Multimodal Affective Computing Using Temporal Convolutional Neural Network and Deep Convolutional Neural Networks

Issa Ayoub

Thesis submitted to the University of Ottawa
in partial Fulfillment of the requirements for the
Master of Applied Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

Affective computing has gained significant attention from researchers in the last decade due to the wide variety of applications that can benefit from this technology. Often, researchers describe affect using emotional dimensions such as arousal and valence. Valence refers to the spectrum of negative to positive emotions while arousal determines the level of excitement. Describing emotions through continuous dimensions (e.g. valence and arousal) allows us to encode subtle and complex affects as opposed to discrete emotions, such as the basic six emotions: happy, anger, fear, disgust, sad and neutral.

Recognizing spontaneous and subtle emotions remains a challenging problem for computers. In our work, we employ two modalities of information: video and audio. Hence, we extract visual and audio features using deep neural network models. Given that emotions are time-dependent, we apply the Temporal Convolutional Neural Network (TCN) to model the variations in emotions. Additionally, we investigate an alternative model that combines a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN). Given our inability to fit the latter deep model into the main memory, we divide the RNN into smaller segments and propose a scheme to back-propagate gradients across all segments. We configure the hyperparameters of all models using Gaussian processes to obtain a fair comparison between the proposed models. Our results show that TCN outperforms RNN for the recognition of the arousal and valence emotional dimensions. Therefore, we propose the adoption of TCN for emotion detection problems as a baseline method for future work. Our experimental results show that TCN outperforms all RNN based models yielding a concordance correlation coefficient of 0.7895 (vs. 0.7544) on valence and 0.8207 (vs. 0.7357) on arousal on the validation dataset of SEWA dataset for emotion prediction.

Acknowledgment

Undertaking my master's degree would not have been possible without the support I received from my supervisor, my family and friends. I am greatly indebted to my supervisor, Dr. Hussein Al Osman, for his continuous efforts, support and guidance. I am sincerely grateful for Dr. Al Osman for assisting me in the planning and implementation phases of my thesis over the past 2 years. He was always ready to answer my questions, provide me with his expertise and push me to do better when needed. Moreover, Dr. Al Osman was my mentor and played an essential role in my growth personally and academically.

I am extremely grateful for having my family who supported me unconditionally during this journey. I would like to thank my mother Sanaa, my grandmother Maria, my father Samir and my sisters Hala and Ghada. They were the central point of my life, the main source of motivation, cooperation, support and love.

I cannot be more fortunate without my best friend, Charbel Tawk, who was the first one to share the ups and downs of my research life, and for helping me reach the best version of myself. Having my friend's support and motivation was another positive factor throughout this life-changing experience. Lastly, I would like to thank all my friends who were beside me throughout my Master's journey.

Dedication

To my Parents and sisters.

To my supervisor Dr. Hussein Al Osman

To my best friend Charbel Tawk

Table of Contents

Chapter 1. Introduction 1
1.1. Motivation 1
1.2. Problem Statement 2
1.3. Contributions 5
1.4. Thesis Outline 5
Chapter 2. Background 7
2.1. Multimodal Features 7
2.1.1. Audio Features 7
2.1.2. Video Features 8
2.1.3. Textual Feature 9
2.1.4. Physiological Features 9
2.2. Datasets 9
2.2.1. RECOLA Dataset 10
2.2.2. AFEW Dataset 11
2.2.3. AFEW-VA Dataset 12
2.2.4. Aff-Wild dataset 12
2.2.5. SEWA Dataset 13
2.3. Hyperparameter Optimization 15
2.4. Recurrent Neural Networks 18
2.5. Temporal Convolutional Neural Networks 21
2.5.1. Causal Convolution 22
2.5.2. Dilated Convolutions 23
2.5.3. Residual Connection 24
Chapter 3. Related Work 26
3.1. Related Work on the RECOLA Dataset 26
3.2. Related Work on SEWA Dataset 30
3.3. Related Work on Aff-Wild Dataset 34
Chapter 4. Proposed Method 35
4.1. Multimodal Features 36
4.1.1. Audio Features 36

4.1.2. Word Features 37
4.1.3. Video features 38
4.2. Memory-Efficient Backpropagation for Recurrent Neural Network 39
4.3. Applying TCN in Unimodal setting 45
4.4. Applying TCN in Multimodal Settings 46
4.4.1. Training TCN using Feature Level Fusion 46
4.4.2. Training TCN under the Model Level Like-Fusion Paradigm 46
4.4.3. Training LSTM and GRU model in Multimodal Setting 48
Chapter 5. Results and Discussion 49
5.1. Results of TCN with unimodal features 49
5.2. Results of TCN with Multimodal features 52
5.2.1. Results Obtained Using Feature Level Fusion 52
5.2.2. Results Obtained Using Model level Fusion 55
5.3. Results of Training AffWildNet Using Memory-Efficient Backpropagation for Recurrent Neural Networks 60
5.4. Memory-Efficient Backpropagation for Recurrent Neural Networks Evaluation and Results 61
Chapter 6. Conclusion and Future Work 64
6.1. Conclusion 64
6.2. Future Work 65
References 66

Table of figures

Figure 1 - Grid Search Vs. Random Search for Hyperparameter Optimization.. 16

Figure 2 - The processes of hyperparameter optimization using Gaussian Process. 18

Figure 3 - Recurrent Neural Network cell Unfolded Over Multiple Time Steps 19

Figure 4 - Predictions at any time step depends on previous inputs if all. 23

Figure 5 - Example of a Dilated Convolution Neural Network..... 24

Figure 6 - Residual Unit..... 25

Figure 7 - Extracting Audio Features Pipeline 37

Figure 8 - Emotion detection procedure starting from feature extraction. 41

Figure 9 - Initial forward pass for the RNN. All inputs are fed sequentially..... 43

Figure 10 - Gradient propagation within one segment 44

Figure 11 - Our TCN architecture following Decision Level Fusion.. 47

Figure 12 - The accuracy of Arousal and Valence when segment size is 2,..... 63

Figure 13 - This figure shows the memory vs. time requirement , displayed..... 63

Table of Tables

Table 1 - Emotion Related databases and their limitations..... 14

Table 2 - Summary of our custom model used before the RNN 45

Table 3 - The Performance of Different TCN models with audio features on the. 50

Table 4 - the Performance of Different TCN models with video features on the..... 50

Table 5 – Performance results of the best TCN model on the arousal 51

Table 6 - Performance results of the best TCN model on the arousal. 52

Table 7 – TCN model configurations trained using feature level fusion..... 53

Table 8 - Comparison between the best models achieved while training..... 53

Table 9 - Performance of different TCN architectures trained in the 54

Table 10 - TCN model trained with a model level fusion setting..... 56

Table 11 - Comparison of performance of different TCN models trained 57

Table 12 - Reported Results of TCN trained using model level fusion against 58

Table 13 - Reported Results of TCN trained using model level fusion against. 58

Table 14 - Performance of different CNN architectures while utilizing 1 layer GRU 60

Glossary of Terms

RNN: Recurrent Neural Network

LSTM: Long Short-term Memory

GRU: Gated Recurrent Unit

TCN: Temporal Convolutional Neural Network

SEWA: Automatic Sentiment Analysis in the Wild

RECOLA: The REmote COLlaborative and Affective.

AVEC: Audio/Visual Emotion Challenge.

eGeMAPS: Extended Version of Geneva Minimalistic Acoustic Parameter Set.

MFCC: Mel-frequency Cepstral Coefficients.

LGBP-TOP: Local Gabor Binary Patterns from Three Orthogonal Planes.

HOG: Histogram of Gradients.

MSDF: Multiscale Dense SIFT.

BoW: Bag of Words

CNN: Convolutional Neural Network

RMSE: Root Mean Square Error

MSE: Mean Square Error

CCC: Concordance Correlation Coefficient

SVM: Support Vector Machine

SVR: Support Vector Regression

GP: Gaussian Processes

EWE: Evaluator Weighted Estimator.

Chapter 1. Introduction

The field of affect detection has been gaining increased attention due to its applicability to various domains such as human-computer interaction and healthcare [1]. Today, machines can recognize affects by analyzing multiple modalities of information that include the human voice, facial expressions, and body gestures.

Conventionally, human emotions are divided into six discrete classes: happiness, sadness, fear, neutral, disgust, and surprise. However, to express the subtlety of human emotions, researchers are increasingly describing affects using a 2D or 3D dimensional model of emotion where typically the x-axis represents valence and the y-axis corresponds to arousal [2]. Commonly, valence and arousal range between -1 to +1. For valence, +1 signifies the peak of positive emotions and -1 reflects the maximum negative emotion that can be recorded by the model. Similarly, +1 for arousal reflects the complete engagement of the subject in an interaction and -1 reflects the absence of expressions or clues about the underlying emotions. For 3D models, dominance is habitually the third dimension [3][4]. Evidently, due to the expressive power of dimensional models of emotion, compared to discrete categories of emotion, measuring the former is considerably more challenging compared to the latter [5]. This is mainly due to the large spectrum of emotions that can be encoded using the dimensional model ranging from subtle to pronounced.

1.1. Motivation

Computer-based human emotion estimation is a challenging endeavor that has gained significant attention from researchers in the last decade due to its envisioned applicability to various technological systems [6] such as computer tutoring applications that personalize learning based on the user's affective response or automated systems that assist therapists in the diagnosis of

depression [7] or bipolar disorder [7]. Picard, in her 1997 book on Affective Computing [8], described how barometers could be installed in classrooms to provide feedback regarding the students' level of engagement [8]. Picard [8] also describes how emotionally intelligent agents can possibly assist individuals with autism to navigate difficult social situations [8].

Affect recognition in the wild refers to the detection of the emotional state of subjects engaging in spontaneous and natural interaction. In contrast, detection of acted emotions refers to the recognition of the emotions expressed by subjects intentionally for the purpose of collecting a dataset. Moreover, detection of induced emotions refers to the estimation of the affective state of subjects reacting to strong emotional content, such as watching an emotionally charged video clip. Acted and induced emotions tend to be prominent and devoid of the subtlety that often characterizes natural interactions. Several algorithms have achieved impressive results for the detection of emotions on acted datasets [9][10][11]. However, there remain challenges in the recognition of spontaneous emotions expressed in the wild.

Given the wide spectrum of affective computing applications, we are motivated to pursue the development of affect recognition algorithms. We believe that the technology has not achieved its peak potential, especially for affect recognition in the wild. Therefore, we will focus our effort in this thesis on the latter problem.

1.2. Problem Statement

Emotion recognition algorithms can process multiple modalities of information, such as facial expression, speech, physiological signal and gesture [12], to produce their output. Although many algorithms rely on a single modality of information to detect emotions, these techniques fail when the modality's features are missing such as when the audio is missing as the target speaker is silent

or when the face is occluded. As a result, multimodal solutions are more reliable for the detection of emotions. Moreover, multimodal solutions are typically more accurate due to the increase in the considered relevant information.

We can typically achieve modality fusion in one of three ways: feature, decision, and model level fusion. In decision level fusion [13], multiple models are trained on unimodal features. The models feed their outputs into a second level model such as a Support Vector Machine (SVM) [14] or a linear regression model [15]. In either case, the interaction between multimodal features is ignored, which is a drawback as features may be correlated [16]. In feature level fusion [17][13], features from multiple modalities are concatenated and fed into the model to compute the predictions. Rozgic et al. [18] achieved improved results in emotion prediction when they applied feature level fusion compared to decision level fusion. However, this technique can suffer from the curse of dimensionality [19], especially when a single classifier is fed with a high number of features after concatenating multimodal features. As a result, overfitting may occur if we train with a small dataset. Additionally, features should be extracted at the same time, such as every 100 ms, for synchronization. Model level fusion [20] techniques often fuse intermediate representations of features, such as concatenating hidden layers from different modalities [21], kernel fusion for kernel classifiers [22], and novel forms of feature interactions in Hidden Markov Modal (HMM) classifier [23]. As a result, model level fusion solves the asynchrony dilemma among multimodal features while preserving their natural correlation over time.

Predicting spontaneous emotions during natural interaction is more challenging than estimating acted or induced emotions. During natural interaction, the same facial expression can depict different emotional states as affect can best be appreciated from the analysis of multiple modalities of information [12]. Moreover, a rater could be biased and provide different labels for the same

facial expression and vocal tone [24] patterns. This increased complexity can be eased by transforming input features into a different space which makes the prediction mechanism easier. In other words, using neural networks to automatically learn the hidden representation, or features, can outperform engineered features. These hidden features learned by neural networks are the result of transforming the input into a different space as mentioned above. Thus, neural networks, a particular class of machine learning, is the most recently employed technique [5][4] given its success and robustness in the field of affective computing.

Emotions are continuously evolving. Hence, they can be described as a time series of dependent values, i.e., the valence/arousal measure at *t* is dependent on the measure at $\{t-k, \dots, t-1\}$ where *k* is a variable number of time steps. Predictions are done using regression methods such as SVMs, deep belief networks with temporal pooling, multimodal temporal fusion [25], and recurrent neural networks [19]. Wollmer et al. [26] proposed a Long Short Term Memory-Recurrent Neural Network (LSTM-RNN) for continuous emotion prediction yielding the best average recognition performance. Additionally, Zhao et al. [5] performed experiments to evaluate the performance of SVM and LSTM-RNN. Their results show that LSTM-RNN outperforms SVM methods in emotion recognition challenges. Recently, researchers have been adopting LSTM-RNN as a consistently robust regression method for continuous emotion prediction.

Therefore, in this thesis, we will address the following problems:

- Choosing the neural network architectures that allow us to extract robust features to enable accurate affect estimation.
- Choosing the most suitable time series model to predict emotions and selecting the optimal model hyperparameters.

1.3. Contributions

1- Typically, when we have a limited memory capacity, an RNN is trained after truncating the sequence data into segments of length n. However, this will impose a limitation on capturing long-term dependencies. As a result, we provide a new way to train an RNN on any length sequential data.

2- We adopted the Temporal Convolutional Neural network (TCN) [27] architecture, an architecture that outperforms RNNs on most time series applications. Our work outperforms the state of the art on the emotion prediction challenge on the SEWA dataset defined in Section 2.2.5.

3- We apply Gaussian processes on top of TCN and LSTM to perform hyperparameter optimization.

1.4. Thesis Outline

The rest of the thesis is organized as follows:

Chapter 2: We present the thesis background. In particular, we discuss the features used for emotion prediction, datasets employed in related work, hyperparameter optimization, and time series models, namely RNN and TCN.

Chapter 3: We provide an overview of the related work. We discuss previous approaches and techniques used for dimensional emotion prediction.

Chapter 4: We describe the TCN architectures that we train using feature level fusion and model level fusion. Additionally, we present our approach to fitting LSTMs modelling long-term time series into memory.

Chapter 5: We present our results, mainly the accuracy for the arousal and valence estimation using TCN, LSTM, and Gated Recurrent Unit (GRU) models. Moreover, we discuss our results and conclusion.

Chapter 6: We summarize the thesis findings and provide insights into future work.

Chapter 2. Background

Performing emotion recognition from audio-visual features is a tedious process where the accuracy depends on the extracted multimodal features. These features are extracted using tools that have been studied for decades. Recently, using representation learning, features are learned automatically after defining the objective function and the dataset to perform the recognition task [28]. A noticeable effort has been made in the field of affective computing to learn audio/video representation [29][30].

Most commonly, the features used for emotion recognition are extracted from audio, video, physiological, and text modalities. We discuss these modalities in the following sub-sections.

2.1. Multimodal Features

2.1.1. Audio Features

AVEC [7][32][33] challenges provide a set of audio features that are divided into supervised expert-level features, semi-supervised and unsupervised features. The first set includes features computed over low-level descriptors (LLD) with a set of functionals such as median, mean, standard deviation, min, max, etc., computed over a fixed window length. The LLDs and their statistical measures are extracted using the openSMILE toolkit [33]. This includes the extended version of Geneva Minimalistic Acoustic Parameter set (eGeMAPS) [34], which contains 23 acoustic LLD, extracted using a window length of 10 ms, and their functionals giving a total of 88 features. LLDs cover cepstral, spectral, prosodic and voice quality information. Previous work proved the robustness of this feature set in affective computing [36][37]. Another set of low-level descriptors' audio features is the Mel-frequency Cepstral Coefficients (MFCC) 1-13 along with

their functionals, first and second derivative. The second set includes the Bag-of-X-Words (BoW) as semi-supervised representations extracted from text and audio using the openXBOW toolkit [38].

2.1.2. Video Features

Visual features that are used for emotion detection pertain to all relevant information from facial expressions, eye gaze and blinking, pupil diameter, and hand and body gestures and poses [12]. Visual features appear in two forms according to Al Osman et al. [12], appearance and geometric features. Geometric features are related to the detected landmarks and their first and second derivative to estimate their speed and direction of motion as the facial expression develops [12]; in addition, the head pose and eye gaze direction can be taken into account. Appearance features are those related to the overall texture information that results from the deformation of the neutral expression [12].

Action units are one example of appearance features. Action units taxonomize variations in facial muscles and are used to describe emotions [12]. Some action units are a combination of other action units and the same action unit might have different intensity levels [39].

Local Gabor Binary Patterns from Three Orthogonal Planes (LGBP-TOP) and Histogram of Gradients (HOG) are appearance features. LGBP-TOP are extracted from a sequence of video frames to capture dynamic information. Hence, the faces should be detected and aligned according to the mean and standard deviation of the landmarks before the calculation of the LGBP_TOP. HOG are computed by dividing an image into a grid and calculating the distribution and the intensity of the gradients or edges within each cell.

According to Chao et al. [40], predicting the valence emotional dimension is best accomplished using visual features.

2.1.3. Textual Feature

Textual features include bag of words (BoW), i.e., word frequencies as features, which are learned from the training partitions of the dataset after removing the stop words and keeping the unigrams as the dictionary. The textual modality is best used for sentiment analysis and has proved its usability in predicting the likability dimension of emotion [5][4]. Word vector is another set of textual features. In this approach, each word is mapped into an n-dimensional vector representing the semantics of the word. This mapping takes place using a neural network trained on a massive dataset [45].

2.1.4. Physiological Features

Various physiological signals are utilized for the emotion detection task due to the presence of certain biological patterns that are reflective of subtle underlying emotional behavior [19, 20]. Biological patterns are present in electrocardiography (ECG), electromyography (EMG), electroencephalograph (EEG), skin conductance (EDA) and skin conductance response (SCR), respiration rate, heart rate (HR), heart rate variability (HRV) and skin temperature [12] signals. Signals are captured using non-invasive sensors affixed to the human body. The RECOLA [47] database, discussed in Section 2.2.1, introduces physiological features in their database.

2.2. Datasets

The application of multimodal emotion detection requires a large collection of sensory data obtained from a large number of subjects. Additionally, comparing models trained on different

dataset does not lead to reliable conclusions given the difference in the experimental setup. Therefore, researchers have published publically available datasets to expedite the algorithm validation process and allow researchers to compare their results consistently. We divide these datasets into three categories: posed, induced and natural emotional database. In the posed databases, subjects are asked to act out an emotion. These databases are not introduced or discussed in the thesis since they do not reflect spontaneous emotions. In induced databases, subjects are asked to watch a movie clip while recording their emotional reaction. Likewise, the displayed emotions are not entirely spontaneous and do not reflect emotions displayed between individuals interacting in real life. For example, the AffWild [51][52] dataset belongs to this category. Finally, with a natural emotional database, subjects interact naturally with another peer. Subjects are asked to have a discussion on a certain topic, such as a watched advertisement, and their emotional reaction is recorded. For instance, both RECOLA [47] and SEWA [8] are natural emotional databases. Additionally, datasets differ in the emotion's labels, that is, discrete vs. continuous emotions. In our work, we describe few databases in terms of their advantages, disadvantages, and the category to which they belong. .All details are presented in Table 1.

2.2.1. RECOLA Dataset

The REmote COLlaborative and Affective (RECOLA) corpus introduced by Ringeval et al. [47] is concerned with spontaneous and natural emotions along the arousal and valence emotional dimensions. Each video was recorded for 5 minutes and the arousal/valence annotations are provided by 6 French-speaking subjects, 3 males and 3 females. The corpus contains four modalities: video, audio, ECG, and electro-dermal activity (EDA) recorded synchronously from 27 French-speaking subjects. The subjects are diverse and originate from Germany, France, and Italy. The subjects are assigned to 3 sets; training, validation, and testing dataset where each set is

composed of 9 subjects such that the age, gender, and mother tongue are balanced. The ratings of arousal and valence are recorded every 40 ms frame. Additionally, the corpus provides the inter-rater reliability measured by Cronbach's α and inter-class correlation coefficient. Finally, the ratings are concatenated among all participants and the RMSE, Pearson Correlation Coefficient (CC) and concordance correlation coefficient (CCC) values are averaged over all possible pair of raters. The CCC is defined as:

$$\rho_c = \frac{2\rho\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2 + \left(\mu_x - \mu_y\right)^2}$$

Where μ_x and μ_y are the means of the sequences *x* and *y*, and σ_x and σ_y are the corresponding standard deviations. ρ is the Pearson Correlation Coefficient (PCC) between *x* and *y*. Generally, *x* and *y* reflect the model predictions and gold standard labels respectively.

2.2.2. AFEW Dataset

The AFEW dataset [48] corpus is adopted in the EmotiW challenges [49][50] which focus on classifying audiovisual clips into 7 facial expressions (Disgust, Fear, Anger, Sadness, Happiness, Surprised and neutral). The dataset contains 1809 videos, presenting dynamic temporal facial expressions, extracted from reality TV shows and movies. Subjects in the video range from 1 to 77 years old. The videos are divided into 3 sets, training set (773 video clips), validation set (383 video clips) and the testing set (653 video clips). The dataset has a limitation in terms of the small number of annotators (3 annotators were used), the small number of frames in total, and the restriction of emotions to 7 basic categories, among which fear, surprise, and disgust are scarce in the dataset.

2.2.3. AFEW-VA Dataset

Part of the AFEW dataset has been annotated in the valence and arousal dimension hence giving a new dataset named AFEW-AV. The new dataset contains 600 video clips that are considered natural and realistic, and include many variations of illuminations, background, head pose and free movements from the subject. Each video ranges from 10 frames to more than 120 frames, giving a total of 30,000 frames. Furthermore, the annotations are provided on a frame by frame basis. Finally, the annotations range from -10 to +10 for valence and arousal.

The AFEW-AV includes some limitations due to the limited number of frames, the small number of annotators (only 2) and the use of discrete values for the emotional dimensions which provides a coarse approximation of the subject's behavior in real life and poor modeling of emotional state, contrast to time continuous emotions that can better model the richness and expressiveness of the emotional state.

2.2.4. Aff-Wild dataset

The Aff-Wild dataset [51][52] introduced in the Affect-in-the-wild challenge (Aff-Wild Challenge) in conjunction with CVPR 2017 contains recordings that describe the human emotional state in terms of arousal and valence. The dataset presents a diverse set of subjects and a large number of frames and 8 annotators. Its recordings show subjects with a variety of head poses and gestures while the illuminations and background differ from one video to another. However, the emotions displayed by participants may not be subtle as the participants were reacting to an emotional stimulus such as watching a disturbing clip or reacting to a prank.

2.2.5. SEWA Dataset

The SEWA database [8] which contains recordings made in the wild, in contrast to the RECOLA [41] database. The dataset consists of 64 German subjects divided into 3 sets, 34 for training, 14 for validation and 16 for testing. A major extension to the SEWA database is done by adding 66 Hungarian subjects to the testing dataset. The aim is to test the generalizability of the machine learning model and exploring how the knowledge of emotion recognition can be transferred from one culture to the other. This open research challenge has gained increased attention in affective computing [53][54] research community.

The recordings include dyadic human-human interaction where each person appears in a separate video. The videos were recorded using personal devices such as a personal PC with a webcam. Since two people can be heard in each recording, additional features were added to the dataset to identify which participant is talking at a given time during the recording, and this is referred to as turn information. Each recording lasts for a maximum of three minutes in which the subjects discuss a product commercial they have watched.

In addition to the video and audio modalities, the transcription of the speech along with the duration and timestamps of utterances are provided by a native speaker.

Finally, the videos are labeled by six raters across three dimensions, arousal, valence, and liking; liking refers to the participant's appreciation of the advertised product. Labels are recorded every 100 ms and the gold standard from the six raters is calculated using the evaluator weighted estimator (EWE).

Table 1 - Emotion Related databases and their limitations

Database	Model of affect	Category	Total number of frames	Number of videos	Number of annotators	limitation
RECOLA	Valence-arousal (continuous)	Natural Emotions	202,500	27	6	- Videos collected in lab environment - Small number of frames -Small number of subjects
AFEW	Seven universal facial expressions	Natural and induced Emotions	113,355	1809	3	- Seven basic expressions - Limited number of frames - Small number of annotators - Imbalanced expression categories
AFEW-AV	Valence-arousal (discrete)	Natural and induced Emotions	30,050	600	2	- Small number of frames - Discrete arousal and valence values - Very small number of annotators
SEWA	Valence-arousal (continuous)	Natural Emotions	228,410	130	6	- limited variations in participants' ethnicities (German and Hungarian) -Small number of frames
Aff-Wild	Valence-arousal (continuous)	Natural and induced Emotions	1,224,100	298	8	-Not fully naturalistic emotions

Since we are concerned about natural emotions, we used the SEWA and the RECOLA datasets throughout the thesis to train and evaluate different models.

Given the noise in the labels provided by annotators, obtaining higher inter-rater agreement is a daunting task especially with dimensional emotions [24]. Ringeval et al. [55] discussed several post-processing techniques to enhance the inter-rater agreement between annotators. Re-annotation causes blanks or jumps to appear in the annotations and thus creates unwanted

variabilities. To fix this problem, Ringeval et al. [55] proposed a piecewise cubic interpolation scheme to estimate missing data points for less than 20 seconds of recording. Furthermore, to increase inter-rater reliability, they proposed two normalization techniques. The first is the zero normalization technique, which reduces any bias in the annotation. The second scheme pertains to synchronization due to a time lag in the reaction to annotate the emotion. In fact, Ebrahimi et al. [56] have shown that time-shifting the annotation from -2 to +2 seconds has minimized inter-rater agreement mean square error for each pair of the annotators. Additionally, they showed that applying synchronization and zero mean normalization provides more balanced instances for both valence and arousal. Finally, they performed mean filtering on the 6 annotators to compute the ground truth labels.

2.3. Hyperparameter Optimization

The process of designing neural network architectures can be tedious and require practitioners with years of experience to set the values of hyperparameters manually. One way to perform hyperparameter optimization is through the grid search algorithm. This method requires a researcher to define a range for each parameter a priori. However, the amount of time required to find the optimal parameter values scales exponentially as the number of parameters increases.

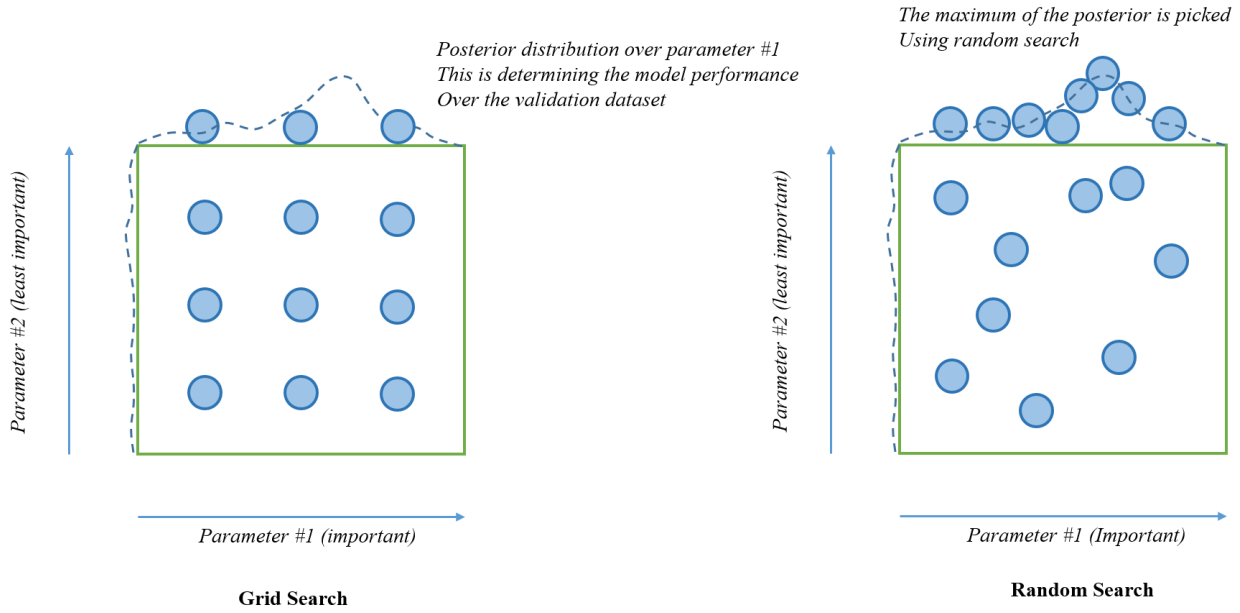

Figure 1 - Grid Search Vs. Random Search for Hyperparameter Optimization. Each square in the figured above represent a 2-dimensional space, relating to 2 parameters. The process of hyperparameter optimization defines a method that looks in this space for the best parameter values that lead to the smallest error on the validation dataset. The image to the left describes the searching processes following the Grid search algorithm. As a result, the image demonstrates how the grid search isn't able to find the maximum of the posterior, where the peak lies between values picked by the algorithm. The dots represent the values chosen for each pair of parameters. However, the right image shows how random search is able to find the maximum of the posterior. Additionally, this figure shows that the performance is mostly affected by the parameter across the x-axis while the parameter across the y-axis contributes less to the performance of the model on the validation dataset.

Bergstra et al. [104] proved that hyperparameter optimization using random search can be superior to grid search optimization. Larochelle et al. [105] provided empirical evidence to prove the disadvantage of grid search, after comparing hyperparameter optimization using grid search to manual search to configure neural networks. Bergstra et al. [104] found that adopting random search optimization achieved better results compared to grid search within a small fraction of the computation time. Since grid search involves searching over the solution space by stepping exhaustively over the hyperparameters values, increasing the step size may result in skipping the maximum of the objective function (see Figure 1) while decreasing it renders the grid search too long.

Recently, hyperparameter optimization based on Gaussian Processes [106] (GPs) has been adopted as the systematic approach to automatically discover the near optimal machine learning

hyperparameters. Hence, this problem is addressed in Bayesian analysis by sampling from space to find the peak of the posterior distribution, or objective function, where the objective function is the accuracy on the validation dataset, as a function of the model hyperparameters.

A GP is a function over functions. Given a set of points $x_1, \ldots, x_n$ then the finite set of values $f(x_1), \ldots, f(x_n)$ follows a multivariate normal distribution. Typically, $x_1, \ldots, x_n$ are a set of vectors, where each entry corresponds to one hyperparameter. A GP is specified by a mean $m(x)$, where $m(x) = E[f(x)]$, and a covariance matrix $k(x, x')$, the kernel, equal to $Cov(f(x), f(x'))$. The smoothness of the functions generated from a GP is determined by the covariance matrix which determines the correlation between all pairs of output values. As a result, $k(x, x')$ is large when x and x' are close to one another. A common function for $k(x, x')$ is $e^{-\alpha\|x,x'\|}$, where α is a random hyperparameter, could be determined by the maximum likelihood. This distance is denoted as Radial Basis Form (RBF), and the resulting kernel matrix denoted by $K_{XX'}$:

$$K_{XX'} = \begin{pmatrix} k(x_1, x_1) & \cdots & k(x_1, x_N) \\ \vdots & \ddots & \vdots \\ k(x_N, x_1) & \cdots & k(x_N, x_N) \end{pmatrix}$$

Where X is D dimensional; $X \in R^D$.

Typically, given a set data points (observations) $X \in R^D$, representing the model hyperparameters, and their corresponding outputs $f \in R^n$, representing the accuracy over the validation dataset, with n being the total number of input vectors; this will results in tractable posterior distribution [106].

Bayesian optimization is an approach to optimizing objective functions that take too long (minutes and hours) to evaluate. It is best suited for optimization over continuous domains of less than 20 dimensions, and tolerate stochastic noise in function evaluations [108].

A Bayesian optimizer can give new suggestions for hyperparameters in a region of the search-space that is not explored yet, or those that could bring the most improvement. This process is known as expected improvement. Repeating this process a number of times will build a good model of how performance will vary as a function of the hyperparameters. Figure 2 describes this processes:

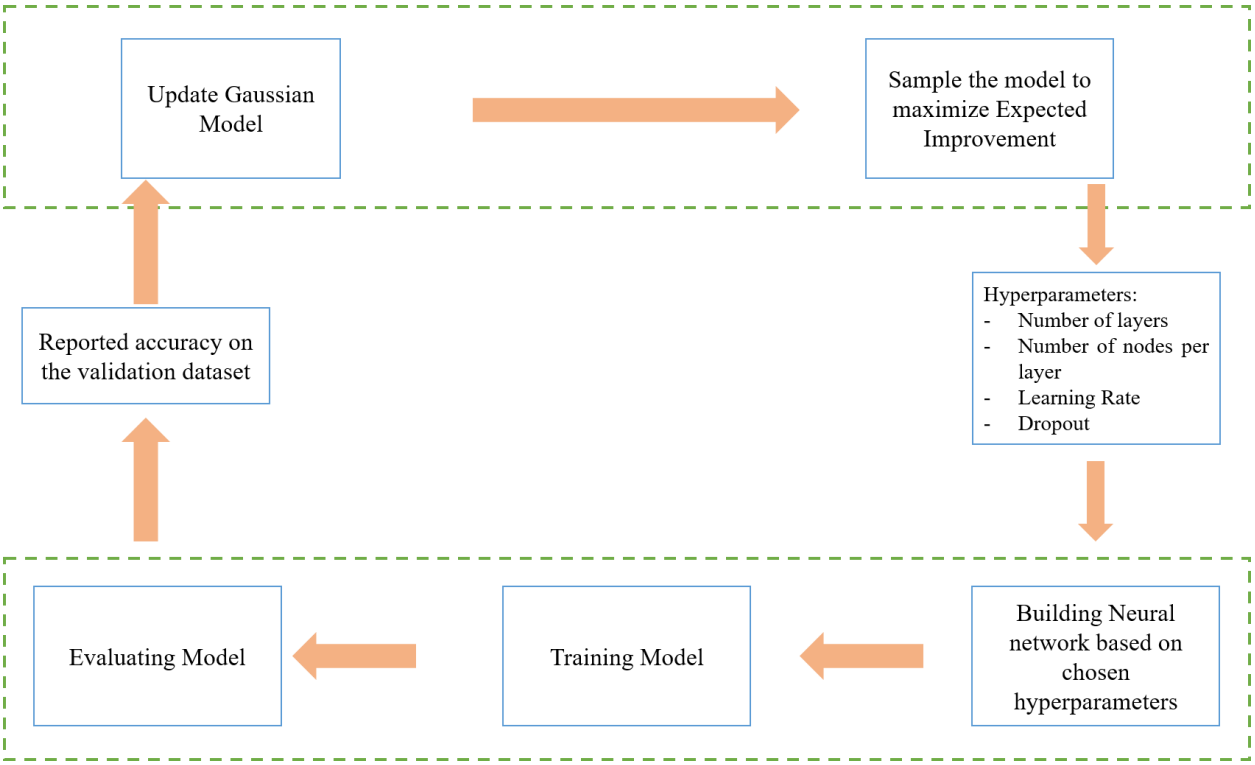

Figure 2 - The processes of hyperparameter optimization using Gaussian Process and Bayesian Optimizer.

2.4. Recurrent Neural Networks

Commonly, ANNs, such as Convolutional Neural Networks (CNNs), consider their input data to be independent and identically distributed [127]. However, RNNs break from this convention by processing input data sequentially [127], which involves the application of the recurrence formula over multiple time steps:

$$h_t = \sigma(W_{hh}h_{t-1} + W_{xh}x_t) \qquad (1)$$

Where σ denotes a non-linear function, h_t is the current state or the hidden vector at time t, which is a function of the hidden state at a previous time step h_{t-1} multiplied by the transition matrix W_{hh} added to the current input x_t multiplied by the weight matrix W_{xh}.

Theoretically, RNNs can be extended to infinitely long sequences. An RNN is similar to a state machine where the output at each time step is a function of the current input and the output of the previous step [127]:

$$y_t = W_{hy}h_t \qquad (2)$$

Where y_t is the output of the RNN cell at time step t, h_t is the current hidden state, and W_{hy} is a weight matrix. Figure 3 shows an example of an RNN.

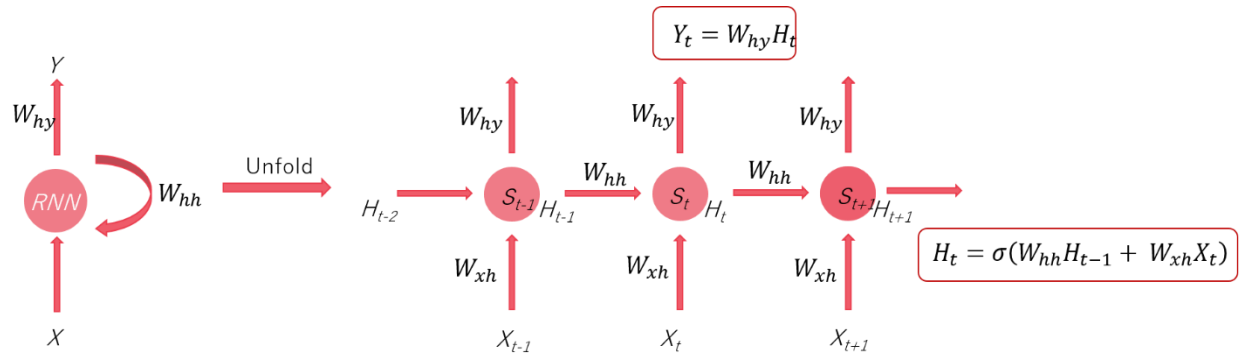

Figure 3 - Recurrent Neural Network cell Unfolded Over Multiple Time Steps

To train an RNN, we apply backpropagation over time [130]. By unraveling the network, we consider the RNN to be a feed-forward network with multiple layers. Hence, we can train the RNN through the conventional backpropagation algorithm. However, in this case, all time steps share the same set of weights.

Theoretically, gradients at each time step are back propagated until time 0 and updating the model parameters is done by summing the gradients followed by optimizing using gradient descent.

As the number of time steps increases, the depth of the RNN, once unraveled, increases as well. Hence, we may encounter the vanishing or exploding gradients problems during model training [131]. We can address the exploding gradients problem by clipping the gradients. However, to solve the vanishing gradients problem, researchers have proposed new RNN architectures such as the Long Short-Term Memory (LSTM) unit [132] and GRU [133][131]. These architectures solve the vanishing gradient problem by applying an identity function to the input of the LSTM or GRU cells. The identity function has a derivative of 1, hence gradients are back propagated without vanishing [131].

Given the requirement needed by the standard backpropagation through time (BPTT) algorithm, a memory constraint arises when we unroll the RNN for a large number of time steps [134][135] as the internal states and activations of all cells are stored in memory to compute the gradients. As a result, researchers commonly employ truncated-BPTT by dividing the network into a fixed number of segments of length S [136][130]. The last hidden state from a segment is fed as an initial hidden state to the next segment. However, dividing the network into a fixed number of segments prohibits the RNN cells from capturing long-term dependencies [137]. This problem arises since RNN gradients are summed up during training. Summing S gradients from a single segment is not equivalent to summing all gradients in the network. Therefore, when the RNN is truncated, the gradients only back propagate within each segment as opposed to across the entire network. We propose a solution to this problem in Section 4.2.

2.5. Temporal Convolutional Neural Networks

Deep learning practitioners commonly adopt RNN methods and their variations such as LSTM and GRU for time series prediction. However, today, convolutional architectures are being applied for applications where conventionally RNNs were used such as word-level language modeling [101], audio synthesis [102] and machine translation [103]. Consequently, a new CNN based model named, Temporal Convolutional Neural network (TCN) was proposed [27] in an attempt to adopt CNNs instead of RNNs for sequential modeling. Bai et al. [27] presented a systematic evaluation across different RNN architectures and generic convolutional models. They performed testing on a broad range of tasks and datasets that are commonly used to benchmark recurrent networks. Their results show that TCNs are superior to RNNs on most tasks and datasets. Hence, they proposed the adoption of CNN as a new time series model for future research involving time series prediction.

The distinguishing characteristics of TCNs include:

1) Fast training and convergence capability
2) Dependence on causal convolutional layers. Hence, there is no information leakage from future to past.
3) A simpler and clearer architecture. The model now is a simple sequence of convolutional layers.
4) The ability to take input sequences of any length and map them to an output sequence of the same length, just as with RNNs.

Finally, the key advantage of TCNs is their ability to look very far into the past to make a prediction using a combination of very deep networks (augmented with residual layers) and dilated convolutions.

2.5.1. Causal Convolution

Given that TCNs are based upon 2 key principles: no leakage of information from the future to the past and their ability to produce an output of equal length as the input, some factors need to be satisfied. First, TCNs depend on a 1D Fully-Convolutional Network (FCN). All layers are of the same length and inputs are zero padded by (kernel size - 1) to keep subsequent layers the same length as previous ones. Second, the TCN causal convolution is what prevents the leakage of the future information (inputs) to predict the current output at time *t*. Figure 4 demonstrates the concept of causal convolution.

Hence, the prediction of any point y_t depends on previous inputs $x_{t-n}, \dots, x_t$.

However, to capture long-term dependencies, or long effective history, the network should be very deep due to the linear relationship between the network depth and the effective history size. Thus, this limitation makes it hard to apply a 1D FCN on a sequence modeling task. Fortunately, Bai et al. [27] proposed using dilated 1D-FCN which supports an exponentially large receptive field.

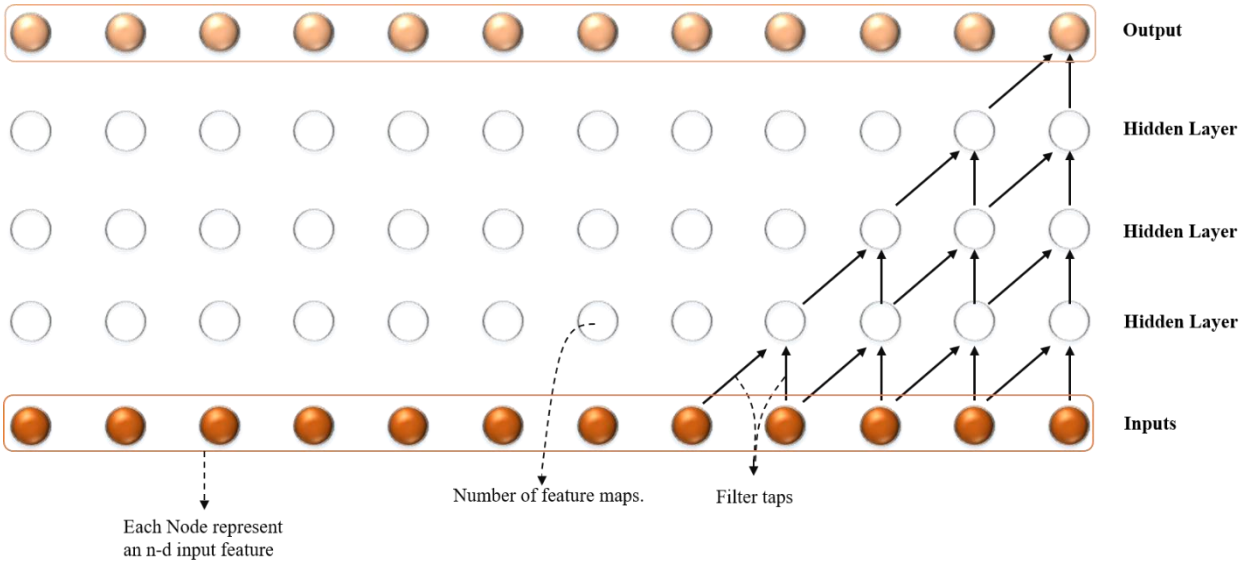

Figure 4 - Predictions at any time step depends on previous inputs if all layers are 1D convolutional causal layers. White nodes represents the output hidden convolutional layers.

2.5.2. Dilated Convolutions

Dilated convolution is the technique which allows us to model long-term dependency with fewer layers in the 1D-FCN [27]. The dilation factor *d* introduces a fixed step between every 2 adjacent filter taps of the convolutional filters which allows nodes of hidden layers to span a wider range of inputs, thus enabling long term effective history. When $d = 1$, a dilated convolution is treated as a regular convolution. Using larger dilations enables an output at the top level to represent a wider range of inputs, thus effectively expanding the receptive field of a convolutional network. Figure 5 provides an example of a dilated convolutional neural network.

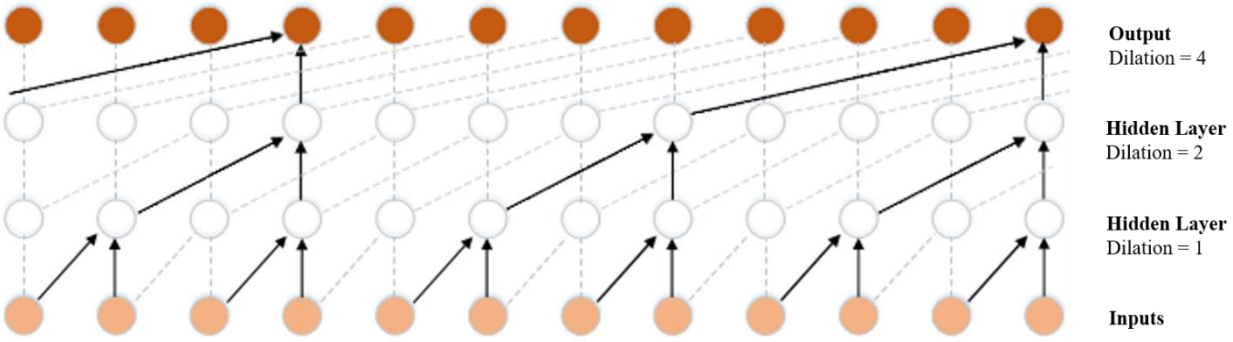

Figure 5 - Example of a Dilated Convolution Neural Network

To enable long-term dependency or to increase the receptive field of the TCN, we can either choose to increase the filter size *k*, dilation factor *d*, or the network depth. Commonly, we should increase the dilation factor after the addition of each hidden layer as shown in the Figure 5, i.e., $d = O(2^i)$ at level i of the network. However, in our work, we did not add any restrictions to the depth to attempt to achieve better time-dependent deep learning features. We will evaluate this design decision in Section 5 of the thesis. Finally, the receptive field size of any layer is (k - 1) * d.

2.5.3. Residual Connection

Adding more layers to neural networks shows improved performance in accuracy [61]; however, there exists a threshold for the number of layers after which the accuracy peaks and then degrades rapidly during training. Remarkably, this is not related to overfitting since the degradation in performance occurs on the training dataset [61].

If we consider a shallow network and its deeper counterpart such that the added layers are identity mappings, then the deeper model should produce an error not higher than its shallower counterpart. Therefore, inspired by this idea, He et al. [61] proposed a residual unit which consists of an identity mapping (shortcut connection, x) and a mapping function F(x), added together, as shown in Figure 6. This way, it is easier to derive a mapping function F(x) to output zeros if the identity mapping

is the ideal solution. However, if the ideal solution requires having F(x) close to zero, then this learning should be easier than learning the entire transformation function which consists of few stacked layers, as with VGG models, which they refer to as “plain network”.

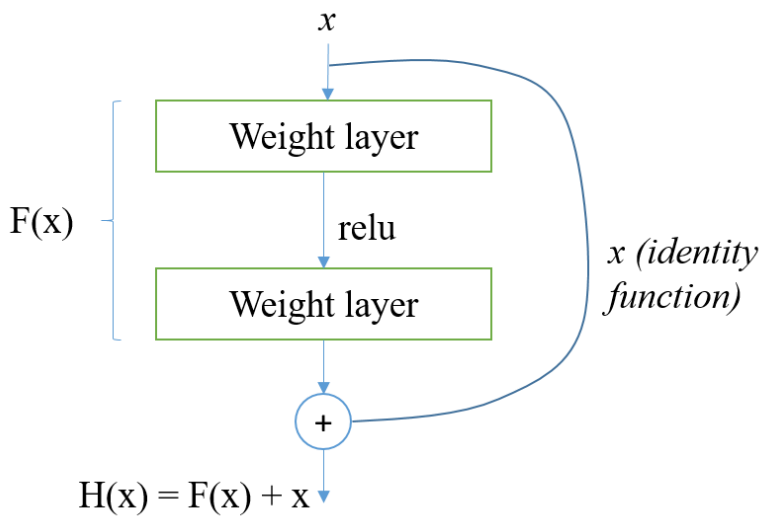

Figure 6 - Residual Unit

Therefore, instead of stacking layers to learn the function transformation H(x), the residual framework will require learning an easier function transformation $F(x)$ such that $H(x) = F(x) + x$. Obviously, identity shortcut connections do not add any parameters or computational complexity. Moreover, the entire network can be trained using stochastic gradient descent with backpropagation. He et al. [61] evaluated ResNets after conducting comprehensive experiments and their results show how ResNets were easy to optimize compared to “plain” nets, which consist of stacked layers, and that considerable accuracy improvements are achieved with stacking more layers. As a result, Bai et al. [27] applied residual connections to TCNs and the results outperformed all RNN (including LSTM and GRU) model configurations.

Chapter 3. Related Work

In this section, we discuss different unimodal and multimodal emotion prediction regression algorithms proposed in the literature. We review different time series models such as Kalman filters and RNN as well as the different set of features fed into the time series models which include handcrafted features, and automatically learned features by neural networks.

3.1. Related Work on the RECOLA Dataset

Povolny et al. [33] introduced a multimodal emotion detection algorithm using audio features with bottleneck (BN) and text-based features along with the same set of features introduced by Velstar et al. [41]. The text-based features consisted of word embedding [57][45] and sentiment lexicon. BN features are extracted from a narrow hidden layer of a neural network trained toward phonetic targets. They have been recently integrated with automatic speech recognition systems and their multilingual variants [58]. The set of visual features in [41] were complemented with CNN features, extracted from hidden layers, after training the CNN on a landmark localization task [59]. The CNN features were found to encode geometric and appearance based visual features. Finally, Povolny et al. [33] proposed multiple linear regression systems, trained on individual feature sets, for predicting the arousal and valence emotional dimensions. The individual systems, trained on each modality separately, are followed by additional single linear regression system, as the decision level fusion technique, for the final arousal/valence predictions. CCC on the arousal increased from 0.699 to 0.833 after the inclusion of BN features on the validation dataset of the RECOLA dataset. On the other hand, Sun et al. [15] complemented the visual features in [41] using the Multiscale Dense SIFT (MSDF) features and deep visual features. Sun et al. [15] utilized MSDF features as they showed considerable potential in the facial expression recognition domain

[60]. They obtained deep visual features from AlexNet [60] and from Deep Residual Neural Networks [61] for each input frame of the RECOLA videos. Finally, they trained a SVR model for each set of input features followed by multiple linear regression to perform final prediction as their decision level fusion method. The results showed that Povolny et al. [33] outperformed Sun et al. [15] in the arousal dimension of the testing dataset, scoring an accuracy (CCC) of 0.719 compared to 0.683. However, Sun et al. [15], achieved better results in the valence dimension, scoring 0.642 compared to 0.596.

In contrast, Somandepalli et al. [62] proposed Kalman filters as the decision level fusion method compared to [33][15]. First, they performed predictions from unimodal features using a support vector regression (SVR) where they treated predictions as noisy estimates of arousal and valence. They fed the SVR models' output to the Kalman filters for fusion. To leverage the correlation between arousal and valence [63], they considered arousal predictions as additional noisy observation to enhance the prediction accuracy of valence. They proposed a facial posture cues and voicing probability scheme to account for the multimodal nature of the problem. Hence, they detected the presence and absence of each input modality to select the appropriate Kalman filter while performing online tracking. Their results demonstrated additional improvements on the validation and testing dataset compared to the baseline results of AVEC2016, scoring 0.703 for arousal and 0.681 for valence.

Different CNN architectures including AlexNet have been applied on emotion recognition where results demonstrate obvious performance improvement [64][65]. Recent studies show the benefits of applying deep convolutional networks for facial feature extraction. Tao et al. [66] investigated the effectiveness of various modalities for predicting dimensional components of emotions. They achieved the best results for the arousal dimension estimation using the audio modality which

consists of the Geneva Minimalistic Acoustic Parameter Set [35] features. For valence, they achieved the best results from the FACE-CNN features extracted from the final layer of the CNN. These results are consistent with previous studies from Gunes et al. [67][68]. Additionally, their results showed that appearance feature sets render better results compared to geometric features for both arousal and valance prediction. In contrast, they obtained the least accurate results when they used physiological features. Gunes and Schuller [67] trained two separate deep CNNs pre-trained on a large dataset and then fine-tuned on their audio and video dataset.

Given the superiority of RNN in [69][70][55] and the success of LSTM in continuous emotion prediction, [71][55][72][16], Gunes et al. [71] proved the success of LSTMs in dimensional emotion prediction and achieved promising results. Additionally, Ringeval et al. [55] successfully applied LSTM-RNN for regression on dimensional emotional recognition utilizing the visual, audio, and physiological modalities. Perhaps, the main feature that makes LSTMs appropriate for emotion prediction is their robustness in the presence of outliers and ability to describe temporal structure and variations in the emotions over time [72]. Chen et al. [16] used LSTMs to capture the long term inter-dependency within a multimedia signal's segments. They proposed a new conditional attention fusion scheme where modalities are weighted depending on the current input feature and its recent history.

Given the difficulty of training deep neural network, the ResNet model is proposed by [73] to ease the training. The main intuition behind ResNet model is, if a shallow network followed by identity mapping should be at least as good as the shallow network, therefore, learning an identity mapping is easier than learning the full function. As a result, training a ResNet comes down to train residual units as described in [73]. It has been shown that ResNet models achieve better performance with deeper layers and are able to generalize better on the testing dataset [73].

Several works combined CNN and LSTM models to resolve human emotion. Tzirakis et al. [72] used a Resnet-50 to capture robust features from both raw audio and video signals. The extracted features were then fed into a two layer LSTM. The model was trained from end-to-end as opposed to training individual components separately. Their work outperformed traditional approaches based on baseline handcrafted features (LGBP-TOP, geometric features and pixel coordinate of the 49 detected landmarks) in the RECOLA dataset used in AVEC 2016. Deep ResNets takes advantage of stacked residual blocks of the form:

$$y_k = \mathcal{F}(x_k, \{\mathcal{W}_k\}) + h(x_k) \qquad (3)$$

Where *x* and *y* are the input and the output from the residual unit. F is the function applied by the residual block and h is an identity function.

Huang et al. [74] employed a Deep Neural Network (DNN) and hypergraphs for emotion recognition using facial information. After training the CNN, they extracted facial features from the last fully connected layer and treated them as attributes for the hypergraph. Another study was performed by Ebrahimi et al. [56] where the CNN was used to extract features that were fed into a Recurrent Neural Network (RNN). The purpose was to categorize emotions in a video. At the end, the model produces a single value that reflects the emotion in a video clip. Brady et al. [75], winner of AVEC 2016, proposed training a CNN followed by an RNN in an end to end manner. The RNN is used as a regressor where predictions are performed in the arousal and valence dimensions. Given the robustness of sparse coding method in the field of computer vision and object recognition tasks [76][77], their second contribution depends on extracting higher level features from raw features using supervised and unsupervised learning method based on deep learning and sparse coding, to ease the learning of the SVM regressor which was chosen as the baseline regressor in past AVEC challenges [9][10]. Finally, they proposed predicting continuous

emotion dimensions using a state space approach such as Kalman filters [11] where measurements and noise are treated as Gaussian distribution. They achieved a CCC of 0.770 and 0.687 in the arousal and valence respectively on the testing dataset of the RECOLA database.

3.2. Related Work on SEWA Dataset

Recently, researchers have been increasingly adopting an LSTM-RNN architecture for emotion prediction. Haung et al. [79], trained different LSTM-RNN models separately on different features types while utilizing temporal pooling and annotation delay. The annotation delay caused by the reaction lag of the annotators observing the videos [80] is handled by shifting forward the annotations before training the model. An Epsilon-insensitive loss function is used to avoid small errors while the absolute value is applied to the large errors. This cost function was proven to be superior to other cost functions [66]. LSTM is followed by SVR as the decision level fusion method to predict the final output for each dimension of the emotion. The speaking turn information is utilized to decrease the influence of the interlocutor on the target speaker.

Recently, phonetic features, Phone Log-Likelihood Ratio (PLLR), have outperformed the eGeMAPS feature set in continuous emotion prediction with both Relevant Vector Machines (RVM) and SVR [81], suggesting useful phonetic information that aid in emotion prediction. Therefore, Dang et al. [82] utilized this feature set in their emotion detection algorithm.

Also, it was assumed that certainty in continuous emotion prediction does not vary over time. However, it is hard to use hard labels for emotion modeling [83] due to the inherent uncertainty in emotions. Therefore, characterizing emotions as distributions with mean and standard deviation, reflecting intensity and confidence, is a more promising [84] approach. Finally, adopting multiple regression models reduces the uncertainty in emotion prediction. An effective method for

multimodal fusion is the Output-Associative RVM (OA-RVM) [85][86][87] which is capable of capturing temporal information and dimensional affect dependencies. Therefore, the predictions of multiple regressors in the arousal and valence dimensions can be included in the OA metrics which was the main motivation for Dang et al. [82] to apply this fusion approach. Hence, the OA-RVM is adopted as a technique for multimodal fusion with multiple modalities and feature types. As a result, Dang et al. [82] achieved significant improvement over the baseline results in AVEC 2017 by 39.5%, 17.6% and 29.3% for arousal, valence and liking.

Chen et al. [4], investigated different deep learned features from the acoustic, visual and textual modalities. For the acoustic features they used the Soundnet [88], a CNN based model, to extract low-level acoustic features. Aytar et al. [88] had shown that extracted acoustic features from the Soundnet CNN outperformed MFCC features in an audio event recognition task. Additionally, Chen et al. [4] explored several strategies to assess the influence of the interlocutor on the target speaker. These strategies take into account the noise in the audio features when the interlocutor is speaking and the target speaker is silent. Hence, one way to reduce the noise is by multiplying audio features by zeros whenever the interlocutor speaks, and keep the audio features unmodified when the target speaker speaks. As a result, Chen et al. [4] took advantage of the turn information provided with the dataset. Visual features are CNN features extracted from the VGGFace [89] CNN pre-trained on FER+ [90] dataset, a new annotation for the standard emotion FER dataset where each image is labeled by 10 crowd-sourced taggers. Moreover, they used another feature set they extracted from the DenseNet [91] model pre-trained on the FER+ dataset and fine-tuned on the SEWA dataset. As for the textual modality, they utilized word-embedding features on the speech transcriptions in the original language (German) and translated language (English) in addition to the conventional bag-of-words textual representation. Since arousal and valence are

highly correlated, they trained them simultaneously in a multi-task learning framework. Their results showed that the fusion of modalities can benefit the arousal and valence prediction, but unimodal solutions that only used textual features generalized well for the liking prediction. Finally, they compared the LSTM and SVR models and their results showed that the temporal LSTM model significantly outperformed the non-temporal SVR model.

Wataraka et al. [92] presented a new approach for predicting arousal and valence that does not depend on low and high-level features. Instead, predicting arousal and valence depends on sparse information, salient events and gestures embedded in human speech such as laughter, moaning and fumbling. In emotion psychology [93], some clearly identifiable events can trigger the non-verbal expression of affect in both voice and face. This behavior is defined as affect bursts. As a result, Wataraka et al. [92] considered the non-verbal vocal events they extracted from speech transcripts. Such events were previously largely disregarded in previous emotion prediction frameworks [94][95]. Each type of event defines a different fluctuation in the emotional state (arousal and valence). This was perceived to be consistent among subjects/speakers. Their experimental results highlighted the effect of laughter and slight laughter as salient events, or vocal affect bursts, in the subject's speech. Huang et al. [19], utilized audio, video and textual modalities along with an RNN-LSTM as a regressor for emotion prediction. Due to the difficulty in capturing continuous emotions, and due to the shortage in training data, they applied data augmentation by creating overlapping video segments extracted from the original videos. This method shows better results compared to models trained on the original data. Additionally, they compared two fusion modalities, feature-level fusion and the decision-level fusion. Their results showed that both methods have comparable accuracy in the arousal dimension. However, feature-level fusion

outperforms decision-level fusion in valence dimension and decision level fusion achieve better results in the liking dimension.

Zhao et al. [5], explored the deep learning features of different modalities while utilizing the LSTM network to model long-term temporal relations between the data point, followed by a regressor. Due to the high correlation in the arousal and valence dimension between both speaker interacting through a conversation, and similar to [4], Zhao et al. [5] proposed that interlocutor features can influence the emotions of the target speaker. Similar to [4], when the target speaker is talking, the audio features are considered totally without any modification. However, when the "Doubled" strategy is applied, the audio features are doubled with one half to represent audio features from the target speaker and the other half for the interlocutor. In addition, Zhao et al. [5] propose a Facial expression interaction strategy. In this method, they ignore the interlocutor facial expression information when the speaker is silent, while extending the facial features to include that of the interlocutor's whenever the target subject speaks. Combing both strategies results in the following case scenarios: when the target subject speaks, inputs features are the target speaker's audio and facial features, as well as facial features of the interlocutor and zeros for the interlocutor audio features. And when the interlocutor speaks, the input features consists of the interlocutor audio features, zeros for interlocutor facial features, the target speaker facial features and zeros for the target speaker's audio features. Finally, textual features are treated the same way as audio features. They extracted acoustic features from VGGish [96], a CNN based model, instead of handcrafted features. They utilized the same visual and textual features adopted by [4]. The textual modality demonstrated direct influence of the liking prediction while acoustic and video feature solely affect the arousal and valence dimension of emotions.

3.3. Related Work on Aff-Wild Dataset

Dimitrios et al. [28] proposed a neural network architecture, AffWildNet, that was trained end-to-end and performed prediction of the emotional dimensions, arousal and valence, based on input faces extracted from the video recordings of the Aff-Wild dataset. Their architecture consisted of a CNN followed by an RNN. The CNN was responsible for extracting facial features while the RNN was used for modeling the temporal dynamics of the emotions. The model was trained on the GTX TITAN X GPU for 5 days. The CNN model was based on the VGG-Face [89] and ResNet-50 [61] models while the RNN consisted of 2 Gated Recurrent Unit (GRU) [97] layers. Their results show that VGG-Face outperforms ResNet-50. The proposed architecture outperforms other trained architectures on the same dataset. These models are the MM-NET [98], a variation of the ResNet architecture followed by multiple memory networks to model the temporal behavior of emotions. Finally, predictions from an ensemble of MM-NET models were combined for the final prediction. Another model named FATAUVA-Net [28]. This model consisted of convolutional layers, followed by one layer to extract facial features, then by another one to extract facial action units and a final layer to predict arousal and valence. Last model is based on the Inception-ResNet [99], named DRC-Net [100] is redesigned for the task of facial affect estimation.

Chapter 4. Proposed Method

Recently, numerous researchers have adopted an architecture composed of a deep CNN followed by RNN for modeling emotions across the arousal and valence dimensions. The best performance on the SEWA dataset has been achieved by the same group of researches over the past two years in [4] and later in their improved model in [5]. Dimitrios et al. [28] achieved the best overall performance on the AffWild dataset. Their architecture is composed of deep CNN model, mainly ResNet-50, followed by a two-layer GRU model. The state of the art on the arousal dimension of the RECOLA dataset was recorded in [109] which adopt a multi-task deep bidirectional LSTM-RNN (BLSTM-RNN).

Inspired by the aforementioned previous work, we implemented the architecture proposed by Dimitrios et al. [28]. Likewise, we implemented the state of the art model proposed by Zhao et al. [5] on the SEWA dataset. However, after Bai et al. [27] proved the superior performance of TCN compared to LSTM and GRU on different benchmarks and tasks, as discussed in Section 2.5, we adopt a TCN to replace the RNN. To the best of our knowledge, this is the first time a TCN is used for emotion prediction across the arousal and valence dimensions.

In the following sub-sections, we will present our detailed work and implementation for the TCN. First, we describe the employed audio, video, and textual features. Second, we propose our first contribution, memory-efficient backpropagation for recurrent neural network, which enabled us to implement previous work [28], originally applied on AffWild dataset, with the SEWA dataset. Third, we describe our implementation of TCN. Finally, we provide a thorough description of the application of LSTM and GRU from previous work [4][5][28], as well as discuss the application of Gaussian processes for hyperparameter optimization on all models (TCN, LSTM, and GRU).

4.1. Multimodal Features

4.1.1. Audio Features

We adopted the VGGish [96] features as the acoustic feature set similar to Zhao et al. [5] who achieved the state of the art on the SEWA dataset. The VGGish model is trained on a large scale dataset and is able to learn a rich audio representation. Extracted features represent short-term acoustic features. Features extraction takes place by first dividing the videos into overlapping video frames (i.e. video segments). The length of each video segment is set to 0.98 seconds. The overlapping window length is set to 100 ms. Each frame is then transformed into log-mel spectrogram features before feeding it to the VGGish model. Audio Embedding Features are then extracted from the last fully connected layer with a dimensionality of 128. Similar to [5], we refer to these features as "vggish.100ms". The term 100ms is used since the labels, arousal and valence, are provided, by annotators, every 100ms. Furthermore, to reduce the influence of the interlocutor on the target speaker, we adopt the same technique discussed in [4][5] by taking advantage of the turn information and zero-out acoustic features when the target speaker is silent. Hence improving the accuracy in the arousal dimension of emotions. Finally, after considering the interlocutor influence, the obtained audio features are named "vggish100ms.empty".

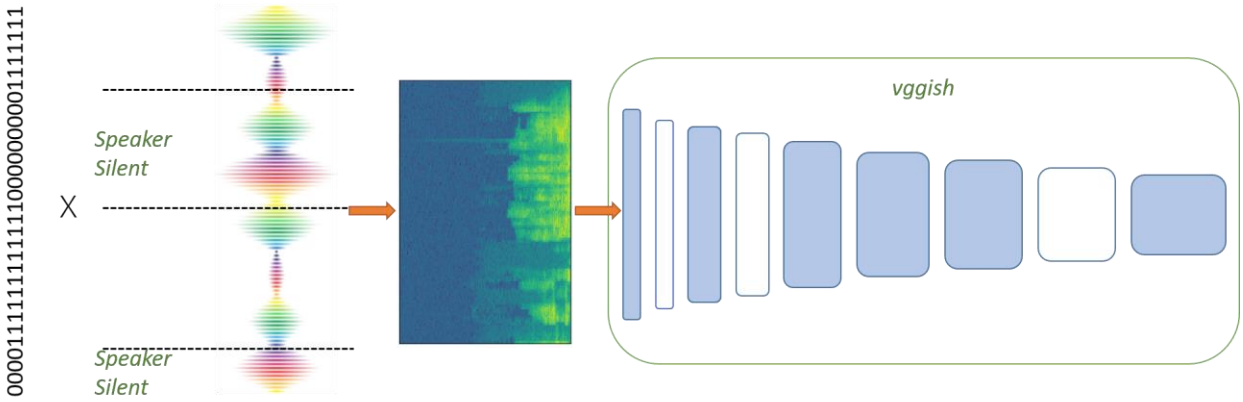

Figure 7 - Extracting Audio Features Pipeline

4.1.2. Word Features

Word features are textual features related to the semantic meaning of the word. Similar to Zhao et al. [5], we translate German text (speech transcription as described in the SEWA Section 2.2.5) into English using Google translator, then we extract word level features with 300 embedding dimensions using the Google English word embedding model [5], [46]. We apply mean pooling, or smoothing using the average, to each 5-word embedding features. Given that there is no time frame for each word, we repeat the extracted features within the time frame for which the target speaker is speaking before the interruption from the interlocutor. Previous work [4][5] showed that textual features were most conducive for the prediction of the liking dimension and add a minimal contribution to arousal and valence estimation. In this work, we conducted few experiments where we considered word features but then dropped it to obtain faster training process and to reduce the feature dimensionality, especially that we could not see a discernable improvement in the performance attribute to the inclusion of this modality.

4.1.3. Video features

Before extracting video features, we detect the faces using the Dlib [110] machine learning library. To better detect the subject's face in each video frame, per subject, we utilize the CNN-based face detector instead of the HOG-based detector from the Dlib library. Despite applying both face detection algorithm, we still failed to detect the face in a number of frames in each video in the dataset. After performing face detection, we align the faces using the Dlib after detecting the 5 facial landmarks that correspond to the nose, left-eye corner, right eye corner, left-mouth corner, and right-mouth corner.

To accelerate the training speed of neural networks, a batch normalization layer [111] can be applied to adjust and scale the input features to the activation unit in any hidden layers of the neural network. Thus, reducing the covariance shift according to [111]. The batch normalization layer consists of 2 trainable parameters named Gamma and Beta which determine the amount of scaling and shifting of the input features. However, normalizing features according to the mini-batch statistics is not desirable during inference. Therefore, normalization is done according to the population mean and standard deviation which is a running average over the mini-batch mean and standard deviation. As a result, the former will be affected whenever the mini-batch statistics are noisy, that is, contains a large number of black images (a black image is used whenever no face is detected within a frame). Therefore, increasing the training speed requires less black images, or the application of a high-performing face detection algorithm, which is why we use the CNN-based face detection algorithm from the Dlib library.

After the detection and alignment phase, we scale the frames to 112 by 112 pixels. Faces that are not detected, we replace the frame by a black image.

The state of the art [4][5] on the SEWA dataset extract deep visual features from the VGGFace. Therefore, we extract visual features from the VGG-Face [89] network after training the latter on the FER+ [90] dataset to classify emotions into 8 categories (Happy, surprise, anger, disgust, sad, fear, neutral, contempt). The VGG-Face model is composed of a set of convolutional layers of small filter sizes, interleaved with max pooling and dropout layers. The number of output feature maps after every convolutional layer doubles whenever its dimensions reduce by half. Our implemented VGG-Face model achieved 81.5% on the testing dataset of the FER+ testing dataset. Though VGG-Face predicts the 8 discrete emotions as opposed to arousal and valence, deep visual features contain useful facial information reflecting the facial emotional expression. Hence, we extract features from conv5 and conv6 layers as deep visual features. However, since conv5 generalizes better than conv6, we used conv5 features throughout the thesis. Hence, we name the extracted video features, vgg.conv5.

4.2. Memory-Efficient Backpropagation for Recurrent Neural Network

We would like to apply an LSTM/GRU-RNN on significantly long time series data to capture long-term dependencies. This task can be challenging given the memory limitation on most graphics processing unit.

An RNN can be truncated into several, typically equally sized, segments to relieve the training memory requirements [120][121]. The segment size can affect the calculation of the loss and gradients if they are calculated from each data segment independently as opposed to the entire sequence. Consequently, a small segment size can prevent the model from capturing long-term dependencies [122]. Conversely, a large segment size would better learn long-term dependencies at the expense of increased memory resource requirements. The segment size can be increased

until the number of segments is reduced to 1, in which case the RNN is no longer truncated and the memory resource requirements are the most stringent. The calculation of some loss functions, such as the CCC, require using the outputs for the entire data sequence. Hence, calculating the loss based on each segment separately is not feasible in this case.

The proposed method uses a truncated RNN training strategy. However, we introduce a mechanism to ensure that the calculation of the loss matches that of non-truncated RNNs. Furthermore, when we back propagate gradients from the final time step to time step 0, the sum of gradients with respect to the model parameters at time step 0 is equal to the sum of gradients we would obtain if the RNN was not truncated. Hence, the modeler can benefit from the less stringent memory requirement that a truncated RNN enables while achieving training results that reproduce those of a non-truncated RNN.

We summarize the proposed approach as follows:

- Step 1: Truncate the RNN into *m* segments of size *S*.
- Step 2: Feed the data points into the RNN and record predictions and the last hidden state for each segment. The last recorded hidden state in a segment is fed to the next segment. Concatenate predictions from all segments (See Figure 9).
- Step 3: Given the ground truth, compute the loss and the gradients of the loss with respect to the predictions (L_t and $\frac{\partial E_t}{\partial o_t}$ in Figure 9). Where L_t is the sum of E_t at each time step.

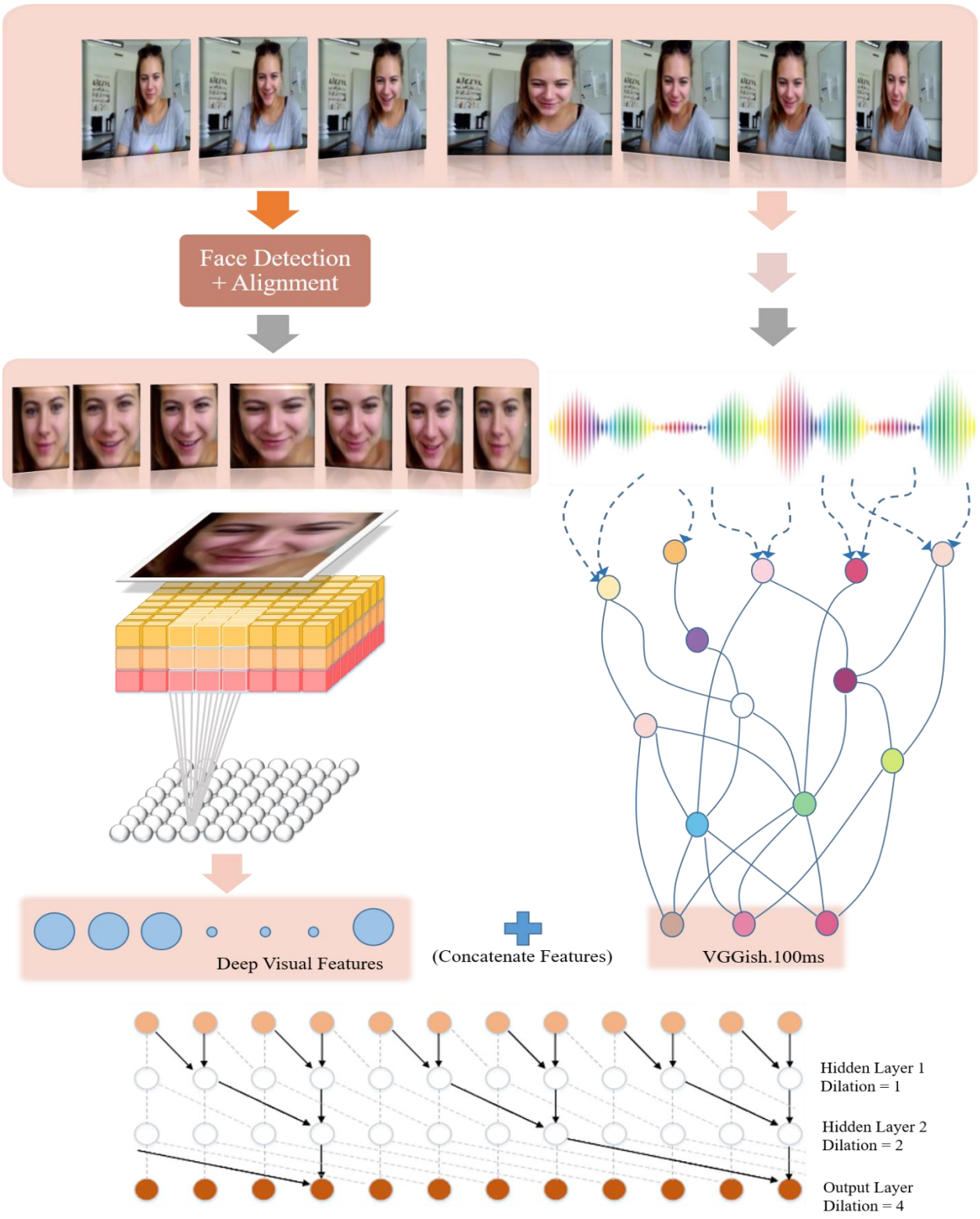

Figure 8 - Emotion detection procedure starting from feature extraction. First, faces are detected in each frame. Then, images are aligned according to the detected landmarks as mentioned before in the proposed method, video features section. Images are then fed into a deep CNN architecture to extract deep visual features. On the other hand, acoustic features are extracted in parallel using another Deep CNN architecture which in our case the Vggish model

- Step 4: Feed the data points into the RNN starting from segment *m* and traversing backward towards segment 0 while at each segment (See Figure 10):
  - A- Back propagate the recorded gradients of predictions with respect to the model parameters which we refer to as local parameters. Then, sum the local gradients according to equation (4):

$$\frac{\partial E}{\partial w} = \sum_{t=1}^{S} \frac{\partial E_t}{\partial w} = \sum_{t=1}^{S} \left(\frac{\partial E_t}{\partial o_t} * \frac{\partial o_t}{\partial h_t} * \left(\prod_{k=t+1}^{S} \frac{\partial h_k}{\partial h_{k-1}} \right) * \frac{\partial h_t}{\partial w} \right) \qquad (4)$$

    Where *E* is the error, *w* are the weights, *o* and *h* are the predictions and hidden states at time step *t*.
  - B- Back propagate the gradients of the last hidden state with respect to the model parameters using equation (5). We refer to these as the future gradients.

$$\frac{\partial h_S}{\partial w} = \sum_{t=0}^{S} \frac{\partial h_t}{\partial w} * \prod_{k=t+1}^{k=S} \frac{\partial h_k}{\partial h_{k-1}} \qquad (5)$$

  - C- Back propagate the gradients of the predictions with respect to the initial hidden states according to equation (6).

$$\frac{\partial E}{\partial h_0} = \sum_{t=0}^{S} \frac{\partial E_t}{\partial o_t} * \frac{\partial o_t}{\partial h_t} * \prod_{k=1}^{k=t} \frac{\partial h_k}{\partial h_{k-1}} \qquad (6)$$

  - D- Add local and future gradients to obtain the total gradients.
  - E- Back propagate the gradients from the last hidden state, if it exists, to the initial hidden state and add them to the gradients collected at C.
  - F- Repeat step 4 until we reach segment 0.
- Step 5: Apply the gradients to the model parameters.

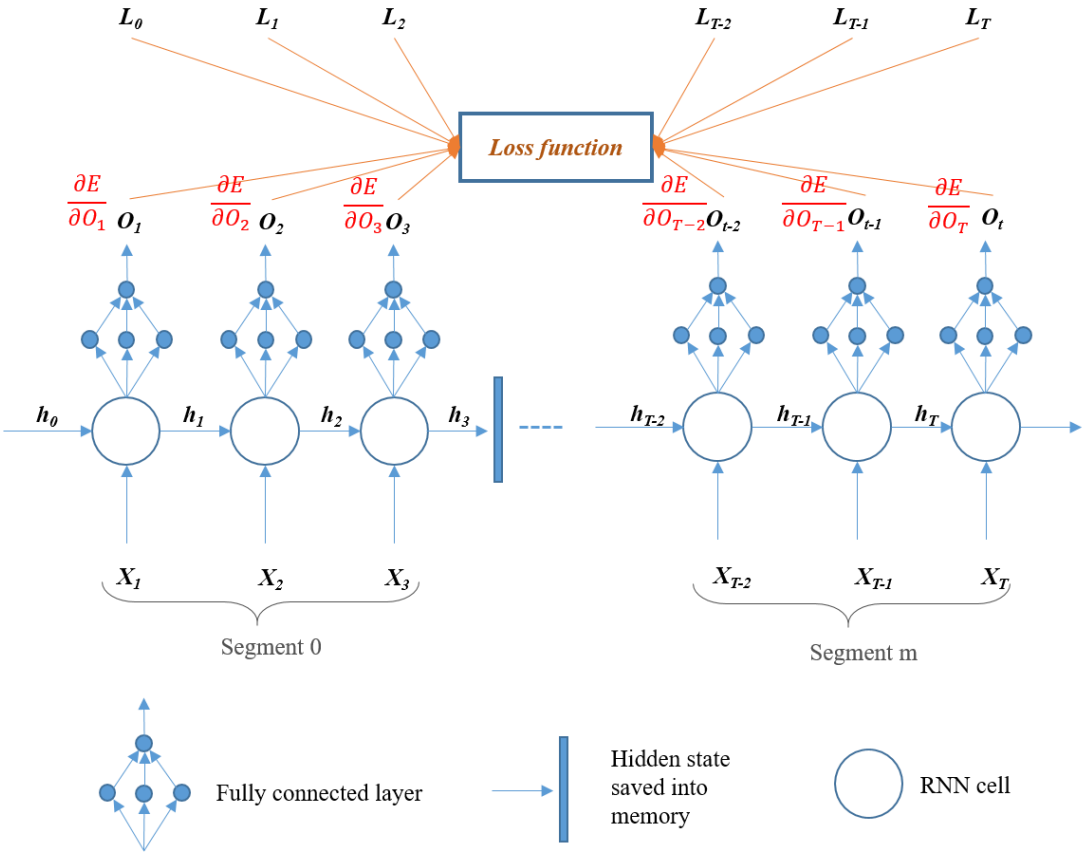

Figure 9 - Initial forward pass for the RNN. All inputs are fed sequentially starting from segment 0 to segment m, where the length of a segment is 3 time steps in this case. At the boundaries between segments, the hidden states are saved and used as the initial hidden state in the next iteration. Predictions are recorded and saved in memory. At the end, predictions, O_t, are concatenated and the partial derivative of the error E is computed with respect to the predictions given the Labels L_t.

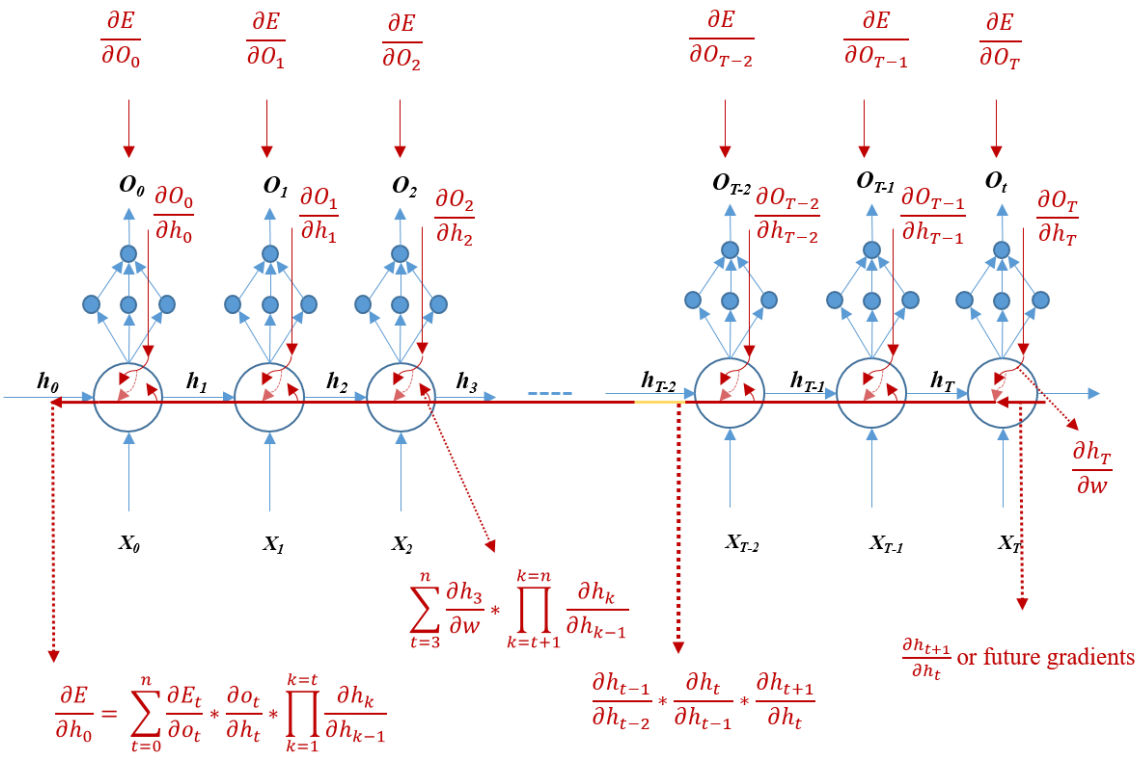

Figure 10 - Gradient propagation within one segment

We apply Memory-Efficient Backpropagation for Recurrent Neural Network in 2 ways. First, to achieve the same behavior while training a truncated RNN, gradients must not back propagate at some point across RNN cells. Generally, when an RNN is truncated into 100 time steps, or the "original" segment size is set to 100, and when the model cannot fit in memory especially if it contains a CNN before the RNN layer, we back propagate the gradients using the, memory-efficient back propagation technique, within the actual "original" segment (of size 100) by dividing the later into smaller sub-segments of size S' << 100. Therefore, we back propagate gradients within the original segment, but not across "original" segment**s**. We will refer to this procedure as Efficient-Truncated back propagation through time (E-TBPTT).

Secondly, to back propagate gradients across the whole data sequence, we divide it into m segments, each of size S', and back propagate gradients across the RNN cells of each segment and across the segments as well. We will refer to this method as Complete-BPTT (C-BPTT).

Inspired by AffWildNet, we tried out different CNN architectures, followed by a single GRU layer. In all cases, we set the cell size to 64 and we tried different CNN architectures mainly: ResNet-18, ResNet-24, Vgg-16 and a customized CNN model whose configurations is summarized in Table 2. This is done to test out the best CNN architecture to choose before modifying the GRU cell size and the number of layers used.

In our work, we trained a CNN+RNN model in an end to end manner using the E-TBPTT method but it did not converge even after few days of training. Probably the model requires more time to be trained given that AffWildNet was trained for 5 days. Therefore, we trained the AffWildNet and similar models to AffWildNet using C-BPTT which allows us to achieve convergence. All results are summarized in Table 10, Section 5.3.

Table 2 - Summary of our custom model used before the RNN

CNN – Custom 1
Conv2d (filters = 32, kernel_size = 5, strides = 2, padding= "same")
Relu
Max_pool2d (pool_size = 3, strides = 2, padding= "same"
Conv2d (filters = 64, kernel_size = 3, strides = 2, padding= "same")
Relu
Reshape (new_shape = [-1, feature_size])
Dense (units = 128)
Relu

4.3. Applying TCN in Unimodal setting

To study the effect of TCN in a unimodal setting, we trained a TCN on Vggish100ms, Vggish100ms.empty and Vgg.conv5 separately. After training few models to obtain a rough estimate of which ones work well for TCN, we obtained a model composed of 5 layers, the number

of feature maps in each layer is set to 125, the filter size is set to 5 and the dropout rate is set to 0.25. Finally, we applied hyperparameter optimization using Gaussian processes in an attempt to achieve higher accuracy. We discuss the results in Section 5.1.

4.4. Applying TCN in Multimodal Settings

4.4.1. Training TCN using Feature Level Fusion

After performing prediction on a single modality of features, we experimented with multimodal features. The later consists of audio and visual features (vggish100ms.empty and Vggface.conv5). In this case, we concatenated input features before feeding them into the model.

4.4.2. Training TCN under the Model Level Like-Fusion Paradigm

We split the TCN into 2 sub-models that join in the middle as shown in Figure 11. This view is close to the model level fusion technique where we allow both sub-models to train simultaneously on different feature sets while feeding their outputs (representations), after concatenating them, into a third level TCN model.

Additionally, we added the linguistic feature, word vectors features as described in the wordvec features in Section 4.1.2, to test out any opportunity to get higher accuracy. Whenever the textual modality is included, they are merged with audio features given the synchrony between both of them (audio features are only present when the target subject is speaking). We divided the parameters into three different sets. First, for the audio-related sub-model, we have the number of layers (*num_layers_audio*), number of filters per layer (*num_filters_audio*) and the filter size in each layer denoted as (*filter_size_audio*). Similarly, the second set consists of *num_layers_video*, *num_filters_video* and *filter_size_video*. Third set include, *num_layers_rest*, *num_filters_rest* and

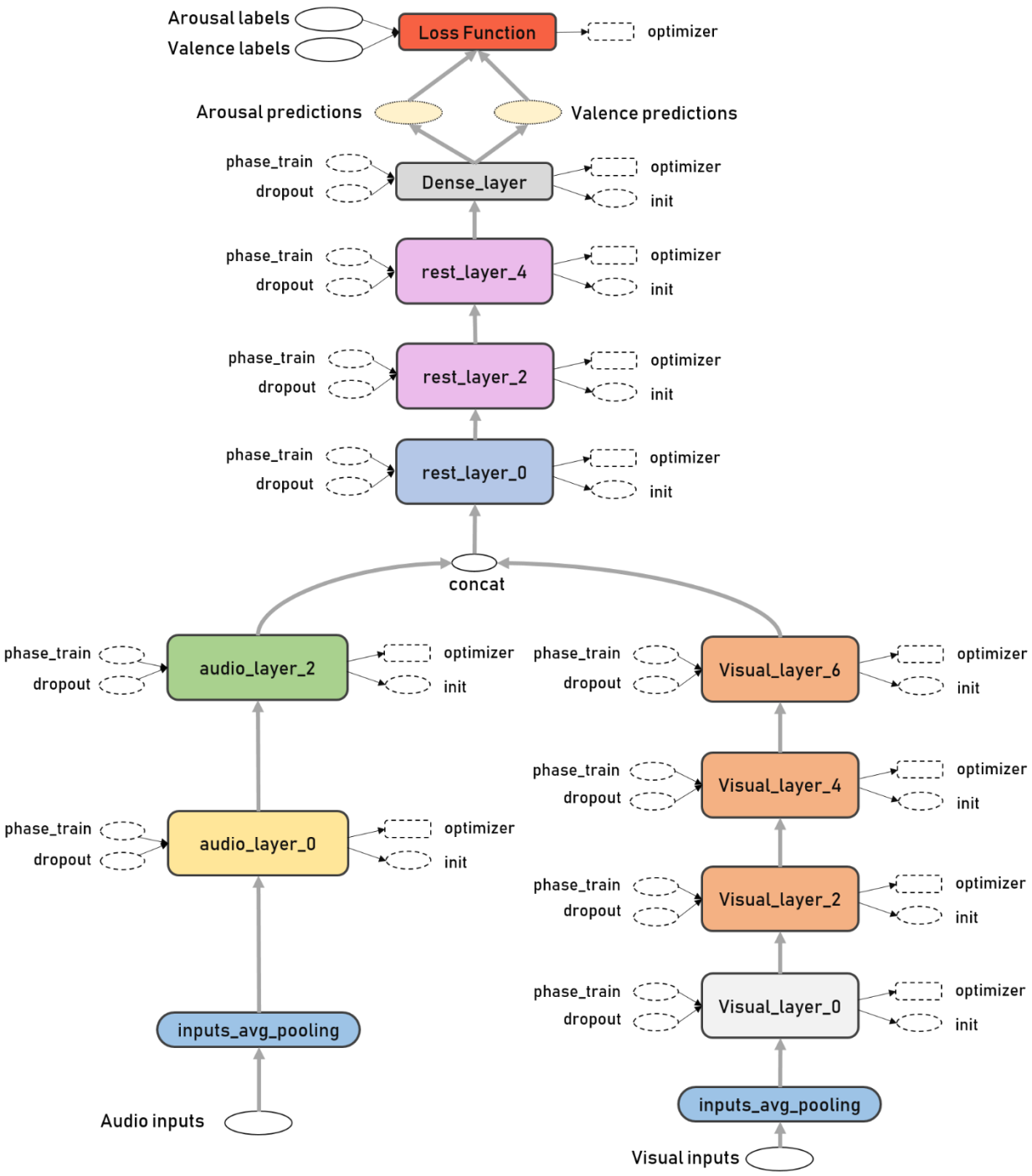

Figure 11 - Our TCN architecture following Decision Level Fusion. Layers are denoted by the curve edged rectangle. Nodes are denoted by the ellipse. Change in color take place when the input or the output dimension changes for specific layer. Layers having the same architecture have the same color. Arrows colored in gray represent tensor transfer between nodes and layers. Phase train is the place holder informing each layer whether we are in the train phase or not. Dropout is another placeholder defining the rate of the dropout in each layer. Numbers are even because the TCN consists of residual units where each contains 2 layers. The init node is responsible for initializing the weights in the layers.

filter_size_rest, where "rest" denotes for "the rest of the model". The final parameter is the dropout rate. Then we performed hyperparameter optimization using Gaussian processes while aiming to increase the average score on both valence and arousal dimensions. Tables 8 and 9 summarizes our obtained results.

4.4.3. Training LSTM and GRU model in Multimodal Setting

In order to perform a firm comparison between the TCN and RNN based time series models, we implemented the time series model which consists of an LSTM layer, along with the same pre-and-post-processing steps discussed by Zhao et al. [5]. Furthermore, we ran Gaussian processes to perform hyperparameter optimization on RNN models. Additionally, we tested out 2 more different models, one based on 2 layer GRU and the other is based on 1 layer GRU units. Generally, GRU layers are better than LSTM in many different time series prediction tasks [28][123] and are less complex and faster to train. The variable parameters are the cell size in each layer, the dropout rate and the learning rate. More formally, we compared first one layer GRU to one layer LSTM. Since the former outperforms the later, we implement 2 layer GRU and we report our results in Section 5.2.2.

All models (TCN and RNN) are implemented using the tensorflow framework. All models are trained for 120 epochs and optimized using the Adam Optimizer starting with a learning rate of 0.01. The learning rate is then reduced by half every 50 epochs. Predictions and input features are smoothed by averaging within a fixed window of length 5. To optimize hyperparameters, Gaussian processes were applied for each model setting and the training lasted for 3-4 days for each model separately.

Chapter 5. Results and Discussion

In this section, we compare our results, after applying TCN, to the state of the art methods on RECOLA, SEWA and Aff-Wild datasets on the arousal and valence dimensions. More formally, the state of the art on the SEWA and RECOLA dataset adopts the LSTM [4][5][72] as the time series model for regression, whereas the state of the art on Aff-wild adopts the GRU [28] as the time series model. Consequently, we compared our results with the state of the art on the SEWA dataset by Zhao et al. [5], likewise, we replaced the LSTM by 1 layer GRU, and then by 2 layer GRU similar to Dimitrios et al. [28]. We adopt the CCC for our performance metric. The results are reported in Sections 5.1 - 5.3.

In Section 5.4, we provide more insight into the advantage of adopting the memory-efficient backpropagation for RNN proposed in Section 4.2.

5.1. Results of TCN with unimodal features

In this section, we compare our achieved results after training the TCN on one feature modality at a time (in unimodal settings), to the results achieved by Zhao et al. [5].

Table 3 presents the CCC for the arousal and valence for two TCN-based networks that estimate emotions using the audio modality. We obtained model parameters through a trial and error process. The first network, which we call TCN_audio_1 uses the vggish.100ms features set while the second one, named TCN_audio_2, uses the vggish.100ms.empty features. We notice that taking into account the turn information, by utilizing the vggish100ms.empty, which reduces the effect of the interlocutor on the target speaker enhances the performance of both arousal and valence.

Table 3 - The Performance of Different TCN models with audio features on the arousal and valence dimensions of the SEWA dataset. The performance metric used is the CCC.

Model Name	Model Params	Features	Arousal (CCC)	Valence (CCC)
TCN_audio_1	- Num of layers: 5 - Num of filters: 125 - Filter size: 5 - Dropout: 0.25	Vggish.100ms	0.5520	0.5010
TCN_audio_2	- Num of layers: 5 - Num of filters: 125 - Filter size: 5 - Dropout: 0.25	Vggish100ms. empty	0.6262	0.5442

Table 4 - the Performance of Different TCN models with video features on the arousal and valence dimensions of the SEWA dataset. The performance metric used is the CCC.

Model Name	Model Params	Hyperparameter optimization applied?	Features	Arousal (CCC)	Valence (CCC)
TCN_video_1	- Num of layers: 5 - Num of filters: 125 - Filter size: 5 - Dropout: 0.25	No	Vgg.conv5	0.6657	0.6343
TCN_video_2	- Num of layers: 14 - Num of filters: 26 - Filter size: 2 - Dropout: 0.05	Yes	Vgg.conv5	0.6796	0.7353
TCN_video_3	- Num of layers: 12 - Num of filters: 29 - Filter size: 5 - Dropout: 0.05	Yes	Vgg.conv5	0.7356	0.6978

Table 4 presents the CCC for the arousal and valence for three TCN-based networks that estimate emotions using the video modality. All networks use the vgg.conv5 features. We resolved the model parameters of the first network, TCN_video_1, through a trial and error process, while we

adopted the Gaussian processes for hyperparameter optimization for TCN_video_2 and TCN_video_3. The hyperparameter optimization allowed us to improve the results that we obtained through the trial and error process.

In Table 5, we compare the most accurate audio-based unimodal network from Table 3, TCN_audio_2, to the corresponding LSTM-based network by Zhao et al. [5]. The proposed TCN architecture outperformers that of Zhao et al. [5] for the arousal and valence emotional dimensions.

Table 5 – Performance results of the best TCN model on the arousal and valence dimensions, while being fed with audio features, compared with the state of the art by Zhao et al. [5]. The performance metric used is the CCC.

Features	**TCN_audio_2**		**Zhao et al. [5]**	
	Arousal	Valence	Arousal	Valence
Vggish100ms.empty	0.6262	0.5442	0.6041	0.5104

In Table 6, we compare two of the video-based unimodal networks from Table 4 to the corresponding method by Zhao et al. [5]. Hence, we present the results of TCN_video_1 (the network we developed without hyperparameter optimization), and TCN_video_2 (the most accurate network we obtained after performing hyperparameter optimization with Gaussian processes). For TCN_video_1, we achieved a higher CCC compared to Zhao et al. [5] on the arousal dimension. However, we were not able to realize a superior result on the valence dimension. For TCN_video_2, we achieved a higher CCC compared to Zhao et al. [5] on both the arousal and valence dimensions.

Table 6 - Performance results of the best TCN model on the arousal and valence dimensions, while being fed with video features, compared with the state of the art by Zhao et al. [5]. The performance metric used is the CCC.

Features	**TCN_video_1**		**TCN_video_2**		**Zhao et al. [5]**	
	Arousal	Valence	Arousal	Valence	Arousal	Valence
Vgg.conv5	0.6657	0.6343	0.6796	0.7353	0.6224	0.7006

We should note that the reported results are achieved at the same epoch for all tested models.

Also, similar to previous findings [5][19][4][79], we achieved the best accuracy on the valence dimension with Vgg.conv5 features which is a visual feature (even before applying hyperparameter optimization).

5.2. Results of TCN with Multimodal features

5.2.1. Results Obtained Using Feature Level Fusion

In this section, we report our results on the arousal and valence dimension when training the TCN on multiple features combined. We use feature level fusion to fuse the input modalities.

Table 7 presents the CCC for the arousal and valence for two TCN-based networks that estimate emotions using audio and visual features combined. The first row presents TCN_f_1, a model with the same parameters as TCN_audio_1 (from Table 3). In this case, we used "f" to indicate that this model is trained using a feature level fusion technique. We obtain the TCN_f_1 model parameters through a trial and error process. However, we obtain the parameters of the second model, TCN_f_2, using Gaussian processes while performing hyperparameter optimization. As with our previous results in Table 4 for the unimodal networks, the hyperparameter optimization allows us to achieve a higher CCC for the arousal and valence dimensions.

Table 7 – TCN model configurations trained using feature level fusion.

Model Name	Model Params	Features	Hyperparameter Optimization applied?	Arousal (CCC)	Valence (CCC)
TCN_f_1	- Num of layers: 5 - Num of filters: 125 - Filter size: 5 - Dropout: 0.25	vggish100ms.empty + vgg.conv5	No	0.7763	0.7318
TCN_f_2	- Num of layers: 12 - Num of filters: 57 - Filter size: 2 - Dropout: 0.045	vggish100ms.empty + vgg.conv5	Yes	0.7819	0.7785

Table 8 - Comparison between the best models achieved while training using unimodal features and feature level fusion.

Model Name	Features	Feature Fusion Technique	Hyperparameter Optimization Applied?	Arousal (CCC)	Valence (CCC)
TCN_video_2	Vgg.conv5	Unimodal	Yes	0.6796	0.7353
TCN_audio_2	vggish100ms.empty	Unimodal	No	0.6262	0.5442
TCN_f_1	vggish100ms.empty + vgg.conv5	Feature Level Fusion	No	0.7763	0.7318
TCN_f_2	vggish100ms.empty + vgg.conv5	Feature Level Fusion	Yes	0.7819	0.7785

In Table 8, we compare the proposed unimodal and multimodal networks. We achieve better performance in the multimodal setting compared to the unimodal setting. This performance increase is expected since the input features dimension is increased. Even before applying

hyperparameter optimization (with TCN_f_1), we are able to achieve better results on the arousal dimension, but not on the valence dimension, compared to the models trained in the unimodal setting. TCN_audio_2 and TCN_f_1 are exactly the same model. However, after applying hyperparameter optimization using Gaussian processes, we obtain model TCN_f_2, which outperform TCN_f_1 and TCN_video_2 on both arousal and valence dimensions.

Table 9 - Performance of different TCN architectures trained in the Multimodal Settings. Models are trained using feature level fusion, i.e., features are concatenated before being fed to the model. The performance metric used is the CCC.

Features	**TCN_f_1**		**TCN_f_2**		**Zhao et al. [5]**	
	Arousal (CCC)	Valence (CCC)	Arousal (CCC)	Valence (CCC)	Arousal (CCC)	Valence (CCC)
vggish100ms.empty + vgg.conv5	0.7763	0.7318	0.7819	0.7785	0.7692	0.7599

In Table 9, we compare our multimodal methods that employ feature level fusion to the corresponding models presented by Zhao et al. [5]. Before applying hyperparameter optimization, TCN_f_1, outperformed Zhao et al. [5] in the arousal dimension, but not in valence. However, after applying Gaussian processes for hyperparameter optimization, we outperformed the state-of-the-art in both dimensions.

Generally, after analyzing the performance measure on the arousal and valence dimensions while training with Gaussian processes, we found that there is a tradeoff in the performance of any model on the arousal and valence dimensions. Increasing the accuracy of prediction of one emotional dimension detrimentally affects the other. Hence, feature level fusion may not be the best

approach. Performing late fusion or model level like-fusion is a better alternative. For this reason, we split the TCN model and adopt the approach discussed in Section 4.4.2.

5.2.2. Results Obtained Using Model level Fusion

In this section, we present the results that we achieved after training the TCN model using the model level fusion approach, which depends on two TCN sub-models followed by another TCN sub-model for the final prediction of arousal and valence. In table 10, we present several network architectures, whose model parameters are obtained after applying hyperparameter optimization using Gaussian processes. TCN_m_1 and TCN_m_2 are models that are fed with video, audio and textual features. However, the remaining models are fed with audio and video features only. We eliminated textual features to obtain faster training time while performing hyperparameter optimization.

As we can see in Table 10, we achieved the highest accuracy on the valence dimension only when accounting for the wordvec features. Clearly, textual features include useful information that improves the accuracy of the valence dimension. Perhaps, affect bursts [92] such as, laughing, grimacing, frowning, etc., reflect effectively the underlying emotions, as was noted in [92].

Even though wordvec features showed enhanced performance on the valence dimension, we decided to reduce the input dimensionality for faster training of TCN_m_3. For TCN_m_3, we obtained better results in the arousal dimension while the valence CCC dropped slightly. However, TCN_m_3 is a simpler model compared to TCN_m_1 and TNC_m_2 as it only considers the audio and video modality.

Since we do not have access to the testing partition of the SEWA dataset, this imposes a limitation on the work presented in this thesis. Therefore, if performing hyperparameter optimization on the

validation dataset leads to overfitting the validation dataset, then that should be the case with RNN based models as well. Consequently, we attempt to remove the TCN model and replace it with RNN based model. From Table 10, first we compared to 1 layer LSTM (similar to the Zhao et al. [5] architecture) 1 layer GRU. Since GRU model outperform LSTM, we trained a 2 layer GRU model (similar to Dimitrios et al. [28]). The 1 layer LSTM, GRU and the 2 layer GRU are all trained while applying hyperparameter optimization. Therefore, if we are overfitting the validation dataset with TCN, then we should be overfitting the validation dataset with RNN based models. However, the achieved results did not outperform the TCN.

Table 10 - TCN model trained with a model level fusion setting. Multimodal features are utilized and the models are configured using Gaussian processes. Models are trained using a model level fusion approach. The performance metric used is the CCC.

Model Name	**Model params**	**Features**	**Arousal (CCC)**	**Valence (CCC)**
TCN_m_1	- *num_layers_audio*: 2 - *num_filters_audio*: 16 - *filter_size_audio*: 6 - *num_layers_video:10* - *num_filters_video:16* - *filter_size_video:2* - *num_layers_rest:10* - *num_filters_rest: 100* - *filter_size_rest: 6* - *Dropout*: 0.01	vggish100ms.empty + vgg.conv5 + English-wordvec	0.7704	0.7895
TCN_m_2	- *num_layers_audio*: 2 - *num_filters_audio*: 130 - *filter_size_audio*: 5 - *num_layers_video:10* - *num_filters_video:83* - *filter_size_video:3* - *num_layers_rest:5* - *num_filters_rest: 30* - *filter_size_rest: 6* - *Dropout*: 0.028656	vggish100ms.empty + vgg.conv5 + English-wordvec	0.8168	0.7809

TCN_m_3	- *num_layers_audio*: 2 - *num_filters_audio*: 105 - *filter_size_audio*: 2 - *num_layers_video:8* - *num_filters_video:43* - *filter_size_video:4* - *num_layers_rest:8* - *num_filters_rest: 138* - *filter_size_rest: 4* - *Dropout*: 0.1718	vggish100ms.empty + vgg.conv5	0.8207	0.7739
2 Layer GRU	- *cell_size_1:312* - *cell_size_2: 610* - *dropout: 0.0954* - *learning_rate: 0.0001*	vggish100ms.empty + vgg.conv5	0.7357	0.7542
1 Layer GRU	- *cell_size:505* - *dropout: 0.0858* - *learning_rate: 0.00047*	vggish100ms.empty + vgg.conv5	0.7076	0.7544
1 Layer LSTM	- *cell_size: 120* - *num_time_steps: 100* - *dropout: 0.5* - *learning_rate: 0.01*	vggish100ms.empty + vgg.conv5	0.6589	0.6853

Table 11 - Comparison of performance of different TCN models trained using feature level fusion and model level like-fusion

Model Name	**Features**	**Arousal (CCC)**	**Valence (CCC)**
TCN_f_2	vggish100ms.empty + vgg.conv5	0.7819	0.7785
TCN_m_1	vggish100ms.empty + vgg.conv5 + English-wordvec	0.7704	0.7895
TCN_m_2	vggish100ms.empty + vgg.conv5 + English-wordvec	0.8168	0.7809
TCN_m_3	vggish100ms.empty + vgg.conv5	0.8207	0.7739

In Table 11, we compare the best performing network that employs feature level fusion, TCN_f_2 to the model level fusion networks. TCN_m_3 significantly outperformed TCN_f_2 on the arousal dimension. However, they achieved comparable accuracy on the valence dimension. Therefore, training TCN using model level like-fusion is advantageous compared to feature level fusion for emotion estimation. Consequently, we should distinguish between the audio and the video deep learned representations which play a significant role in enhancing the performance. Probably, both visual and audio features should experience some transformation before concatenating them. Also, when performing early level fusion, the first few convolutional layers will be consuming directly visual and audio features. Thus, it is necessary to distinguish between both features and apply few transformation layers before concatenating the features.

Table 12 - Reported Results of TCN trained using model level fusion against the state of the art by Zhao et al. [5]. The performance metric used is the CCC. Input features are video, audio and textual features.

Features	**TCN_m_1**		**TCN_m_2**		**Zhao et al. [5]**	
	Arousal	Valence	Arousal	Valence	Arousal	Valence
vggish100ms.empty + vgg.conv5 + English-wordvec	0.7704	0.7895	0.8168	0.7809	0.7914	0.7823

Table 13 - Reported Results of TCN trained using model level fusion against the state of the art by Zhao et al. [5]. The performance metric used is the CCC. Input features are video and audio features.

Features	**TCN_m_3**		**Zhao et al. [5]**	
	Arousal	Valence	Arousal	Valence
vggish100ms.empty + vgg.conv5	0.8207	0.7739	0.7692	0.7599

In Table 12, we compare TCN_m_1 and TCN_m_1 to the multimodal network that uses audio, video and text features by Zhao et al. [5]. TCN_m_1 outperforms Zhao et al. [5] for the valence, but not the arousal dimension. However, TCN_m_2 outperformed Zhao et al. [5] on the arousal dimension. Additionally, in Table 13, we compare our bimodal network TCN_m_3 which uses audio and video features to that of Zhao et al. [5]. TCN_m_3 outperforms Zhao et al. [5] in both arousal and valence dimension. Moreover, TCN_m_3 outperforms TCN_m_1 in the arousal dimension, and TCN_m_2 in the valence dimension. However, none of the TCN models outperformed Zhao et al. [5] on both arousal and valence simultaneously. One possible solution could involve dividing the audio TCN sub-model into 2 separate TCN sub-models, one for audio and the other one for textual features. As a result, we end up having three different TCN sub-models followed by a final TCN sub-model.

Applying dilation convolution in the TCN allows us to model long-term dependency. Generally, the effective history length achieved without dilation is equal to the number of layers. For instance, a TCN without dilation, of depth 10 will have an effective history of size 10. However, with dilation, a TCN of depth 10 will reach an effective history of size $(k - 1) * d$, where d is the dilation rate and k is the filter size. Hence, having a filter size of 3, and $d = 2^{10}$, the effective history will be equal to 3072.

We found that using only audio and visual features, TCN models outperform any RNN based model, even those trained while performing hyperparameter optimization. Hence, we suggest adopting TCN as the future baseline time series model in emotion detection tasks.

5.3. Results of Training AffWildNet Using Memory-Efficient Backpropagation for Recurrent Neural Networks

As discussed in Section 3.3, the AffWildNet [28] achieved the highest accuracy on the AffWild dataset. This model is composed of a VGG-Face or ResNet models followed by a 2 layer GRU for the arousal and valence prediction. Hence, we aimed at training the same model on the SEWA dataset. In [28], the batch size is set to 4. However, since the SEWA database consists of 34 subjects in the training dataset, we set our batch size to 34. Nonetheless, this imposes a stringent requirement on the number of time steps, segment size *S*, which is set to 80 in [28]. In our case, with a batch size of 34, the maximum number of time steps allowed is less than 10 to be able to fit the model into memory. Therefore, using C-BPTT, we were able to train the AffWildNet on the whole dataset. Table 14 summaries our finding based on different CNN architectures tried out.

Table 14 - Performance of different CNN architectures while utilizing 1 layer GRU

Model Name	**CCC (Arousal)**	**CCC (Valence)**
CNN - Custom 1	0.4611	0.5279
ResNet-18	0.5441	0.4902
ResNet-34	0.3737	0.5059
Vgg-16	0.5	0

In this experiment, we were not able to perform hyperparameter optimization using Gaussian Processes since training a single model under one configuration settings takes more than one day to converge. While training, the shallow network, CNN – Custom 1 was fast to train, around 14 hours. ResNet-18 took 1 day to converge and ResNet-34 took 2 days. However, when we adopt Vgg-16, our model required 3 days to start converging on the arousal dimension but never converged on the valence dimension. However, we noticed a slight improvement in the valence

dimension after day 4 before we concluded the training. Therefore, maybe setting a longer training time might enable us to train this model.

Again, our realization in Section 5.2.2. is confirmed in this section. Even training a CNN model followed by a 2 layer GRU model in an end to end manner did not result in a model that outperforms the TCN one. Additionally, training the AffWildNet requires days of training which impedes the application of hyperparameter optimization.

5.4. Memory-Efficient Backpropagation for Recurrent Neural Networks Evaluation and Results

To evaluate our method, we build a 2 layer LSTM model followed by a fully connected layer to predict the two dimensions of emotions. The model is trained using the Adam optimizer for 100 epochs. The metric used for evaluating the model is the CCC whose inverse is employed to calculate the loss during training.

We trained 3 different models (2 layer LSRM), truncated into various segment sizes and compare them to a non-truncated 2 layer LSTM in terms of the accuracy, execution time, and memory consumption during training. Hence, the objective of the evaluation is to:

1. Illustrate the relationship between memory consumption and execution time during training; and

2. Demonstrate that all compared models achieve the same accuracy at the end of each epoch and thus exhibit identical training behavior.

As a case study, we will employ an affective computing problem to demonstrate the performance of the proposed method. However, our training technique can be used to train RNN models corresponding to any other domain.

In Figure 12, we depict the CCC accuracy plotted against the number of epochs for three models that estimate arousal (Figure 12a), valence (Figure 12b). The first model is a truncated RNN with a segment size of 2-time steps. The second model is a truncated RNN with a segment size of 100 time steps. We apply the proposed method for training these truncated models. The third model is a non-truncated RNN that considers the entire data sequence (which is composed of 1500 time steps). Weight initialization for all three models is set to glorot normal initializer, and the initiale seed is set to "123" for all models. The results in Figure 12 are rendered using tensorboard. Although we are training three models, the CCC curves are fully overlapping across valence and arousal. This demonstrates that the three models are behaving identically during training.

Figure 13 illustrates the relationship between the time and memory requirement as the number of segment size decreases from 1500 to 2. Note that a segment size of 1500 corresponds to a non-truncated RNN as a single segment covers the entire data sequence. The inverse relationship between the memory consumption and execution time during training is evident. This tradeoff gives modelers the flexibility of reducing memory consumption at the cost of increasing the training time without limiting their ability to capture long term dependencies.

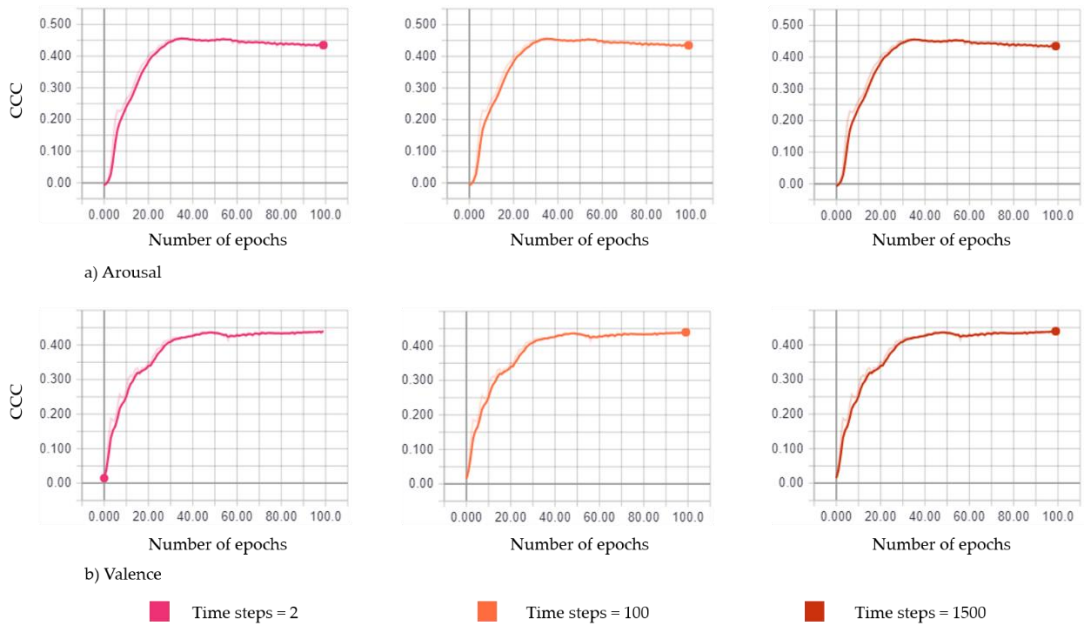

Figure 12 - The accuracy of Arousal and Valence when segment size is 2, 100 and 1500 time steps; given the same random initialization and random seed for all models.

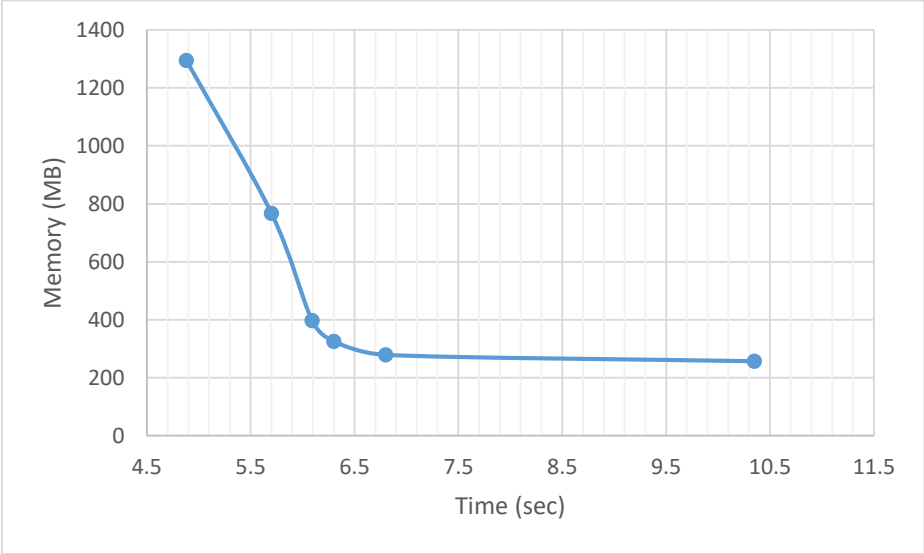

Figure 13 - This figure shows the memory vs. time requirement , displayed on y-axis, for training different RNN models unrolled into different segment sizes, displayed on x-axis

Chapter 6. Conclusion and Future Work

In this chapter, we finalize our work with the conclusion and future work.

6.1. Conclusion

The goal of this thesis was to improve on the emotion prediction task across the valence and arousal dimensions. This problem is of high complexity due to the difficulty of estimating spontaneous emotions compared to acted or induced ones.

We extracted audio, visual, and textual features from deep neural network models, and performed time series modeling for feature fusion and regression through the temporal convolution neural network instead of RNN-LSTM, the state of the art regression and time series model for emotion prediction. We summarize our contributions as follows:

1- The ability to train recurrent neural networks to capture long term dependencies. Models that were unable to fit in memory can now be trained while setting the number of time steps to be equal the size of sequential data.

2- We adopt the temporal convolution neural network as model based fusion technique which replaces the previous state-of-the-art time series models, RNN-LSTM, on emotion prediction tasks. Thus, we use convolutions for modeling time series data and for regression. Our work outperforms the state-of-the-art as reported in Sections 5.1-5.3

3- We apply Gaussian processes and Bayesian inference to perform hyperparameter optimization on the validation dataset of the SEWA dataset, for TCN models and other RNN based models. This enables us to perform a fair comparison between alternative models.

Finally, we were able to achieve better results with our TCN based model on the arousal and valence dimensions compared to RNN models. Hence, we recommend adopting TCN as the baseline model for emotion prediction in future research in the field.

6.2. Future Work

We summarize our future work as follows:

1- Applying different time series models such as the Wavenet [119] for the emotion prediction task. If TCN resembles ResNet models, then Wavenet resembles DenseNet models, a CNN based model that outperformed ResNets [102].

2- Improving the residual unit in TCN given its importance in the ResNet model. For instance, we can adopt the Wide-ResNet residual unit.

3- Utilizing unsupervised deep learning based features for emotion prediction similar to [125]. For instance, a variational auto encoder may be able to detect facial action units as latent variable after being trained on massive unlabeled dataset such as YouTube videos [126]. As a result, this may enable us to achieve better results on small datasets.

4- Improving on the natural language processing we performed to extract emotions from text more effectively.

5- Performing data augmentation by: 1) training time series models on overlapping video segments cropped from the original videos provided with the dataset, and 2) merging more datasets.

References

[1] R. W. Picard, “Affective Computing for HCI.,” in *HCI (1),* 1999, pp. 829–833.

[2] S. Marsella and J. Gratch, “Computationally modeling human emotion,” *Commun. ACM*, vol. 57, no. 12, pp. 56–67, 2014.

[3] B. Schuller, M. Valstar, F. Eyben, G. Mckeown, R. Cowie, and M. Pantic, “AVEC 2011-The First International Audio/Visual Emotion Challenge ⋆.”

[4] S. Chen, Q. Jin, J. Zhao, and S. Wang, “Multimodal Multi-task Learning for Dimensional and Continuous Emotion Recognition,” pp. 19–26, 2017.

[5] J. Zhao, R. Li, S. Chen, and Q. Jin, “Multi-modal Multi-cultural Dimensional Continues Emotion Recognition in Dyadic Interactions,” in *Proceedings of the 2018 on Audio/Visual Emotion Challenge and Workshop*, 2018, pp. 65–72.

[6] M. Wöllmer *et al.*, “Abandoning emotion classes - Towards continuous emotion recognition with modelling of long-range dependencies,” in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2008, pp. 597–600.

[7] Fabien Ringeval and Björn Schuller and Michel Valstar and Roddy Cowie and Heysem Kaya and Maximilian Schmitt and Shahin Amiriparian and Nicholas Cummins and Denis Lalanne and Adrien Michaud and Elvan Çiftçi and Hüseyin Güleç and Albert Ali Salah and Maja, “AVEC 2018 Workshop and Challenge: Bipolar Disorder and Cross-Cultural Affect Recognition,” in *Proceedings of the 8th International Workshop on Audio/Visual Emotion Challenge, AVEC’18, co-located with the 26th ACM International Conference on Multimedia, MM 2018*, 2018.

[8] R. W. Picard, “Affective computing MIT press,” *Cambridge, Massachsusetts*, 1997.

[9] M.-I. Georgescu, R. T. Ionescu, and M. Popescu, “Local learning with deep and handcrafted features for facial expression recognition,” *arXiv Prepr. arXiv1804.10892*, 2018.

[10] E. Barsoum, C. Zhang, C. Canton Ferrer, and Z. Zhang, “Training Deep Networks for

Facial Expression Recognition with Crowd-Sourced Label Distribution," in *ACM International Conference on Multimodal Interaction (ICMI)*, 2016

[11] T. Connie, M. Al-Shabi, W. P. Cheah, and M. Goh, "Facial expression recognition using a hybrid CNN--SIFT aggregator," in *International Workshop on Multi-disciplinary Trends in Artificial Intelligence*, 2017, pp. 139–149.

[12] H. Al Osman and T. H. Falk, "Multimodal affect recognition: Current approaches and challenges," *Emot. Atten. Recognit. Based Biol. Signals Images*, pp. 59–86, 2017.

[13] D. Lahat, T. Adali, and C. Jutten, "Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects," *Proc. IEEE*, vol. 103, no. 9, pp. 1449–1477, Sep. 2015.

[14] K. Brady *et al.*, "Multi-modal audio, video and physiological sensor learning for continuous emotion prediction," in *Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge*, 2016, pp. 97–104.

[15] B. Sun, S. Cao, L. Li, J. He, and L. Yu, "Exploring multimodal visual features for continuous affect recognition," in *Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge*, 2016, pp. 83–88

[16] S. Chen and Q. Jin, "Multi-modal conditional attention fusion for dimensional emotion prediction," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 571–575.

[17] A. Metallinou, A. Katsamanis, and S. Narayanan, "A hierarchical framework for modeling multimodality and emotional evolution in affective dialogs," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 2401–2404.

[18] V. Rozgić, S. Ananthakrishnan, S. Saleem, R. Kumar, A. N. Vembu, and R. Prasad, "Emotion recognition using acoustic and lexical features," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012

[19] J. Huang, Y. Li, J. Tao, Z. Lian, M. Niu, and M. Yang, "Multimodal Continuous Emotion Recognition with Data Augmentation Using Recurrent Neural Networks," in *Proceedings of the 2018 on Audio/Visual Emotion Challenge and Workshop*, 2018, pp. 57–64.

[20] C.-H. Wu, J.-C. Lin, and W.-L. Wei, “Survey on audiovisual emotion recognition: databases, features, and data fusion strategies,” *APSIPA Trans. signal Inf. Process.*, vol. 3, 2014.

[21] Z. Wu, Y.-G. Jiang, J. Wang, J. Pu, and X. Xue, “Exploring inter-feature and inter-class relationships with deep neural networks for video classification,” in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 167–176

[22] J. Chen, Z. Chen, Z. Chi, and H. Fu, “Emotion recognition in the wild with feature fusion and multiple kernel learning,” in *Proceedings of the 16th International Conference on Multimodal Interaction*, 2014, pp. 508–513

[23] K. Lu and Y. Jia, “Audio-visual emotion recognition with boosted coupled HMM,” in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, 2012, pp. 1148–1151.

[24] B. Schuller, M. Valstar, F. Eyben, R. Cowie, and M. Pantic, “AVEC 2012-The Continuous Audio/Visual Emotion Challenge,” 2012.

[25] L. Chao, J. Tao, M. Yang, Y. Li, and Z. Wen, “Multi-scale temporal modeling for dimensional emotion recognition in video,” in *Proceedings of the 4th International Workshop on Audio/Visual Emotion Challenge*, 2014, pp. 11–18

[26] M. Wöllmer, M. Kaiser, F. Eyben, B. Schuller, and G. Rigoll, “LSTM-Modeling of continuous emotions in an audiovisual affect recognition framework,” *Image Vis. Comput.*, vol. 31, no. 2, pp. 153–163, 2013.

[27] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *arXiv Prepr. arXiv1803.01271*, 2018.

[28] D. Kollias *et al.*, “Deep affect prediction in-the-wild: Aff-wild database and challenge, deep architectures, and beyond,” *Int. J. Comput. Vis.*, pp. 1–23, 2019

[29] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, 2013.

[30] N. Cummins, S. Amiriparian, G. Hagerer, A. Batliner, S. Steidl, and B. W. Schuller, “An image-based deep spectrum feature representation for the recognition of emotional speech,” in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 478–484

[31] N. Cummins, S. Amiriparian, S. Ottl, M. Gerczuk, M. Schmitt, and B. Schuller, “Multimodal Bag-of-Words for cross domains sentiment analysis,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4954–4958.

[32] F. Ringeval *et al.*, “Avec 2017: Real-life depression, and affect recognition workshop and challenge,” in *Proceedings of the 7th Annual Workshop on Audio/Visual Emotion Challenge*, 2017, pp. 3–9.

[33] F. Povolny *et al.*, “Multimodal emotion recognition for AVEC 2016 challenge,” in *Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge*, 2016, pp. 75–82

[34] F. Eyben, M. Wöllmer, and B. Schuller, “Opensmile: the munich versatile and fast open-source audio feature extractor,” in *Proceedings of the 18th ACM international conference on Multimedia*, 2010, pp. 1459–1462

[35] F. Eyben *et al.*, “The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing,” *IEEE Trans. Affect. Comput.*, vol. 7, pp. 190--202, 2015.

[36] F. Ringeval *et al.*, “Automatic Analysis of Typical and Atypical Encoding of Spontaneous Emotion in the Voice of Children,” in *Proceedings INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association (ISCA)*, 2016, pp. 1210–1214.

[37] F. Ringeval, E. Marchi, M. Mehu, K. Scherer, and B. Schuller, “Face Reading from Speech-Predicting Facial Action Units from Audio Cues,” in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015

[38] M. Schmitt and B. Schuller, “OpenXBOW: introducing the passau open-source

crossmodal bag-of-words toolkit," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 3370–3374, 2017.

[39] M. Valstar *et al.*, "AVEC 2013-The Continuous Audio/Visual Emotion and Depression Recognition Challenge," in *Proceedings of the 3rd ACM international workshop on Audio/visual emotion challenge*, 2013, pp. 3--10.

[40] L. Chao, J. Tao, M. Yang, Y. Li, and Z. Wen, "Long short term memory recurrent neural network based multimodal dimensional emotion recognition," in *Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge*, 2015, pp. 65–72.

[41] M. Valstar *et al.*, "AVEC 2016-Depression, Mood, and Emotion Recognition Workshop and Challenge," in *Proceedings of the 6th international workshop on audio/visual emotion challenge*, 2016, pp. 3--10.

[42] F. Ringeval *et al.*, "AVEC 2018 workshop and challenge: Bipolar disorder and cross-cultural affect recognition," in *Proceedings of the 2018 on Audio/Visual Emotion Challenge and Workshop*, 2018, pp. 3–13.

[43] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "Openface: an open source facial behavior analysis toolkit," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016, pp. 1–10.

[44] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, "Incremental face alignment in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1859–1866.

[45] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[46] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv Prepr. arXiv1301.3781*, 2013.

[47] F. Ringeval, A. Sonderegger, J. Sauer, and D. Lalanne, "Introducing the RECOLA multimodal corpus of remote collaborative and affective interactions," in *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*,

2013, pp. 1–8.

[48] A. Dhall, R. Goecke, S. Ghosh, J. Joshi, J. Hoey, and T. Gedeon, “From individual to group-level emotion recognition: EmotiW 5.0,” in *Proceedings of the 19th ACM international conference on multimodal interaction*, 2017, pp. 524–528.

[49] A. Dhall, R. Goecke, J. Joshi, J. Hoey, and T. Gedeon, “Emotiw 2016: Video and group-level emotion recognition challenges,” in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, 2016, pp. 427–432.

[50] A. Dhall, R. Goecke, J. Joshi, K. Sikka, and T. Gedeon, “Emotion recognition in the wild challenge 2014: Baseline, data and protocol,” in *Proceedings of the 16th international conference on multimodal interaction*, 2014, pp. 461–466

[51] S. Zafeiriou, D. Kollias, M. A. Nicolaou, A. Papaioannou, G. Zhao, and I. Kotsia, “Aff-Wild: Valence and Arousal’In-The-Wild’Challenge,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 34–41.

[52] D. Kollias, M. A. Nicolaou, I. Kotsia, G. Zhao, and S. Zafeiriou, “Recognition of affect in the wild using deep neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 26–33.

[53] H. A. Elfenbein and N. Ambady, “On the universality and cultural specificity of emotion recognition: a meta-analysis.,” *Psychol. Bull.*, vol. 128, no. 2, p. 203, 2002.

[54] A. Esposito, A. M. Esposito, and C. Vogel, “Needs and challenges in human computer interaction for processing social emotional information,” *Pattern Recognit. Lett.*, vol. 66, pp. 41–51, 2015.

[55] F. Ringeval *et al.*, “Pattern Recognition Letters Prediction of Asynchronous Dimensional Emotion Ratings from Audiovisual and Physiological Data,” *Pattern Recognit. Lett.*, vol. 66, pp. 22–30, 2015.

[56] S. Ebrahimi Kahou, V. Michalski, K. Konda, R. Memisevic, and C. Pal, “Recurrent neural networks for emotion recognition in video,” in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, 2015, pp. 467–474.

[57] D. E. Holmes and L. C. Jain, *Innovations in machine learning*. Springer, 2006.

[58] F. Grézl, E. Egorova, and M. Karafiát, "Further investigation into multilingual training and adaptation of stacked bottle-neck neural network structure," in *2014 IEEE Spoken Language Technology Workshop (SLT)*, 2014, pp. 48–53.

[59] Y. Wu, T. Hassner, K. Kim, G. Medioni, and P. Natarajan, "Facial landmark detection with tweaked convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3067–3074, 2018.

[60] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[62] K. Somandepalli, R. Gupta, M. Nasir, B. M. Booth, S. Lee, and S. S. Narayanan, "Online affect tracking with multimodal kalman filters," in *Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge*, 2016, pp. 59–66.

[63] M. A. Nicolaou, H. Gunes, and M. Pantic, "Continuous prediction of spontaneous affect from multiple cues and modalities in valence-arousal space," *IEEE Trans. Affect. Comput.*, vol. 2, no. 2, pp. 92–105, 2011.

[64] B. Sun *et al.*, "Combining multimodal features within a fusion network for emotion recognition in the wild," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, 2015, pp. 497–502.

[65] B. Sun, L. Li, G. Zhou, and J. He, "Facial expression recognition in the wild based on multimodal texture features," *J. Electron. Imaging*, vol. 25, no. 6, p. 61407, 2016.

[66] M. Yang, J. Tao, Z. Wen, Y. Li, and L. Chao, "Long Short Term Memory Recurrent Neural Network based Multimodal Dimensional Emotion Recognition," pp. 65–72, 2015.

[67] H. Gunes and B. Schuller, "Categorical and dimensional affect analysis in continuous

input: Current trends and future directions ☆," *IMAVIS*, vol. 31, pp. 120–136, 2013.

[68] H. Gunes and M. Pantic, "Automatic, Dimensional and Continuous Emotion Recognition," *Int. J. Synth. Emot.*, vol. 1, no. 1, pp. 68–99, 2010.

[69] S. Chen and Q. Jin, "Multi-modal dimensional emotion recognition using recurrent neural networks," in *Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge*, 2015, pp. 49–56.

[70] L. He, D. Jiang, L. Yang, E. Pei, P. Wu, and H. Sahli, "Multimodal affective dimension prediction using deep bidirectional long short-term memory recurrent neural networks," in *Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge*, 2015, pp. 73–80.

[71] H. Gunes, M. Piccardi, and M. Pantic, "From the Lab to the Real World: Affect Recognition Using Multiple Cues and Modalities," in *Affective Computing*, I-Tech Education and Publishing, 2008, pp. 185–218.

[72] P. Tzirakis, G. Trigeorgis, M. A. Nicolaou, B. Schuller, and S. Zafeiriou, "End-to-End Multimodal Emotion Recognition using Deep Neural Networks," *IEEE J. Sel. Top. Signal Process.*, vol. 11, no. 8, pp. 1301--1309, 2017.

[73] S. Wu, S. Zhong, and Y. Liu, "Deep residual learning for image steganalysis," *Multimed. Tools Appl.*, vol. 77, no. 9, pp. 10437–10453, 2018.

[74] Y. Huang and H. Lu, "Deep learning driven hypergraph representation for image-based emotion recognition," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction - ICMI 2016*, 2016, pp. 243–247.

[75] K. Brady *et al.*, "Multi-modal audio, video and physiological sensor learning for continuous emotion prediction," in *Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge*, 2016, pp. 97–104.

[76] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?," *Vision Res.*, vol. 37, no. 23, pp. 3311–3325, 1997.

[77] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised

feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 215–223.

[78] F. Ringeval *et al.*, "AV + EC 2015-The First Affect Recognition Challenge Bridging Across Audio, Video, and Physiological Data," in *Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge*, 2015, pp. 3--8.

[79] J. Huang *et al.*, "Continuous multimodal emotion prediction based on long short term memory recurrent neural network," in *Proceedings of the 7th Annual Workshop on Audio/Visual Emotion Challenge*, 2017, pp. 11–18.

[80] S. Mariooryad and C. Busso, "Correcting time-continuous emotional labels by modeling the reaction lag of evaluators," *IEEE Trans. Affect. Comput.*, vol. 6, no. 2, pp. 97–108, 2015.

[81] Z. Huang and J. Epps, "A PLLR and multi-stage staircase regression framework for speech-based emotion prediction," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2017, pp. 5145–5149.

[82] T. Dang *et al.*, "Investigating word affect features and fusion of probabilistic predictions incorporating uncertainty in AVEC 2017," in *Proceedings of the 7th Annual Workshop on Audio/Visual Emotion Challenge*, 2017, pp. 27–35.

[83] E. Mower *et al.*, "Interpreting ambiguous emotional expressions," in *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, 2009, pp. 1–8.

[84] T. Dang, V. Sethu, J. Epps, and E. Ambikairajah, "An investigation of emotion prediction uncertainty using Gaussian Mixture Regression," *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, vol. 2017-Augus, pp. 1248–1252, 2017.

[85] M. A. Nicolaou, H. Gunes, and M. Pantic, "Output-associative rvm regression for dimensional and continuous emotion prediction," *Image Vis. Comput.*, vol. 30, no. 3, pp. 186–196, 2012.

[86] Z. Huang *et al.*, "Staircase regression in oa rvm, data selection and gender dependency in avec 2016," in *Proceedings of the 6th International Workshop on Audio/Visual Emotion*

Challenge, 2016, pp. 19–26.

[87] Z. Huang *et al.*, “An investigation of annotation delay compensation and output-associative fusion for multimodal continuous emotion prediction,” in *Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge*, 2015, pp. 41–48

[88] Y. Aytar, C. Vondrick, and A. Torralba, “Soundnet: Learning sound representations from unlabeled video,” in *Advances in neural information processing systems*, 2016, pp. 892–900.

[89] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep Face Recognition,” in *bmvc*, 2015, vol. 1, no. 3, p. 6.

[90] E. Barsoum, C. Zhang, C. Canton Ferrer, and Z. Zhang, “Training Deep Networks for Facial Expression Recognition with Crowd-Sourced Label Distribution,” in *ACM International Conference on Multimodal Interaction (ICMI)*, 2016.

[91] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[92] K. Wataraka Gamage, T. Dang, V. Sethu, J. Epps, and E. Ambikairajah, “Speech-based Continuous Emotion Prediction by Learning Perception Responses related to Salient Events: A Study based on Vocal Affect Bursts and Cross-Cultural Affect in AVEC 2018,” in *Proceedings of the 2018 on Audio/Visual Emotion Challenge and Workshop*, 2018, pp. 47–55.

[93] S. H. M. Van Goozen, N. E. de Poll, J. A. Sergeant, J. A. Sergeant, and S. H. M. Van Goozen, *Emotions: Essays on emotion theory*. Psychology Press, 2013.

[94] C. N. Anagnostopoulos, T. Iliou, and I. Giannoukos, “Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011,” *Artif. Intell. Rev.*, vol. 43, no. 2, pp. 155–177, 2012.

[95] K. W. Gamage, V. Sethu, and E. Ambikairajah, “Modeling variable length phoneme sequences - A step towards linguistic information for speech emotion recognition in wider world,” *2017 7th Int. Conf. Affect. Comput. Intell. Interact. ACII 2017*, vol. 2018-Janua,

pp. 518–523, 2018.

[96] S. Hershey *et al.*, “CNN architectures for large-scale audio classification,” in *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, 2017, pp. 131–135.

[97] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” pp. 1–9, 2014.

[98] J. Li *et al.*, “Estimation of Affective Level in the Wild with Multiple Memory Networks,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2017-July, pp. 1947–1954, 2017.

[99] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[100] B. Hasani and M. H. Mahoor, “Facial Affect Estimation in the Wild Using Deep Residual and Convolutional Networks,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2017-July, pp. 1955–1962, 2017.

[101] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. van den Oord, A. Graves, and K. Kavukcuoglu, “Neural machine translation in linear time,” *arXiv Prepr. arXiv1610.10099*, 2016.

[102] A. Van Den Oord *et al.*, “WaveNet: A generative model for raw audio.,” *SSW*, vol. 125, 2016.

[103] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, “Language modeling with gated convolutional networks,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017, pp. 933–941.

[104] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *J. Mach. Learn. Res.*, vol. 13, no. Feb, pp. 281–305, 2012.

[105] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, “An empirical evaluation of deep architectures on problems with many factors of variation,” in

Proceedings of the 24th international conference on Machine learning, 2007, pp. 473–480.

[106] C. E. Rasmussen, “Gaussian processes in machine learning,” in *Summer School on Machine Learning*, 2003, pp. 63–71.

[107] C. E. Rasmussen, “Gaussian processes in machine learning,” in *Summer School on Machine Learning*, 2003, pp. 63–71. Springer, Berlin, Heidelberg.

[108] P. I. Frazier, “A tutorial on Bayesian optimization,” *arXiv Prepr. arXiv1807.02811*, 2018.

[109] D. Le, Z. Aldeneh, and E. M. Provost, “Discretized Continuous Speech Emotion Recognition with Multi-Task Deep Recurrent Neural Network.,” in *INTERSPEECH*, 2017, pp. 1108–1112.

[110] D. E. King, “Dlib-ml: A Machine Learning Toolkit,” *J. Mach. Learn. Res.*, vol. 10, pp. 1755–1758, 2009.

[111] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” 2015.

[112] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in *European conference on computer vision*, 2016, pp. 630–645.

[113] S. Zagoruyko and N. Komodakis, “Wide residual networks,” *arXiv Prepr. arXiv1605.07146*, 2016.

[114] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv Prepr. arXiv1409.1556*, 2014.

[115] R. K. Srivastava, K. Greff, and J. Schmidhuber, “Highway networks,” *arXiv Prepr. arXiv1505.00387*, 2015.

[116] T.-Y. Lin *et al.*, “Microsoft coco: Common objects in context,” in *European conference on computer vision*, 2014, pp. 740–755.

[117] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” 2009. Technical report, University of Toronto.

[118] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, “Reading digits in natural images with unsupervised feature learning,” 2011.

[119] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in *CVPR09*, 2009.

[120] P. J. Werbos, “Backpropagation Through Time: What It Does and How to Do It,” *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[121] R. J. Williams and J. Peng, “An efficient gradient-based algorithm for on-line training of recurrent network trajectories,” *Neural Comput.*, vol. 2, no. 4, pp. 490–501, 1990.

[122] M. Jaderberg *et al.*, “Decoupled Neural Interfaces using Synthetic Gradients,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017, pp. 1627--1635.

[123] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv Prepr. arXiv1412.3555*, 2014.

[124] A. Van Den Oord *et al.*, “WaveNet: A generative model for raw audio.,” *SSW*, vol. 125, 2016

[125] P. V Rouast, M. Adam, and R. Chiong, “Deep Learning for Human Affect Recognition: Insights and New Developments,” *IEEE Trans. Affect. Comput.*, 2019.

[126] D. Linh Tran, R. Walecki, S. Eleftheriadis, B. Schuller, M. Pantic, and others, “Deepcoder: Semi-parametric variational autoencoders for automatic facial action coding,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3190–3199.

[127] C. E. Rasmussen, *Gaussian Processes for Machine Learning*. Summer School on Machine Learning, 2003.

[128] P. I. Frazier, “A Tutorial on Bayesian Optimization,” *arXiv Prepr. arXiv1807.02811*, 2018.

[129] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[130] P. J. Werbos, “Backpropagation Through Time: What It Does and How to Do It,” *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990

[131] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *International conference on machine learning*, 2013, pp. 1310--1318.

[132] S. Hochreiter and J. Urgen Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997

[133] K. Cho *et al.*, “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,” *arXiv Prepr. arXiv1406.1078*, 2014.

[134] M. Jaderberg *et al.*, “Decoupled Neural Interfaces using Synthetic Gradients.”

[135] A. Gruslys, R. Munos, G. Deepmind, I. Danihelka, M. Lanctot, and A. Graves, “Memory-Efficient Backpropagation Through Time,” in *Advances in Neural Information Processing Systems*, 2016, pp. 4125--4133.

[136] R. J. Williams and J. Peng, “An Efficient Gradient-Based Algorithm for On-Line Training of Recurrent Network Trajectories,” *Neural Comput.*, vol. 2, no. 4, pp. 490–501, 1990.

[137] H. Jaeger and H. Jaeger, “A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach,” 2002.