Automatic detection of human affective behavior in dyadic conversations

by

Bhavan Jasani

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Robotics

in the

Graduate Division

of the

Carnegie Mellon University, Pittsburgh

Committee in charge:

Professor Jeffrey Cohn, Chair
Dr. Laszlo Jeni
Professor Louis Philippe Morency
Rohit Girdhar

Summer 2019

CMU-RI-TR-19-53

The thesis of Bhavan Jasani, titled Automatic detection of human affective behavior in dyadic conversations, is approved:

Chair _____ Date _____

_____ Date _____

_____ Date _____

_____ Date _____

Carnegie Mellon University, Pittsburgh

Automatic detection of human affective behavior in dyadic conversations

Abstract

Automatic detection of human affective behavior in dyadic conversations

by

Bhavan Jasani

Master of Science in Robotics

Carnegie Mellon University, Pittsburgh

Professor Jeffrey Cohn, Chair

Within the past decade, major strides have been made in automatic emotion detection. Most research has focused on frame-level detection of emotion or facial action descriptors (i.e. action units in Facial Action Coding System). More recently, attention has focused on prediction of session-level descriptors, such as depression severity, from automated analysis of emotion. This thesis addresses two challenges. One is the detection of emotion descriptors when unknown latency exists between the onset of an event and its time stamp. Latency of this type occurs when continuous manual annotation is performed without stopping and reviewing video to determine onsets and offsets with temporal precision. This problem has been addressed to a limited extent in the continuous annotation of valence and arousal but never before for coding multiple categorical descriptors (e.g., happy, angry, sad). The second challenge is the detection of session-level characteristics (e.g., gender) from video. Session-level descriptors provide a unique challenge for machine learning because the total amount of data per person is limited and at the same time each individual data (video) is relatively long (average of 20 mins in our data). This is challenging as temporal models such as Long short-term memory (LSTM) are poorly suited to long videos. To address these challenges we pursue both hand-crafted and-deep approaches.

List of topics: Human affective behaviour, dyadic conversation, behaviour science, computer vision, machine learning

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank many people who have been a great support during the course of my master's program at Carnegie Mellon University. First of all, I would like to start with my advisers Prof. Jeffrey Cohn and Dr. Laszlo Jeni for giving me the opportunity to work on a very interesting research topic and for their continued support. I would like to thank my parents and sister who have been supportive through out the entire duration of my master's program. I would like to thank Itir Onal Ertugrul, Torsten Wortwein and Prof. LP Morency for all the technical discussions related to the PANAM project. I would like to thank the other team members of PANAM project - Nick Allen, Lisa Sheeber, Nicki Silvering, Kenny Wenk, Leon Geon, Graham Neubig and Bhargavi Paranjape. I would like to thank Tejas Khot for all the academic and other discussions. I would like to thank Gunnar Atli Sigurdsson and Rohit Girdhar for the technical discussions. I would like to thank Naman Gupta, Joao Fonseca, Ting Che Lin and Rohan Reddy for being great roommates and all the support.

Chapter 1

Introduction

1.1 Human affective behaviour

Emotions are a way people express themselves and reveal their mental state. Emotions are multi-modal signals that may involve body posture, facial expression, voice tone, and speech. The term affect refers to the subjective experience of emotion. This thesis is primarily concerned with the behavior or signaling components of emotion, which is referred to variously as emotion or affective behavior. The last decade has witnessed major advances in automatic detection of emotion. Automatically recognizing emotion has many applications including but not limited to social robots, mental health, advertisement and education.

There are two major approaches to quantifying affective behaviour. In dimensional models, value is chosen over a continuous scale such as valence or arousal [47]. Valence expresses how positive or negative an event appears to be while arousal quantifies how exciting or soothing an event appears to be. Valence and arousal are inferred from affective behavior.

While dimensional approaches emphasize similarities among emotions (such as variation in valence), a discrete approach emphasizes their differences. In an influential discrete approach, Ekman et al. [13] defined six "basic emotions" that differed from each other on multiple criteria (e.g., emotion-specific differences in physiology, presence in non-human primates, and universal recognition).

Discrete emotions may be defined by specific signs (e.g., brow furrowing) or by inferences about or judgments of what a person is feeling. In a sign-based approach, descriptive actions

are mapped to specific emotions. For instance, the Facial Action Coding System (FACS) [12] is a system to code anatomically-based facial muscle movements, which are referred to as action units. Combinations of action units can then be mapped to discrete emotions based on context or prior research. By contrast, a judgment-based approach is informed by inferences from behavior about subjective experience [6].

This thesis is concerned with a judgment based approach to discrete emotions in which emotions are inferred from behavior. Because judgments are inherently subjective (unlike a sign-based approach that defines emotions in terms of specific action units), they are more difficult with which to achieve inter-observer reliability. For this reason the reliability of any single judge may be low. Judgment based approaches often aggregate ratings from multiple judges or coders [45]. By aggregating many ratings or coding across coders, high effective reliability may be achieved even when reliability between any two coders is low. An alternative for judgment based approaches is to rely on extensive training prior to annotating video. In our data, the latter was the approach taken. Each video was coded by a single trained coder; and a subset of the video was coded by two trained coders in order to assess reliability. Nevertheless, error occurred. This thesis explores the type of error, its consequences, and ways to ameliorate it for classifier training.

1.2 Contribution of this thesis

Our emotions are heavily dependent on our surrounding environment including our fellow human beings. Hence there has been a significant focus of affective behaviour in social settings. In this thesis, we focus on detecting affective behaviour in dyadic conversations between parents and their adolescent children at two levels of analysis: the event level and the session level. Event level refers to the duration of discrete emotions, which may last only a few seconds each. A conversation of 20 minutes or so, as in our data, may include hundreds of discrete emotions. Session level refers to characteristics of the person or to summary measures of the entire conversation. Gender and depression severity are session level characteristics that are invariant over a 20-minute conversation.

This thesis makes two contributions:

1) Automatic detection of event level affective behavior in the context of error in ground

truth. We focus on three emotions: Aversive, dysphoric, and positive emotion. Aversive and dysphoric are negative emotions that differ in whether negative affect is directed toward the other person (aversive) or the self (dysphoric). Positive includes a wide range of positive affect. Each emotion was annotated using a judgment based system (described later) appropriate for dyadic (mother and adolescent child) and triadic (mother, father, and adolescent child) conversations.
2) Automatic detection of session level descriptors of gender and depression severity. Gender and depression severity are characteristics that remain constant over the course of a dyadic interaction.

For both event level and session level prediction, features were obtained from automated measures of facial action units, facial landmarks, head pose, and in some cases voice quality. These were obtained using custom software as described below.

Event and session level prediction present unique challenges that this thesis addresses. Video was annotated in real time using a judgment based system that resulted in at least two sources of error: 1) Latency error and 2) Individual difference between annotators.

Latency error occurs when there is a lag between when an event begins or ends and the time at which it is denoted by a time stamp. Latency error is inescapable when coding is done in real time without stopping and starting of the video to identify precise onset and offset time [33]. This is because perception that an event has occurred and the motor response to tag it take time during which the video continues to advance. Thus, the time stamp for an event necessarily lags its actual onset. The second source of error results from differences between coders in the definition of an event, in the time necessary for its perception, and in the time necessary to register it. Coders may disagree on the label of an event or even whether an event has occurred. They may differ on its start time and its duration. Most supervised machine learning algorithms require correctly annotated data and assume that error in the ground truth is minimal. We evaluate that assumption in real-time coded discrete emotions and consider methods to increase classifier robustness to latency and individual difference error in ground truth.

In stop frame coding (non-real time) like FACS [12] this latency is minimized by asking coders to go back and correct the temporal precision of the annotations which in turn leads to higher inter-rater reliability. Prior work has dealt with the issue of latency for the case

of continuous annotations like valence. Prior work also has focused on using annotations from multiple coders to increase reliability and precision of measurement. In this thesis, we try to address the problem for the case of annotations that consists of multiple categorical descriptors (e.g., positive, aversive, and dysphoric) and for the case when we only have annotations from a single coder. We show the importance of accounting for the latency and individual differences in the annotations and show how much it can improve the classification results.

For the case of session level prediction, latency is not a concern but another factor is. The total number of data samples (number of subjects) is less, while each individual data sample is temporally very long (average 20 minutes). This is challenging as temporal machine learning models like Long short-term memory (LSTM) are not well suited for long videos and need more data for training.

We carry out our study on the Adolescent Study Dataset (ADS) [49,55,56] which consists of dyadic interaction tasks between children and parents. Video was annotated using the Living in Family Environment Coding (LIFE) System [23, 24]. LIFE is a judgement based system specifically created to characterise human affective behaviour for dyadic and triadic tasks. It's a novel way of categorizing human affective behaviour based on multi-modal human signals that combine verbal and nonverbal modalities. We focus on nonverbal affective behavior coded with LIFE.

The ADS dataset was manually coded with Living in Family Environment Coding (LIFE) System. Manual coding of these is time consuming process and requires experts who have been specifically trained for the purpose. We aim to build an automated system using computer vision and machine learning to detect the affective behaviour on event level (which are the LIFE codes) and session level (e.g., the gender of the children). Building algorithms that can automatically detect affective behaviour would result in high-impact use for research and clinical practice.

Human affective behaviour can be detected using supervised and unsupervised machine learning approaches. We have worked on both the approaches but in this thesis, we limit ourselves to supervised approaches. Figure 5.1 in the Appendix shows the overview of my entire master's research work.

1.3 Adolescent Development Study Dataset (ADS)

Figure 1.1: Example from ADS dataset. Left figure is for PSI task (showing anger) and right figure is from EPI task (showing happiness).

The Adolescent Development Study (ADS) [49, 55, 56] dataset was funded by the Australian Research Council and consists of audio-video recordings of dyadic conversations between adolescents and their parents, with a total of 202 parent and adolescent child dyads. It consists of two tasks which are recorded separately for each dyad - event planning (EPI) and problem solving (PSI) tasks. The dataset has been created for studying the onset of major depressive disorder in adolescents It has separate recordings from two cameras each facing one of the subject's face in the dyadic conversations. The two tasks have been shown to be powerful elicitors of emotion [49,55,56]. Event planning task evokes positive emotions while the problem solving task evokes negative as well as positive emotions. In the event planning task, the subjects plan a festive enjoyable activity like going for a trip, whereas in the problem solving task, the subjects attempt to resolve a mutual conflict.

Adolescents' age ranges from 11 to 14 years, with 50% male and 50% females. The average duration of each task and hence of each video recording is 20 minutes.

1.4 Living In Family Environment Coding (LIFE) System

The LIFE coding system [23, 24] is specifically designed to study dyadic and triadic conversations between parents and adolescents based on multiple modalities - visual clues, vocal clues and spoken words. It consists of specific verbal codes (called as content codes) and non-verbal codes (called as affect codes). The annotators watch the dyadic/triadic videos (average length 20 mins) of parent-child together and code for the onset of new affect or verbal behaviour for either of the subjects in real time, without pausing the videos. Annotators look for specific changes based on the face, voice and body posture of the subjects.

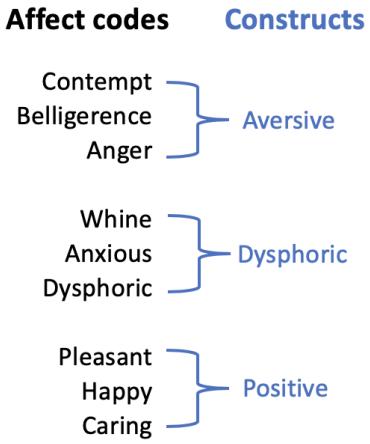

Figure 1.2: LIFE constructs: Mapping the affect codes (low-level) to constructs (high-level)

Based on the low-level content and affect codes, three high-level constructs are defined - dysphoric, aversive, positive. The dysphoric construct consists of anxious, dysphoric and whine affect codes. Aversive construct consists of contempt, belligerence and anger affect codes. While the positive construct consists of happy, caring and pleasant affect codes. Figure 1.2 shows the mapping between the affect codes and the LIFE constructs. An important thing to mention is that the LIFE constructs like the individual affect codes are different emotions. Dysphoric construct is an emotional state of unease or general dissatisfaction with life, similar to a depressed state. Aversive construct is an emotional state when one has a strong dislike or disinclination towards something. Positive construct is equivalent to the

happy emotion. The reliability of the constructs is higher than that of the individual affect codes; we are interested in predicting the three higher-level constructs. Particular sequences of occurrence of the LIFE constructs between parents and adolescents are [49, 55, 56] to be predictive of depression in adolescents.

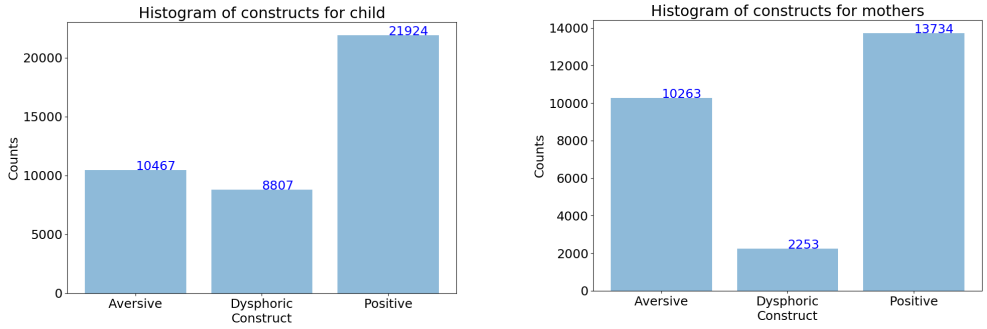

Figure 1.3: Distribution of LIFE constructs for mothers and children

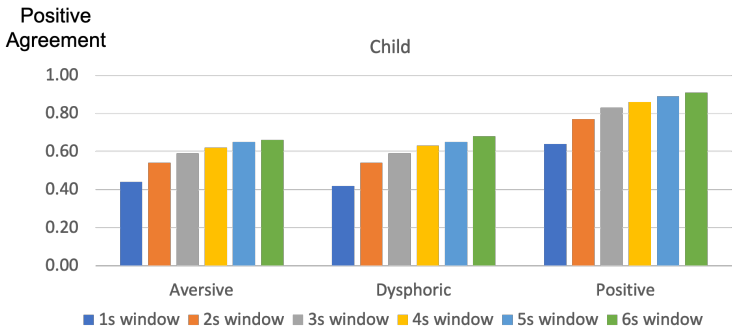

Figure 1.4: Inter-rater reliability of annotations for children

Figure 1.3 shows the distribution of LIFE constructs for mothers and children. For a subset of data (78 videos which is about 10.6% of the total videos), we have annotations from 2 sets of coders. We use this subset of data to compute the inter-coder reliability of the annotations for different window sizes. Given a code from one coder, we want to detect if one can detect that code within a window of certain duration for the second coder. Figures 1.4 and 1.5 shows the inter-coder reliability for different window sizes for the three constructs. As can be seen the reliability increases sharply up to 4 seconds after which the increment is low, indicating that the annotations are reliable for a 4 second window.

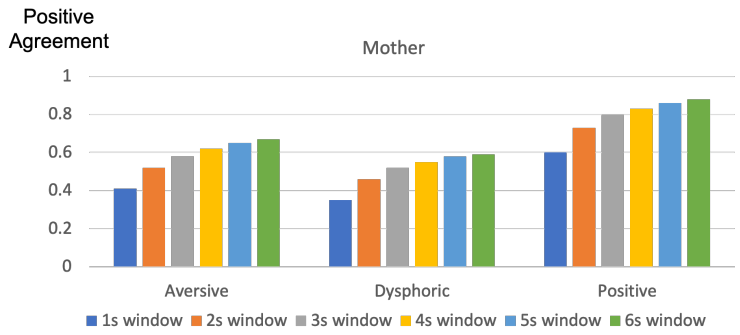

Figure 1.5: Inter-rater reliability of annotations for mothers

1.5 Thesis organization

In chapter 2 we describe our work on detecting human affective behaviour on event level. We describe two different supervised machine learning approaches to detect LIFE constructs. The first one is based on handcrafted features inspired from prior research in behavioural sciences, resulting into an interpretable model. While for the second approach we dont use pre-computed features rather we use data-driven approach to learn the features from raw visual data.

Based on the performance of our classifiers in both the approaches, we motivate about the presence of annotator latency and propose an approach which gives us a better estimate of the labels by trying to account for the latency.

In chapter 3 we describe our work on session level, of learning a mapping between automated measures of behaviour and gender of adolescents. We look at the entire dyadic conversation and learn a mapping to the child's gender. Finally, we conclude the thesis with a discussion in chapter 4.

Chapter 2

Event level behaviour prediction

In order to predict the LIFE constructs, we explore two different approaches. The first approach is based on hand-crafted features inspired from prior research in behavioural sciences, resulting in an interpretable model. The second approach is based on learning distributions. The main emphasis here is of using a data-driven model using deep learning to learn the features directly from the visual data. For this second approach we first briefly motivate it with 2 algorithms which learn distributions but use handcrafted features - Support distribution [43, 52] and time series kernels [8, 30].

2.1 Approach 1 - Hand crafted features

Below we describe our model inspired from prior research in behaviour sciences which predicts the three LIFE constructs from the videos. From the videos we compute the following visual features: 1) Facial landmarks 2) Head orientation 3) Facial Action units [12]. Additionally, we also compute audio features using OpenSMILE [15, 16]. These features are computed per frame and so we use their summary statistics to get a fixed dimensional feature representation which is fed to an SVM based classifier. We describe each of the parts in detail below.

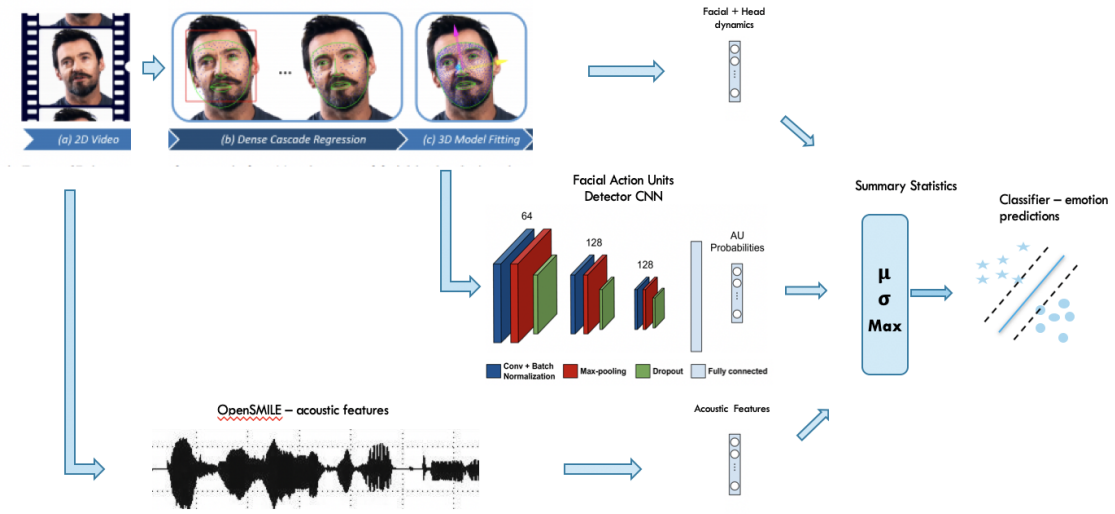

Figure 2.1: Approach 1: Based on prior research in behaviour sciences. Parts of figure taken from [14, 26, 27]

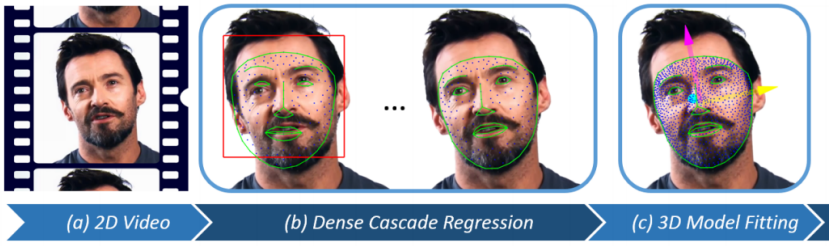

Figure 2.2: Zface: Extracting 3D facial landmarks and head pose from videos. Image from [26, 27]

2.1.1 Facial landmarks and Head Orientation

We use Zface [26, 27], a 3D face tracking tool which detects and tracks 49 3D facial landmarks from 2D videos. It also provides 3 degrees of rigid head movements (yaw, pitch and roll). Zface takes as input a single 2D image and locates the face using Viola-Jones [53] face detector. The bounding box of the detected face provides an initial configuration for the 49 facial landmarks. Local binary features [40] are extracted around these initial landmarks and a sequence of linear regressor matrices are used to update their positions to get a better estimate of the landmarks. Next, an iterative method is used to register a denser 3D model on the 2D landmarks. 3D shape and the 3D pose are iteratively refined until convergence.

This results in 49 3D facial landmarks and 3D pose in every frame.

Since the 3D facial landmarks are highly correlated, instead of using the raw values directly we apply PCA and use the top 26 PCA components. This way for 49 landmarks each with (x,y,z) values we go from 49x3 raw values to top 26 PCA components. Previous work such as [21, 22] have also used a similar approach to reduce the dimensionality.

2.1.2 AU classifier

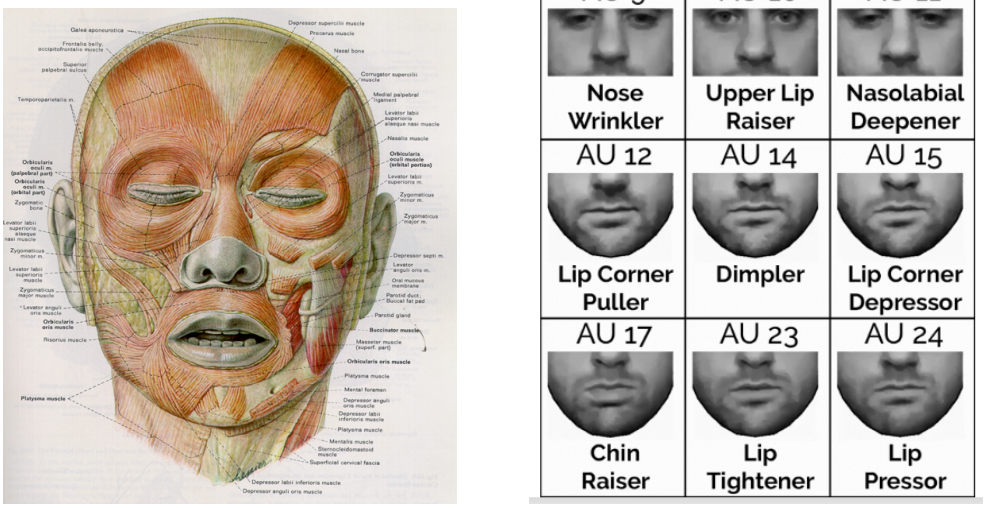

Figure 2.3: Different facial muscles (left figure) and different Facial Action Units (right figure). Images from [6]

Facial Action Coding System (FACS) [12] provides a system to taxonomize different human facial movements based on their appearance on face. FACS defines action units (AUs) which are a contraction or relaxation of one or more facial muscles. The combination of these action units can then be mapped to emotions and hence they are an important

indicator of the behaviour of the subjects. For example the presence of AU6 (cheek raised) and AU12 (lip corner puller) are associated with a smile and hence are indicative of happiness.

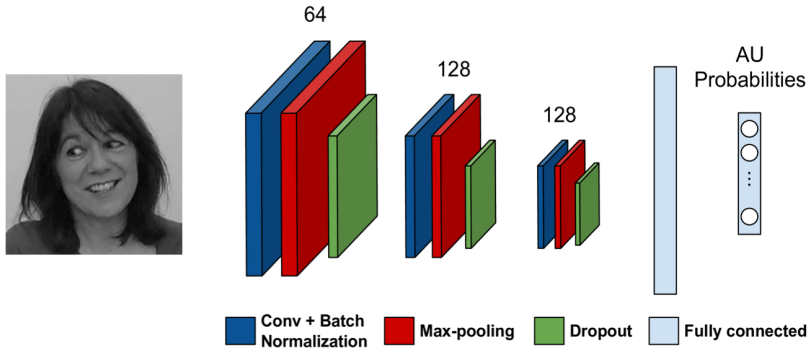

Figure 2.4: CNN based AU classifier which provides AU probabilities for 12 AU's. Part of figure from [14]

We use a convolutional neural network (CNN) based AU detector used in [14] that gives per frame independent occurrence probability values for 12 different AUs. Table 2.1 describes the 12 AUs and which are used in our experiments. The tracked face from Zface is first normalized by a similarity transformation between the detected facial landmarks and the landmarks of an average frontal looking face. The CNN takes as input a gray-scale image of this normalized face. The CNN consists of 3 convolutional layers followed by 2 fully connected layers. Each convolutional layer sequentially consists of a convolutional filter, batch normalization, max-pooling and finally dropout layer. Rectified linear unit (ReLU) is used as the non-linearity. The output from the last convolutional layer is connected to a fully connected layer of size 400. The output from this is connected to another fully connected layer having 12 neurons, each of which corresponds to the probability of 12 AUs shown in 2.1. Binary cross-entropy loss is used as the loss function to do multi-label AU classification. The ADS dataset doesn't have annotations for AU's hence we don't train our CNN on ADS, rather it's trained on data from 200 participants from Extended BP4D+ (EB+) dataset which is an extension of BP4D+ [58] dataset.

It's important to discuss the performance of the AU classifier as that impacts the performance of action units for the classifier. The average performance of the AU classifier across the 12 AU's, on EB+ for different evaluation matrix are as follow: 1) Free-margin kappa

Action Unit	Base rate	Free margin kappa	AUC	F1	Description
1	0.09	0.787	0.811	0.468	Inner Brow Raiser
2	0.07	0.856	0.816	0.437	Outer Brow Raiser
4	0.07	0.873	0.879	0.526	Brow Lowerer
6	0.43	0.685	0.925	0.821	Cheek Raiser
7	0.63	0.646	0.894	0.864	Lid Tightener
10	0.59	0.713	0.926	0.881	Upper Lip Raiser
12	0.53	0.736	0.945	0.876	Lip Corner Puller
14	0.42	0.566	0.853	0.749	Dimpler
15	0.10	0.776	0.808	0.408	Lip Corner Depressor
17	0.14	0.643	0.791	0.344	Chin Raiser
23	0.14	0.722	0.852	0.569	Lip Tightener
24	0.03	0.943	0.895	0.245	Lip Pressor
Average	**0.27**	**0.745**	**0.866**	**0.599**	

Table 2.1: Details of AU's predicted by our CNN classifier, results taken from [14]. These results are for 5 fold cross validation on EB+ dataset.

= 0.745, 2) Area under curve = 0.866, 3) F1 score = 0.599. For use of AU classifier in observational research in psychology, free margin kappa of around 0.7 is expected. For most of the AU's the individual free-margin kappa are within this acceptable range as can be seen in 2.1. Base rates of AU's affect their individual performance. Lesser base rate means less amount of positive examples for training the classifier, which in turn typically leads to lower performance as shown in [7,14]. For BP4D+ dataset, 7 out of 12 AU's occur in less than 15% of the total frames and hence there is class imbalance and hence performance of some AU's would be a lot better than others. Additionally, an important thing to mention is that since AU classifier is trained on a different dataset but used on ADS, there is a domain difference, which can lower its performance as analysed in [7,14]. They show that performance decreases when the model is trained on one dataset and evaluated on another dataset in comparison to training and testing on the same dataset i.e. for cross-domain experiments. When the model is trained on GFT [18] and evaluated on EB+ there is a decrease of 0.145 and 0.173 for AUC and F1 score respectively, in comparison to training and evaluating on EB+. While in the other direction the decrease is 0.053 and 0.018 for AUC and F1 score, which is lower than the previous case. Therefore, a model trained on GFT does not generalize well to EB+

dataset but the same model trained on EB+ generalizes well to GFT dataset.

2.1.3 Acoustic features

The voice tone of speakers are an important source of understanding their emotions. We use OpenSMILE [15, 16] to extract acoustic features. "SMILE" stands for "Speech and Music Interpretation by Large-space Extraction". It is open-source software for automatic extraction of features from audio signals and for classification of speech and music signals. It is a widely used tool in the affective computing research community for emotion recognition from audio. It is capable of recognizing the characteristics of the given speech rather than the spoken content. The OpenSMILE features can characterize the speakers age, emotion, gender, personality, and speaker states like depression and intoxication.

Researchers [28, 42] have used a large number of acoustic parameters which are indicative of emotions. This includes parameters in the time domain (ex:- speech rate), the frequency domain (ex:- fundamental frequency (F0) or formant frequencies), the amplitude domain (ex:- intensity or energy), and the spectral distribution domain (ex;- relative energy in different frequency bands). OpenSMILE provides various acoustic low-level descriptors which are frame-wise features. These are then mapped onto a vector of fixed dimensionality by computing various fuctionals over a certain sized sliding window (4 sec in our case). Examples of low-level descriptors include pitch, jitter, loudness while functionals include arithmetic mean, standard deviation and percentile. Specific details of these can be found in [15].

By applying different functionals to different low-level descriptors there are more than 6000 features possible, but not all are relevant. Using these large brute-forced feature sets have known to overfit on training data and reduced their generalisation capabilities to unseen test set [48]. Minimalistic parameter sets might reduce this danger and lead to better generalisation in cross-corpus experiments and ultimately in real-world test scenarios. Geneva Minimalistic Acoustic Parameter Set (GeMAPS) [17] is an interdisciplinary attempt to agree on a minimalistic parameter set based on multiple source, interdisciplinary evidence and theoretical significance.

GeMAPS [17] provides two versions of the acoustic features: 1) Minimalistic set (62 parameters from 18 low-level descriptors) which implements prosodic, excitation, vocal tract,

and spectral descriptors found to be most important in prior work 2) Extension to the minimalistic set (26 extra parameters from 7 low-level descriptors), which contains a small set of cepstral descriptors that are known to increase the accuracy of automatic affect recognition over a pure prosodic and spectral parameter set. We use the extended GeMAPS with a total of 88 features for our model.

2.1.4 Encoding dynamics

Zface [26, 27] provides per frame static head pose and location of facial landmarks. Dynamics of these are very important indicators of human emotions. To encode the dynamic information we compute the velocity and the acceleration of the head movements and facial features. Velocity is computed by taking the difference of current and previous frame of the raw features and acceleration by taking the difference between the current and previous velocity frames. [21, 22] use similar approach to encode the dynamics.

Unlike in the case of facial landmarks and head pose we don't use temporal dynamics (velocity and acceleration) of AU's in our model. The reason for this is our AU classifier was trained to predict the occurrence of AU's, i.e. their probability values and not the intensity values. If our AU classifier were trained to predict the intensity values, then we would have used the velocity and acceleration of the AU intensity values as additional features.

2.1.5 Summary statistics

Previously computed feature are frame level. Given a video segment of particular window size, we compute summary measures over the frame level features to get a fixed dimensional feature vector, irrespective of the number of frames in the video segment. We take the frame level features over all the frames in a given window and we compute their mean, standard deviation and the max values. We concatenate all these summary measures to get a fixed dimensional feature representation irrespective of the size of the window.

Feature	Summary measure	No. of dimensions
AU [1,2,4,6,7,10,12,14,15,17,23,24]	Mean, Max, Std	36
Head pose	Mean, Max, Std	9
Head pose velocity	Mean, Max, Std	9
Head pose acceleration	Mean, Max, Std	9
PCA of facial landmarks	Mean, Max, Std	78
PCA of facial landmarks velocity	Mean, Max, Std	78
PCA of facial landmarks acceleration	Mean, Max, Std	78
OpenSMILE	Mean	88

Table 2.2: Different features given as input to the classifier

2.1.6 Classifier

We use linear SVM as the classifier to predict the three constructs. The linear SVM takes as input the previously computed fixed dimensional features. We first concatenate all the visual and acoustic features before passing it through the classifier.

2.1.7 Results

Modality	Only Audio	Only Video	Audio + Video
Weighted Accuracy	46.92%	58.47%	62.09%
Kappa	0.198	0.419	0.450

Table 2.3: Results for different modalities - audio, video and audio + video

We use five fold family independent split with three folds for training, one for validation (to tune the C parameter of SVM) and one for testing. The folds are stratified so that each has an almost similar distribution of the three constructs.

We compute the confusion matrix and normalize it so that every row sums to one, by dividing every row by the total samples of that class. Weighted accuracy is computed as the average of diagonals of the normalized confusion matrix. Class imbalance or skew is a very important factor which can bias the evaluation metrics and hence should be be accounted for,

Modality	Audio vs Video	Video vs Audio+Video	Audio vs Audio+Video
Weighted Accuracy	t = 16.68 p = 3.7e-53	t = 6.26 p = 6.4e-10	t = 23.53 p = 1.5e-91
Kappa	t = 25.68 p = 4.3e-104	t = 6.30 p = 5.0e-10	t = 31.19 p = 2.1e-136

Table 2.4: Statistical analysis (T-test) for Table 2.3. The results indicate the performance of classifier is significantly different ($p << 0.05$) across all modalities

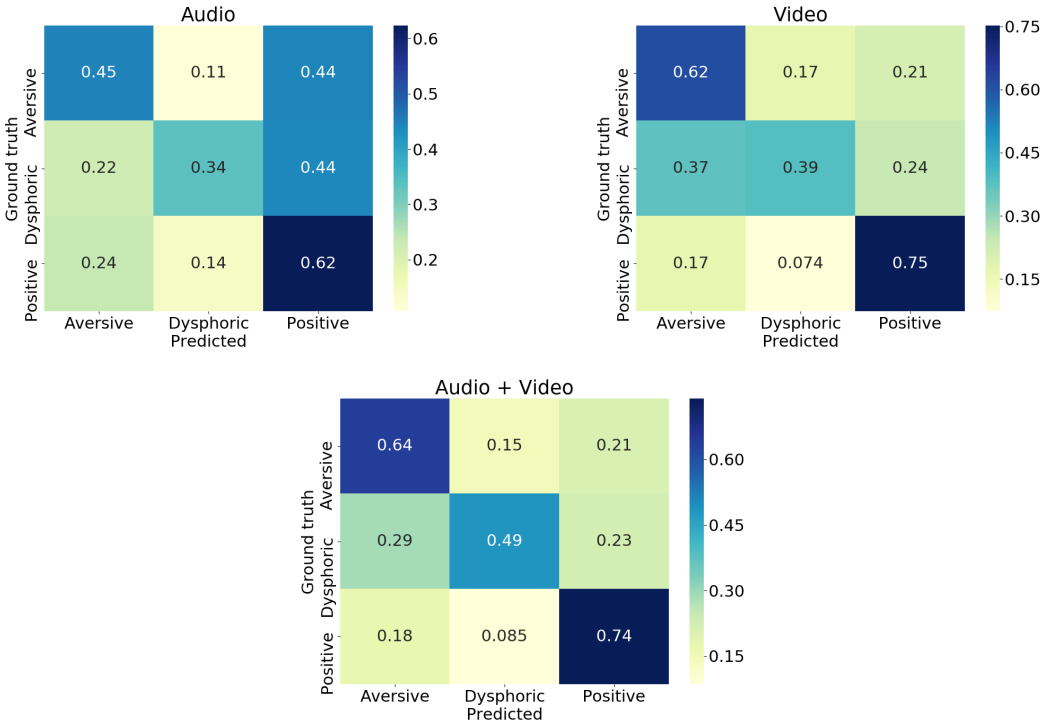

Figure 2.5: Normalized confusion matrix for different modalities - audio (top left), video (top right) and audio + video (bottom)

as shown in [25] with experiments on AU detection with different amount of class imbalance. Weighted accuracy tries to take into account class imbalance. This is because when the confusion matrix is normalized, the diagonal entries in the original confusion matrix are divided by the number of samples of that class. Figure 2.5 shows the normalized confusion matrix and table 2.3 compares performance for different modalities.

As can be seen, the performance of audio-only model is poor, and video only model does

better than it. And the best results are when one uses both the audio and video modalities. Positive construct is detected reasonably well from both audio and video modalities. Aversive is also detected reasonably with the video modality, although not as good as the positive construct. While dysphoric has the worst performance, but combining audio and video improves its detection performance in comparison to individual modalities.

2.2 Approach 2 - Learning distribution (deep learning)

In this section, we describe a different approach which is based on learning distributions. The aim here is to learn distributions of the features such that distance between pair of samples from the same constructs would be low while it would be large for samples from different constructs. Thus, in higher dimensional feature space there would be distinct clusters for every construct. We first briefly motivate this section with two techniques - support distribution machine and time series kernels, which utilize the previously used handcrafted features. Unfortunately, both of these resulted in chance level accuracy but they are an important motivation for the deep learning model, which is the main focus. The deep model avoids using handcrafted features rather directly learns from the raw visual data so as to have separate clusters for every construct in the higher dimensional space.

To clarify, the first two techniques even though use handcrafted features, differ from Approach 1 (which uses handcrafted features with a linear SVM). This is because, in Approach 1 we use fixed summary measures (average, standard deviation and max) to get a fixed dimensional feature vector. This has some limitations. These 2 techniques avoid using predefined summary measures, rather they learn what's the best way to summarize the data. Unfortunately, they lead to chance level performance.

2.2.1 Support Distribution Machine

The first approach is based on support distribution machines [43,52], which is an extension of SVMs to operate on sets (where each set can have different number of data points) as against a fixed dimensional feature vector. Each video segment with its corresponding per

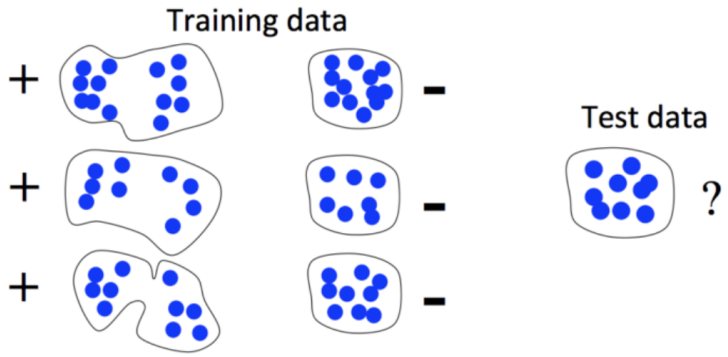

Figure 2.6: Learning distributions. Image taken from [43, 52]

frame handcrafted features (visual and audio) is a set, and the aim is to predict the label of this set which is the construct. Essentially two sets of the same construct would have similar distributions of the visual and audio features. This technique tries to learn that the samples of same class label would have similar distributions. Unfortunately, this method gave close to chance accuracy. In the previous model, we compute summary measures (mean, max and std) over the per frame features which results in a single fixed dimensional vector which is feed to a classifier.

In some cases, computing this pre-defined summary measures might not be enough for the classification task. SDM avoids using such a fixed number of pre-defined summary measures on the video segments (sets), rather it directly compares the amount by which the different sets (video segments here) overlap. The 2 sets from same LIFE construct should overlap more in comparison to sets from opposite LIFE constructs. It is done by treating the per-frame features as samples from an unknown probability distribution and then statistically estimating the distance between those distributions like KL divergence, L2 distance or another such distance metrics.

Figure 2.6 shows an example of SDM. The sets with positive labels have points spread up whereas the set with negative labels have points close by. The test set has points close by and hence it would be classified as a negative label. Unfortunately, this method gave close to chance accuracy.

2.2.2 Time series kernel

The second technique is based on time series kernels [8,30]. This is a kernel extension to SVM which works on time series data. It accounts for shortcomings of earlier method which neglects the temporal nature of data. We use the time series kernels to compute the Gram Matrix, which is then passed to an SVM. Unfortunately, this method again gave close to chance accuracy.

2.2.3 Siamese Network

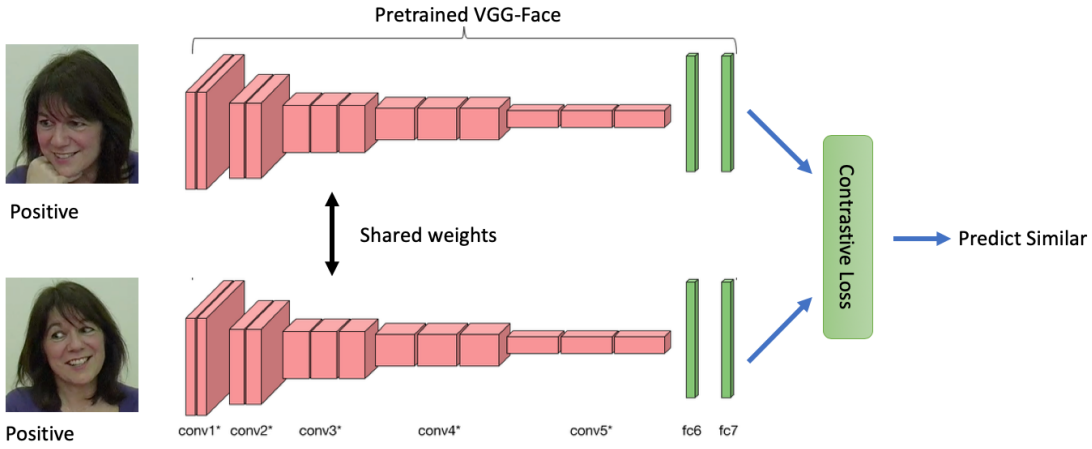

Figure 2.7: Siamese Network

The third technique is based on Siamese networks [5]. The previous two methods use hand-crafted features, on the other hand this approach is based on deep learning wherein the features are learnt directly from the data itself. This approach can avoid the shortcomings of hand-crafted features which may not be robust in conditions like huge head motions of the subjects. Siamese network is a special type of neural network architecture which, instead of learning to classify an input, learns to differentiate between two pairs of inputs. It learns the similarity between the inputs. Input pairs of the same class should be similar whereas input pairs of a different class should be very dissimilar in some high dimensional space. An important thing to mention is that our model based on Siamese network learns the deep

features only from video but not from audio. Hence this section only focuses on visual modality.

Siamese network [5] is a special type of neural network architecture which takes two inputs. Instead of a neural network learning to classify its inputs, Siamese network learns to differentiate between the two inputs. It learns the similarity between them. Architecture wise it consists of two identical neural networks (both have common weights), each taking one of the two inputs. The last layer from both the networks is then fed to contrastive loss, which calculates the similarity between the two inputs. Contrastive loss is used as the loss function. The aim here is not to classify the inputs, rather see how similar they are.

The contrastive loss is defined as below:

$$(1 - Y)1/2(D_w)^2 + Y(1/2)max(0, m - D_w)^2$$
$$D_w = \sqrt{G_W(X_1) - G_W(X_2))^2}$$

Here, X_1 and X_2 are the 2 inputs and $G_W(X)$ is the output from one of the networks. Y = 0 if the inputs are of the same class and Y = 1 if the inputs are of the opposite class. m is the margin value, typically a value of 0.2 is used.

If the two inputs are from the same class then we want the loss value to be as small as possible. While if the two inputs are from the opposite class and they differ by atleast 'm' then they are very dissimilar and so don't contribute to the loss (loss will be zero). Otherwise, they contribute to the loss and will be proportional to their difference from the margin m. So we are optimizing the network so that both the outputs from the sister networks have close to zero Euclidean distance if the inputs belong to the same class. While for inputs of different classes we want the outputs from both the sister networks to differ by atleast the margin m.

2.2.3.1 Network details

We use the VGG network [50] as the backbone for the Siamese network. For every video segment, we sample 10 frames uniformly and pass through the VGG network and then average pool the features from the last fully connected layer to get video level features. The VGG network has been pre-trained first for the task of face recognition task on the VGGFace dataset [41], which consists of 2.6 million images of 2600 different people. This provides the model with knowledge of how human faces look. The same network has then been fine-tuned on Facial Expression Recognition 2013 (FER-2013) [19] dataset for the task of recognizing the 7 emotion - anger, disgust, fear, happiness, sadness, surprise and neutral. This provides the model with knowledge of how human emotions look like. This model is based on the work in [2].

2.2.3.2 Sampling data and training

We divide the data into 70% for training and 30% for testing with the splits being family independent. As can be seen in Figure 1.3, the distribution of LIFE constructs, the three constructs are in different proportions, with positive being the majority construct. During training, we need to pass randomly picked pairs of videos and give it as input to the Siamese network. Since there are three different constructs it results in 6 different combinations: positive-positive, aversive-aversive, dysphoric-dysphoric, positive-aversive, positive-dysphoric, dysphoric-aversive. Since the constructs are unevenly distributed, during training we sample the pairs so that these 6 different combinations always happen to occur in the same proportion.

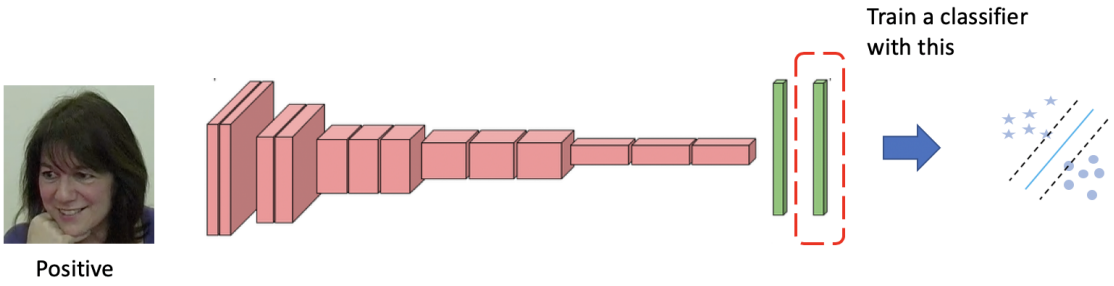

Figure 2.8: Extracting deep features from Siamese network

2.2.3.3 Testing

Once the Siamese network has been trained, the weights are kept fixed and we just pass the video segments (from both train and test splits) to one of the network to get the deep features. We then use deep features of the train set to train a linear SVM which acts as the classifier. After training, the linear SVM is tested on the deep features of the test set.

2.2.3.4 Visualization of deep features

To visualize the deep features from Siamese network we compute the Gram matrix (using L2 distance) and t-SNE for 400 randomly picked samples of each of the three constructs. For each of the 400x3 samples, we compute their L2 distance using all the possible pairs, resulting in the Gram matrix and visualize it using heatmap. We should ideally expect close to zero L2 distance when the two samples are the from same construct and large distance when they are from different constructs. For the Gram matrix, we stack all 400 samples of aversive followed by dysphoric followed by positive construct. Hence we should ideally see three square looking shapes of 400x400 along the diagonal.

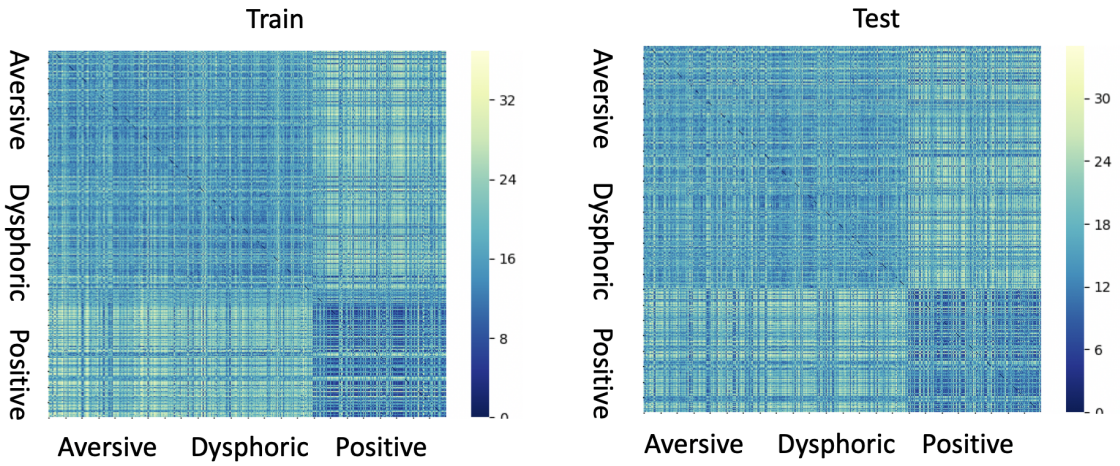

Figure 2.9: Gram matrix for 400 randomly picked samples from each of the three constructs. Left image is from samples picked from train set and right image for samples from test set

Another technique we use to visualize the high dimensional deep features are using t-SNE [31], abbreviated as T-distributed Stochastic Neighbor Embedding. It is a nonlinear

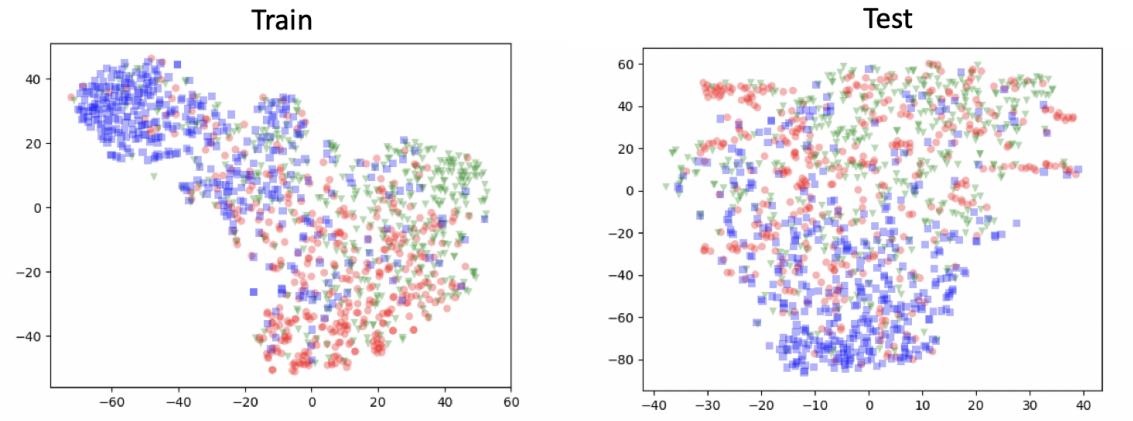

Figure 2.10: tSNE visualization for 400 randomly picked samples from each of the three constructs. Left image is from samples picked from train set and right image for samples from test set. Blue samples are from positive, red from aversive and green from dysphoric class

dimensionality reduction technique suited for embedding high-dimensional data to a low-dimensional space of two or three dimensions for visualization purpose. It models each high-dimensional object by a two- or three-dimensional point in such a way that similar objects are modelled by nearby points and dissimilar objects are modelled by distant points with high probability. Hence we should ideally expect three distant clusters corresponding to each of the constructs.

2.2.3.5 Results

Only visual features	SVM on deep features (from Siamese network)	SVM on hand crafted features
Weighted Accuracy	55.53%	54.88%
Kappa	0.3608	0.3556

Table 2.5: Comparing performance of deep and hand crafted features

We compare the performance of the Siamese network with handcrafted features (Approach 1) in table 2.5 and show the confusion matrices in 2.11. Note since the Siamese network only learns from visual modality we compare it's performance only with the visual handcrafted features. The performance of deep features and hand-crafted features when

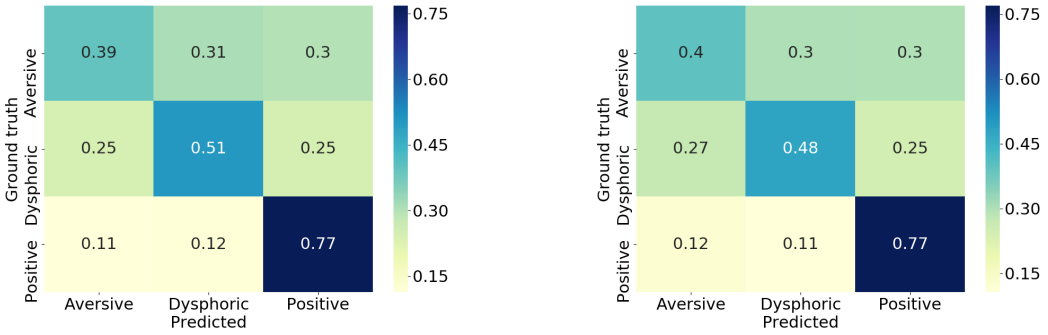

Figure 2.11: Normalized confusion matrix for SVM on deep (left) and hand crafted (right) features

using visual clues are very similar. The results show that both the models are doing well at detecting samples of positive constructs but are getting confused between aversive and dysphoric. The Gram matrix in figure 2.9 as well as the t-SNE visualization in figure 2.10 shows that samples from the positive class are very distinct while the samples from aversive and dysphoric look quite similar to each other. This explains not so good performance for aversive and dysphoric classes.

Sample type	Accuracy
Aversive vs Aversive	69.64%
Dysphoric vs Dysphoric	73.28%
Positive vs Positive	70.74%
Aversive vs Dysphoric	27.80%
Aversive vs Positive	53.82%
Dysphoric vs Positive	52.84%
Overall	60.19%

Table 2.6: Performance of Siamese network as a pairwise predictor

Since Siamese network is trained as a pairwise predictor, we also show in Table 2.6 the performance of Siamese network to classify if the input samples are from the same construct or not on the test set. The results show that Siamese network is doing fairly good (accuracy of around of 70%) if the pairs belong to the same construct irrespective of which specific construct they belong to. While if the pairs are from separate construct then it does decent

and has similar performance to identify Aversive vs Positive (accuracy of 53.82%) and Dysphoric vs Positive (accuracy of 52.84%), but very poor for Aversive vs Dysphoric (accuracy of 27.80%). This indicates Aversive and Dysphoric look similar but different from Positive.

Only visual features	End to end single CNN	SVM on features from end to end CNN
Weighted Accuracy	57.42%	54.14%
Kappa	0.4043	0.3478

Table 2.7: Comparing performance of a single end to end CNN and SVM on the deep features extracted from this CNN

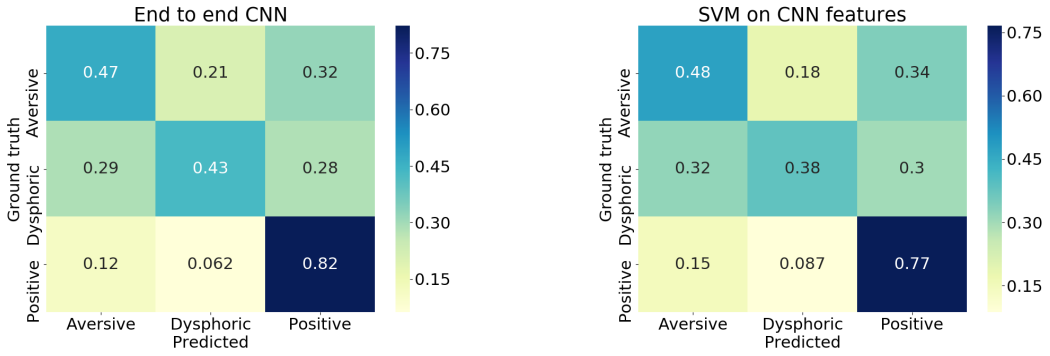

Figure 2.12: Normalized confusion matrix for a single end to end CNN which directly predicts the three constructs (left) and SVM on the deep features extracted from this end to end CNN (right)

As an ablation study we also check the performance of a single network CNN trained end to end which predicts the three constructs directly and the performance of a SVM on the deep features extracted from this CNN. The architecture of this CNN is exactly the same as the ones used in Siamese network. Table 2.7 shows these results and Figure 2.12 shows the corresponding confusion matrices. The results show that SVM trained with deep features from Siamese network has slightly better performance than the same SVM trained on deep features from a single CNN trained end to end. At the same time a single network CNN trained end to end performs slightly better than the SVM on deep features from Siamese network. These results show Siamese features are slightly better than features from a single network CNN while an end to end deep model performances better than both.

Deep learning has recently shown to learn better representations of the data and outperform traditional hand-crafted machine learning models in a variety of tasks in computer vision and natural language processing on images, videos, audio and text domains [29]. The reason for this is that deep models are more powerful and have better capabilities to learn from the data in comparison to hand-crafted models. In our case we don't see much improvement with a deep model in comparison to the model based on hand-crafted features, especially the samples of aversive and dysphoric don't seem to be distinguishable from each other even for the deep features. One probable reason for this lack of improvement in performance can be because there is something wrong in the annotations and the data because of which the deep model is not able to learn separable features, especially for the case of aversive and dysphoric. We discuss more about this in the next section.

2.3 Latency issue

As can be seen from the previous results, we get very similar performance for both our models and they are not great results specifically for the case of aversive and dysphoric constructs. This can be attributed to two factors:
1) Problem with the models
2) There is something in the data and the labels which we assumed

Given that we are getting similar results for both the models, it indicates that both the models are facing a common issue. Hence there is a strong possibility for the second factor. Since we are using supervised machine learning approaches, they rely on the correct annotation of the labelled data. This need not be the case always. In case of LIFE codes, we discovered this assumption to be wrong later on. We attribute this because of the two sources of errors: 1)latency introduced in the real-time annotation process and 2) the individual differences of the annotators. Both of these result in low inter-rater reliability of the annotations. We describe these two things next.

When coding human behaviour from video in real time, there is an inherent delay [20, 33, 38] between an action of the target person and its associated time stamp of an annotator or coder. Time is required for the coder to process the action, determine its occurrence, and

execute a motor response to tag the event. All this time the video is advancing. The more time required to perceive an event, the longer the resulting latency.

The second source of error is lack of agreement in the annotations for the same data amongst the coders because of individual differences (we also call these as variance in the annotations). This happens because human emotions are subjective. Given two different annotators they can have individual differences in the annotation of a given event due to 1) They might not agree on the label (category) of the event 2) They might differ on the start time of the event 3) They might differ on the duration of the event. Typically in behaviour sciences by training the annotators about the annotation process, the individual differences are minimized but this still can manifest as an error in the annotations.

Most supervised machine learning algorithms assume the existence of reliable labels corresponding to the event it is coded for. If unaccounted for the latency and individual differences, these labels are less reliable and can impact the performance of supervised learning algorithms. Typically for problems related to human behaviour, the annotations are done by multiple coders to account for individual differences of coders. These annotations are fused to get a better estimate of the annotations. The issue of latency has been investigated to a limited way in prior work for the case of dimensional coding, example coding of valence. A question arises what about when events are represented as multiple discrete codes and annotations are available only from a single coder? These are the challenges which we face with the LIFE codes and next we address these.

For a subset of data we have annotations from two pairs of the coders. To investigate variance (or individual differences) we first do windowed inter-coder reliability, to estimate the variance of the onsets by the two pair of coders. Given this information, for the full dataset, we next carry out experiments by training classifiers to determine windowed coder-classifier reliability. We carry these by varying the size of the window and by looking at symmetric (centred) and asymmetric windows (left and right sided), around the coded location of the onset. To investigate latency, we take a window of fixed size and temporally shift (by various amount) around the coded location of the event. We observe the effects of these experiments on the performance of our classifier. Figure 2.14 shows this overall approach.

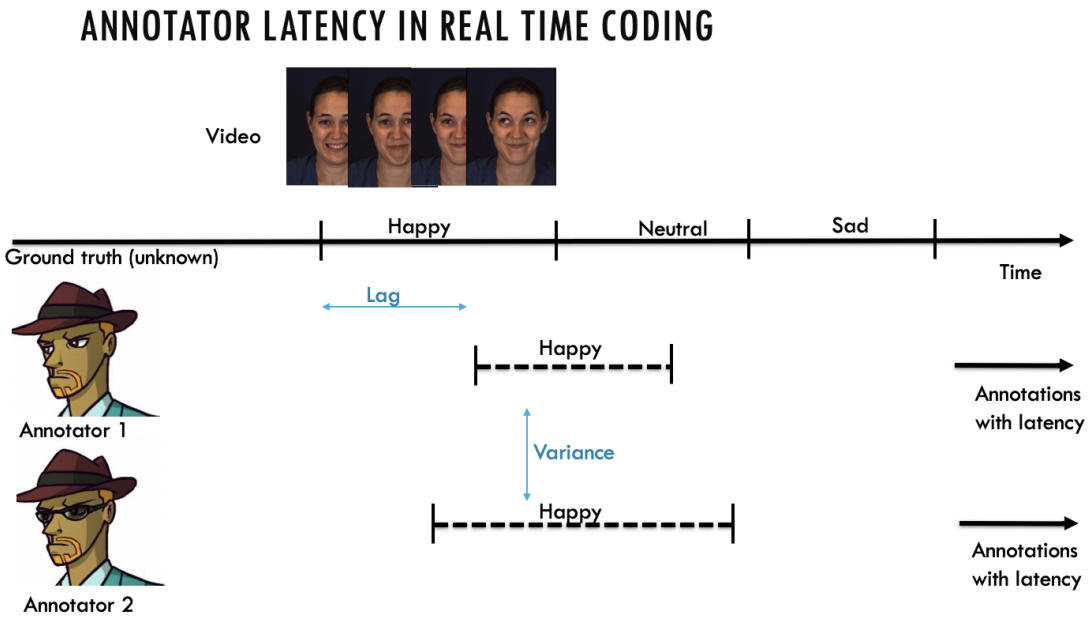

Figure 2.13: Annotator latency in real time coding, characterized by two things 1) Temporal lag 2) Individual differences. Images from BP4D [57] dataset

2.3.1 Related Work

Method	continuous/ discrete	real time/ stop frame	multi-class	no. of annotators
Marioryad et al [32, 33]	continuous	real time	single	2-8
Gupta et al [20]	continuous	real time	single	28
Nicolaou et al [38]	continuous	real time	single	-
FACS [12]	discrete	stop frame	multi	-
LIFE codes [23, 24]	discrete	real time	multi	1-2

Table 2.8: Taxonomy

Table 2.8 shows the different kinds of coding schemes. Prior work mainly focuses on coding which is done for a single dimensional variable in real time on a continuous scale. For example, only valence is coded at a time. Prior work also focuses on fusing annotations from multiple coders to get a more reliable estimate of the ground truth. The idea is individual coder would have their own latency's and noise but combining them would give a better estimate of the actual annotations. In [51] authors carry out annotations with naive/non-

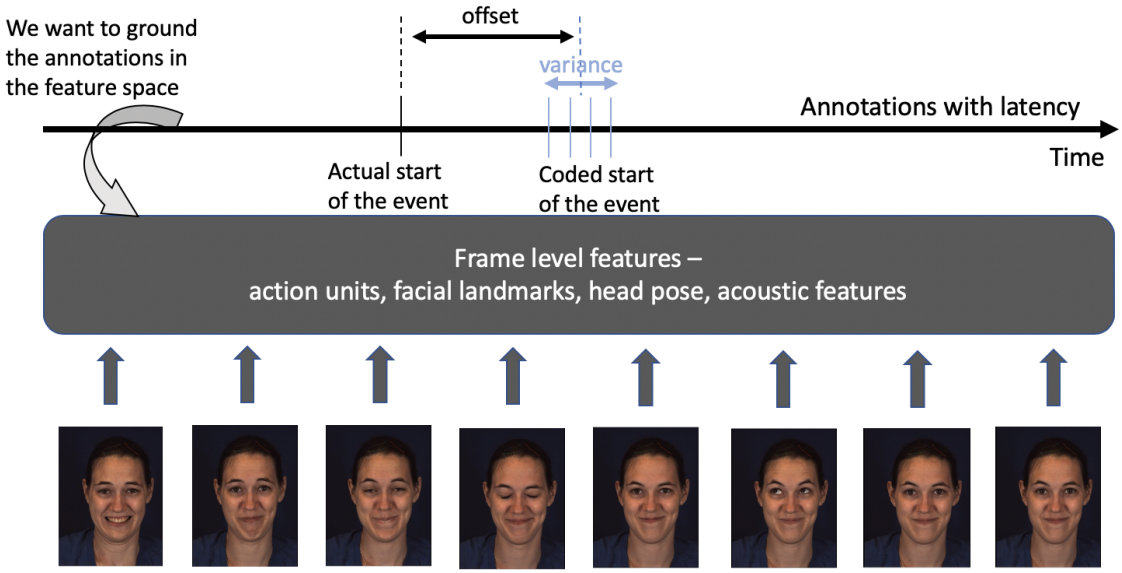

Figure 2.14: Overview of the approach. The latency is characterized by two factors: (1) Variance: There is variance in the coded location of the event. (2) Offset: There is time shift between the actual onset of the event (which is unknown) and the average annotated onsets. Images from BP4D [57] dataset

experts and compare their results with FACS trained expert coders for infant and parent emotions for early autism risk. They found that ratings averaged across 10 non-expert coders exhibited high concordance with expert facial-action codes for the case of infant emotions, and 20 non-experts were required for reliable parent ratings. They mention how intuitive non-expert ratings can be used as an alternative to complex and costly behavioral coding systems. Rosenthal [46] describes more broader issues in annotation of nonverbal behaviour in affective sciences.

There has been a lot of earlier work to find the latency in case of continuous annotations (valence and arousal). Marioryad et al [32, 33] focuses on using mutual information criteria to find the time shift that maximizes the mutual information between the single-dimensional continuous annotations (valence and arousal separately) and expressive behaviours characterised by facial action units [12] which are used as video features. They also carry experiments with different amount of delays in annotations and show that it improves the performance of the classifier. Too longer delay deteriorate the performance while too less

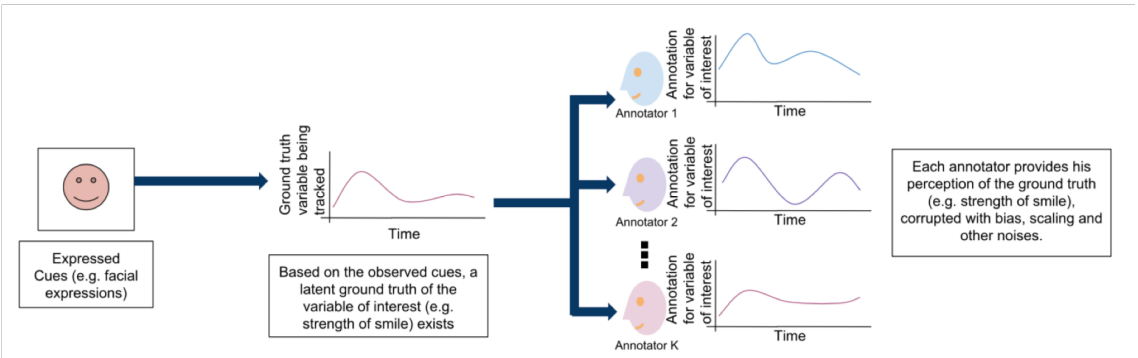

Figure 2.15: Fusing multiple annotations. Figure taken from [20]

delay don't show much performance improvement. Lag of around 2 sec was found to be optimal in their study on SEMAINE dataset [35]. Nicolle et al [39] did similar work on SEMAINE dataset. They assumed a linear relationship between the user's facial features and the annotated scores. They proposed a correlation-based measurement to find the delay by comparing the correlation of facial features with the delayed emotional annotations across different delay intervals. Further [32] followed a similar approach as [32, 33] but found that the optimal delay depends on the the device used for annotation. Specifically, they found in their study that when annotations are done using joystick in comparison to mouse (which was the case in [32, 33]), the optimal delay amount is less, around 250 ms.

Gupta et al [20] as shown in figure 2.15 focuses on modelling annotations from multiple coders (N = 28) as noisy distortions of the ground truth signal. They propose that simply taking the average of the multiple annotator's ratings doesn't provide an accurate representation of the actual phenomenon. They condition the ground truth on a set of features extracted from the raw data and assume that the annotators provide their ratings as a noisy modification of the ground truth, with each annotator having specific distortion. They train the model using an Expectation-Maximization algorithm [9]. Nicolaou et al [38] focuses on the similar problem of fusing multi annotations to account for the latency. They use dynamic probabilistic canonical time warping based approach which is build upon their earlier work [37].

Contrary to prior work, our's focuses on analyzing latency when there are multiple discrete annotations (like happy, angry, sad, anxious) from a single annotator. In contrary to

coding a single dimensional signal like valance, coding multiple discrete labels is very challenging because the coder has to decide between different behaviours (there can be sudden jump) and code them in real time. An important thing to mention is our work focuses when we have annotations from only a single coder, unlike all the previous works described. We aim to provide the correct way to read annotation from a single coder such that the latency factor is accounted for.

Another important thing to mention is the Facial Action Unit Coding System. When coding the occurrence of multiple action units, coders typically don't do it in real time. They pause and rewind videos to correct for temporal precision, due to this nature action unit coding doesn't have the latency issue. And the reliability of the annotations will be very high.

2.3.2 Our Approach

In this section, we describe different experiments we carried out with our classifier (based on approach 1 which uses handcrafted features) which predicts the LIFE constructs. The crucial part is the correct way to read the LIFE annotations such that it accounts for the coder latency and individual differences. Classifier trained with data which accounts for these should give better performance.

We start with a preliminary analysis to show the shift between the actual onset and the annotated onset. Then we carry out experiments to first demonstrate variance between two sets of coders on a subset of dataset followed by variance analysis between the codes and the classifier for different window sizes. After this, we carry experiments by shifting the window temporally to determine the offset.

2.3.2.1 Preliminary analysis

Figure 2.16 shows the actual start of the event which is unknown and the coded location of the event which happens after the event starts due to the latency. Based on this it's evident that the actual event started before the coded location of the event and continues afterwards. The peak behaviour of the event would most likely happen after the coded location. So one needs to look at both the sides of the coded location of the event to fully

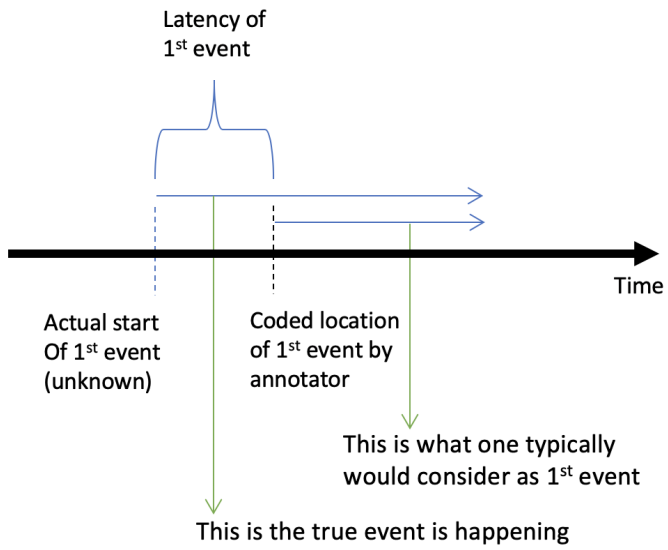

Figure 2.16: Actual start of the event and the coded location of the event. This shows the latency in the annotations

capture the actual event. Without this insight (of coder's latency) naturally one might have just started to look from the coded location of the event. This would result in missing the earlier portion of the event.

2.3.2.2 Variance: Inter-rater reliability

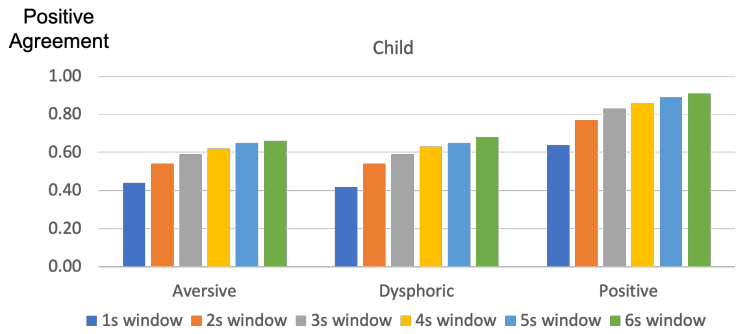

Figure 2.17: Inter-rater reliability of annotations for children

For a subset of data, we have annotations from 2 sets of coders. We use this subset of data to compute the inter-coder reliability of the annotations for different window sizes.

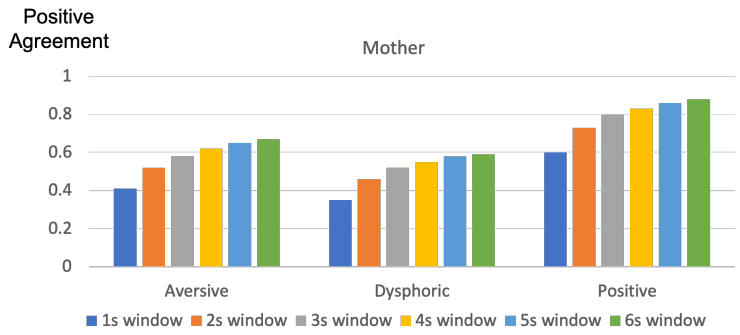

Figure 2.18: Inter-rater reliability of annotations for mothers

Given a code from one coder, we want to detect if one can detect that code within a window of certain duration for the second coder. Figures 2.17 and 2.18 also shown earlier in Chapter 1, shows the inter-coder reliability for different window sizes for the three constructs. As can be seen the reliability increases sharply up to 4 seconds after which the increment is low, indicating that the annotations are reliable for a 4 second window.

2.3.2.3 Variance: Window size analysis

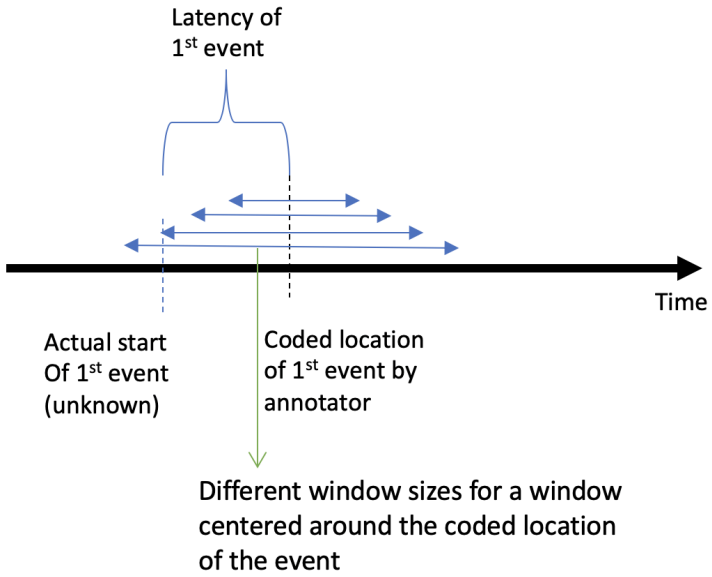

Figure 2.19: Experiments with different window sizes

As we proposed earlier it's important to look on both sides of the coded location event, but the natural question arises, what size of window one should look for? To answer this question we perform experiments with different window sizes around the coded location of the event. Since one does not know the actual duration of an event we globally estimate them by a fixed length window, whose duration we experimentally determine. The window size which provides use with the best performance of the classifier would globally approximate as the average duration of all the events.

Figure 2.22 and 2.23 shows the performance of the classifier for different window sizes around the coded location of the event. Looking at the columns, we can observe that the accuracy increases as we increase the window size till a certain window size and then it starts to decrease. A too small window might not capture all the information of the event and a too big window captures information of the neighbouring event. Different codes might have different average duration but we do not have such information and so we approximate everything with common window size.

2.3.2.4 Variance: Window asymmetry analysis

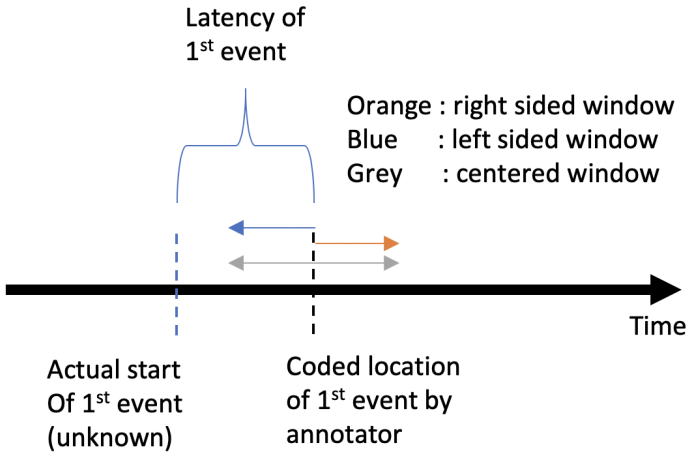

Figure 2.20: Experiments to compare performance of a left sided vs right sided vs centered window

Next, we experiment and compare the performance of 1) window just on the left side of the coded location 2) window just on the right side of the coded location 3) window centred

on the coded location.

Plots in figure 2.24 and 2.25 indicate that looking at the right side of the coded location always has better accuracy than the left side while the difference is not significant. This confirms our earlier claim that the event starts before the coded location but the peak behaviour happens after the coded location. Interestingly for small window sizes, the right-sided window has better accuracy than the centered window which in turn is better than left sided window. But for larger window sizes the centred window gives the best accuracy. This behaviour again indicates that for a small window the right side of the coded location contains the most information in comparison to the left side as the peak behaviour happens on the right side. But the overall highest accuracy comes from a centred window of length 6 sec (i.e +- 3 sec), indicating it's necessary to capture information from both sides.

2.3.2.5 Offset: Window shift analysis

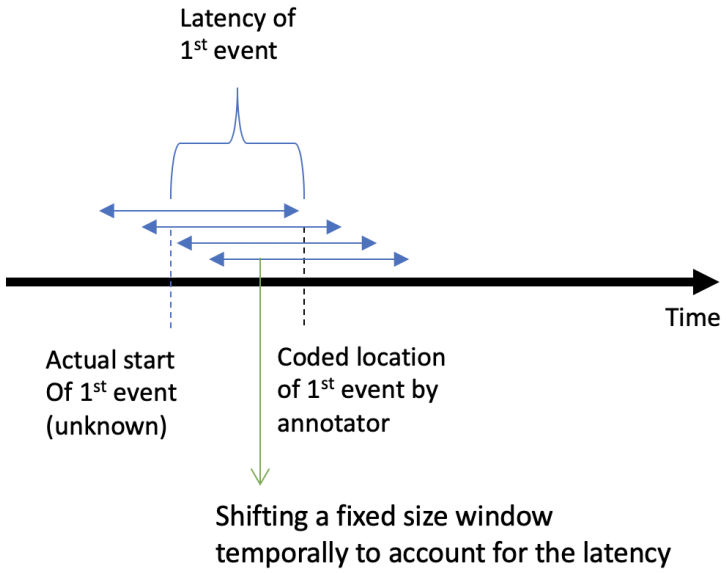

Figure 2.21: Experiments with different amount of temporal shift in the windows

Since there is this inherent latency, another thing we carry out is to shift a fixed size window temporally by various different amounts to account for the latency. One might expect that a non-zero shift value equal to actual latency would result in classifier giving

the best result. Rows in figure 2.22 and 2.23 shows the performance of the classifier for various window sizes shifted by different amounts. Contrary to expectation, the no shift window always performs the best. This indicates that looking at both the sides for a window around the coded location of the onset is sufficient to incorporate the latency. Also for small shifts, there's not much degradation in accuracy, but for larger shifts, the results go down significantly. This indicates that the peak behaviour happens around the coded location of behaviour.

2.3.2.6 Conclusion

We systematically carry out experiments and provide a methodology to correctly read categorical annotations which are coded in real time without pausing for a single annotator. There are two sources of error in the annotation - latency and individual differences. Based on the performance of our classifier for these various combinations we conclude that a fixed window (the size is experimentally determined) around the annotation of the event without any temporal shift provides the best performance for our classifier. We initially had the hypothesis that a left-sided window should have better performance than a right-sided window, as that would include the latency, but we observed the other way. The best performance came from a centered window. The correct way to account individual differences would be through a centered window because the variation of the event can be on either side of the coded location. While the correct way to account for latency would be to look at left-sided window or shift the centered window temporally to left. The results indicate that individual difference is a more dominant source of error than latency and centered window accounts for both latency as well as the individual difference. The inter-rater reliability analysis shown in figures 2.17 and 2.18 also indicate a need for a large centered window (greater than 4 seconds) for aversive and dysphoric constructs to have good reliability and account for individual differences. Another conclusion can be that there is latency but the event just starts (person starts to smile) before the coded location of the onset but it truly develops (full-fledged smile) right after the onset. Hence it's important to look at both sides of the coded onset. An important point to mention is we could have done pair wise statistical significance test (T-test) for windows of different size and shift. But we are interested in pattern of the

findings rather then pairwise differences and hence we don't pairwise T-test.

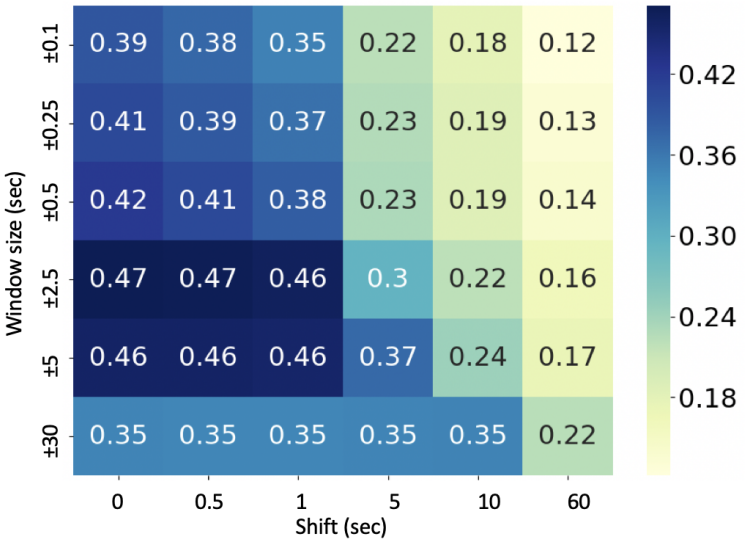

Figure 2.22: Comparing the performance (Kappa) of our classifier for different values of shift vs window sizes.

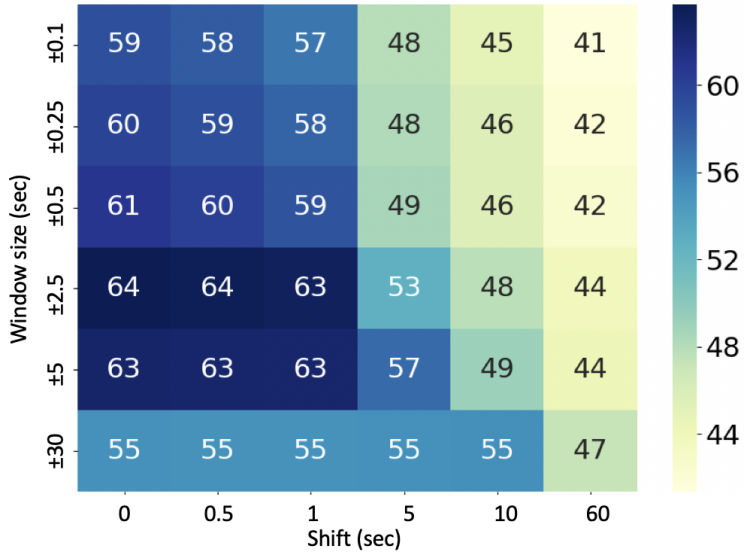

Figure 2.23: Comparing the performance (Weighted accuracy) of our classifier for different values of shift vs window sizes.

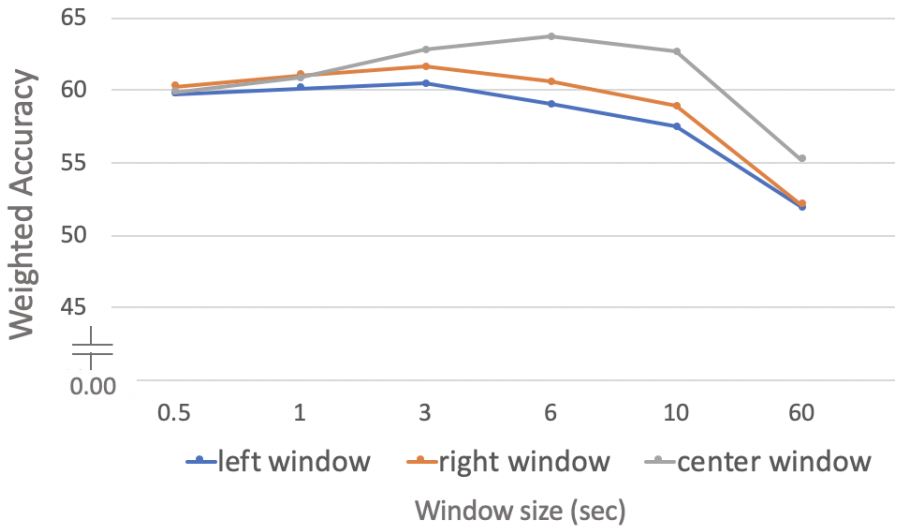

Figure 2.24: Comparing the performance (Weighted Accuracy) of our classifier for left vs right vs centered window.

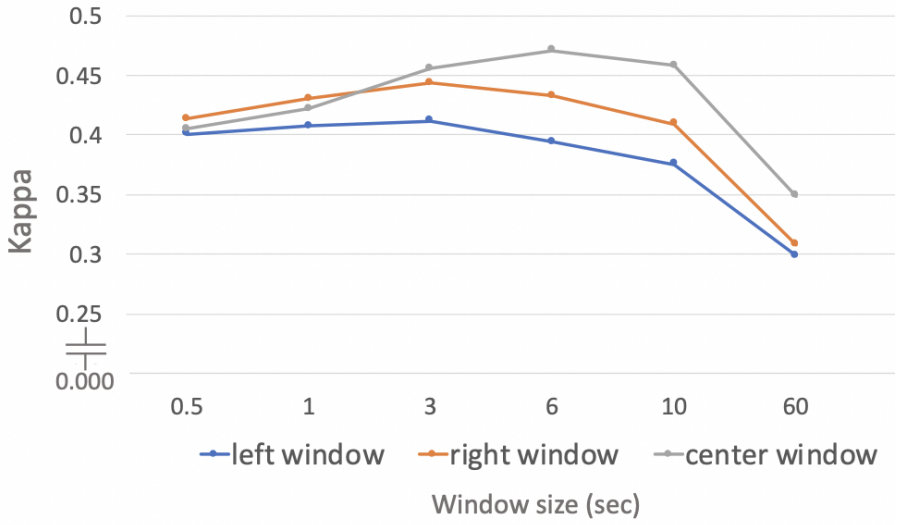

Figure 2.25: Comparing the performance (Kappa) of our classifier for left vs right vs centered window.

2.3.2.7 Limitations

Our analysis provides important insight on how to correctly read the annotations in a way which tries to account for the inherent latency introduced in real-time coding of human behaviour. Our proposed approach to account for latency has certain limitations. First, it assumes constant lag for all the constructs. This need not be true as some constructs can be identified much sooner than others and hence there can be construct specific latency. Second different coders may have different efficiency and hence there can be coder specific latency. This can also depend on external factors like the time of day and if they drank coffee before starting annotating. Third, even for the same coder, they might be more efficient at the beginning of the video than at the end when they might be tired. Hence even for the same coder and same construct, latency can vary across the video. Nonetheless, our analysis shows that it's important to take into account the annotator latency when training machine learning models as their performance can benefit from it.

Another limitation is we consider all the frames in a given window to be displaying the associated construct but this might not be the case. Frames at the start or at the end of the window segment might be from an adjacent construct. Multiple instance learning [10] is one way to address this limitation. In multiple instance learning the window (which is the bag) is labelled with the associated construct but not all the frames (which are called instances) in that window would necessarily belong to the construct. So some frames can be from another construct. Instead of receiving a set of instances which are individually labelled, the learner hence receives a set of labelled bags, each containing many instances. From a collection of labelled bags (labels are the constructs in our case), the learner tries to either (i) induce a concept that will label individual instances (i.e. the frames) correctly or (ii) learn how to label bags (i.e. the windows) without inducing the concept.

2.3.2.8 Comparison

	±3 Second window (accounting for latency)	Original segment (neglecting latency)	t-value	p-value
Weighted Accuracy	63.70%	62.09%	1.7647	0.0778
Kappa	0.4708	0.4509	2.429	0.0152

Table 2.9: Comparison between model with and without accounting for latency.p-values indicate significant difference for the case of kappa values

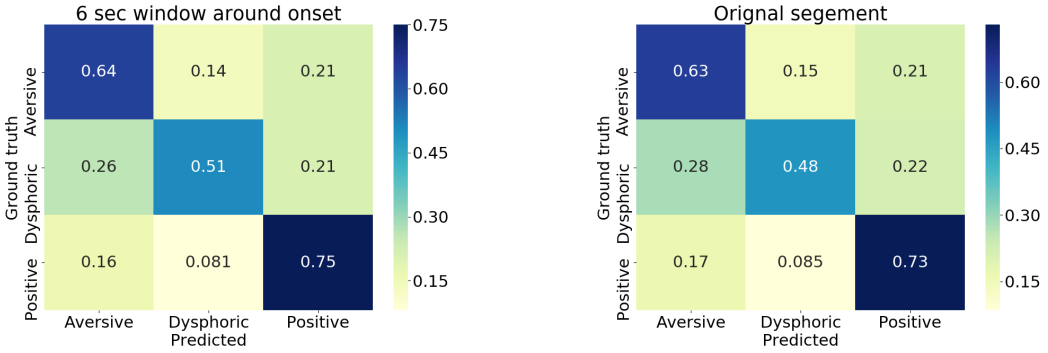

Figure 2.26: Comparison between model with and without accounting for latency

Chapter 3

Session level behaviour prediction

In this chapter, we explore the gender differences in behaviours of boys and girls towards their mothers in dyadic conversations in Adolescent Development Study (ADS). Rather than looking at events which last for a couple of seconds (to predict LIFE constructs) as we did earlier, here we look at holistic behaviour of children towards their mothers for the entire duration of the dyadic task (average 20 minutes). We use automated measures to calculate action units, headpose and facial landmarks using our model (approach 1) described earlier which is based on prior research in behavioural sciences.

What we are interested is to explore the question that whether the gender difference lies only on the appearance of a person or is there something deep down in our behaviour which has evolved our time. As we become older we tend to adjust ourselves with the norms of society. On the other hand, children tend to be less influenced by the norms imposed by the society. Hence wouldn't it be very interesting to study gender based behaviour difference in children? And infact talking to one's mother for a child is a very common day to day thing and so wouldn't it be even more interesting to explore in this kind of a very common social context if there are behaviour based differences in boys and girls? If there are any behaviour differences then they wouldn't be rare situation based differences, rather they would be something which is naturally occurring and has evolved over time. That's exactly what we are trying to explore in this chapter. We hope this work can open new research directions and shade light on behaviour differences in children.

Most of the previous research on identifying gender or analyzing gender differences like

[3, 34, 54] have focused on non-social context wherein a person is told to perform some task like watching TV advertisements [34] which is then used for gender identification or analyzing differences. These are typically short recordings of about a minute or so and in lot of work which focus on the appearance of the person [36, 44], are based on stationary images of people. And further, these works usually focus on adults. Our work differs in comparison to these earlier ways in multiple ways and which makes it more interesting. First, we bring in a very naturally occurring social context of children's talking with their mothers having a general day to day conversation without much restrictions. Secondly, we analyze interactions of longer duration, about 20 minutes long and hence are very informative for studying holistic behaviour based differences. And we don't focus on appearance based differences. Third, we study behaviour differences in children as opposed to adults, children tend to be less aware of the social norms imposed by our society and hence it can reveal more fundamental behaviour differences. Additionally, we also analyze the behaviour differences of a mother towards their children based on the gender of the child. It's also important to mention we focus on analyzing the behaviour differences of children towards their mother and not focus on identifying the gender of children.

We train separate classifiers to differentiate the behaviour for the positive and negative tasks. We examine predicting the child's gender based behaviour differences by looking at the child's videos and also looking at mother's videos.

3.1 Related Work

In [54] authors investigate expression based gender recognition in 3-D faces. They demonstrate that facial expressions can influence the gender patterns in 3D face images, and gender recognition system can have better performance when trained expression specific. They show that gender can be recognized with considerable accuracy in happy and disgust expressions, while sad and surprise expressions do not convey much gender-related information.

In [3] authors proposed an approach for gender estimation, based on facial behavior in video-sequences capturing smiling subjects. Their behavioral approach quantifies gender dimorphism of facial smiling behavior and is complementary when faces are occluded. [34] carried out a large-scale study that examines whether women are consistently more expressive

than men or whether the effects are dependent on the emotion expressed. They studied 2,000 viewers who watched a set of video advertisements in their home environment, which were recorded using webcams and then used an automated Action Unit [11] classifier. They found that generally women express actions more frequently than men, and in particular express more positive valence actions but expressiveness is not greater in women for all negative valence actions and is dependent on the discrete emotional state.

In [1] authors carried out experiments to be able to identify the gender and emotion of a person apart from other things based on point light displays. They carried out experiments with one male and one female actor to create videos of point light display of body key points which are then showed to participants to identify the gender of the actors in one experiment and the emotion (out of 6 basic categories) in another set of experiment.

In [4] authors mention that during conversations, women tend to node their heads more than men and also an individual speaking with a woman tends to node more than when speaking with a man. Authors study whether this phenomenon happens because of the coupled motion dynamics or the apparent sex of the person. They use avatars to dissociate sex of the subjects.

3.2 Experiments

3.2.1 Statistical analysis

We carried out independent sample T-test for boy vs girl features (which our classifier uses), separately for the EPI and PSI task. In tables 3.3, 3.4, 3.5 we show the T-test results for AU's, head dynamics and face dynamics respectively. The direction of the T-test is from boys to girls, hence a significantly differing (p¡0.05) positive value indicates that boys have a higher mean value of that feature then girls and vice-versa. Comparing EPI and PSI columns in the tables it indicates boys and girls tend to differ more in the EPI task than PSI task.

Table 3.3 indicates the mean value of AU6 (which corresponds to cheek raiser) differ significantly for boys and girls and indicates girls tend to smile more often. Lots of significantly differing positive values in table 3.4 example, the mean pitch and mean yaw velocities indicate boys, in general, tend to have more head motions than girls.

3.2.2 Tracking analysis

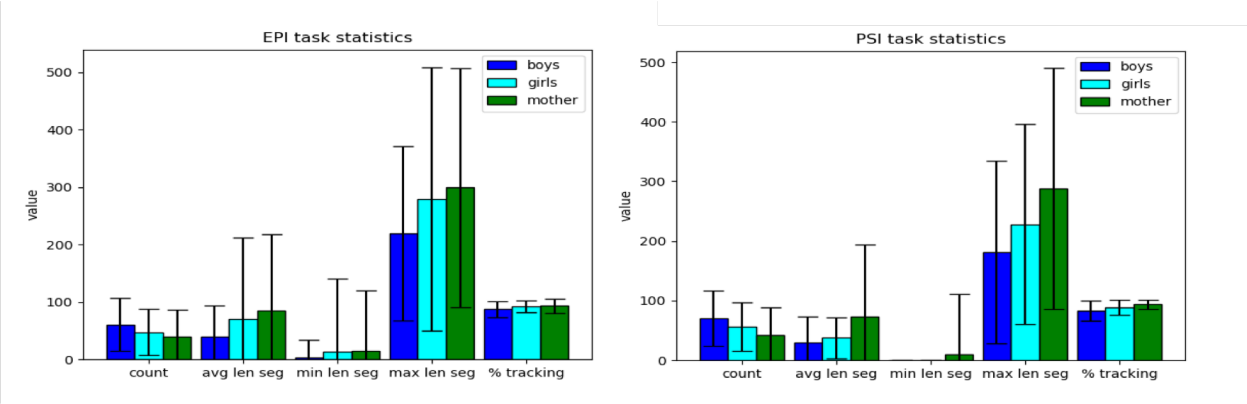

Figure 3.1: Tracking statistics for EPI and PSI tasks

Children usually move a lot and due to large head rotations from the frontal facing camera, our tracker fails to track the faces. During these periods we don't have the AU probabilities and head and facial features. In figure 3.1 we show the statistics of tracking for boys, girls and mothers. There is a sequence of tracked segment followed by a non-segment which is where we don't have tracking. It's interesting to see that boys have a lot of head rotations and in general, the tracking is better for girls as compared to boys. The tracking is best for mothers.

3.2.3 Model

We use the model (approach 1) which we described earlier which is based on prior research in behaviour sciences and meant to predict the LIFE constructs. We modify it so that instead of taking events (segments of videos) as input and predicting the LIFE constructs, it takes the whole video as input and predicts the gender of the child.

3.2.4 Experimental setup

We use 10 fold cross validation with folds divided based on the subject ID's and having an almost equal distribution of the child's gender. We use one fold as validation fold for

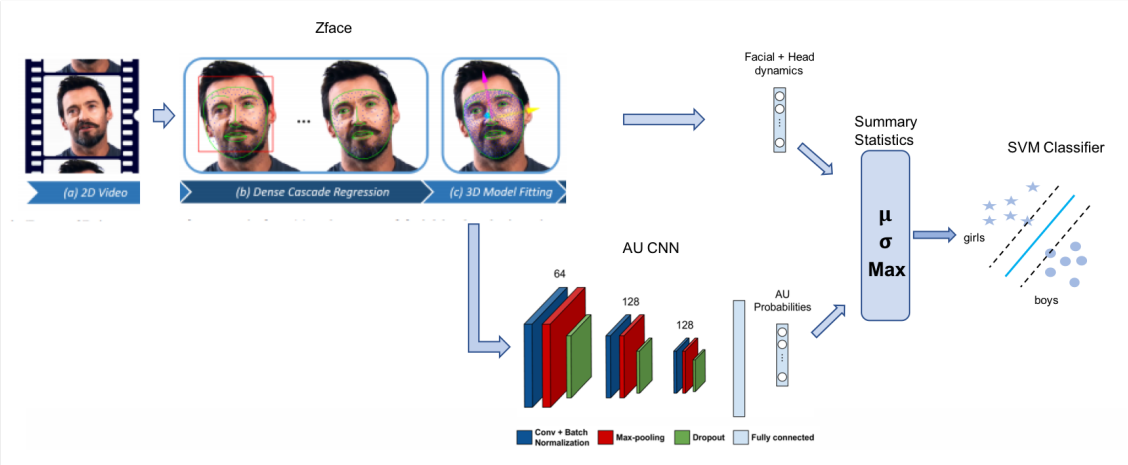

Figure 3.2: Model for gender prediction. Parts of figure taken from [14, 26, 27]

finding the best hyper-parameter, 8 folds for training and one fold as the test fold. We use the validation fold to find the best 'C' parameter of Logistic Regression. We train separate classifiers for event planning (EPI) which is a positive task and problem solving (PSI) which is a negative task.

3.2.5 Results

Feature	EPI Accuracy	EPI Kappa	PSI Accuracy	PSI Kappa
AU	58.54%	0.171	51.1%	0.019
Head dynamics	62.65%	0.253	59.27%	0.184
Face dynamics	57.17%	0.147	59.8%	0.194
Head + Face dynamics	65.15%	0.304	59.93%	0.198
AU + Head dynamics	68.67%	0.372	51.55%	0.03
AU + Head + Face dynamics	68.67%	0.373	61.32%	0.225

Table 3.1: Logistic Regression results for child features

Table 3.1 shows the results of predicting a child's gender by using child's video features with logistic regression for event planning (EPI) and problem solving (PSI) tasks separately.

Feature	EPI Accuracy	EPI Kappa	PSI Accuracy	PSI Kappa
AU	48.64%	-0.025	58.37%	+0.168
Head dynamics	55.19%	+0.104	50.24%	+0.005
Face dynamics	50.20%	+0.002	47.08%	-0.059
Head + Face dynamics	54.39%	+0.088	49.89%	-0.004
AU + Head dynamics	50.42%	+0.010	54.88%	+0.098
AU + Head + Face dynamics	46.86%	-0.064	47.43%	-0.052

Table 3.2: Logistic Regression results for mother features

We show an ablation study with different combinations of AU, head dynamics and face dynamics as the features. Table 3.2 similarly shows the results of predicting a child's gender using mother's video features.

3.3 Discussion

Deep learning [29] based approaches with Convolutions Neural Networks (CNN) have shown to learn very good representations of the data and have shown very good performances in computer vision applications like object detection, emotion recognition and action classification. But in our work, we are interested in finding the behaviour differences and not the appearance based differences of the children. So we didn't use Convolutions Neural Networks as they can learn to focus on the appearance features of the children in the RGB image frames as opposed to learning behaviour features. That's also the reason why we don't use raw RGB image data with our classifiers.

Another approach to model and learn time series data is to use recurrent neural networks like LSTM's. These have shown to have vanishing gradient problem for a very long sequence of data and hence they cannot learn anything beyond certain time samples (around 100). In our case, the videos are on average 20 minutes long and are 25 frames per second. So this amounts to an average of 30000 frames or time stamps which are extremely long for LSTM's to learn from. We did experiment with thin slicing the data wherein we segment the videos into smaller segments and then train LSTM's on these smaller segments. That resulted in poor results, the issue we faced here with such thin slicing is that the particular

behaviour patterns would not be evident in any such small segments rather they are more of holistic natures and hence information in such individual segment may not be descriptive of the behaviour of the person over the entire conversation.

Based on our study with the ADS dataset and the results in tables 3.1 and 3.2 we conclude that:

1) Mother's behaviours are similar towards their children in a dyadic conversation and doesn't depend on the gender of the child.

2) Boys and girls behave differently with their mother and summary of their behaviour over the full dyadic conversation can be used for differentiating the genders.

3) The behaviour of boys and girls towards their mothers differ more significantly in a positive task like event planning in comparison to a negative task like problem solving task.

Feature	EPI t-value	EPI p-value	PSI t-value	PSI p-value
AU6 mean	**-2.038**	**0.043**	-1.462	0.146
AU10 mean	-1.706	0.090	-1.882	0.062
AU12 mean	-1.732	0.085	-0.739	0.461
AU14 mean	**-2.161**	**0.032**	-1.817	0.071
AU15 mean	-0.610	0.543	-1.355	0.177
AU17 mean	-0.692	0.490	-1.589	0.114
AU23 mean	0.035	0.972	0.110	0.913
AU24 mean	1.738	0.084	0.719	0.473
AU6 std	**-3.020**	**0.003**	**-2.424**	**0.017**
AU10 std	**-2.349**	**0.020**	**-2.628**	**0.009**
AU12 std	**-2.477**	**0.014**	-1.531	0.128
AU14 std	**-2.228**	**0.027**	-1.971	0.051
AU15 std	-1.166	0.245	**-2.039**	**0.043**
AU17 std	0.113	0.910	-0.148	0.882
AU23 std	0.316	0.752	0.200	0.842
AU24 std	**2.666**	**0.009**	1.559	0.121
AU6 max	-1.791	0.075	-1.455	0.148
AU10 max	-1.754	0.081	0.762	0.447
AU12 max	-1.754	0.081	-1.073	0.285
AU14 max	-1.489	0.139	-0.606	0.546
AU15 max	**-2.948**	**0.004**	**-2.121**	**0.036**
AU17 max	0.492	0.624	-0.963	0.337
AU23 max	0.827	0.410	-0.161	0.872
AU24 max	**2.672**	**0.008**	**1.974**	**0.050**

Table 3.3: T-test for AU's. Direction is from boys to girls

Feature	EPI t-value	EPI p-value	PSI t-value	PSI p-value
Head P amp mean	-0.781	0.436	-1.351	0.179
Head R amp mean	1.615	0.108	1.061	0.290
Head Y amp mean	**-2.115**	**0.036**	-1.268	0.207
Head P vel mean	**2.790**	**0.006**	0.999	0.320
Head R vel mean	0.052	0.958	0.065	0.948
Head Y vel mean	**2.670**	**0.008**	**1.984**	**0.049**
Head P acc mean	**2.148**	**0.033**	**2.399**	**0.018**
Head R acc mean	-0.310	0.757	-0.920	0.359
Head Y acc mean	**2.125**	**0.035**	0.562	0.575
Head P amp std	**1.998**	**0.048**	0.529	0.598
Head R amp std	-0.486	0.627	-1.657	0.100
Head Y amp std	**2.721**	**0.007**	1.372	0.172
Head P vel std	0.839	0.403	0.739	0.461
Head R vel std	0.101	0.920	-0.120	0.905
Head Y vel std	**2.366**	**0.019**	**2.248**	**0.026**
Head P acc std	1.919	0.057	1.579	0.116
Head R acc std	1.914	0.058	1.215	0.226
Head Y acc std	2.084	0.039	1.912	0.058
Head P amp max	1.604	0.111	-0.451	0.652
Head R amp max	0.161	0.872	0.262	0.794
Head Y amp max	-0.111	0.912	0.564	0.574
Head P vel max	0.178	0.859	-0.840	0.402
Head R vel max	-0.688	0.493	-0.660	0.510
Head Y vel max	-0.102	0.919	0.200	0.841
Head P acc max	0.484	0.629	-1.463	0.146
Head R acc max	-0.663	0.508	-0.805	0.422
Head Y acc max	-1.208	0.229	-0.510	0.611

Table 3.4: T-test for head dynamics. Direction is from boys to girls. (P=Pitch, Y=Yaw)

Feature	EPI t-value	EPI p-value	PSI t-value	PSI p-value
Face 1 vel mean	0.494	0.622	-0.075	0.941
Face 2 vel mean	-1.372	0.172	**-2.824**	**0.005**
Face 3 vel mean	0.037	0.971	0.440	0.661
Face 4 vel mean	-0.946	0.346	0.117	0.907
Face 5 vel mean	1.087	0.279	0.686	0.494
Face 22 vel mean	1.343	0.181	1.213	0.227
Face 23 vel mean	0.155	0.877	0.432	0.666
Face 24 vel mean	**-2.680**	**0.008**	**-3.141**	**0.002**
Face 25 vel mean	-0.278	0.782	-0.179	0.858
Face 26 vel mean	**-3.392**	**0.001**	**-3.593**	**0.000**
Face 1 acc mean	-0.227	0.821	-1.431	0.155
Face 2 acc mean	0.539	0.591	0.647	0.519
Face 3 acc mean	-0.669	0.505	0.571	0.569
Face 4 acc mean	-0.015	0.988	0.367	0.714
Face 5 acc mean	0.832	0.407	0.188	0.851
Face 22 acc mean	0.672	0.503	-0.234	0.816
Face 23 acc mean	-0.798	0.426	-0.899	0.370
Face 24 acc mean	-0.048	0.962	0.617	0.538
Face 25 acc mean	0.335	0.738	1.114	0.267
Face 26 acc mean	-0.673	0.502	-0.998	0.320
Face 1 vel std	0.915	0.362	0.676	0.500
Face 2 vel std	-1.212	0.227	-0.989	0.324
Face 3 vel std	1.253	0.212	0.769	0.443
Face 4 vel std	-0.212	0.833	-0.837	0.404
Face 5 vel std	0.662	0.509	0.532	0.595
Face 22 vel std	1.316	0.190	1.244	0.215
Face 23 vel std	1.272	0.205	1.160	0.248
Face 24 vel std	1.197	0.233	0.857	0.393
Face 25 vel std	**2.058**	**0.041**	1.443	0.151
Face 26 vel std	1.937	0.055	1.566	0.119
Face 1 acc std	1.712	0.089	1.218	0.225
Face 2 acc std	0.146	0.884	0.107	0.915
Face 3 acc std	1.594	0.113	0.984	0.327
Face 4 acc std	0.676	0.500	-0.161	0.873
Face 5 acc std	1.783	0.077	1.297	0.196
Face 22 acc std	1.826	0.070	1.637	0.104
Face 23 acc std	1.665	0.098	1.445	0.150
Face 24 acc std	1.528	0.129	1.073	0.285
Face 25 acc std	**2.549**	**0.012**	1.796	0.075
Face 26 acc std	**2.135**	**0.034**	1.663	0.098

Table 3.5: T-test for face dynamics. Direction is from boys to girls

Chapter 4

Discussion

In this thesis, we focus on building classifiers to detect human affective behaviour at the event and session levels. We propose two models, one which uses handcrafted features based on prior research in behaviour science and the other one is data-driven using deep learning.

For the case of event level detection, we address two challenges, both concern the ground truth of expert annotation to be used for learning algorithms. One is latency. Latency refers to the offset between when an emotion begins and its timestamp in the annotations. When annotators work in real time without stopping and replaying segments of the recording, latency becomes a critical source of error. The other source of error is individual differences between annotators. Even with training, annotators may disagree with how emotion is defined and when it occurs. We analyze the contributions of these two sources of errors to the performance of our classifier by reading ground truth annotations with variable length temporally shifted windows around the annotated timestamps. Our results indicate the importance to account for these sources of error in ground truth annotations when annotations are done in real time. We show accounting for these errors helps in the performance of the classifiers for detecting emotions.

For the case of session level detection, we dealt with the case of predicting the gender of the children based on their conversation with their mothers in the dyadic tasks. Boys and girls behave differently with their mother and summary of their behaviour over the full dyadic conversation can be used for differentiating the genders while mother's behaviour is alone not indicative for predicting child's gender. The behaviour of boys and girls towards

their mothers differ more significantly in a positive task like event planning in comparison to a negative task like problem solving task.

Currently our best model achieves an accuracy of 64% for predicting the three constructs. There are lots of future directions to get further improvement. One of the main limitations is the error in the annotations especially for aversive and dysphoric constructs. Manually looking at the video segments and the corresponding annotations, this discrepancy is very much evident. The inter-rater reliability study shows the low reliability for aversive and dysphoric constructs and this limits the performance which one can get from machine learning algorithms.

The most important future direction to further improve the performance of our classifier is getting better annotation. Real-time annotation results in latency and individual differences as major sources of errors resulting in a poor quality of ground truth annotations. Stop frame coding ((i.e. when annotators pause and rewind videos to temporally fix their annotations)) can be used to reduce these errors, resulting into higher inter-rater reliability of the annotations.

Stop frame coding is time consuming and expensive process and might not be possible for huge datasets. So what additional things can be done for real-time annotated datasets? One thing is our error analysis showed how it affects the performance of our classifier. It has certain limitations because of the following assumption: 1) All three constructs have same window size and latency 2) No annotator specific errors 3) No drift across the video for the same annotator (i.e. errors are not temporally dependent).

A subset of the dataset can be coded in a different way to overcome these limitations and better understand the errors (latency and individual differences). First, if the coding on a subset is done with stop frame then one can compare those annotations with the current real-time ones and better model the errors in annotations. Second, coding a subset of data in real-time with multiple coders (around 4-5) and re-coding the same videos by the same annotators can also reveal more about the errors in annotations. Third, last few minutes of the videos can be coded separately to understand the impact of drift on annotations (i.e. are the errors same at the end of the video when coders might be tired, in comparison to the beginning of the videos). This knowledge about errors in annotations can then be incorporated into classifiers when reading the original annotations in the full dataset.

Another limitation is we consider all frames of the video segment (window) to belong to a particular construct. From the algorithm point of view as mentioned earlier Multiple instance learning based approach might be a way to tackle this limitation. More can be explored in this direction. One can also consider algorithms which can learn and find which annotations are noisy and select only the good annotations for training the classifier. Reinforcement learning based approach might be relevant in this direction. Apart from these imbalanced data is also a possible contributing factor for the lower performance of the classifiers, data augmentation to generate more data for the classes with fewer training samples might help to overcome this limitation.

Chapter 5

Appendix

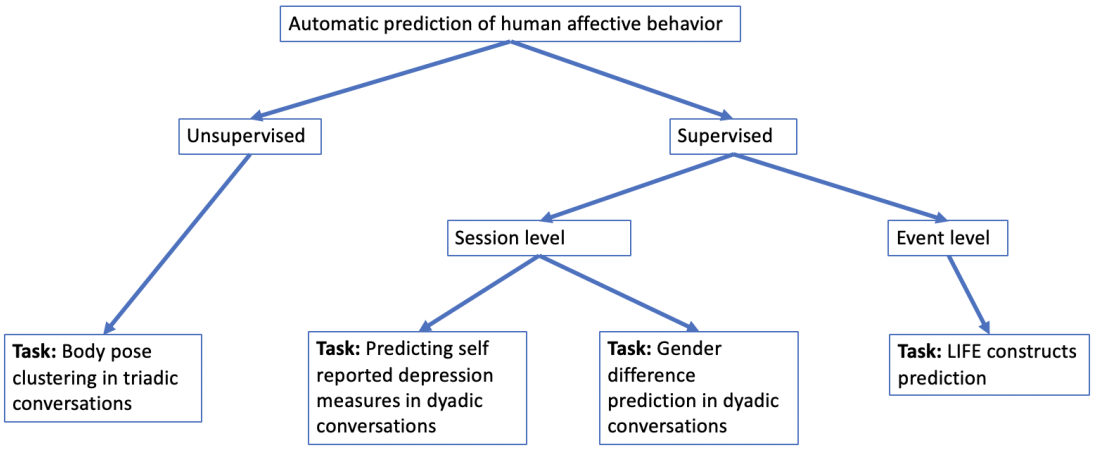

Figure 5.1: Overview of my master's work

Bibliography

[1] Kaat Alaerts, Evelien Nackaerts, Pieter Meyns, Stephan P Swinnen, and Nicole Wenderoth. Action and emotion recognition from point light displays: an investigation of gender differences. *PloS one*, 6(6):e20989, 2011.

[2] Samuel Albanie and Andrea Vedaldi. Learning grimaces by watching tv. *arXiv preprint arXiv:1610.02255*, 2016.

[3] Piotr Bilinski, Antitza Dantcheva, and François Brémond. Can a smile reveal your gender? In *2016 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6. IEEE, 2016.

[4] Steven M Boker, Jeffrey F Cohn, Barry-John Theobald, Iain Matthews, Michael Mangini, Jeffrey R Spies, Zara Ambadar, and Timothy R Brick. Something in the way we move: motion dynamics, not perceived sex, influence head movements in conversation. *Journal of Experimental Psychology: Human Perception and Performance*, 37(3):874, 2011.

[5] Sumit Chopra, Raia Hadsell, Yann LeCun, et al. Learning a similarity metric discriminatively, with application to face verification. In *CVPR (1)*, pages 539–546, 2005.

[6] Jeffrey F Cohn, Zara Ambadar, and Paul Ekman. Observer-based measurement of facial expression with the facial action coding system. *The handbook of emotion elicitation and assessment*, pages 203–221, 2007.

[7] Jeffrey F Cohn, Itir Onal Ertugrul, Wen-Sheng Chu, Jeffrey M Girard, László A Jeni, and Zakia Hammal. Affective facial computing: Generalizability across domains. In *Multimodal Behavior Analysis in the Wild*, pages 407–441. Elsevier, 2019.

[8] Marco Cuturi. Fast global alignment kernels. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 929–936, 2011.

[9] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[10] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997.

[11] P. EKMAN. Facial action coding system (facs). *A Human Face*, 2002.

[12] P Ekman, WV Friesen, and JC Hager. Facial action coding system: the manual. research nexus, div. *Network Information Research Corp., Salt Lake City, UT*, 1:8, 2002.

[13] Paul Ekman and Wallace V Friesen. Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2):124, 1971.

[14] Itir Onal Ertugrul, Jeffrey F Cohn, László A Jeni, Zheng Zhang, Lijun Yin, and Qiang Ji. Cross-domain au detection: Domains, learning approaches, and measures. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pages 1–8. IEEE, 2019.

[15] Florian Eyben, Felix Weninger, Florian Gross, and Björn Schuller. Recent developments in opensmile, the munich open-source multimedia feature extractor. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 835–838. ACM, 2013.

[16] Florian Eyben, Martin Wöllmer, and Björn Schuller. Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1459–1462. ACM, 2010.

[17] Schuller BW Sundberg J Andr E Busso C Devillers LY Epps J Laukka P Narayanan SS Truong KP Eyben F, Scherer KR. The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing. *IEEE Transactions on Affective Computing*, 2015.

[18] Jeffrey M Girard, Wen-Sheng Chu, László A Jeni, and Jeffrey F Cohn. Sayette group formation task (gft) spontaneous facial expression database. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 581–588. IEEE, 2017.

[19] Ian J Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, et al. Challenges in representation learning: A report on three machine learning contests. In *International Conference on Neural Information Processing*, pages 117–124. Springer, 2013.

[20] Rahul Gupta, Kartik Audhkhasi, Zach Jacokes, Agata Rozga, and Shrikanth Narayanan. Modeling multiple time series annotations based on ground truth inference and distortion. *IEEE Transactions on Affective Computing*, (99):1–1, 2016.

[21] Zakia Hammal, Jeffrey F Cohn, Carrie Heike, and Matthew L Speltz. What can head and facial movements convey about positive and negative affect? In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 281–287. IEEE, 2015.

[22] Zakia Hammal, Jeffrey F Cohn, and Daniel S Messinger. Head movement dynamics during play and perturbed mother-infant interaction. *IEEE transactions on affective computing*, 6(4):361–370, 2015.

[23] H Hops, A Biglan, A Tolman, J Arthur, and N Longoria. Living in family environments (life) coding system: Reference manual for coders. *Eugene, OR: Oregon Research Institute*, 1995.

[24] Hyman Hops, Betsy Davis, and Nancy Longoria. Methodological issues in direct observation: Illustrations with the living in familial environments (life) coding system. *Journal of Clinical Child Psychology*, 24(2):193–203, 1995.

[25] László A Jeni, Jeffrey F Cohn, and Fernando De La Torre. Facing imbalanced data–recommendations for the use of performance metrics. In *2013 Humaine association*

conference on affective computing and intelligent interaction, pages 245–251. IEEE, 2013.

[26] László A Jeni, Jeffrey F Cohn, and Takeo Kanade. Dense 3d face alignment from 2d videos in real-time. In *2015 11th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, volume 1, pages 1–8. IEEE, 2015.

[27] László A Jeni, Jeffrey F Cohn, and Takeo Kanade. Dense 3d face alignment from 2d video for real-time use. *Image and Vision Computing*, 58:13–24, 2017.

[28] Patrik N Juslin and Petri Laukka. Communication of emotions in vocal expression and music performance: Different channels, same code? *Psychological bulletin*, 129(5):770, 2003.

[29] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[30] Andras Lorincz, Laszlo Jeni, Zoltan Szabo, Jeffrey Cohn, and Takeo Kanade. Emotional expression classification using time-series kernels. In *Proceedings of the IEEE Conference on computer vision and pattern recognition workshops*, pages 889–895, 2013.

[31] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[32] Soroosh Mariooryad and Carlos Busso. Analysis and compensation of the reaction lag of evaluators in continuous emotional annotations. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pages 85–90. IEEE, 2013.

[33] Soroosh Mariooryad and Carlos Busso. Correcting time-continuous emotional labels by modeling the reaction lag of evaluators. *IEEE Transactions on Affective Computing*, 6(2):97–108, 2014.

[34] Daniel McDuff, Evan Kodra, Rana el Kaliouby, and Marianne LaFrance. A large-scale analysis of sex differences in facial expressions. *PloS one*, 12(4):e0173942, 2017.

[35] Gary McKeown, Michel Valstar, Roddy Cowie, Maja Pantic, and Marc Schroder. The semaine database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent. *IEEE Transactions on Affective Computing*, 3(1):5–17, 2011.

[36] Baback Moghaddam and Ming-Hsuan Yang. Learning gender with support faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):707–711, 2002.

[37] Mihalis A Nicolaou, Hatice Gunes, and Maja Pantic. Continuous prediction of spontaneous affect from multiple cues and modalities in valence-arousal space. *IEEE Transactions on Affective Computing*, 2(2):92–105, 2011.

[38] Mihalis A Nicolaou, Vladimir Pavlovic, and Maja Pantic. Dynamic probabilistic cca for analysis of affective behaviour. In *European Conference on Computer Vision*, pages 98–111. Springer, 2012.

[39] Jérémie Nicolle, Vincent Rapp, Kévin Bailly, Lionel Prevost, and Mohamed Chetouani. Robust continuous prediction of human emotions using multiscale dynamic cues. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, pages 501–508. ACM, 2012.

[40] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7):971–987, 2002.

[41] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition. In *bmvc*, volume 1, page 6, 2015.

[42] Sona Patel and Klaus R Scherer. Vocal behaviour. *Handbook of nonverbal communication. Berlin: Mouton-DeGruyter*, pages 167–204, 2013.

[43] Barnabás Póczos, Liang Xiong, Dougal J Sutherland, and Jeff Schneider. Nonparametric kernel estimators for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2989–2996. IEEE, 2012.

[44] Rajeev Ranjan, Vishal M Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):121–135, 2017.

[45] Robert Rosenthal. Conducting judgment studies: Some methodological issues. *The new handbook of methods in nonverbal behavior research*, pages 199–234, 2005.

[46] Robert Rosenthal. Conducting judgment studies: Some methodological issues. *The new handbook of methods in nonverbal behavior research*, pages 199–234, 2005.

[47] James A Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

[48] Bjorn Schuller, Bogdan Vlasenko, Florian Eyben, Martin Wollmer, Andre Stuhlsatz, Andreas Wendemuth, and Gerhard Rigoll. Cross-corpus acoustic emotion recognition: Variances and strategies. *IEEE Transactions on Affective Computing*, 1(2):119–131, 2010.

[49] Orli S Schwartz, Michelle L Byrne, Julian G Simmons, Sarah Whittle, Paul Dudgeon, Marie BH Yap, Lisa B Sheeber, and Nicholas B Allen. Parenting during early adolescence and adolescent-onset major depression: A 6-year prospective longitudinal study. *Clinical Psychological Science*, 2(3):272–286, 2014.

[50] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[51] Mohammad Soleymani, Sadjad Asghari-Esfeden, Yun Fu, and Maja Pantic. Analysis of eeg signals and facial expressions for continuous emotion detection. *IEEE Transactions on Affective Computing*, 7(1):17–28, 2015.

[52] Dougal J Sutherland, Liang Xiong, Barnabás Póczos, and Jeff Schneider. Kernels on sample sets via nonparametric divergence estimates. *arXiv preprint arXiv:1202.0302*, 2012.

[53] Paul Viola, Michael Jones, et al. Rapid object detection using a boosted cascade of simple features. *CVPR (1)*, 1:511–518, 2001.

[54] Baiqiang Xia. Which facial expressions can reveal your gender? a study with 3d faces. *arXiv preprint arXiv:1805.00371*, 2018.

[55] Marie BH Yap, Nicholas B Allen, and Cecile D Ladouceur. Maternal socialization of positive affect: The impact of invalidation on adolescent emotion regulation and depressive symptomatology. *Child development*, 79(5):1415–1431, 2008.

[56] Marie BH Yap, Orli S Schwartz, Michelle L Byrne, Julian G Simmons, and Nicholas B Allen. Maternal positive and negative interaction behaviors and early adolescents' depressive symptoms: Adolescent emotion regulation as a mediator. *Journal of Research on Adolescence*, 20(4):1014–1043, 2010.

[57] Xing Zhang, Lijun Yin, Jeffrey F Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, Peng Liu, and Jeffrey M Girard. Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. *Image and Vision Computing*, 32(10):692–706, 2014.

[58] Zheng Zhang, Jeff M Girard, Yue Wu, Xing Zhang, Peng Liu, Umur Ciftci, Shaun Canavan, Michael Reale, Andy Horowitz, Huiyuan Yang, et al. Multimodal spontaneous emotion corpus for human behavior analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3438–3446, 2016.