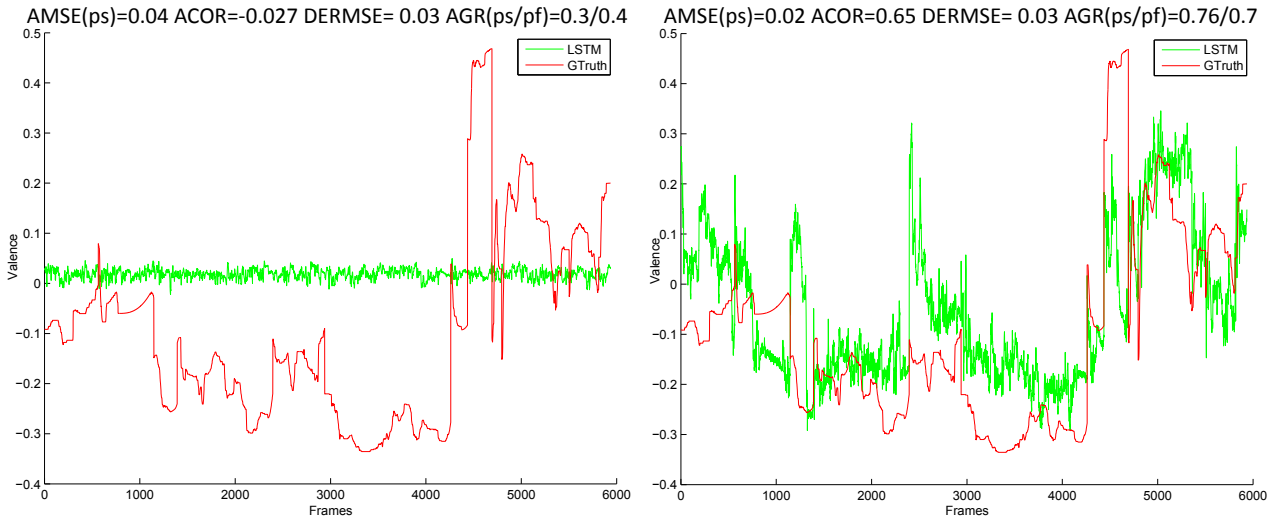


(b)



(c)

Figure 5.1: Pathologies of the MSE: In this figure we can see the arousal estimated and ground truth values (for one fold) of: (a) SVR with a polynomial kernel (b) SVR with an RBF kernel (c) LSTM. The metrics depicted are: Averaged mean squared error per sequence (AMSE), averaged correlation per fold (ACOR), derivative averaged mean squared error (DERMSE) and agreement per sequence and per fold respectively (AGRps/pf).



(a) (b)

Figure 5.2: A fold for valence estimation, using LSTM networks. In (a), a network achieves the best test MSE in the first few epochs, producing an unnatural estimation with a very low correlation. (b) A network providing a good estimation for the same fold. Notice that the correlation for this fold is 0.65.

The issue we described is in fact still an open research question in the field of dimensional emotion recognition [84]. The MSE has been used by research work being done with LSTM and SVR [199], while other work also including SVR [117, 101] used the mean error and the correlation coefficient. From our experimentation, we believe that the MSE, the correlation and the agreement level are typically characteristic evaluation metrics for the emotion estimation distribution. We have noted that a rise in correlation does not always imply a rise in the agreement level (and vice-versa) since such scenarios were observed during experimentation. These two metrics should be used in conjunction with the MSE for a better evaluation of the estimation. Furthermore, not as important but discriminative is the mean squared derivative error, which is an indication of the smoothness variation of the prediction with the ground truth. A measure that can similarly be used is the mean error in the variances of the values at hand, although the last two metrics come second with respect to the MSE, the correlation and the agreement.

To elaborate on how the correlation coefficient and the agreement are related, we will refer to a last example, presented in Fig. 5.3. It can be observed that the correlation for this fold is significantly low: At 0.03, it competes with the 0.01 correlation of the polynomial kernel which is essentially approximating the mean of the distribution (Fig. 5.1) [5]. The estimated values presented in Fig. 5.3 though, are more naturalistic, approximating in some cases the peaks of the ground truth. Despite this, the MSE also points to the direction of choosing the SVR-P (Fig. 5.1) which has a MSE of 0.07 in contrast to 0.11 of the plot in Fig. 5.3. Contradicting (and perhaps restoring the balance) the previous metrics, the agreement per fold for 5.3 is 0.75. much higher than the 0.4 of the SVR-P estimation, and thus we have a good indication that despite the significantly low correlation, this predication is not useless. Again, there is no agreed evaluation function that considers all MSE, correlation and agreement in order to allow us to rank these results generally. From what follows, we will not pay much attention to the MSE variances when they are approximately equivalent,

[5] We are comparing different ground truth values and different folds, but the essence of our description lies in general characteristics of the prediction not a specific comparison

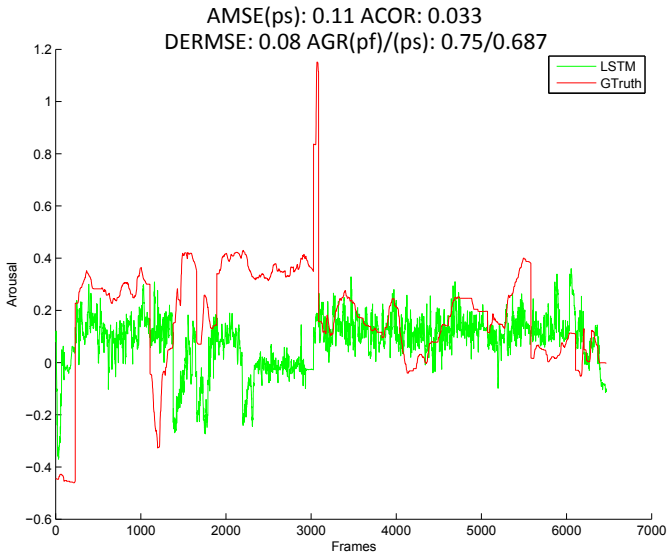


Figure 5.3: An example of an arousal fold where the correlation is extremely low but the agreement levels are quite high

i.e. we will mostly pay attention to the two first decimal points. We will though pay much more attention to the correlation, which seems to be highly descriptive of the estimator's performance. We should note that we make use of both *per fold* and *per sequence* metrics in order to capture both long and short range characteristics of the estimation.

5.2.4 MSE, Correlation & Agreement Evaluation

Following our previous observations, we re-evaluated our experiments based on more than just the MSE. More specifically, we rejected experiments that although provided a good MSE, did not provide such a good correlation or agreement with the ground truth. The difficulty in this case is that there is no predefined optimisation function upon which to base the ranking of the results and would take into account the three aforementioned metrics. Such a function could e.g. impose a threshold upon the correlation and reject any experiments which rank below that threshold or even attempt to maximise the weighted average of the metrics in question. Here, the weights would balance the relative importance amongst the metrics (a trade of).

In this section, we will provide results for the re-evaluation of our experiments, favouring especially the correlation and secondly the agreement levels, and tolerating a small MSE increase in compensation. In this case, we were willing to trade-off a small MSE increase for correlation and agreement in order to avoid issues as described in Section 5.2.3.

Results for our experiments for valence estimation are presented in Table 5.13, while the corresponding results for arousal presented in Table 5.14. Firstly, commenting on the MSE, we observe that it has generally been limited to values of 0.071 and below for valence, while the worst MSE for arousal is 0.098. Observing the correlation results, the first comment relates to the SVR-P results. Notice how the polynomial SVR kernels produce the minimum correlation with respect to the other classification methods, while for arousal the correlation even acquires a negative value. We now have a metric beyond the MSE which can discriminate amongst our methods, and a negative correlation with the test data is certainly not a good sign for the prediction. We refer the reader to Fig. 5.1 and Section 5.2.3 were this behaviour was discussed.

We will proceed with some observations on the valence results. Specifically for LSTMs, we can observe that the largest correlation values were achieved for the single audio and face cues. We note the low correlation generated for the fusion of all cues and for the fusion of the face/shoulder cues. For SVR-RBF, the best values are attained for the fusion of all cues and the fusion of the face and audio cues. This is actually a case where the audio cues perform worse than their fusion with the facial expression cues. Also, it should be noted that the facial expression cues *outperform* the audio cues. This is a very important observation. As we have aforementioned, the facial expressions are considered to provide better results for valence estimation. This is experimentally confirmed in our case as well. Furthermore, we should also notice how this is not the case for LSTMs. In LSTMs, the best correlation is provided by the audio cues. This is an indication of our earlier comments on the justification of the performance of the audio cues (Section 5.2.2), being strongly influenced by the larger size of training data for the audio cues. We should also note how the shoulder cues have a low correlation and agreement in both the SVR-RBF and LSTM.

Proceeding with the valence evaluation, we can see that for LSTMs the cues which achieve the best correlation also achieve the best agreement per fold and per sequence. When rounded to the first decimal place, the agreement per sequence never drops below 0.5, i.e. at least half the positive/negative emotional state estimations with respect to the ground truth are correct. In SVR-RBF, the agreement per fold and per sequence is quite similar for both face and audio cues, while the maximum correlation is achieved with the fusion of all cues: a very important observation for the information provided by modality fusion for emotion recognition. Commenting on the SVR-P results, we should state that in accordance with the correlation being very low, the agreement results attained are in general worse as well. Finally, we denote the low performance with respect to the correlation of the shoulder cues. As far as the mean squared derivative error is concerned (DMSE), it provides us with good information with respect to the structure of the predictions in relation to the ground truth but does not seem to contain any further indications that would assist in the evaluation in any other way. It does though discriminate the SVR-RBF from the rest methods, since the spiked distribution[6] produced by the algorithm provides the largest MSE on average. In general, the best correlation achieved by the LSTMs is 0.4 by the audio cues, with a corresponding agreement of 0.7. It is is the best combination of these two metrics attained all over the experiments with both LSTM and SVR. For RBFs, the best correlation has been achieved by the fusion of the face and audio cues (1.9). A further comment, is that for the feature level fusion of all cues the SVR-RBF provided results with a higher correlation than the respective LSTM networks for the same cues.

The arousal results demonstrate the clear dominance of the audio cues for arousal estimation. In both SVR-RBF and LSTM results, the audio achieves the best correlation, of 0.376 and 0.511 respectively, again demonstrating the crucial characteristic of LSTMs to capture long range temporal dependencies. In both classifiers, the fusion of the audio cues with either the shoulder or face cues provides the second best results with respect to the correlation, while the fusion of all cues negatively influences the resulting correlation. It is important to note though, that in this case the LSTM results for the fusion of all cues is better than the respective one for SVR-RBF. A characteristic that is constant for both valence and arousal is the low correlation produced by the shoulder cues. Regarding the agreement values, again the superiority of the audio cues for arousal recognition are confirmed, by achieving 0.51 correlation and 0.75 agreement accuracy for

[6] A justification for the spiked distribution produced by SVR-RBF is that they are static learning methods

Table 5.13: Valence results for LSTM & SVR, considering the mean squared error (MSE), the correlation per fold COR_{pf}, the derivative mean squared error ($DMSE$) and the agreement per fold (AGR_{pf}) and per sequence (AGR). The two best values for each metric (per method) are presented in bold.

Valence						
		MSE	**COR_{pf}**	**DMSE**	**AGR_{pf}**	**AGR**
LSTM	**F**	0.067	**0.205**	0.011	**0.551**	**0.552**
	S	0.056	0.026	**0.001**	0.438	0.467
	A	**0.052**	**0.397**	0.197	**0.672**	**0.674**
	FS	**0.053**	0.050	**0.003**	0.485	0.512
	SA	0.071	0.154	0.070	0.516	0.526
	FA	0.067	0.150	0.062	0.520	0.537
	FSA	0.059	0.076	0.024	0.441	0.475
		MSE	**COR_{pf}**	**DMSE**	**AGR_{pf}**	**AGR**
SVR-RBF	**F**	0.054	0.159	**0.092**	0.516	**0.549**
	S	0.057	0.059	0.125	0.477	0.496
	A	**0.053**	0.124	0.265	**0.518**	0.542
	FS	0.056	0.115	**0.100**	0.494	0.518
	SA	0.062	0.091	0.942	0.509	0.520
	FA	**0.053**	**0.186**	0.202	0.510	0.542
	FSA	0.058	**0.161**	0.262	**0.526**	**0.547**
		MSE	**COR_{pf}**	**DMSE**	**AGR_{pf}**	**AGR**
SVR-P	**F**	**0.054**	0.025	0.003	**0.456**	0.490
	S	**0.053**	0.028	**0.002**	0.423	0.465
	A	**0.054**	0.017	**0.001**	0.412	0.450
	FS	**0.054**	0.028	0.005	**0.465**	**0.494**
	SA	**0.054**	**0.029**	0.003	0.417	**0.456**
	FA	**0.054**	**0.029**	0.003	0.417	0.456
	FSA	**0.054**	**0.035**	0.005	0.455	0.481

Table 5.14: Arousal experiments for LSTM & SVR, considering the mean squared error (MSE), the correlation per fold COR_{pf}, the derivative mean squared error ($DMSE$) and the agreement per fold (AGR_{pf}) and per sequence (AGR). Again, the two best values for each metric (per method) are presented in bold.

Arousal						
		MSE	**COR_{pf}**	**DMSE**	**AGR_{pf}**	**AGR**
LSTM	**F**	**0.080**	0.237	**0.008**	0.681	**0.693**
	S	0.098	0.041	**0.003**	0.524	0.522
	A	**0.072**	**0.511**	0.153	**0.747**	**0.720**
	FS	0.098	0.174	0.108	0.592	0.617
	SA	0.084	**0.460**	0.194	**0.697**	**0.693**
	FA	0.081	0.356	0.206	0.668	0.676
	FSA	0.087	0.270	0.092	0.645	0.638
		MSE	**COR_{pf}**	**DMSE**	**AGR_{pf}**	**AGR**
SVR-RBF	**F**	0.090	0.091	0.135	0.594	0.618
	S	0.089	0.019	**0.002**	**0.648**	**0.682**
	A	**0.084**	**0.376**	0.730	**0.696**	**0.655**
	FS	0.089	0.014	**0.008**	0.619	0.652
	SA	0.092	**0.298**	1.563	0.643	0.620
	FA	**0.084**	0.211	0.363	0.636	0.642
	FSA	0.087	0.172	0.233	0.617	0.627
		MSE	**COR_{pf}**	**DMSE**	**AGR_{pf}**	**AGR**
SVR-P	**F**	0.088	-0.002	**0.004**	0.643	0.678
	S	**0.087**	**0.012**	**0.004**	**0.649**	**0.683**
	A	**0.087**	0.002	**0.004**	**0.650**	**0.684**
	FS	0.088	**0.009**	0.009	0.638	0.674
	SA	**0.087**	0.001	0.007	**0.650**	**0.683**
	FA	0.088	-0.010	**0.006**	0.645	0.680
	FSA	0.088	**0.009**	0.008	0.640	0.675

the LSTM networks. For the RBFs, the results per fold again denote the audio cues as the best for agreement per fold. It should be noted that despite of the shoulder cues having a very low correlation, they achieve good agreement accuracy. For the agreement per fold, the shoulder accuracy is even better than the audio cues. The results for RBF-P are completely analogous to the valence results with respect to the correlation, which now even attains negative values, something which is a very bad indication. Intuitively, it means that not only the distributions do not change together, but sometimes when the one increases the other decreases. We should note that the agreement values, despite the negative correlation appear to be equally high as the other two methods. This is due to the fact that the correlation is a much more strict evaluation metric than the agreement. In Table 5.16, we present the per fold correlation for two LSTM networks, trained with the fused audio and shoulder cues (AS) and audio cues (A) alone respectively. Finally, in order to provide a complete comparison, in Table 5.15 we present the average correlation for valence and arousal for LSTMs and SVR-RBF, where the clear dominance of LSTMs is obvious.

Table 5.15: Comparing the correlation of the averaged results for all cue/modality combinations with feature level fusion for LSTMs and SVR-RBF.

	V	**A**	**AVG**
LSTM	**0.1513**	**0.2927**	**0.2220**
SVR-RBF	0.1278	0.1686	0.1482

5.2.5 Performance of Audio & Shoulder Cues: Theoretical Expectations & Experimental Results

This section will firstly explore the reasons behind the high performance of the audio cues with respect to the valence estimation. Previously, we commented on the good performance of the audio cues in terms of valence, which is contradictory to the theoretical expectations which consider the facial expressions to be better performing for valence. We hypothesised (Section 5.2.2) that this was due to the audio sequences being double in size than the video sequences, since features were extracted for the audio frame rate, which was 0.02 seconds. To provide evidence for this hypothesis, we extracted the audio features from the audiovisual data, adjusting the parameters so as to produce one set of features for each 0.04 seconds, equivalent to one video frame. In Table 5.17, we present the results which we compare with the performance of other single cues. It is observed that now, the correlation of the audio cues drops from 0.397 to 0.076. The facial expression cues are now dominant in the valence recognition, providing the best correlation with the ground truth. Furthermore, it is highly important to observe that both the correlation and agreement values drop for the arousal estimation but only slightly. With respect to the valence

Table 5.16: The fused audio/shoulder cues (AS) and the audio cues (A) correlation accuracy for each fold over 10-fold cross validation

Fold	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
AS*cor*	0.6	0.6	0.5	0.7	0.3	0.4	0.3	0.03	0.4	0.7
A*cor*	0.8	0.7	0.5	0.6	0.4	0.1	0.7	0.3	0.6	0.6

decrease, the arousal decrease is barely noticeable. This is another indication that audio cues are better for arousal than valence estimation.

Finally, it should be noted that the correlation and agreement values for the audio is still better than the correlation and agreement for the shoulders. This brings us to the second issue discussed in this section. The information regarding affect carried by the shoulder cues has not been obvious. Since there has not been any other work on continuous recognition with shoulder cues, we can hypothesise on the reasons behind this phenomenon from our results alone. Firstly, we assume that the variations and motion performed with the shoulders might be very limited and thus not descriptive enough for mapping into the entire valence/arousal space with continuously varying values. Secondly, the scenario under which the audiovisual material is taken could have a crucial role in this: The individuals were talking and using facial expressions throughout the video, while their shoulder motion was not continuous. If we were to think of the corresponding scenario in terms of facial expressions, it would be as if the facial features of the individual would not significantly change positions during the video, or if the person did not speak for most of the video. Under those scenarios the results would have been quite different.

Table 5.17: Statistics for the performance of all single cues with LSTMs, when the audio features have been extracted with a frame rate equivalent to the video frame rate (row $\mathbf{A}^{vf}$). The audio results using the audio frame rate are presented in the last row for comparison

	Valence				**Arousal**			
LSTM	**MSE**	**COR**$_{pf}$	**AGR**$_{pf}$	**AGR**	**MSE**	**COR**$_{pf}$	**AGR**$_{pf}$	**AGR**
F	0.067	**0.205**	**0.551**	**0.552**	0.080	0.237	0.681	0.693
S	**0.056**	0.026	0.438	0.467	0.098	0.041	0.524	0.522
$\mathbf{A}^{vf}$	0.057	0.076	0.525	0.526	**0.079**	**0.468**	**0.742**	**0.718**
A	*0.052*	*0.397*	*0.672*	*0.674*	*0.072*	*0.511*	*0.747*	*0.720*

5.2.6 Dimensionality Reduction

From observing the results discussed in the previous section (especially for valence and LSTMs), we hypothesised that one possible negative factor could be the large dimensionality of the feature vectors. This negative effect can manifest as the generation of more complicated (and thus over-fitted) models, while also due to the dimensionality, the algorithm could focus on features which are irrelevant to the given problem (the *curse of dimensionality*). The dimensionality problem is common in automatic emotion recognition [84], since in many cases many features are extracted from multiple modalities.

Thus, in this section we will present some experimentation with dimensionality reduction with LSTMs. Principal component analysis (PCA) is an orthogonal linear transformation, which maps the input data into a new coordinate system. The characteristics of the new system are such that the greatest variance of any projection of the data lie on the first coordinate, the second greatest variance on the second component and so on. The i-th component is called the i-th principal component [98]. It is also noted that each dimension is linearly independent. For more details on PCA the reader is referred to [98].

Our experiments included reducing the facial expression features from 40 to the range $[10, 20]$, the

shoulder features from 10 to [2,6] and the audio features from 15 to [5,10]. For the experiments we will present, we chose the facial expression features to have 14 dimensions , while the audio and shoulder cues were reduced to 4 dimensions each. Our experimentation was targeted at the fusion of the cues. Nevertheless, we provide the results for single cues in Table 5.18. Essentially, the performance is worse, and the only observation worth mentioning is the improvement of the metrics regarding the shoulder cues besides the MSE, indicating that not all original feature dimensions offer useful information for the regression.

Table 5.18: PCA vs. no dimensionality reduction for single cues. A cell for the PCA results is bold if it improves the performance achieved with no dimensionality reduction. Similarly for the cells with results attained with no dimensionality reduction.

		PCA				**No Dim. Reduction**			
		MSE	**COR**$_{pf}$	**AGR**$_{pf}$	**AGR**	**MSE**	**COR**$_{pf}$	**AGR**$_{pf}$	**AGR**
V	**F**	0.069	0.145	0.485	0.516	**0.067**	**0.205**	**0.551**	**0.552**
	S	0.086	**0.030**	**0.553**	**0.526**	**0.056**	0.026	0.438	0.467
	A	0.079	0.093	0.565	0.533	**0.052**	**0.397**	**0.672**	**0.674**
A	**F**	0.104	0.212	0.593	0.609	**0.080**	**0.237**	**0.681**	**0.693**
	S	0.117	**0.072**	**0.556**	**0.587**	**0.098**	0.041	0.524	0.522
	A	0.082	0.477	0.704	0.690	**0.072**	**0.511**	**0.747**	**0.720**

The results for the PCA experiments with feature-level fusion are presented in Table 5.19. We will analyse each fusion setting and estimated value separately. For the valence prediction and the fusion of the facial expression and audio cues the dimensionality reduction does improve all the metrics in contrast to the full dimensionality. The same applies for the fusion of the facial expression cues with the shoulder cues, where we observe a significant rise in the correlation (from 0.05 to 0.233) and a small increase in the agreement although in this case we observe an increase in the MSE. The shoulder/audio fusion is a case where the dimensionality reduction does not offer any improvement except a small decrease in the MSE. The fusion of all cues though, provides us with an improvement for both agreement and correlation metrics, while slightly worsening the MSE. In general, we can say that where facial expression cues where used (which have a large dimensionality of 40), the reduction the correlation and agreement parameters. In the case where the dimensionality was already rather small (e.g. the shoulder and audio cues fusion), no significant improvement was noticed.

As far as the arousal results are concerned, the fusion of facial expression and shoulder cues provides an improvement in all parameters. For the rest of the fusion settings for arousal, there is a worsening of the correlation observed, although for the fusion of all cues the agreement increases. There are essentially no solid conclusions we can draw for the arousal values. However, we can claim that the very good performance of the audio cues when no dimensionality reduction is imposed are indications that the entire set of features is required for achieving this and that not all shoulder feature dimensions contain important information for continuous emotion recognition.

5.2.7 Capturing Correlations and Temporal Patterns in Valence and Arousal

In this section, we will refer to the issue of capturing existing correlations between valence and arousal values, while also learning manifesting temporal patterns. Current state-of-the-art in

Table 5.19: PCA vs full dimensionality. The subscript for the fused cues demonstrates whether valence or arousal is being estimated (v or a)

	PCA				**No Dim. Reduction**			
	MSE	**COR**$_{pf}$	**AGR**$_{pf}$	**AGR**	**MSE**	**COR**$_{pf}$	**AGR**$_{pf}$	**AGR**
FA$_v$	**0.063**	**0.181**	**0.559**	**0.569**	0.067	0.150	0.520	0.537
FS$_v$	0.066	**0.233**	**0.589**	**0.595**	**0.053**	0.050	0.485	0.512
SA$_v$	**0.067**	0.145	0.513	0.516	0.071	**0.154**	**0.516**	**0.526**
FSA$_v$	0.064	**0.188**	**0.590**	**0.579**	**0.059**	0.076	0.441	0.475
FA$_a$	0.096	0.186	0.620	0.623	**0.081**	**0.356**	**0.668**	**0.676**
FS$_a$	**0.095**	**0.216**	**0.612**	**0.646**	0.098	0.174	0.592	0.617
SA$_a$	**0.077**	0.395	0.692	0.682	0.084	**0.460**	**0.697**	**0.693**
FSA$_a$	**0.087**	0.173	0.629	**0.643**	**0.087**	**0.270**	**0.645**	0.638

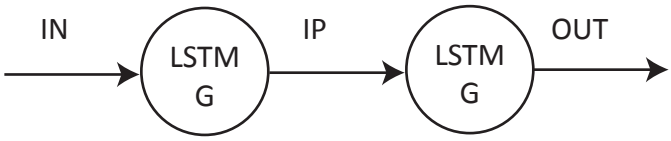


Figure 5.4: Scenario I: A network trained on a ground truth G provides the intermediate estimation (IE) which is used as the input for a second network

dimensional and continuous emotion recognition does not present any work which attempts to capture any such dependencies. We performed a set of experiments with two scenarios, one for each of the following cases: The first scenario is presented in Fig. 5.4. The setup includes two networks, each one trained with a target ground truth G which can be valence or arousal (the same for both networks). The training proceeds typically, with input consisting of features from a combination of cues. In this scenario, we assume that a set of patterns which manifest in the valence or arousal annotations are also exhibited in the intermediate prediction (IP) of the first network (LSTM G1), and that these patterns can be captured by the network which is trained with these predictions (LSTM G2), while furthermore the network will be able to learn which of these dependencies correspond to a set of dependencies in the ground truth. If successful, the network can learn the true dependencies that correspond to actual temporal patterns in the ground truth and ignore the false indications. The second scenario depicted in Fig. 5.5, contains 3 networks. The first two networks, labelled 1 and 2, are trained on predicting valence and arousal respectively. The (intermediate) outputs of both these networks are then fed into a third network (labelled 3), which produces the final output which can be a prediction on either valence or arousal. Again, we assume that characteristics of the valence and arousal ground truth manifest in the intermediate outputs, and thus correlations between the valence/arousal values could be captured by the third network.

In Table 5.20, we present our experimental results using both scenarios I & II, which we compare with the previous estimation. We will begin our analysis with the results from the audio cues, since the original networks used expose the greatest correlation with the ground truth. This is an indication that patterns and correlations emerging in the corresponding arousal/valence ground truth do appear in the estimated values. For the audio cues we can observe the following:

(1) Predicting valence with a network trained on valence (Scenario I): All result metrics are

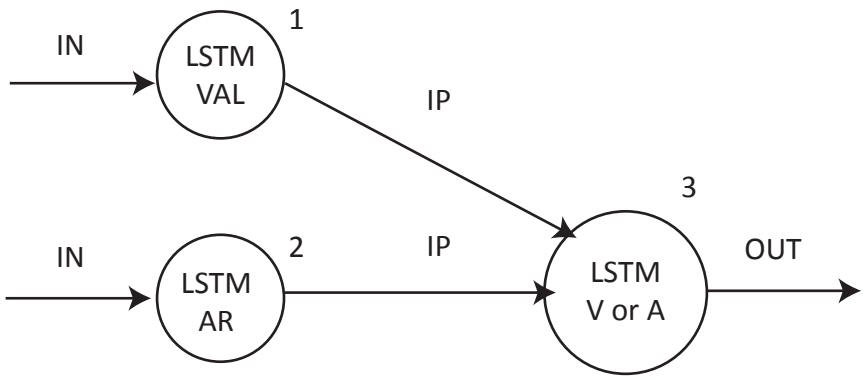


Figure 5.5: Scenario II: Networks (1) and (2) are trained for predicting valence and arousal respectively. The intermediate output (IP) from these networks is fed into network (3).

improved (or not decreased for the agreement per sequence).

(2) Predicting valence with Scenario II (inputs are estimations of networks trained on valence and arousal respectively): All result metrics improve even more than (1).

(3) Predicting arousal with a network trained on arousal (Scenario I): The correlation increases noticeably as well as the agreement per sequence. There is a drop in the agreement per fold and an increase for the MSE.

(4) Predicting arousal values following Scenario II: All result metrics improved.

The previous observations provide us with strong indications that such training can discover and learn temporal patterns in the valence and arousal values. Especially for the case of training with both arousal and valence as inputs, the results demonstrate an improvement of *all* the result metrics, irrelevantly of which value we are trying to predict.

Prediction with the facial expression cues confirms the first observation from the audio cues: All result metrics of valence estimation improve, both with the first and second scenarios, only in this case there is no clear dominance of any of the two scenarios. For the arousal prediction, training with another network predicting arousal (Scenario I) improves the correlation but decreases the agreement, while training with scenario II improves all result metrics. It is important to note that the increase is not as great as in the audio case. We believe that this is in direct relation with the fact that the correlation metric of the original network for the facial expression cues is not as strong as in the audio cues case. Thus, many false dependencies/correlations emerging from the prediction could negatively affect the performance of the second network which operates with the prediction as input. Nevertheless, there is still an improvement.

The third scenario relates to the shoulder cues. Observe that the shoulder cues have the least correlation from all the cues. Based on the previous experiments, we can deduce that the improvement will not be great. Indeed, by observing the results, we can see that the only improvement that can be found relates to the agreement values. In fact, there is a great drop in the correlation for the valence, and a small drop for arousal.

In conclusion, what we can say about using networks to capture the correlations and patterns in valence and arousal is that we can expect great improvement only if these patterns manifest in a good degree in the intermediate prediction. For example, the audio cues have shown an

Table 5.20: Comparing scenarios I & II with the original predictions, for capturing correlations and patterns in the valence/arousal values. Values are in bold when they provide some improvement (or no worsening) compared to the metrics in the original network

		Valence				Arousal			
		MSE	**COR_pf**	**AGR_pf**	**AGR**	**MSE**	**COR_pf**	**AGR_pf**	**AGR**
A	**Original**	0.052	0.397	0.672	0.674	0.072	0.511	0.747	0.720
	Scen. I	**0.048**	**0.403**	**0.685**	**0.674**	0.087	**0.529**	0.706	0.678
	Scen. II	**0.046**	**0.425**	**0.696**	**0.680**	**0.069**	**0.577**	**0.785**	**0.755**
		MSE	**COR_pf**	**AGR_pf**	**AGR**	**MSE**	**COR_pf**	**AGR_pf**	**AGR**
F	**Original**	0.067	0.205	0.551	0.552	0.080	0.237	0.681	0.693
	Scen. I	**0.054**	**0.209**	**0.595**	**0.566**	**0.078**	**0.247**	0.672	0.680
	Scen. II	**0.054**	**0.213**	**0.551**	**0.564**	**0.078**	**0.257**	**0.688**	**0.696**
		MSE	**COR_pf**	**AGR_pf**	**AGR**	**MSE**	**COR_pf**	**AGR_pf**	**AGR**
S	**Original**	0.056	0.026	0.438	0.467	0.098	0.041	0.524	0.522
	Scen. I	0.061	0.001	**0.534**	**0.527**	0.099	0.031	**0.568**	**0.601**
	Scen. II	0.066	0.004	**0.438**	**0.474**	0.099	0.035	**0.563**	**0.595**

improvement of 2.8% and 6.6% for the correlation in valence and arousal prediction respectively. We can not expect significant results though if the intermediate estimation is not good enough. We have though experimented with incorporating both valence and arousal in decision level fusion, following a setting similar to Scenario II, only for more than one set of cues. We describe this approach in the next section.

5.2.8 Decision-Level fusion & Valence Arousal Correlations

We will now describe some experiments with decision-level fusion using LSTM networks. Our setting, includes the three best networks initially obtained (Section 5.2.4), i.e. with no PCA or other attempt to improve the performance. We firstly perform decision level fusion on each of the arousal/valence values, by using networks which predict the corresponding value as presented in Fig. 5.6. Furthermore, in other experiments we attempt to fuse not only the networks which predict the required value but both valence and arousal values from all the networks, as presented in Fig. 5.7. This follows our experiments in learning correlations between the valence and arousal values in Section 5.2.7. We will call this decision level fusion as the second type of decision-level fusion (type II or 2) in order to discriminate it from the first one (type I or 1).

Our results are presented in Table 5.21 along with a comparison to our earlier feature-fusion attempts, including PCA and SVR-RBF. Our first conclusion is that for the fusion of any combination of cues except the face and shoulder cues there is a clear dominance of decision level fusion in both settings, improving all the metrics. Furthermore, in this selection of cues and specifically for arousal a further improvement is observed in *all* metrics by using the second type of decision level. For valence, there is always an increase in the correlation while a small decrease is observed comparing to the first type of decision level fusion is observed only for the agreement in the face/shoulder fusion. It is very important to notice the amount that the correlation increases to comparing to feature level fusion: For example, in the fusion of all cues/modalities the increase from feature level fusion is from 0.076 to 0.422. For the decision level fusion of the face and shoulder the behaviour is more inconsistent. We believe that this is due to the very low

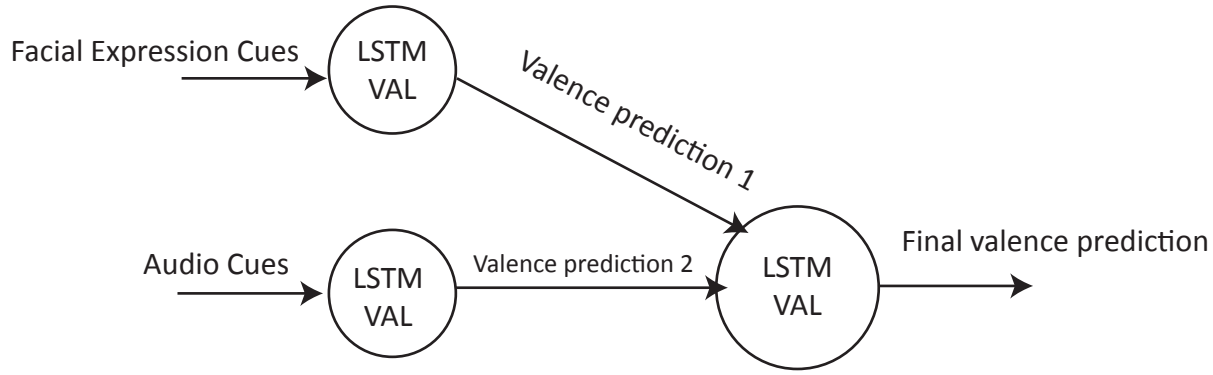


Figure 5.6: Decision level fusion (type 1): In the example, the predicted valence from facial expression cues and audio cues is fused by using a third network which outputs the final prediction. This example generalises to all combinations of cues and both valence/arousal.

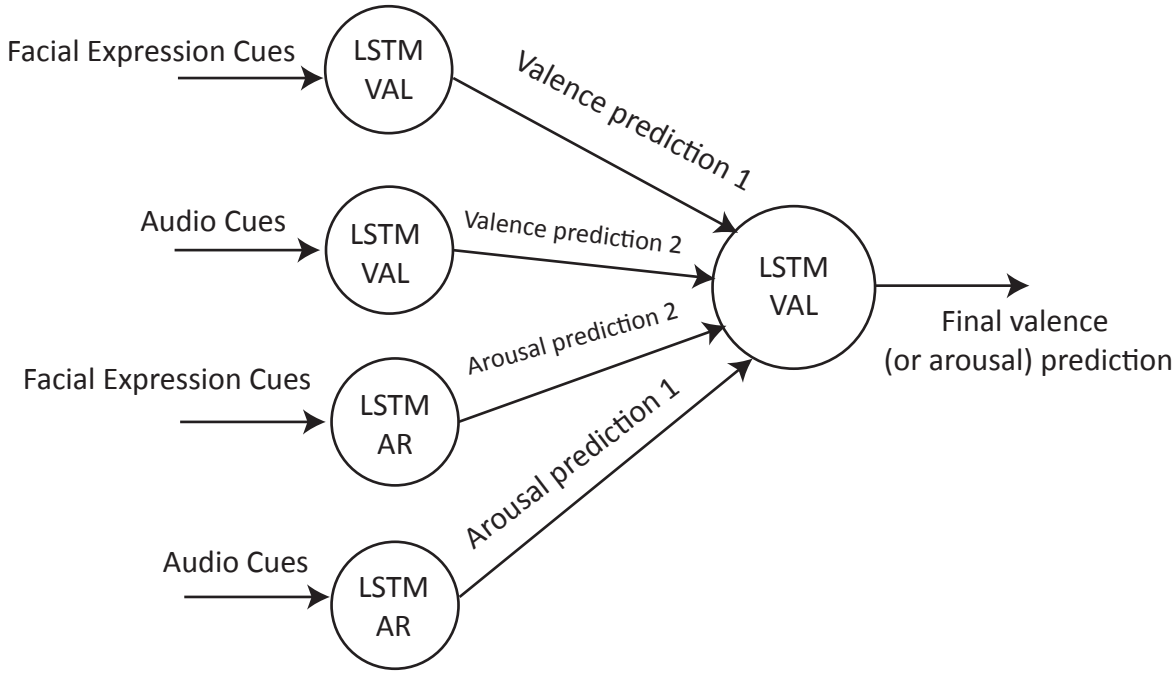


Figure 5.7: Decision level fusion (type 2) by using both arousal and valence from every set of cues: In this example, the predicted valence and arousal values from both facial expression and audio cues are fused, again by using a third network which outputs the final prediction. This example generalises to all combinations of cues and both valence/arousal as final predictions.

correlation that the shoulder cues expose, and the absence of the very good correlation that the audio cues offer to balance the problem. The correlation still increases for training with both valence and arousal (second decision-level fusion type) in both valence and arousal prediction, while training with the first decision-level fusion type increases the correlation only in the valence prediction. Agreement increases in both decision-level fusion types for arousal prediction, while for valence prediction it increases only with the first type of decision-level fusion.

What is very important to observe, is that the best correlation along with the best agreement is achieved with the fusion of the face and audio cues and the fusion of all modalities/cues. There is a very slight advantage to the fusion of the face and audio cues, but typically the difference between the two is very small. This confirms our previous hypothesis (Section 5.2.7) that the incorporation of the shoulder cues with the low correlation they provide slightly decreases the overall performance. Finally, in Table 5.22 we present the averaged improvement[7] for each of the techniques compared in Table 5.21. The dominance of decision-level fusion for LSTMs is obvious against the compared techniques, while decision-level fusion (type 2, with both valence and arousal as inputs) presents us with the best recognition accuracy.

5.2.9 Continuous to Discrete Emotion Recognition

The final experiment that we will refer to is a different approach for discrete emotion recognition. We essentially performed an experiment in determining how accurate the valence estimations would be in determining the positive or negative classification of a sequence. The procedure we followed referred to slicing the continuous valence estimation into a fixed number of windows n. Then, we calculated the mean μ and standard deviation σ of the valence predictions that within each temporal window. The pairs of these values for the n time frames composed an input example to an SVM classifier as follows:

$$\langle \sigma, \mu \rangle_1, \ldots, \langle \sigma, \mu \rangle_n$$

where the vector was included by the corresponding label for positive or negative classification. We performed this type of experiment only with the valence estimated by the audio cues. After a grid search for the parameters, the best performance achieved in classifying a sequence as a positive or negative emotional state was 61.4% (Parameters: $c = 1.7, g = 0.038, E = 0.8$), slightly surpassing the 61.2% recognition rate of the HMM model trained in Section 5.1. This is highly significant, as it can be observed that at least for this case this method (slightly) overperforms typical HMM models. It is also noted that models such as SVMs can output probability estimates, and by feeding these estimates into another classifier, a similar procedure to the classification in the likelihood space (Section 5.1.2) could be performed - in this case by training one SVM for each class. In conclusion, such experimentation would be interesting, and would also provide another type of evaluation of the prediction of a continuous classifier. It is noted though, that this is heavily dependent on the specific problem: In our case, we used positive/negative emotional states and that suited the transition well. In other cases, e.g. where continuous emotion recognition is performed on continuous streams of data, such a mapping is more difficult. As a final comment, we should note that we also performed experiments in sequence classification with LSTMs, where

[7] We denote that when we refer to percentages with respect to the correlation, since the (non-negative) correlations we are interested in are scaled from 0 to 1, we multiply with 100 in order to attain the percentage value. Then, the percentage increase is the subtraction of the highest from the lowest value.

Table 5.21: Decision-Level (DLev, type I) fusion compared to feature-level fusion (FeatLev), feature-level fusion with PCA (featPCA), feature-level fusion with SVRs (fLSVR) and Decision-Level fusion using both arousal and valence for each fused cue (VADLev, type II). The two best results for each case are in bold.

		Valence				Arousal			
		MSE	**COR$_{pf}$**	**AGR$_{pf}$**	**AGR**	**MSE**	**COR$_{pf}$**	**AGR$_{pf}$**	**AGR**
FSA	featPCA	0.064	0.188	0.590	0.579	0.087	0.173	0.629	0.643
	FeatLEV	0.059	0.076	0.441	0.475	0.087	0.270	0.645	0.638
	fLSVR	0.058	0.161	0.526	0.547	0.087	0.172	0.617	0.627
	Dlev	**0.051**	**0.420**	**0.698**	**0.678**	**0.083**	**0.528**	**0.752**	**0.730**
	VADLev	**0.050**	**0.422**	**0.708**	**0.690**	**0.077**	**0.570**	**0.781**	**0.745**
		MSE	**COR$_{pf}$**	**AGR$_{pf}$**	**AGR**	**MSE**	**COR$_{pf}$**	**AGR$_{pf}$**	**AGR**
FA	featPCA	0.063	0.181	0.559	0.569	0.096	0.186	0.620	0.623
	FeatLEV	0.067	0.150	0.520	0.537	0.081	0.356	0.668	0.676
	fLSVR	**0.053**	0.186	0.510	0.542	0.084	0.211	0.636	0.642
	Dlev	**0.052**	**0.424**	**0.706**	**0.681**	**0.073**	**0.538**	**0.758**	**0.730**
	VADLev	**0.053**	**0.426**	**0.679**	**0.659**	**0.066**	**0.586**	**0.786**	**0.751**
		MSE	**COR$_{pf}$**	**AGR$_{pf}$**	**AGR**	**MSE**	**COR$_{pf}$**	**AGR$_{pf}$**	**AGR**
SA	featPCA	0.067	0.145	0.513	0.516	0.077	0.395	0.692	0.682
	FeatLEV	0.071	0.154	0.516	0.526	0.084	0.460	0.697	0.693
	fLSVR	0.062	0.091	0.509	0.520	0.092	0.298	0.643	0.620
	Dlev	**0.055**	**0.399**	**0.694**	**0.676**	**0.086**	**0.521**	**0.714**	**0.698**
	VADLev	**0.052**	**0.418**	**0.703**	**0.689**	**0.078**	**0.539**	**0.750**	**0.723**
		MSE	**COR$_{pf}$**	**AGR$_{pf}$**	**AGR**	**MSE**	**COR$_{pf}$**	**AGR$_{pf}$**	**AGR**
FS	featPCA	**0.053**	0.050	0.485	0.512	0.098	0.174	0.592	0.617
	FeatLEV	0.066	0.233	**0.589**	**0.595**	0.095	**0.216**	0.612	0.646
	fLSVR	**0.056**	0.115	0.494	0.518	**0.089**	0.014	0.619	0.652
	Dlev	0.065	**0.234**	**0.551**	**0.549**	**0.089**	0.204	**0.663**	**0.675**
	VADLev	0.064	**0.242**	0.420	0.465	**0.086**	**0.259**	**0.679**	**0.683**

Table 5.22: Improvement of averaged correlation over all fused cues for PCA-LSTM (PCA), feature-level SVR (fLSVR), feature-level LSTM (FeatLEV) against decision level fusion type I (DLev) and decision level fusion type II (VADLev - using both valence and arousal as inputs). Notice that type II fusion is better than type I on every single comparison.

	Dlev			VADLev		
	Valence	**Arousal**	**AVG**	**Valence**	**Arousal**	**AVG**
PCA	22.83%	21.55%	22.19%	22.91%	25.65%	24.28%
fLSVR	23.11%	27.40%	25.26%	23.19%	31.50%	27.35%
FeatLEV	21.59%	12.20%	16.90%	21.68%	16.30%	18.99%

the networks output a class label for each sequence instead of the continuous estimation. Results did not show any improvement over the standard HMMs for the face and shoulder cues for which we experimented with.

5.3 Discussion

In this section we will attempt to summarise some of the conclusions we arrived at during the previous discussions and experiments described. There has been a variety of experiments performed, thus this discussion will focus on the most important conclusions.

Firstly, with experiments in discrete emotion recognition:

- Subject-independent recognition is very difficult with four subjects. We confirm previous work on the SAL database that only provides subject-dependent results (e.g. [199]). This applies to continuous emotion recognition as well.
- The removal of background speech improves the performance of the system by nearly 6%, while the noise reduction increases it by another 3.73%. In total, the improvement was almost 10%.
- In subject dependent emotion recognition from CHMMs, the fusion of all cues provides the best results, while the facial expression and shoulder cues (visual modality) are better performing than the audio for positive/negative emotional state classification. Again, this confirms the theoretical expectations [84].
- For classification in the likelihood space, a linear kernel by using SVM can improve the performance of a separating line found with gradient descent. Furthermore, RBF kernels specific for each combination of cues have been shown to increase the classification rate of CHMMs by 11.9%, and the performance of a gradient descent found separating line by an additional 5.5%. Most importantly, the results are the theoretically expected ones, with the fusion of the all cues providing the best classification and the single cues which related to the visual modality performing better than the audio cues. We denote that classification rates of 88.2%, 86.7% and 85.1% were attained, while with maximum likelihood classification the maximum accuracy was 79.1% and with a separating line determined by gradient decent 85.1%. Finally, we observe that 6 of the 8 combinations of cues/modalities produce an accuracy of over 79-80% (Table 5.8). We believe the high recognition rates demonstrate that our segmentation process has successfully discriminated between negative/positive valence. This is an indication towards the reliability of the ground truth.

and for continuous emotion recognition:

- The length of each temporal sequence which is destined for training sequence learning techniques and especially for LSTMs which have the ability to learn long range patterns in the input data is crucial. Although theoretically the facial expressions are better indications for the valence dimension [84], the double length sequences of the audio cues provide better results. When the features from the audio signal are extracted at a frame rate equivalent to that of the video, the facial expression cues perform much better than the audio cues (again for valence).
- Following the latter comparison, of the audio cues reduced to a video frame rate, we can still notice that they perform better than the shoulder cues, which do not provide good results in general, either in mono-cue or multi-cue recognition. We assume this is due to limited variances in the shoulder motion (See discussion in Section 5.2.5).

- The averaged MSE alone is not a proper indicator of the accuracy of the estimated values for valence and arousal, as we have shown in Section 5.2.3. It should be used primarily with the correlation and agreement metrics in order to provide a more objective evaluation, which considers not only the numerical error but also the patterns of the distributions, as well as the agreement level in terms of estimating a positive (negative) emotional state as a positive (negative) for valence, similarly with active/passive for arousal.

- By training a SVM with the estimated "windowed" valence prediction of an LSTM network for applying discrete emotion recognition, we achieve a performance of 61.4% slightly surpassing the 61.2% results reported by the HMMs. This is a very important observation and despite the fact that more experiments with fusion and other cues are required to provide more solid conclusions, it is crucial that the results compete with models specifically modelled for discrete emotion recognition, such as the HMMs.

- Using a polynomial kernel in SVR does not provide naturalistic results for continuous emotional recognition. The RBF kernel provides much better results, considering the correlation and agreement. It is noted that the polynomial kernel in some cases presents us with a negative correlation with the ground truth.

- For single cues, the LSTMs perform better than the SVR-RBF (by 8.3% on average), except for the shoulder cues for valence estimation, which as we have mentioned appear to be problematic for continuous recognition. For feature-level fusion, the LSTMs provide substantially better results for arousal, while for valence the SVR-RBF performs better - this could be an indication of certain feature dimensionality size issues with LSTMs. The correlation of the audio cues with respect to the arousal ground truth for LSTMs is 0.511, even improving the 0.505 average correlation that the human coders have for estimating arousal. The respective valence correlation for audio is 0.397, slightly lower than the 0.45 correlation that human coders have with the ground truth.

- For single cues dimensionality reduction does not provide any significant improvement. The only slight improvement is with the shoulder cues. Dimensionality reduction with the fusion of all cues does provide some improvement when the shoulder cues are involved, except when they are fused with the audio cues alone. We believe that this is due to the deterioration of the performance of the audio cues due to the dimensionality reduction.

- Capturing correlations between the valence/arousal values and detecting patterns in the distributions can improve the performance, considering that the initial estimation is moderately accurate. In our experiments, the only estimations that did not provide significant improvement had a correlation of 0.035 or below with the ground truth. The maximum improvement for the correlation of mono-cue recognition was 6.6%.

- Decision-level fusion with LSTMs, generally outperforms feature-level fusion with either LSTMs or SVRs and feature-level fusion with PCA. Moreover, by using both arousal and valence values for determining the output (which in an instance of networks can be either arousal or valence) again provides very significant improvement. The largest increase can be observed for the fusion of all cues: Feature-level fusion with LSTMs provides with a correlation of 0.076, which is boosted to 0.42 with decision level fusion. The accuracy of

a human coder with respect to the correlation with the ground truth is 0.505 for arousal. This is again outperformed with decision level fusion by the fusion of all cues (0.57), the face/audio cues (0.586) and the fusion of the shoulder/audio cues (0.539). The maximum coder correlation with the ground truth for arousal is 0.62, closely approximated by our 0.59 results. The agreement of the coder with the ground truth regarding arousal is 0.82, a value which is very closely approximated by the fusion of all cues (0.78) the fusion of facial expression and shoulder cues (0.79) and the fusion of shoulder/audio cues (0.75). Also, the correlation of the human coders with the valence ground truth is 0.45, approximated by values such as 0.426 achieved by the facial expression/audio cues fusion. In general, with decision-level fusion we improve feature-level fusion by 22% and 12% for valence and arousal respectively, while with decision-level fusion by using both estimated arousal and valence as features during training, the improvement is 22% and 16% respectively. When compared to feature-level SVR fusion, the improvement reaches a 32% average increase for arousal and 23.2% for valence.

We will conclude this chapter with two final discussions: Firstly, it is significant to remind the reader that the optimal fusion method for emotion recognition is still an open research issue (Chapter 2). It is claimed that feature-level fusion can allow the algorithm to learn inter-cue correlations, while with decision-level fusion this is not possible. However, this generally applies to discrete emotion recognition, where the output of a classifier is typically a small integer indicating the class assigned. If we consider continuous emotion recognition though, decision-level fusion can be seen from a much more different perspective. Firstly, the output of the algorithm is a real number (limited to a finite set only by the machine accuracy). We believe that although the correlations amongst the cues can not be explicitly learnt in decision-level fusion, these characteristics of the cues can be implicitly encoded in the estimated values. Thus, by providing the decision-level algorithm with a set of these patterns can imply the significant improvement observed. Furthermore, decision-level fusion with valence (or arousal) estimations can allow the learning technique to identify *common* and *correct* patterns across the predictions, and can *disregard* patterns which appear not to offer any improvement to the estimated value during training. Furthermore, the decision-level fusion of *both* arousal and valence estimations from the set of cues being fused provides the learning technique with the opportunity to detect not only correlations between valence (or arousal) estimations from different cues, but also correlations and patterns between the valence and arousal estimations, thus providing a robust, effective and more accurate performance improvement.

Secondly, we should comment on the fact that the correlation of our prediction with the ground truth with respect to the arousal, is higher (0.586) than the average correlation (0.505) of the coders with respect to the ground truth. We should state that we do not claim that the performance is in general better. The coder annotations provide a minimum MSE of 0.02 with respect to the ground truth, while for the corresponding experiments our MSE was more than 0.07. Secondly, the average agreement of the coder with the ground truth is 0.825, while the maximum agreement we achieve is 0.786. In other words, we can not say that we generally perform better than the average coder, but just refer to the correlation. We also should mention the fact that the coder with the maximum correlation with respect to the ground truth has a correlation of 0.616, higher than our predicted 0.586.

Our assumption is that the distribution of the arousal ground truth can be easier predicted than the distribution of the valence ground truth. We denote that this is confirmed by other work which presents results for correlation in continuous emotion recognition: Specifically in [117], the predicted correlation is twice and almost three times higher for arousal than the predicted correlation for valence, while in [101], again the arousal correlation is larger. We should also denote that this phenomenon could relate to the issues manifesting in human coder annotations (Section 4.4). Finally, the procedure of normalising and segmenting the annotations was based mainly on synchronising the valence annotations and secondly on imposing thresholds on arousal, since the main focus was on extracting positive and negative segments. This could have introduced a bias, negatively affecting the coder arousal correlation to the ground truth.

Chapter 6

Conclusions & Future Work

In this work, we have provided a detailed description of background research (Chapter 2), covering state-of-the-art emotion recognition systems, providing a discussion on modern research questions while also describing the background in emotion theory. We described a set of state-of-the-art learning techniques suitable for continuous emotion recognition in Chapter 3, from which Support Vector Machines and Long short-term memory neural networks have been used in our experiments. In Chapter 4, we presented a novel segmentation procedure for extracting audiovisual segments from databases which have continuous and dimensional emotion annotations, while we presented our pre-processing experimentation. Finally, in Chapter 5, we present the entire set of our experiments, ranging from discrete to continuous emotion recognition. We described our most significant conclusions for these experiments in Section 5.3.

We believe that one of the most important conclusions one can have from this work, is the crucial and substantial differences that constitute discrete and continuous emotion recognition two vastly different problems. This is not only demonstrated by the different techniques and algorithms required to approach the two problems, but also in the experimental results provided. For example, consider how the double frame rate of the audio feature extraction shifted the balance in favour of the audio cues in continuous emotion recognition, while the same characteristic did not provide any significant variations in the ranking of the performance of each cue in the discrete recognition. Consider how the shoulder movement cues, very descriptive for discrete emotion recognition proved to be weak for predicting continuous emotions.

Continuous emotion recognition is still at its infancy, with the literature in the field providing just a few examples of previous relevant research work. We hope that with this work, we have provided a step towards the generation of fully automatic continuous emotion recognition systems.

6.1 Future Work

In this section we will provide some topics that can be undertaken as future work. We should note that the research issues relating with emotion recognition and continuous emotion recognition are many, and so are the possibilities of research building on this work. We just denote some of the most significant points:

- We have briefly mentioned the topic of moving from continuous to discrete emotion recognition. This is very important for two reasons: Firstly, the accuracy could provide a quite interesting evaluation method of the performance of the continuous estimation. Secondly, as we have seen from just a single experiment conducted with SVMs for the audio cues valence estimation, the performance we achieved for classifying a sequence as positive/negative was slightly better compared to the classification rate of HMMs (Section 5.2.9).

- Training continuous emotion recognition systems to learn and be able to model an individual coder. For example, in the SAL case we have four coders, each providing annotations. A separate classifier can be trained for each of the coders, each one predicting the annotations of the specific coder given as input the same set of features (from the cues used). Then, the estimation that these models of the coders would provide would be fused providing the final estimation. We believe that this technique would be able to better capture the actual behaviour of the coder while annotating and thus would be able to better predict the estimated values.

- We have already demonstrated how the frame rate at which features are extracted is of high importance, along with the actual length of the sequences. It is thus crucial to firstly obtain audiovisual material with a higher video frame rate. It is noted that the issue of how humans perceive visual images is not fully explored in the relevant sciences such as neurology, i.e. there is no "frames-per-second" estimate for the human eye and translation. Thus, the higher the frame rate of the videos we process, the more features we extract per sequence, enforcing crucially the dynamic sequence learning techniques for continuous emotion recognition. In this case, in the future we can work with databases which provide more frames per second, while we can also experiment with extracting longer sessions.

- We have already referred to the baseline problem (Section 2.5). It would be interesting to examine how regression on sequence learning performs without the explicit presence of such a baseline. Although it is typically considered that a baseline is required, it is interesting to observe how the performance degrades in the contrary case, and if the dynamic classification methods would be able to catch up over time and adjust to the changing features in the sequences.

- There is still room for experimenting with new techniques. We plan to work not only with LSTMs and SVR, but also with other techniques for continuous emotion recognition, whether that is with continuous real values (regression) or quantised levels (with assigned labels, as would be done for CRFs). Furthermore, it would be interesting to experiment with learning techniques that can cope with missing values in the learning data, thus making the interpolation stage unnecessary (where the "missing values" problem is manifested)

- Furthermore, in this work we have not fully explored the decision-level fusion options available. It would be interesting to examine how SVR performs with decision-level fusion, while there are many other techniques for experimentation, such as using linear combinations [101].

- As aforementioned, the development of emotion-specific learning techniques is still an open research issue. A quite interesting work would be to evaluate the results that we achieved in

our experiments and determine how modifying or combining modern learning techniques can improve the performance in emotion recognition.

- Improving our normalisation and segmentation procedure: We have demonstrated that our normalisation procedure does decrease the MSE amongst the coders and does promote coder agreement. Experimentation with the segmentation procedure though could provide us with improved segments for the learning. Issues of improvement could relate to more accurate synchronisation and time-shifting for both valence and arousal, more robust matching between transitions from one emotional state to the other, synchronisation based on the peaks in the distributions or even synchronisation based on temporal intervals were missing annotations exist, while minimising the time shifting offset and maximising the number of agreeing coders.

Chapter 7

Appendix

7.1 Toolkits

For the learning techniques involved in this project, we have used the libsvm library for Support Vector Machines and Regression [33], RNNLIB `http://wiki.github.com/alexgraves/RNNLIB` for Long Short-Term Memory networks by Alex Graves, and the library developed for Coupled Hidden Markov Models by S. Kaltwang [100] based on The Bayes Net Toolbox for MATLAB [131]. The source code of each toolkit was modified where it was required, while for Support Vector Machines we experimented with the "Multi-class classification (and probability output) via error-correcting codes" extension, described in [94].

7.2 Algorithm Notation

We present common notation for the algorithms we present. It is noted that other notation may appear if described in the text.

- **a** $\leftarrow$ **b:** Assignment of b to a
- **a** = **b:** Check if b equals a
- **for a in C:** Iterate through the collection (or list, or vector) C, assigning at iteration i the i-th element of C to a
- **sign**(**s**)**:** returns 1 for a positive and -1 for a negative number. Since this function has been used for inputs which do not contain zeros (0), 0 is returned only if the input is not a number (NaN)
- **length**(**C**)**:** refers to the number of objects in collection C
- **int**(**K**)**:** returns the nearest integer of real number K
- **corr**(**coder**)**:** returns the correlation of *coder*

7.3 Useful Vocabulary

We present some of the terminology that is constantly used throughout the project:

- **annotations:** With the term annotations we describe the valence/arousal measurements that the human coders provided in order to evaluate the emotional state of the subjects presented in the SAL database sessions.
- **coder:** A coder is the human who manually performs the annotation of the audiovisual material in SAL.
- **segment, sequence:** In our description, the set of videos along with the ground truth annotations produced by our segmentation (Chapter 4) constitute the set of audiovisual *segments* we use for learning. We use the term sequence essentially to refer to the segment as a *sequence* of frames.
- **session:** By session, we refer to the SAL database sessions, before they were segmented by our method.
- **transition, crossover:** Especially in the segmentation chapter (Chapter 4), we refer to transitions from and to an emotional state to the other (we examine two discrete states: positive and negative). We use both words with essentially the same meaning, with the crossover being used mostly when we describe algorithms.

7.4 Tables

Table 7.1: Coder error with respect to the ground truth, by comparing the annotations before pre-processing

	Valence		**Arousal**	
Coder	**MSE**	**COR**	**MSE**	**COR**
JD	0.04687	0.452134	0.039294	0.530569
cc	0.063639	0.403239	0.038473	0.392515
dr	0.067365	0.483183	0.063935	0.610911
em	0.111626	0.391809	0.093746	0.426104
AVG	**0.072375**	**0.432591**	**0.058862**	**0.490025**

Table 7.2: Correlation, agreement (Equation 4.1) and $trust_1$ (Equation 4.2 for a=b=1) values for each of the sessions and each of the coders, after normalisation (mean=1, std=0). Each of the values presented is the averaged value of each coder towards the rest of the coders, as discussed in Section 4.1.3

Name	**Coder**											
		cc			**dr**			**em**			**JD**	
	COR	**AGR**	**TR1**	**COR**	**AGR**	**TR1**	**COR**	**AGR**	**TR1**	**COR**	**AGR**	**TR1**
edA01	0.62	0.72	0.67	0.52	0.66	0.59	0.52	0.66	0.59	0.62	0.68	0.65
edA02	0.64	0.72	0.68	0.44	0.63	0.54	0.66	0.71	0.69	0.64	0.73	0.68
edA03				0.60	0.69	0.65	0.51	0.68	0.59	0.45	0.66	0.56
edB01				0.66	0.71	0.68	0.74	0.74	0.74	0.62	0.71	0.66
edB02				0.54	0.60	0.57	0.57	0.63	0.60	0.52	0.62	0.57
edB03				0.64	0.68	0.66	0.75	0.70	0.72	0.68	0.66	0.67
ellA01	0.87	0.86	0.86	0.85	0.83	0.84	0.88	0.86	0.87	0.85	0.84	0.84
ellA02	0.73	0.78	0.76	0.66	0.74	0.70	0.72	0.78	0.75	0.51	0.65	0.58
ellA03	0.83	0.74	0.79	0.82	0.75	0.78	0.86	0.77	0.82	0.82	0.75	0.79
ellA04	0.42	0.59	0.50	0.22	0.53	0.37	0.49	0.63	0.56	0.34	0.59	0.46
ellB01	0.52	0.72	0.62	0.58	0.75	0.66	0.52	0.73	0.62	0.53	0.72	0.62
ellB02	0.67	0.73	0.70	0.66	0.70	0.68	0.67	0.70	0.68	0.74	0.75	0.74
ellB03	0.71	0.77	0.74	0.70	0.76	0.73	0.78	0.78	0.78	0.69	0.74	0.72
ellB04	0.62	0.75	0.69	0.67	0.72	0.69	0.67	0.76	0.71	0.53	0.68	0.60
ianA01	0.55	0.76	0.65	0.54	0.74	0.64	0.57	0.75	0.66	0.39	0.71	0.55
ianA02	0.14	0.59	0.37	0.36	0.65	0.51	0.35	0.66	0.51	0.36	0.65	0.51
ianB01	0.42	0.66	0.54	0.24	0.64	0.44	0.27	0.66	0.46	0.28	0.62	0.45
ianB02	0.27	0.69	0.48	0.34	0.69	0.51	0.36	0.69	0.53	0.37	0.67	0.52
rodA01	0.53	0.69	0.61	0.56	0.69	0.63	0.55	0.67	0.61	0.53	0.71	0.62
rodA02	0.70	0.80	0.75	0.67	0.77	0.72	0.72	0.80	0.76	0.62	0.77	0.70
rodA03	0.44	0.68	0.56	0.52	0.69	0.61	0.52	0.71	0.61	0.28	0.64	0.46
rodA04	0.27	0.62	0.44	0.25	0.59	0.42	0.41	0.66	0.54	0.19	0.62	0.41
rodB01	0.06	0.56	0.31	0.16	0.56	0.36	0.15	0.58	0.36	0.16	0.57	0.36
rodB02	0.55	0.69	0.62	0.52	0.70	0.61	0.48	0.66	0.57	0.46	0.67	0.57
rodB03	0.54	0.74	0.64	0.54	0.69	0.61	0.51	0.72	0.61	0.56	0.71	0.63
rodB04	0.22	0.63	0.43	0.14	0.61	0.37	0.32	0.65	0.49	0.19	0.64	0.42
rodB05	0.61	0.70	0.65	0.61	0.67	0.64	0.65	0.71	0.68	0.61	0.68	0.65
AVG	0.52	0.70	0.61	0.52	0.68	0.60	0.56	0.71	0.63	0.50	0.68	0.59

Table 7.3: Range of valence annotations for each session by each coder (cc, dr, em and JD and average), before and after normalising the data to zero mean. Before the normalisation, the average range of values is [0.603 -0.457], while after the normalisation it is [0.598 -0.463]. Narrowed down to two significant digits the range is identical, [0.60 to 0.46]

Valence																				
	no normalization										normalized to avg=1									
name	cc		dr		em		JD		AVG		cc		dr		em		JD		AVG	
	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN
edA01	0.730	0.020	0.700	-0.050	0.950	0.150	0.550	-0.300	0.730	-0.050	0.350	-0.360	0.360	-0.400	0.350	-0.460	0.430	-0.420	0.370	-0.410
edA02	0.650	-0.470	0.230	-0.410	0.720	-0.690	0.560	-0.500	0.540	-0.520	0.760	-0.360	0.420	-0.220	0.960	-0.450	0.790	-0.280	0.730	-0.330
edA03			0.650	-0.330	0.910	-0.390	0.440	-0.280	0.670	-0.330			0.440	-0.530	0.670	-0.620	0.420	-0.310	0.510	-0.490
edB01			0.620	-0.500	0.700	-0.410	0.770	-0.500	0.700	-0.470			0.480	-0.640	0.480	-0.630	0.660	-0.620	0.540	-0.630
edB02			0.180	-0.540	0.630	-0.500	0.620	-0.510	0.480	-0.520			0.380	-0.340	0.830	-0.300	0.810	-0.310	0.680	-0.320
edB03			0.790	-0.530	0.720	-0.380	0.580	-0.550	0.690	-0.490			0.650	-0.670	0.560	-0.540	0.540	-0.580	0.580	-0.600
ellA01	0.720	-0.740	0.680	-0.680	0.650	-0.500	0.760	-0.390	0.700	-0.580	0.730	-0.730	0.690	-0.670	0.670	-0.480	0.760	-0.390	0.710	-0.570
ellA02	0.540	-0.740	0.460	-0.670	0.440	-0.640	0.530	-0.430	0.490	-0.620	0.780	-0.510	0.530	-0.600	0.620	-0.460	0.620	-0.340	0.640	-0.480
ellA03	0.670	-0.710	0.730	-0.640	0.720	-0.580	0.690	-0.640	0.700	-0.640	0.930	-0.450	0.860	-0.510	1.000	-0.290	0.880	-0.450	0.920	-0.420
ellA04	0.450	-0.740	0.150	-0.540	0.560	-0.690	0.520	-0.520	0.420	-0.620	0.700	-0.490	0.340	-0.350	0.890	-0.370	0.770	-0.270	0.670	-0.370
ellB01	0.510	-0.440	0.200	-0.470	0.080	-0.610	0.190	-0.490	0.240	-0.500	0.590	-0.350	0.320	-0.350	0.400	-0.290	0.410	-0.280	0.430	-0.320
ellB02	0.540	-0.510	0.630	-0.580	0.330	-0.660	0.510	-0.460	0.500	-0.550	0.450	-0.600	0.590	-0.630	0.470	-0.520	0.520	-0.450	0.510	-0.550
ellB03	0.670	-0.520	0.770	-0.600	0.690	-0.560	0.900	-0.410	0.760	-0.520	0.430	-0.760	0.570	-0.800	0.520	-0.730	0.700	-0.600	0.560	-0.730
ellB04	0.350	-0.480	0.520	-0.620	0.870	-0.610	0.920	-0.680	0.660	-0.590	0.490	-0.330	0.680	-0.450	1.160	-0.310	1.100	-0.500	0.860	-0.400
ianA01	0.660	-0.200	0.610	-0.320	0.780	-0.190	0.560	-0.290	0.650	-0.250	0.420	-0.440	0.440	-0.490	0.490	-0.480	0.530	-0.320	0.470	-0.430
ianA02	0.360	-0.430	0.520	-0.560	0.890	-0.410	0.600	-0.370	0.590	-0.440	0.310	-0.480	0.550	-0.530	1.020	-0.280	0.700	-0.270	0.650	-0.390
ianB01	0.480	-0.240	0.510	-0.400	0.850	-0.220	0.500	-0.250	0.590	-0.280	0.290	-0.430	0.460	-0.450	0.530	-0.530	0.560	-0.200	0.460	-0.400
ianB02	0.520	0.000	0.660	-0.250	0.740	-0.240	0.580	-0.210	0.630	-0.170	0.220	-0.300	0.320	-0.580	0.330	-0.650	0.500	-0.290	0.340	-0.450
rodA01	0.510	-0.160	0.840	-0.490	0.790	-0.120	0.930	-0.280	0.770	-0.260	0.280	-0.390	0.500	-0.830	0.480	-0.430	0.740	-0.470	0.500	-0.530
rodA02	0.510	-0.250	0.730	-0.650	0.710	-0.670	0.990	-0.590	0.740	-0.540	0.460	-0.310	0.930	-0.450	0.860	-0.520	1.150	-0.430	0.850	-0.430
rodA03	0.700	-0.160	0.790	-0.670	0.690	-0.290	0.970	-0.370	0.790	-0.370	0.370	-0.480	0.720	-0.740	0.490	-0.490	1.030	-0.310	0.650	-0.500
rodA04	0.640	-0.020	0.670	-0.500	0.810	-0.190	1.000	-0.300	0.780	-0.250	0.310	-0.350	0.630	-0.550	0.530	-0.480	0.980	-0.320	0.610	-0.420
rodB01	0.010	-0.740	0.010	-0.850	-0.030	-0.770	-0.120	-0.890	-0.030	-0.810	0.430	-0.320	0.580	-0.280	0.540	-0.200	0.390	-0.390	0.490	-0.300
rodB02	0.630	-0.350	0.780	-0.640	0.470	-0.580	0.990	-0.450	0.720	-0.500	0.500	-0.480	0.900	-0.530	0.700	-0.340	0.740	-0.690	0.710	-0.510
rodB03	0.390	-0.720	0.080	-0.870	0.010	-0.790	0.970	-0.890	0.360	-0.820	0.630	-0.470	0.560	-0.390	0.530	-0.270	1.210	-0.640	0.740	-0.440
rodB04	0.610	0.000	0.690	-0.010	0.600	-0.350	0.960	-0.310	0.720	-0.170	0.250	-0.360	0.330	-0.380	0.370	-0.570	0.700	-0.570	0.410	-0.470
rodB05	0.540	-0.170	0.600	-0.600	0.630	-0.570	0.950	-0.590	0.680	-0.480	0.320	-0.390	0.470	-0.720	0.580	-0.610	0.840	-0.690	0.550	-0.610
AVG	0.539	-0.381	0.548	-0.517	0.626	-0.461	0.682	-0.461	**0.603**	**-0.457**	0.478	-0.441	0.544	-0.521	0.631	-0.456	0.721	-0.422	**0.598**	**-0.463**

Table 7.4: Range of arousal annotations for each session by each coder (cc, dr, em and JD and average), before and after normalising the data to zero mean. Average range before normalisation is [0.630 -0.406], which changes after normalisation to [0.536 -0.500]

Arousal																				
	no normalization										normalized to avg=1									
name	cc		dr		em		JD		AVG		cc		dr		em		JD		AVG	
	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN
edA01	0.470	-0.360	0.630	-0.270	0.770	0.030	0.520	-0.120	0.600	-0.180	0.310	-0.520	0.430	-0.470	0.380	-0.360	0.410	-0.220	0.380	-0.390
edA02	0.710	-0.430	0.840	-0.370	0.790	-0.310	0.720	-0.200	0.760	-0.330	0.490	-0.650	0.470	-0.740	0.360	-0.740	0.460	-0.450	0.450	-0.650
edA03			0.500	-0.510	0.640	-0.370	0.480	-0.240	0.540	-0.370			0.590	-0.420	0.290	-0.710	0.390	-0.330	0.430	-0.490
edB01			0.520	-0.410	0.720	-0.410	0.590	-0.510	0.610	-0.440			0.520	-0.410	0.490	-0.640	0.540	-0.560	0.520	-0.540
edB02			0.870	-0.740	0.650	-0.530	0.830	-0.450	0.780	-0.570			0.770	-0.840	0.450	-0.730	0.720	-0.550	0.650	-0.710
edB03			0.610	-0.370	0.850	-0.350	0.660	-0.170	0.700	-0.290			0.440	-0.540	0.450	-0.750	0.500	-0.320	0.460	-0.540
ellA01	0.460	-0.410	0.720	-0.540	0.610	-0.340	0.630	-0.190	0.600	-0.370	0.520	-0.350	0.680	-0.580	0.410	-0.540	0.470	-0.350	0.520	-0.450
ellA02	0.340	-0.730	0.720	-0.470	0.550	-0.670	0.740	-0.410	0.590	-0.570	0.460	-0.610	0.710	-0.480	0.560	-0.660	0.690	-0.450	0.600	-0.550
ellA03	0.620	-0.550	0.810	-0.580	0.830	-0.550	0.810	-0.450	0.770	-0.530	0.630	-0.540	0.710	-0.680	0.640	-0.740	0.670	-0.600	0.660	-0.640
ellA04	0.610	-0.350	0.900	-0.410	0.760	-0.380	0.840	-0.240	0.780	-0.350	0.440	-0.520	0.520	-0.790	0.370	-0.770	0.540	-0.540	0.470	-0.650
ellB01	0.370	-0.280	0.680	-0.260	0.580	-0.180	0.630	0.000	0.570	-0.180	0.380	-0.270	0.560	-0.390	0.220	-0.540	0.330	-0.300	0.370	-0.380
ellB02	0.240	-0.610	0.540	-0.550	0.450	-0.650	0.640	-0.280	0.470	-0.520	0.420	-0.440	0.650	-0.440	0.490	-0.600	0.520	-0.400	0.520	-0.470
ellB03	0.420	-0.390	0.720	-0.620	0.650	-0.490	0.570	-0.260	0.590	-0.440	0.420	-0.380	0.620	-0.720	0.600	-0.550	0.390	-0.440	0.510	-0.520
ellB04	0.220	-0.470	0.540	-0.520	0.560	-0.730	0.490	-0.340	0.450	-0.510	0.390	-0.300	0.670	-0.390	0.720	-0.570	0.460	-0.370	0.560	-0.410
ianA01	0.290	-0.330	0.640	-0.400	0.650	-0.410	0.490	-0.170	0.520	-0.330	0.280	-0.340	0.600	-0.430	0.280	-0.770	0.430	-0.230	0.400	-0.440
ianA02	0.420	-0.290	0.600	-0.630	0.540	-0.350	0.580	-0.280	0.530	-0.390	0.400	-0.310	0.630	-0.590	0.300	-0.590	0.480	-0.370	0.450	-0.460
ianB01	0.430	-0.300	0.600	-0.450	0.520	-0.490	0.570	-0.130	0.530	-0.340	0.440	-0.280	0.690	-0.360	0.400	-0.620	0.470	-0.230	0.500	-0.370
ianB02	0.310	-0.260	0.520	-0.520	0.630	-0.090	0.490	-0.010	0.490	-0.220	0.320	-0.260	0.640	-0.400	0.320	-0.390	0.380	-0.130	0.410	-0.300
rodA01	0.380	-0.200	0.750	-0.380	0.710	-0.180	0.560	-0.010	0.600	-0.190	0.280	-0.300	0.670	-0.460	0.410	-0.480	0.430	-0.150	0.450	-0.350
rodA02	0.580	-0.310	0.870	-0.750	0.720	-0.540	0.760	-0.470	0.730	-0.520	0.520	-0.360	0.630	-0.990	0.420	-0.850	0.550	-0.680	0.530	-0.720
rodA03	0.270	-0.430	0.530	-0.670	0.460	-0.410	0.270	-0.120	0.380	-0.410	0.440	-0.270	0.700	-0.500	0.510	-0.350	0.210	-0.180	0.460	-0.330
rodA04	0.440	-0.260	0.850	-0.260	0.700	-0.100	0.720	-0.090	0.680	-0.180	0.380	-0.320	0.530	-0.570	0.290	-0.510	0.560	-0.250	0.440	-0.410
rodB01	0.730	-0.710	0.850	-0.780	0.810	-0.840	0.940	-0.870	0.830	-0.800	1.110	-0.340	1.210	-0.420	1.200	-0.450	1.230	-0.580	1.190	-0.450
rodB02	0.440	-0.340	0.840	-0.440	0.710	-0.590	0.580	-0.290	0.640	-0.410	0.420	-0.360	0.750	-0.530	0.770	-0.530	0.510	-0.360	0.610	-0.450
rodB03	0.580	-0.650	0.890	-0.850	0.970	-0.800	0.900	-0.870	0.840	-0.790	0.720	-0.500	1.110	-0.620	1.010	-0.770	0.980	-0.790	0.950	-0.670
rodB04	0.500	-0.290	0.840	-0.360	0.790	-0.440	0.730	-0.030	0.720	-0.280	0.400	-0.400	0.640	-0.550	0.440	-0.800	0.420	-0.340	0.470	-0.520
rodB05	0.500	-0.390	0.870	-0.400	0.790	-0.620	0.660	-0.350	0.700	-0.440	0.360	-0.530	0.750	-0.520	0.460	-0.950	0.410	-0.600	0.500	-0.650
AVG	0.449	-0.406	0.713	-0.500	0.682	-0.437	0.644	-0.280	**0.630**	**-0.406**	0.458	-0.398	0.663	-0.549	0.490	-0.628	0.524	-0.399	**0.536**	**-0.500**

Table 7.5: Audiovisual data and annotations. All videos have a frame rate of 25 fps

	Session Filename	Length (seconds)	Coders
1	edA01	332.04	4,{cc,dr,em,JD}
2	edA02	459.20	4,{cc,dr,em,JD}
3	edA03	493.16	3,{dr,em,JD}
4	edB01	330.78	3,{dr,em,JD}
5	edB02	348.99	3,{dr,em,JD}
6	edB03	282.15	3,{dr,em,JD}
7	ellA01	438.64	4,{cc,dr,em,JD}
8	ellA02	372.36	4,{cc,dr,em,JD}
9	ellA03	595.88	4,{cc,dr,em,JD}
10	ellA04	394.71	4,{cc,dr,em,JD}
11	ellB01	346.52	4,{cc,dr,em,JD}
12	ellB02	410.67	4,{cc,dr,em,JD}
13	ellB03	305.60	4,{cc,dr,em,JD}
14	ellB04	357.51	4,{cc,dr,em,JD}
15	ianA01	237.64	4,{cc,dr,em,JD}
16	ianA02	394.92	4,{cc,dr,em,JD}
17	ianB01	289.92	4,{cc,dr,em,JD}
18	ianB02	220.68	4,{cc,dr,em,JD}
19	rodA01	426.03	4,{cc,dr,em,JD}
20	rodA02	307.30	4,{cc,dr,em,JD}
21	rodA03	483.58	4,{cc,dr,em,JD}
22	rodA04	372.38	4,{cc,dr,em,JD}
23	rodB01	430.56	4,{cc,dr,em,JD}
24	rodB02	335.99	4,{cc,dr,em,JD}
25	rodB03	515.60	4,{cc,dr,em,JD}
26	rodB04	298.40	4,{cc,dr,em,JD}
27	rodB05	357.88	4,{cc,dr,em,JD}

Bibliography

[1] Mehrabian A. and J. A. Russell. *An Approach to Environmental Psychology*. MIT Press, 1980.

[2] A. Aizerman, E. M. Braverman, and L. I. Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

[3] A. N. Ali and P. H. Marsden. Affective multi-modal interfaces: the case of mcgurk effect. In *IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces*, pages 224–226, New York, NY, USA, 2003. ACM.

[4] G. Allport. *Social Psychology*. Houghton Mifflin, Boston, 1924.

[5] N. Ambady and R. Rosenthal. [pdf] half a minute: Predicting teacher evaluations from thin slices of nonverbal behavior and physical attractiveness. *Journal of Personality and Social Psychology*, 1993.

[6] Affect analysis group. `http://www.pitt.edu/~emotion/publications.html`, 2009.

[7] K. Anderson and P.W. McOwan. A real-time automated system for the recognition of human facial expressions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 2006.

[8] A.B. Ashraf, S. Lucey, J.F. Cohn, T. Chen, Z. Ambadar, K. Prkachin, P.Solomon, and B. J. Theobald. The painful face: Pain expression recognition using active appearance models. *Proceedings of the 9th international conference on Multimodal interfaces*, 2007.

[9] P. Baldi, S. Brunak, P. Frasconi, G. Pollastri, and G. Soda. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, 15:937–946, 1999.

[10] P. Baldi, S. Brunak, P. Frasconi, G. Pollastri, and G. Soda. Bidirectional dynamics for protein secondary structure prediction. *Lecture Notes in Computer Science*, 1828:80–104, 2001.

[11] T. Bänziger and K. Scherer. Using actor portrayals to systematically study multimodal emotion expression: The GEMEP corpus. In *ACII '07: Proceedings of the Second International Conference on Affective Computing and Intelligent Interaction*, pages 476–487. Springer, 2007.

[12] S Baron-Cohen and T.H.E. Tead. *Mind reading: The interactive guide to emotion*. Jessica Kingsley Publishers Ltd, 2003.

[13] M.S. Bartlett, J.C. Hager, P. Ekman, and T.J. Sejnowski. Measuring facial expressions by computer image analysis. *Psychophysiology*, 1999.

[14] M.S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Recognizing facial expression: machine learning and application to spontaneous behavior. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2:568–573 vol. 2, June 2005.

[15] M.S. Bartlett, G. Littlewort, M.G. Frank, C. Lainscsek, I. Fasel, and J. Movellan. [pdf] fully automatic facial action recognition in spontaneous behavior. *Proc. Conf. Face & Gesture Recognition*, 2006.

[16] A. Batliner, K. Fischer, R. Huber, J. Spilker, and E. Noth. How to find trouble in communication. *Speech Communication*, 2003.

[17] P. Belin, S. Fillion-Bilodeau, and F. Gosselin. The montreal affective voices: A validated set of nonverbal affect bursts for research on auditory affective processing. *Behavior Research Methods*, 40(2):531–539, May 2008.

[18] Richard E. Bellman. *Adaptive control processes - A guided tour*. Princeton University Press, Princeton, New Jersey, U.S.A., 1961.

[19] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[20] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, September 1999.

[21] R. Bhiksha and S. Rita. Classification in likelihood spaces. *Technometrics*, pages 318–329, 2004.

[22] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition, 1996.

[23] J. Buckworth and R. K. Dishman. *Exercise Psychology*. Human Kinetics Europe, 2002.

[24] D.B. Buller and J.K. Burgoon. Interpersonal deception theory. *Communication Theory*, 1996.

[25] D.B. Buller, J.K. Burgoon, C.H. White, and A.S. Ebesu. Interpersonal deception: Vii. behavioral profiles of falsification, equivocation, and concealment. *Journal of Language and Social Psychology*, 1994.

[26] N. Campbell and P. Mokhtari. [pdf] voice quality: the 4th prosodic dimension. *15 th International Congress of Phonetic Sciences*, 2003.

[27] G. Caridakis, K. Karpouzis, and S. Kollias. User and context adaptive neural networks for emotion recognition. *Neurocomput.*, 71(13-15):2553–2562, 2008.

[28] G. Caridakis, L. Malatesta, L. Kessous, N. Amir, A. Raouzaiou, and K. Karpouzis. Modeling naturalistic affective states via facial and vocal expressions recognition. In *ICMI '06: Proceedings of the 8th international conference on Multimodal interfaces*, pages 146–154, New York, NY, USA, 2006. ACM.

[29] C.Clavela, I. Vasilescu, L. Devillers, G.Richard, and T. Ehrette. Fear-type emotion recognition for future audio-based surveillance systems. *Speech Communication*, 50(6):487–503, 2008.

[30] Hjortsjo C.H. *Man's face and mimic language*. Malmo, Studentlitteratur, 1970.

[31] G. Chanel, K. Ansari-Asl, and T. Pun. Valence-arousal evaluation using physiological signals in an emotion recall paradigm. In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pages 2662–2667, Oct. 2007.

[32] G. Chanel, K. Ansari-Asl, and T. Pun. Valence-arousal evaluation using physiological signals in an emotion recall paradigm. In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pages 2662–2667, 2007.

[33] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[34] Y. Chang, C. Hu, R. Feris, and M. Turk. Manifold based analysis of facial expression. *Image and Vision Computing*, 2006.

[35] J. Chen and N.S. Chaudhari. Capturing long-term dependencies for protein secondary structure prediction. In Fuliang Yin, Jun Wang, and Chengan Guo, editors, *Advances in Neural Networks - ISNN 2004, International Symposiumon Neural Networks, Part II*, volume 3174 of *Lecture Notes in Computer Science*, pages 494–500, Dalian, China, 2004. Springer.

[36] P. Clifford. Markov random fields in statistics, 1990.

[37] J.F. Cohn and K. Schmidt. The timing of facial motion in posed and spontaneous smiles. *Active Media Technology*, 2003.

[38] C. Cortes and V. Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

[39] M. Coulson. Attributing emotion to static body postures: Recognition accuracy, confusions, and viewpoint dependence. *Nonverbal Behavior*, 28(2):117•139, 2004.

[40] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J.G. Taylor. Emotion recognition in human-computer interaction. *Signal Processing Magazine, IEEE*, 18(1):32–80, Jan 2001.

[41] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Cambridge, UK, March 2000.

[42] A. Cruttenden. *Intonation / Alan Cruttenden*. Cambridge University Press, Cambridge [Cambridgeshire] ; New York :, 1986.

[43] C. Darwin. *The expression of the emotions in man and animals /*. New York ;D. Appleton and Co.„ 1872/1916. http://www.biodiversitylibrary.org/bibliography/4820.

[44] J.R. Davitz. Auditory correlates of vocal expressions of emotional meanings. *The Communication of Emotional Meaning*, pages 101–112, 1964.

[45] B. de Gelder, K. B. E. Bfcker, J.i Tuomainen, M. Hensen, and J. Vroomen. The combined perception of emotion from voice and face: early interaction revealed by human electric brain responses. *Neuroscience Letters*, 260(2):133 – 136, 1999.

[46] B.M. DePaulo, J.J. Lindsay, B.E. Malone, L. Muhlenbruck, K. Charlton, and H. Cooper. [pdf] cues to deception. *Psychological Bulletin*, 2003.

[47] R. Descartes. The passions of the soul (1649). *The Philosophical Work*, Vol.

[48] L. Devillers, I. Vasilescu, and L. Vidrascu. Anger versus fear detection in recorded conversations. *Proceedings of speech prosody*, 2004.

[49] E. Douglas-Cowie, R. Cowie, I.n Sneddon, C. Cox, O. Lowry, M. Mcrorie, J. Martin, L. Devillers, S. Abrilian, A. Batliner, N. Amir, and K. Karpouzis. The humaine database: Addressing the collection and annotation of naturalistic and induced emotional data. In *ACII '07: Proceedings of the 2nd international conference on Affective Computing and Intelligent Interaction*, pages 488–500, Berlin, Heidelberg, 2007. Springer-Verlag.

[50] P. Dreuw, T. Deselaers, D. Rybach, D. Keysers, and H. Ney. [pdf] tracking using dynamic programming for appearance-based sign language recognition. *IEEE Automatic Face and Gesture Recognition*, 2006.

[51] R. Duan, W. Jiang, and H. Man. Robust adjusted likelihood function for image analysis. In *AIPR '06: Proceedings of the 35th Applied Imagery and Pattern Recognition Workshop*, page 29, Washington, DC, USA, 2006. IEEE Computer Society.

[52] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, July 1999.

[53] R. Edgeworth, E. Keen, B.and Crane, M. Gross, and A. Arbor. Effect of speed on emotion-related kinematics during walking. *North American Congress on Biomechanics*, 2008.

[54] P. Ekman. *Universals and Cultural Differences in Facial Expressions of Emotion*. University of Nebraska Press, 1971.

[55] P. Ekman. About brows: Emotional and conversational signals. *Human ethology: Claims and limits of a new discipline*, 1979.

[56] P. Ekman. [pdf] darwin, deception, and facial expression. *Ann. NY Acad. Sci*, 2003.

[57] P. Ekman and W. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press, Palo Alto, 1978.

[58] P. Ekman and W. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press, Palo Alto, 1978.

[59] P. Ekman and W. V. Friesen. Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, 17(2):124–129, 1971.

[60] P. Ekman and E.L. Rosenberg. *[BOOK] What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System*. books.google.com, 2005.

[61] R. el Kaliouby, R.W. Picard, and K. Dautenhan. Autism and affective-social computing tutorial, 2007.

[62] C. Elkan. Log-linear models and conditional random fields: Notes for a tutorial at the acm 17th conference on information and knowledge management, 2008.

[63] J. L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

[64] R Fabio, F. Dieter, and D.-W. Hugh. Crf-matching: Conditional random fields for feature-based scan matching. In *Robotics Science and Systems*, 2007.

[65] Z. Fang, Z. Guoliang, and S. Zhanjiang. Comparison of different implementations of mfcc. *J. Comput. Sci. Technol.*, 16(6):582–589, 2001.

[66] I. Fasel, B. Fortenberry, and J. Movellan. A generative framework for real time object detection and classification. *Computer Vision and Image Understanding*, 2005.

[67] Feeltrace. `http://www.dfki.de/~schroed/feeltrace/`, 2009.

[68] W. V. Friesen and P Ekman. Emfacs-7: Emotional facial action coding system, 1984.

[69] T. Fukada, M. Schuster, and Y. Sagisaka. Phoneme boundary estimation using bidirectional recurrent neural networks and its applications. *Systems and Computers in Japan*, 30(4):20–30, 1999.

[70] J. B. Gao, S. R. Gunn, C. J. Harris, and M. Brown. A probabilistic framework for svm regression and error bar estimation. *Mach. Learn.*, 46(1-3):71–89, 2002.

[71] F. Gers. *Long short-term memory in recurrent neural networks*. PhD thesis, EPFL, Lausanne, 2001.

[72] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12:2451–2471, 2000.

[73] S.B. Gokturk, J.Y. Bouguet, C. Tomasi, and B. Girod. [pdf] model-based face tracking for view-independent facial expression recognition. *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, 2002.

[74] D. Grandjean, D. Sander, and K. R. Scherer. Conscious emotional experience emerges as a function of multilevel, appraisal-driven response synchronization. *Consciousness and Cognition*, 17(2):484 – 495, 2008. Social Cognition, Emotion, and Self-Consciousness.

[75] M. Graver. *Cicero on the emotions: Tusculan Disputations 3 and 4*. Chicago: University of Chicago Press, 2002.

[76] A. Graves, S. Fernandez, F.J. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In W. W. Cohen and A. Moore, editors, *ICML*, volume 148 of *ACM International Conference Proceeding Series*, pages 369–376. ACM, 2006.

[77] A. Graves, A. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(5):855–868, 2009.

[78] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18:602–610, 2005.

[79] A. Graves and J. Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Le'on Bottou, editors, *NIPS*, pages 545–552. MIT Press, 2008.

[80] J. A. Gray, editor. *Cognition, emotion, conscious experience and the brain*. Handbook of Cognition and Emotion. Wiley and Sons, New York, 1999.

[81] M. Grimm, K. Kroschel, and S. Narayanan. The vera am mittag german audio-visual emotional speech database. In *ICME*, pages 865–868. IEEE, 2008.

[82] M.M. Gross, E.A. Crane, and B.L Fredrickson. Effect of felt and recognized emotions on gait kinematics. *American Society of Biomechanics Conference, Palo Alto, CA*, 2007.

[83] M.M. Gross, E.A. Crane, and B.L. Fredrickson. Expression of emotion changes gait kinematics. *International Society for Posture and Gait Research, Burlington, VT*, 2007.

[84] H. Gunes and M. Pantic. Automatic dimensional and continuous emotion recognition. *International Journal of Synthetic Emotions*, 2009.

[85] H. Gunes and M. Piccardi. Creating and annotating affect databases from face and body display: A contemporary survey. *IEEE Int. Conf. on Systems, Man and Cybernetics*, pages 2426–2433, 2006.

[86] H. Gunes, M. Piccardi, and M. Pantic. From the lab to the real world: affect recognition using multiple cues and modalities. In J. Or, editor, *Affective computing: focus on emotion expression, synthesis, and recognition*, pages 185–218. InTech Education and Publishing, Vienna, Austria, 2008.

[87] G. Guo and C.R. Dyer. Learning from examples in the small sample case: face expression recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 2005.

[88] N. Hadjikhani and Gelder B. de. Seeing fearful body expressions activates the fusiform cortex and amygdala. *Current Biology*, 13(24):2201–2205, December 2003.

[89] S. Hochreiter. Untersuchungen zu dynamischen neuronalen Netzen. Diploma thesis, Institut für Informatik, Lehrstuhl Prof. Brauer, Technische Universität München, 1991.

[90] S. Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 6(2):107–116, 1998.

[91] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[92] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *PNAS*, 79(8):2554–2558, April 1982.

[93] C.-W. Hsu, C.-C. Chang, and C.-J. Lin. A practical guide to support vector classification, 2000.

[94] Tzu-Kuo Huang, Ruby C. Weng, and Chih-Jen Lin. Generalized bradley-terry models and multi-class probability estimates. *J. Mach. Learn. Res.*, 7:85–115, 2006.

[95] H. Jaeger. The "echo state" approach to analysing and training recurrent neural networks. GMD Report 148, GMD - German National Research Institute for Computer Science, 2001.

[96] L. C. Jain and L. R. Medsker. *Recurrent Neural Networks: Design and Applications*. CRC Press, Inc., Boca Raton, FL, USA, 1999.

[97] H. E. Jean and J. Rouat. Combining pitch and mfcc for speaker recognition systems, 2001.

[98] I. T. Jolliffe. *Principal Component Analysis*. Springer, second edition, October 2002.

[99] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, second edition, February 2008.

[100] S. Kaltwang. Dynamic modeling of human nonverbal behavior from multiple cues and modalities. *Project Report, Universitat Karlsruhe (TH), Institute for Anthropomatics and Imperial College London, Department of Computing*, 2009.

[101] I. Kanluan, M. Grimm, and K. Kroschel. Audio-visual emotion recognition using an emotion recognition space. *16th European Signal Processing Conference*, 2008.

[102] A. Kappas, U. Hess, and K. R. Scherer. *Voice and emotion*. Fundamentals of nonverbal behavior. Cambridge University Press., Cambridge and New York, r.s. feldman and b. rimi (eds.) edition, 1991.

[103] Michael Kass, Andrew P. Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[104] A. Kleinsmith and N. Bianchi-Berthouze. Recognizing affective dimensions from body posture. In *ACII '07: Proceedings of the 2nd international conference on Affective Computing and Intelligent Interaction*, pages 48–58, Berlin, Heidelberg, 2007. Springer-Verlag.

[105] M. Kubat. Neural networks: a comprehensive foundation by simon haykin, macmillan, 1994, isbn 0-02-352781-7. *Knowl. Eng. Rev.*, 13(4):409–412, 1999.

[106] D. Kulic and E. A. Croft. Affective state estimation for human-robot interaction. *IEEE Transactions on Robotics*, 23(5):991–1000, 2007.

[107] R. Laban and L. Ullmann. *The Mastery of Movement*. Princeton Book Company Publishers; 4 Revised edition, 1988.

[108] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data, 2001.

[109] R.J. Larsen and E. Diener. Affect intensity as an individual difference characteristic: A review. *Journal of research in personality(Print)*, 1987.

[110] J. Laver. *Principles of Phonetics (Cambridge Textbooks in Linguistics)*. Cambridge University Press, June 1994.

[111] C.M. Lee and S.S. Narayanan. Toward detecting emotions in spoken dialogs. *IEEE Transactions on Speech and Audio Processing*, 2005.

[112] J. Legge, C. Chai, and W. Chai. *Li chi: book of rites*. University Books New York, 1967.

[113] R. Levenson. Emotion and the autonomic nervous system: A prospectus for research on autonomic specificity. *Social Psychophysiology and Emotion: Theory and Clinical Applications*, pages 17–42, 1988.

[114] M. Lewis. *Handbook of emotions*. Guilford Press, 2008.

[115] R. Lienhart and J. Maydt. [pdf] an extended set of haar-like features for rapid object detection. *IEEE ICIP*, 2002.

[116] G..C Littlewort, M.S. Bartlett, and K. Lee. Faces of pain: automated measurement of spontaneous facial expressions of genuine and posed pain. *Proceedings of the 9th international conference on Multimodal interfaces*, 2007.

[117] Grimm M. and Kroschel K. Emotion estimation in speech using a 3d emotion space concept. In *6th European Signal Processing Conference*, 2008.

[118] Gross M., Gerstner G., Koditschek D., Fredrickson B., and Crane E. Emotion recognition from body movement kinematics. *American Society of Biomechanics, Portland, OR*, 2004.

[119] W. M. Massaro and M. M. Cohen. Fuzzy logical model of bimodal emotion perception: Comment on "the perception of emotions by ear and by eye" by de gelder and vroomen. *Cognition & Emotion*, 14(3):313–320, 2000.

[120] R.A. Masters. Compassionate wrath: Transpersonal approaches to anger. *Journal of Transpersonal Psychology*, 2000.

[121] Piecewise Cubic Hermite Interpolating Polynomial (PCHIP). MATLAB Documentation. [online] `http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/ref/pchip.html`.

[122] 1-D data interpolation: interp1. MATLAB Documentation. [online] `http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/ref/interp1.html`.

[123] H. G. Mayer, F. J. Gomez, D. Wierstra, I. Nagy, A. Knoll, and J. Schmidhuber. A system for robotic heart surgery that learns to tie knots using recurrent neural networks. In *IROS*, pages 543–548. IEEE, 2006.

[124] A. McCallum, D. Freitag, and F. C. N. Pereira. Maximum entropy markov models for information extraction and segmentation. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 591–598, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[125] D. McNeill. *The conceptual basis of language / David McNeill*. Lawrence Erlbaum Associates ; distributed by the Halsted Press, Division of Wiley, Hillsdale, N.J. : New York :, 1979.

[126] D. McNeill. So you think gestures are nonverbal? *Psychological Review*, 92:350–371, 1985.

[127] D. Mcneill. *Language and Gesture (Language Culture and Cognition)*. Cambridge University Press, August 2000.

[128] T. M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[129] S. Mitra and T. Acharya. [pdf] gesture recognition: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions*, 2007.

[130] J. Montepare, E. Koff, D. Zaitchik, and M. Albert. The use of body movement and gestures as cues to emotions in younger and older adults. *Journal of Nonverbal Behavior*, 23:133–152, 1999.

[131] K. P. Murphy. The bayes net toolbox for matlab.

[132] K. P. Murphy. Dynamic bayesian networks: Representation, inference and learning, 2002.

[133] R. Rosenthal N. Ambady. Thin slices of expressive behavior as predictors of interpersonal consequences : a meta-analysis. *Psychological Bulletin*, 111(2):256–274, 1992.

[134] H. Ning, T.X. Han, Y. Hu, Z. Zhang, Y. Fu, and T.S. Huang. [pdf] a realtime shrug detector. *Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 2006.

[135] T. L. Nwe, S. W. Foo, and L. C. De Silva. Speech emotion recognition using hidden markov models. *Speech Communication*, 41(4):603–623, November 2003.

[136] A. Ortony and T. J. Turner. What's basic about basic emotions? *Psychol Rev*, 97(3):315–331, July 1990.

[137] C. E. Osgood, G. Suci, and P. Tannenbaum. *The measurement of meaning*. University of Illinois Press, Urbana, IL, 1957.

[138] M. Pantic and M.S. Bartlett. Machine analysis of facial expressions. In K. Delac and M. Grgic, editors, *Face Recognition*, pages 377–416. I-Tech Education and Publishing, Vienna, Austria, July 2007.

[139] M. Pantic and I. Patras. Dynamics of facial expression: Recognition of facial actions and their temporal segments from face profile image sequences. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 2006.

[140] M. Pantic and L.J.M. Rothkrantz. Facial action recognition for facial expression analysis from static face images. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 2004.

[141] M. Pantic, M. Valstar, R. Rademaker, and L. Maat. Web-based database for facial expression analysis. *IEEE International Conference on Multimedia and Expo, 2005.*, 2005.

[142] I. Patras and M. Pantic. Particle filtering with factorized likelihoods for tracking facial features. In *FGR*, pages 97–104, 2004.

[143] B. Paul. Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound, 1993.

[144] I. Pavlidis, J. Levine, and P. Baukol. Thermal imaging for anxiety detection. In *CVBVS '00: Proceedings of the IEEE Workshop on Computer Vision Beyond the Visible Spectrum: Methods and Applications (CVBVS 2000)*, page 104, Washington, DC, USA, 2000. IEEE Computer Society.

[145] S. Petridis, H. Gunes, S. Kaltwang, and M. Pantic. Static vs. dynamic modeling of human nonverbal behavior from multiple cues and modalities. In *11th ACM Int. Conf. on Multimodal Interfaces*, November 2009.

[146] S. Petridis and M. Pantic. Audiovisual laughter detection based on temporal features. *Proceedings of the 10th international conference on conference on Multimodal interfaces*, 2008.

[147] D. Pinto, A. McCallum, X. Lee, and W. Croft. Table extraction using conditional random fields. In *Proceedings of the 26th ACM SIGIR*, 2003.

[148] Michael K. Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999.

[149] R. Plutchik. A general psychoevolutionary theory of emotion. *Emotion: Theory, research, and experience: Vol. 1. Theories of emotion*, page 3•33, 1980.

[150] R. Plutchik and H. R. Conte. *Circumplex models of personality and emotions*. Washington, DC: American Psychological Association, 1997.

[151] R. Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 2007.

[152] J. Posner, J. A. Russer, and B. S. Peterson. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and Psychopathology*, 17(03):715–734, 2005.

[153] L. Rabiner and B.-H. Juang. *Fundamentals of speech recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[154] A. S. Reber. Implicit learning of artificial grammars. *Journal of Verbal Learning and Verbal Behavior*, 6:855–863, 1967.

[155] R. Rosenthal, K. Scherer, and J. Harrigan. *Vocal expression of affect, In: The New Handbook of Methods in Nonverbal Behavior Research*. Harrigan, 2005.

[156] J. A. Russell and J. D. Fernndez Dols. *The psychology of facial expression*. Cambridge University Press, Cambridge, 1997.

[157] J.A. Russell. Affective space is bipolar. *Journal of Personality and Social Psychology*, 37:345–356, 1979.

[158] J.A. Russell. [pdf] culture and the categorization of emotions. *Psychological Bulletin*, 1991.

[159] Hava T. S. and Eduardo D. S. Turing computability with neural nets. *Applied Mathematics Letters*, 4:77–80, 1991.

[160] D. Sander, D. Grandjean, and K. R. Scherer. A systems approach to appraisal mechanisms in emotion. *Neural Netw.*, 18(4):317–352, 2005.

[161] K. R. Scherer. *Appraisal theory*. Handbook of Cognition and Emotion. Wiley, Chichester, t. dalgleish and m. power (eds.) edition, 1999.

[162] K. R. Scherer. Psychological models of emotion. In Joan C. Borod, editor, *The Neuropsychology of Emotion*, pages 137–166. Oxford University Press US, Oxford, New York, Estados Unidos, 2000.

[163] K.L. Schmidt and J.F. Cohn. Human facial expressions as adaptations: Evolutionary questions in facial expression research. *Yearbook of Physical Anthropology*, 2001.

[164] B. Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. The MIT Press, December 2001.

[165] S. Schotz. Linguistic & paralinguistic phonetic variation in speaker recognition & text-to-speech synthesis. term paper for course. In *in Speech Technology, GSLT. Available on the Web at* `http://www.speech.kth.se/~rolf/gslt_papers/SusanneSchotz.pdf`, 2002.

[166] M. Schuster. *On supervised learning from sequential data with applications for speech recognition*. PhD thesis, Nara Institute of Science and Technology, 1999.

[167] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45:2673–2681, November 1997.

[168] semaine. `http://www.semaine-project.eu/`, 2009.

[169] F. Sha and Fernando C. N. Pereira. Shallow parsing with conditional random fields. In *HLT-NAACL*, 2003.

[170] Y.-S. Shin. Facial expression recognition based on emotion dimensions on manifold learning. In *ICCS '07: Proceedings of the 7th international conference on Computational Science, Part II*, pages 81–88, Berlin, Heidelberg, 2007. Springer-Verlag.

[171] H. T. Siegelmann. *Neural Networks and Analog Computation: Beyond the Turing Limit (Progress in Theoretical Computer Science)*. Birkhäuser Boston, 1 edition, December 1998.

[172] B. Spinoza. Ethics, 1677.

[173] K. N. Spreckelmeyer, M. Kutas, T. P. Urbach, E. Altenmller, and T. F. Mnte. Combined perception of emotion in pictures and musical sounds. *Brain Research*, 1070(1):160 – 170, 2006.

[174] J. Tao and T. Tan. Affective computing: A review. *Lecture Notes in Computer Science*, 2005.

[175] A. Tellegen and D. Watson. Toward a consensual structure of mood. *Psychological Bulletin*, 1985.

[176] R.E. Thayer. *[BOOK] The biopsychology of mood and arousal*. Oxford University Press US, 1989.

[177] T. Thireou and M. Reczko. Bidirectional long short-term memory networks for predicting the subcellular localization of eukaryotic proteins. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 4(3):441–446, 2007.

[178] Y. Tian, T. Kanade, and J. F. Cohn. Evaluation of gabor-wavelet-based facial action unit recognition in image sequences of increasing complexity. In *FGR '02: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, page 229, Washington, DC, USA, 2002. IEEE Computer Society.

[179] P. Tsiamyrtzis, J. Dowdall, D. Shastri, I.T. Pavlidis, M.G. Frank, and P. Ekman. Imaging facial physiology for the detection of deceit. *International Journal of Computer Vision*, 2007.

[180] V. Tyagi and C. Wellekens. On desensitizing the mel-cepstrum to spurious spectral components for robust speech recognition. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 1, pages 529–532, 18-23, 2005.

[181] M. Valstar and M. Pantic. Fully automatic facial action unit detection and temporal analysis. In *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, page 149, Washington, DC, USA, 2006. IEEE Computer Society.

[182] M. Valstar, M. Pantic, and I. Patras. Motion history for facial action detection in video. *2004 IEEE International Conference on Systems, Man and Multimodal interfaces*, 2004.

[183] M. F. Valstar and M. Pantic. Biologically vs. logic inspired encoding of facial actions and emotions. In *in video, Proc. IEEE Int'l Conf. Multimedia and Expo*, pages 325–328, 2006.

[184] M.F. Valstar, H. Gunes, and M. Pantic. How to distinguish posed from spontaneous smiles using geometric features. *Proceedings of the 9th international conference on Multimodal interfaces*, 2007.

[185] M.F. Valstar and M. Pantic. [pdf] fully automatic facial action unit detection and temporal analysis. *Proc. Int'l Conf. Computer Vision and Pattern Recognition*, 2006.

[186] M.F. Valstar, M. Pantic, Z. Ambadar, and J.F. Cohn. Spontaneous vs. posed facial behavior: Automatic analysis of brow actions. *Proceedings of the 8th international conference on Multimodal interfaces*, 2006.

[187] J. Van den Stock, R. Righart, and B. de Gelder. Body expressions influence recognition of emotions in the face and voice. *Emotion*, 7(3):487–494, August 2007.

[188] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[189] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[190] V. Vapnik, S. E. Golowich, and A. J. Smola. Support vector method for function approximation, regression estimation and signal processing. In Michael Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems 9 — Proceedings of the 1996 Neural Information Processing Systems Conference (NIPS 1996), December 2-5, 1996, Dever, CO, USA*, pages 281–287. MIT Press, Cambridge, MA, USA, 1997.

[191] V. N. Vapnik and A. Ya. Chervonenkis. *Theory of Pattern Recognition [in Russian]*. Nauka, USSR, 1974.

[192] P. Viola and M.J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 2004.

[193] D. Vrakas and Ioannis P. Vlahavas. *Artificial Intelligence for Advanced Problem Solving Techniques*. Information Science Reference - Imprint of: IGI Publishing, Hershey, PA, 2008.

[194] Hanna M. Wallach. Conditional random fields: An introduction. Technical Report MS-CIS-04-21, University of Pennsylvania, 2004.

[195] C. M. Whissell. The dictionary of affect in language. *Emotion: Theory, research and experience. The measurement of emotions*, 4:113–131, 1989.

[196] A Wierzbicka. Talking about emotions: Semantics, culture, and cognition. *Cognition and Emotion*, 6:285–319, 1992b.

[197] R. J. Williams and J. Peng. An efficient gradient–based algorithm for on–line training of recurrent network trajectories. *Neural Computation*, 2(4):490–501, 1990.

[198] A.D. Wilson, A.E. Bobick, and J. Cassell. Temporal classification of natural gesture and application to videocoding. *Conference on Computer Vision and Pattern Recognition, 1997*, 1997.

[199] Wollmer, M. and Eyben, F. and Reiter, S. and Schuller, B. and Cox, C. and Douglas-Cowie, E. and Cowie, R. Abandoning emotion classes - towards continuous emotion recognition with modelling of long-range dependencies. In *Proc. 9th Interspeech 2008 incorp. 12th Australasian Int. Conf. on Speech Science and Technology SST 2008, Brisbane, Australia*, pages 597–600. ISCA, 2008. 22.-26.09.2008, ISSN 1990-9772.

[200] J. Xiao, T. Kanade, and J.F. Cohn. Robust full-motion recovery of head by dynamic templates andre-registration techniques. *Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, 2002.

[201] M. Xu, L.-Y. Duan, J. Cai, L.-T. Chia, C. Xu, and Q. Tian. Hmm-based audio keyword generation. In Kiyoharu Aizawa, Yuichi Nakamura, and Shin'ichi Satoh, editors, *PCM (3)*, volume 3333 of *Lecture Notes in Computer Science*, pages 566–574. Springer, 2004.

[202] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys (CSUR)*, 2006.

[203] Z. Zeng, M. Pantic, G.n I. Roisman, and T. S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):39–58, 2009.

[204] Y. Zhang and Q. Ji. Active and dynamic information fusion for facial expression understanding from image sequences. *IEEE Transactions on pattern analysis and machine intelligence*, 2005.

[205] M. Zuckerman, D. T. Larrance, J. A. Hall, R. S. Defrank, and R. Rosenthal. Posed and spontaneous communication of emotion via facial and vocal cues. *Journal of Personality*, 47(4):712–733, 1979.