MULTIMODAL EMOTION RECOGNITION USING TEMPORAL CONVOLUTIONAL NETWORKS

HUSSEIN HARB

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

Over the past decade, the field of affective computing has received increasing attention. With advancements in machine learning, a wide range of methodologies have been developed to better understand human emotions. However, one of the major challenges in this field is accurately modeling emotions on a set of continuous dimensions, such as arousal and valence. This type of modeling is essential to represent complex and subtle emotions, and to capture the full spectrum of human emotional experiences. Additionally, predicting changes in emotions across time series adds another layer of complexity, as emotions can shift continuously.

Our work addresses these challenges using a dataset that includes natural and spontaneous emotions from diverse individuals. We extract multiple features from different modalities, including audio, video, and text, and use them to predict emotions across three axes: arousal, valence, and liking. To achieve this, we employ deep features and multiple fusion techniques to combine the modalities. Our results demonstrate that temporal convolutional networks outperform long short-term memory models in multimodal emotion prediction.

Overall, our research contributes to advancing the field of affective computing by developing more accurate and comprehensive methods for modeling and predicting human emotions.

Acknowledgement

Working towards this degree would have never been possible without the support I received from my supervisor, friends and family.

I am greatly thankful for my supervisor, Prof. Hussein Al Osman, without whom I would not have been able to complete my studies. He was always there and ready to answer my questions and guide me through my degree.

I am very privileged to have amazing parents and siblings who provided me with unconditional support and motivation.

Table of Contents

List of Figures

List of Tables

Glossary of Terms

RNN: Recurrent Neural Network

LSTM: Long Short-term Memory

GRU: Gated Recurrent Unit

CNN: Convolutional Neural Network

DCNN: Deep Convolutional Neural Network

TCN: Temporal Convolutional Network

FCN: Fully Convolutional Network

ESN: Echo State Networks

GAN: Generative Adversarial Network

SEWA: Automatic Sentiment Analysis in the Wild

RECOLA: The REmote COLlaborative and Affective

FER: Facial Expression Recognition

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.

LFPC: Log Frequency Power Coefficient

PLLR: Phone Log-Likelihood features

BoW: Bag of Words

MSE: Mean Square Error

PCA: Principal Component Analysis

CCC: Concordance Correlation Coefficient

SVR: Support Vector Regression

GP: Gaussian Processes

BERT: Bidirectional Encoder Representations from Transformers

ALBERT: A lite BERT

ELMo: Embeddings from Language Model

Chapter 1 Introduction

The field of affective computing has seen significant growth in recent years, primarily due to the advances in deep learning throughout the last decade. Affective computing has various applications in human-computer interaction and has consequently received a lot of attention from researchers. Emotion recognition is one of the most investigated problems in this field, where a machine recognizes the affect of its user using various input types, such as audio and video. This allows the machine to potentially take the user's emotional status into account in its response to their commands.

On a very basic and discrete level, human emotions are divided into six categories: happiness, sadness, fear, neutral, disgust, and surprise. However, dividing emotions into discrete categories severely limits their representation, as real-life emotions are much more complex and nuanced than simple categories. As a result, emotions are increasingly represented on a continuous scale based on the dimensionality theory of emotions [1]. In this case, emotions are often described on at least 2 axes, where the x-axis represents valence and the y-axis corresponds to arousal. These emotions are commonly expressed in the range of -1 to +1. Hence, a valence value of +1 represents maximum positive emotion, while a value of -1 represents maximum negative emotion. Moreover, a +1 arousal value represents maximum emotional intensity, while a -1 value represents the minimum expression of emotion. Another commonly used dimension is dominance, which measures power or control [1]. Naturally, using such a complex model for emotions presents more challenges compared to detecting discrete emotions, as it is capable of representing a much wider array of subtlety.

1.1 Motivation

The field of emotion recognition is growing rapidly, and it has a wide range of potential applications in human-computer interaction. This technology can improve machine feedback to user prompts by modulating its responses to account for the user's emotional state [2][3][4]. Moreover, the technology has been applied to healthcare applications, including the detection of mental illnesses such as bipolar disorder [5] and depression [6].

Rosalind Picard is credited with coining the term "affective computing" which she defined in her seminal book on the subject [7]. In this book, Picard proposes various applications for the technology [7]. For example, she proposes applications in personalized learning, such as tailoring students' experiences to their emotional responses. A teacher can potentially better present material if they are aware of and respond to emotional cues from their students. However, it can be challenging for teachers to monitor emotions in the classroom, especially in large lecture halls where facial and body expressions are not easily discernible. To overcome this challenge, an affective classroom barometer can be used to collect and display students' emotional responses anonymously. This tool can be especially useful in situations where some students are confused or disengaged, as the barometer's level will change without revealing the identity of individual students. Teachers can use this feedback to adjust their teaching strategies and create a more engaging and effective learning environment.

Affective computing can also be applied to machines that engage students in learning experiences. By understanding students' emotions, these machines can adjust their teaching approach to better support individual needs and improve learning outcomes.

Affective computing can also have important applications in supporting individuals with autism in challenging social situations. Although individuals with autism may have strong memories and the ability to recognize patterns, they often struggle to interpret social and emotional cues. As a result, they may inadvertently say or do things that offend others, without understanding the impact of their actions. Teaching individuals with autism to recognize and interpret social cues can be difficult, as they may struggle to generalize their learning to other similar situations. Affective computing can be a valuable tool to support individuals with autism in navigating challenging social situations. By utilizing technology that can understand and respond to emotions, individuals with autism can receive personalized guidance and support to improve their social skills and interactions. This can be particularly helpful as teaching individuals with autism to recognize and interpret social cues can be challenging, given their difficulty in generalizing learning to similar situations.

Affective computing can also be used to help consumers when using software or hardware based on their emotional feedback. Computer users can face many frustrating hurdles while attempting to learn or operate any complex software or hardware. However, having the computer always helping the user is not very constructive, as it can prove to be more of an annoyance than a constructive helper, and might be detrimental to the user experience. Thus, it would be very helpful if the computer detected when its user was frustrated and only offered assistance when the user reaches a certain threshold of emotional distress. For example, if a user is having trouble while learning how to use new software, an affective computer can detect the user's frustration and offer assistance that would resolve their current problems but would not interfere with any further operation.

Affective computing has numerous other potential applications that can enhance the user experience and enable better interactions with computing devices. As a result, there has been significant recent interest in this field, with researchers exploring various methods and datasets to advance the technology.

To realize affective computing applications, we must build machine learning models that can recognize human emotions based on one or more informational channels or modalities, such as facial expressions, voice prosody, and speech sentiment. These models must be trained on datasets. Broadly, emotion datasets can be divided into three categories. These datasets are employed to develop models for emotion recognition. Acted emotions datasets contain records of subjects purposely expressing emotions. Induced emotions datasets contain records of subjects reacting to strong emotional content, such as a disturbing video or a prank. The emotions expressed in these datasets are typically pronounced, and in some cases exaggerated, and lack the subconscious gestures that underlie natural emotions. Hence, models developed using such datasets may not generalize well to real-life situations. Spontaneous datasets comprise affects recorded in the wild (i.e., involving real life interactions). This refers to the practice of capturing the emotional state of subjects behaving naturally. These subjects are never asked to express an emotion or induced by any stimulus to display any affect. Instead, they are asked to engage in spontaneous interactions while being recorded. It is more complex to develop models for such datasets. However, such models are more generalizable to real-life applications.

A significant amount of work has been done on acted datasets, achieving very good results [8][9][10]. Nonetheless, these results may not transfer to real world success. However, recognizing spontaneous and natural emotions remains challenging. Thus, motivated by the diversity and importance of affective computing applications, we focus on spontaneous and continuous emotion

recognition due to its practical usefulness in many applications. There has been a steady improvement in methodologies and results in the literature, but we believe that the technology can be further improved further.

1.2 Problem Statement

Emotion recognition is a complex problem that is made even more challenging by the use of continuous scales to represent emotions. To address this issue, researchers have explored the use of multiple modalities, including audio, video, text, and physiological features. Each modality has varying abilities to predict emotions, and different modalities can provide unique information that others lack. For instance, a subject may remain silent but display an extremely emotional facial expression. Moreover, similar data from one modality can correspond to multiple emotions without additional information from other modalities. For example, the same facial expression can indicate several emotions, and the true underlying emotion can be revealed through other modalities. Thus, combining information from multiple modalities through multimodal fusion can be highly advantageous. There are several approaches to multimodal fusion, including feature-level fusion, decision-level fusion, and model-level fusion.

In feature-level fusion [11][12], multiple features from different modalities are concatenated with each other and then fed to the model. However, this technique can suffer from the curse of dimensionality, meaning that the classifier can overfit the training data, especially when trained on a small dataset with a large number of concatenated features. Moreover, to ensure synchronization, it is necessary to extract the features at the same time.

In decision-level fusion, the outputs of multiple models are combined to render a final output. Hence, multiple models are trained on each input feature, and their outputs are then fused using a

second model such as a linear regression [13] or support vector machines. However, this method ignores interactions between the different types of input features, which can be correlated with each other [14].

In model-level fusion, intermediate feature representations are fused in several ways, such as concatenating the outputs of intermediate layers, kernel fusion, and Hidden Markov Models. As a result, model-level fusion can exploit the correlations between different features.

More recently, temporal convolutional networks (TCNs) have emerged as a promising solution for sequence modeling [15]. TCNs use convolutional layers with dilated filters to model temporal relationships in the input data [16]. These networks have shown to be highly effective in various time-series tasks and have outperformed traditional recurrent models in some cases. However, to the best of our knowledge, no study has comprehensively explored the utilization of TCN with different features and fusion strategies for emotion recognition.

In this study, we aim to investigate the utilization of TCNs with different feature modalities for emotion recognition on the SEWA dataset. Specifically, we explore the use of unimodal solutions based on text, audio, or video modalities. For the text modality, we evaluate the performance of various state-of-the-art embedding technologies, including Word2Vec, BERT, ALBERT, and ELMo, as features for our TCN models. For the visual modality, we explore both VGGFace and DenseNet, and for the audio modality, we use the VGGish feature sets. We also investigate feature-, decision-, and model-level fusion on the best performing features to enhance the recognition performance. This comprehensive exploration aims to provide insights into the effectiveness of TCNs with different feature modalities and fusion strategies and to inform the development of more accurate and robust emotion recognition systems.

1.3 Contributions

The main contributions of this thesis are as follows:

- We propose the use of TCN models to predict the valence, arousal, and liking emotional dimensions from the audio, video, and text modalities.
- We conduct a comprehensive study to assess the performance of TCN models using various visual, audio, and text features for unimodal and multimodal emotion recognition.
- We propose TCN architectures for feature, decision, and model level fusion and compare their performance.

1.4 Thesis Organization

The rest of this thesis is organized as follows:

- Chapter 2: We present the background and related work for our thesis. Specifically, we briefly overview our dataset, TCNs, and hyperparameter optimization.
- Chapter 3: We present the SEWA database and describe the different features it provides, and other common features used in the literature.
- Chapter 4: We discuss our proposed method. We describe the different types of features we extract and our multimodal fusion configurations.
- Chapter 5: We evaluate our proposed models and compare their performance to existing state-of-the-art solutions.
- Chapter 6: We summarize our thesis work and provide insights into future work.

Chapter 2 Background and Related Work

2.1 Background

2.1.1 Deep Learning

Deep learning has emerged in recent years as a solution to many complex machine-learning problems. It has given machines the capability to learn large amounts of information and use it to classify or predict data. This is accomplished using deep neural networks that consist of several layers. The networks attempt to learn complex patterns present in the input data and predict their outputs using them. Those patterns are usually not easily found using regular handcrafted methods. Hence, deep neural networks are very useful when dealing with large amounts of data.

2.1.1.1 Convolutional Neural Networks

Convolutional neural networks (CNNs) are neural networks that utilize convolution, which is very useful in image-related tasks. A convolutional layer consists of a group of filters that are convolved with the input image, called kernels. The kernels are usually small with sizes of about 3 to 7 per dimension. The layer then outputs a stack of the resulting convolutions with each of the filters.

The filters are usually applied on each pixel in the input and its immediate neighbors. However, this is not always the case. A filter's stride refers to the shift the filter does while going over the image. In the simplest case, a stride of 1 means that the filter is applied to all the pixels in the input. Additionally, the filter can be dilated, which creates a gap between adjacent filter taps. A dilation factor of 1 will thus mean that the filter is applied to adjacent pixels, and increasing that means skipping over neighboring pixels and applying the filter to a wider area.

2.1.1.2 Recurrent Neural Networks

Time-series data presents a challenge to regular neural networks. To humans, understanding time series is not difficult, as we are capable of understanding context using previous information. For example, when watching a movie, any event that happens is understood better through the previous context of the film. However, unlike humans, a typical neural network does not have this ability, as it is not capable of storing information about previous time steps to use for later inputs. It can only process data at each time instant separately, which prevents it from taking into account the context of the information.

As a result, recurrent neural networks (RNNs) are very useful for time-series data. They are fed an input in a sequential manner starting from the first data point. The network then preserves information about previous data points in a state. The state is then updated with every new input through a cyclical connection that incorporates the current state information with the new inputs to the network. As a result, the network can learn new information while utilizing context from previous inputs.

However, a simple cyclical connection suffers in the long term. When attempting to capture long-term information, the information about past time steps slowly disappears as a result of vanishing gradients. Hence, gated RNNs were proposed as a solution to this problem. The goal of gated RNNs is to give networks the capability to selectively choose what states to accumulate or forget depending on their usefulness in the task the network is being trained for. The most common gated RNNs are the gated recurrent unit (GRU) and long short-term memory (LSTM).

An LSTM solves this problem with the addition of an inner loop, giving the network the ability to determine whether it needs to retain older information. An LSTM cell is shown in Figure 1 below. It consists of several gates that govern the amount of information entering or leaving the cell. The

input gate uses the previous hidden state to determine the extent to which the current input is to be incorporated into the state. The forget gate is used in the inner loop to determine if the current information stored in the state is to be forgotten. Finally, the output gate decides whether the new state can be passed on to the output.

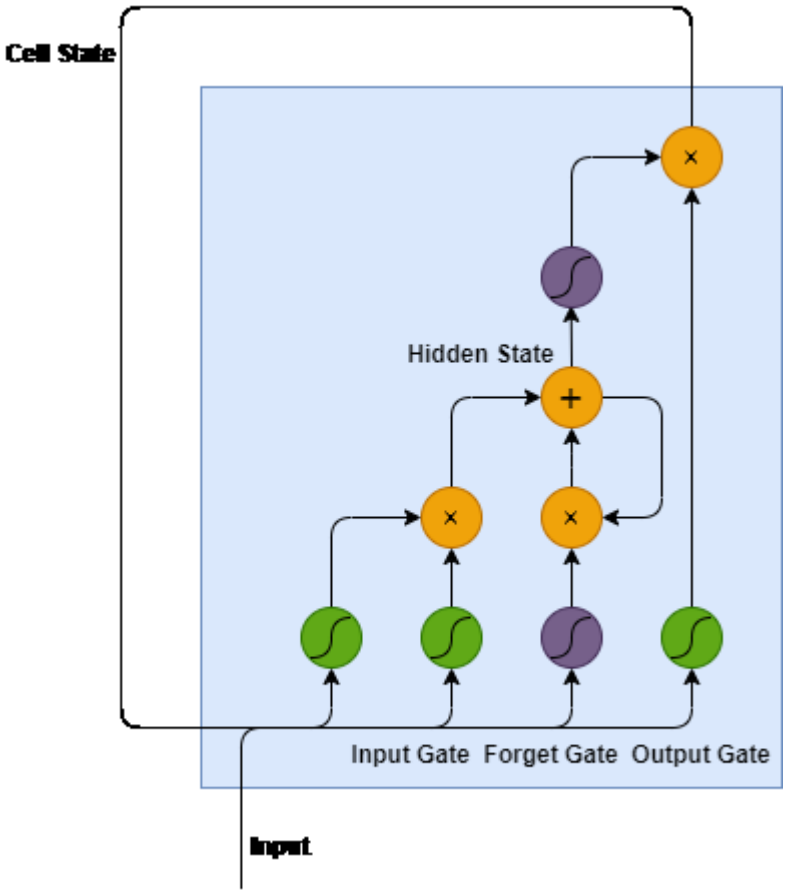

Figure 1: An LSTM diagram. The green operation is a sigmoid, while the purple operation is a tanh.

On the other hand, the GRU architecture is simpler. It has only a reset gate and an update gate. The update gate uses the previous state and the current input to determine what information to keep in the state. The reset gate is similar to the forget gate and determines to what extent the cell forgets its stored information.

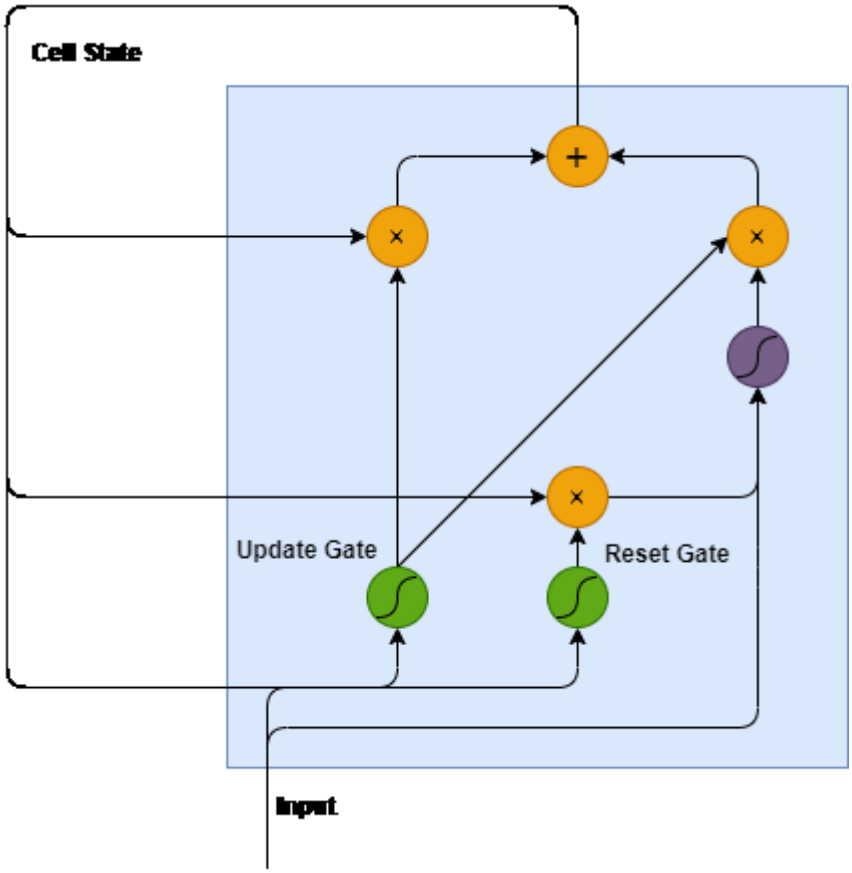

Figure 2: A GRU diagram. The green operation is a sigmoid, while the purple operation is a tanh

2.1.2 Temporal Convolutional Neural Networks

RNNs, such as LSTMs and GRUs, are the most frequently used deep neural network architectures for time series prediction tasks. Convolutional networks, on the other hand, have been successfully applied to a variety of time series tasks in the past with promising results [17][18]. As a result, Bai et al. proposed the temporal convolutional networks model, which is based on CNNs [15]. They conducted a systematic evaluation of the most frequently used convolutional and recurrent networks for sequential tasks. Their evaluation involved a variety of tasks that are frequently used to assess RNNs. Their findings indicate that TCNs outperform RNNs on most of the benchmarks they tested. As a result, they recommended that future time-series prediction systems should use

TCNs rather than other RNN architectures. TCNs have several advantages over other types of recurrent networks. These advantages include the following:

- The network is composed of a series of convolutional layers and therefore it is simple and fast to train.
- Since the network uses convolutional layers, its outputs are computed concurrently. Conversely, RNNs require the network to first determine the outputs for the previous time steps before predicting the current time step.
- Since convolutions use parallel computation, the network consumes significantly less memory than RNNs, as it does not need to store previous states.
- TCNs can look very far into the past when making predictions by utilizing exponentially dilated convolutions and extremely deep residual layers.
- TCNs do not exhibit information leakage from the future when using causal convolutions.

2.1.2.1 Causal Convolutions

The TCN is based on two principles: the output of the network has the same size as the input, and there is no information leakage from the future.

A TCN implements the first principle by utilizing a one-dimensional fully convolutional network (FCN) architecture. All hidden layers are identical in length, and their inputs are zero-padded with (kernel size – 1) zeros to maintain the input length. To prevent information leakage, the TCN employs causal convolutions, where the output of a layer at a time step t only depends on input data from time steps 0 to t.

2.1.2.2 Dilated Convolutions

The TCN employs dilated convolutional layers to capture long-term dependencies with a small number of convolutional layers. A convolutional layer with a dilation factor of d creates a fixed step between adjacent filter taps. Thus, when d=1, the TCN is reduced to a regular convolution. This enables the network to consider a broader range of inputs with fewer convolutional layers, effectively expanding the outputs' receptive field and representing long-term historical dependencies. To augment the size of the receptive field, we can increase the filter size k, the dilation factor d, or the layer count. The dilation factor increases exponentially with the network depth in the TCN.

An example of a series of causal dilated convolutions is shown in Figure 3 below.

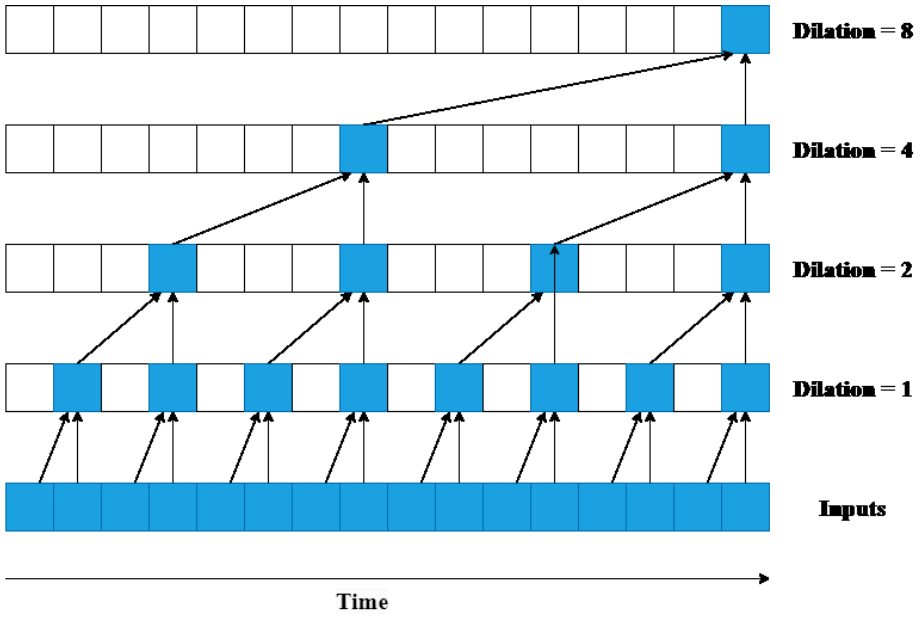

Figure 3: Causal dilated convolutions

2.1.2.3 Residual Connections

Generally, deeper neural networks have shown improved performance in accuracy [19]. However, as more layers are added, the network will reach a point where adding more layers becomes detrimental to the performance, and the accuracy will start to deteriorate. Notably, this decrease is reported on the training data. This means that the network is not overfitting. Instead, there is a decrease in its learning capabilities during the training process compared to shallower networks. Intuitively, if we consider a shallow network with a certain number of layers, we can simply add layers with identity mappings, and the new network should not produce a higher training error compared to its shallow counterpart.

Based on this idea, He et al. [19] propose utilizing a residual unit. Instead of the network layers learning to map the inputs to the outputs, they will learn the difference between an identity mapping and the true mapping. This is accomplished by creating a skip connection over the layers and adding it to their output, as demonstrated in Figure 4**.** Mathematically, let the desired function be H(x), where x is the input. In a residual network, the layers would learn the mapping H(x) – x instead. In this case, assuming the ideal function is an identical mapping, it would be much easier for the residual to be pushed to zero instead of learning the mapping. This configuration does not add any parameters to the network and does not add complexity to the training process, as the network can still be trained using traditional gradient descent. This architecture was shown to improve deep network performance and can lead to accumulated benefits with added layers. Such a TCN unit is shown in Figure 4.

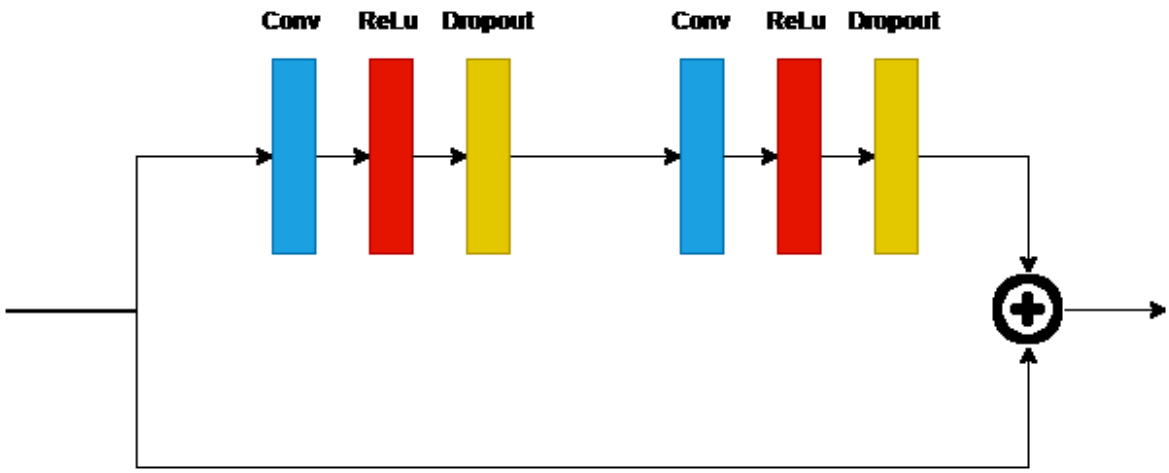

Figure 4: Residual unit

Hence, Bai et al. [15] use stacked residual units in their proposed TCN architecture. In each residual unit, the network has 2 stacked 1-D convolutional layers, each followed by a weight normalization layer, a ReLu layer, and a dropout layer. To ensure that the output of the residual layer and the skip connection have the same shape, a 1×1 convolution can be used on the input before adding it to the residual.

Putting this all together, a TCN residual unit is shown in Figure 5, where the convolutions are all causal and dilated.

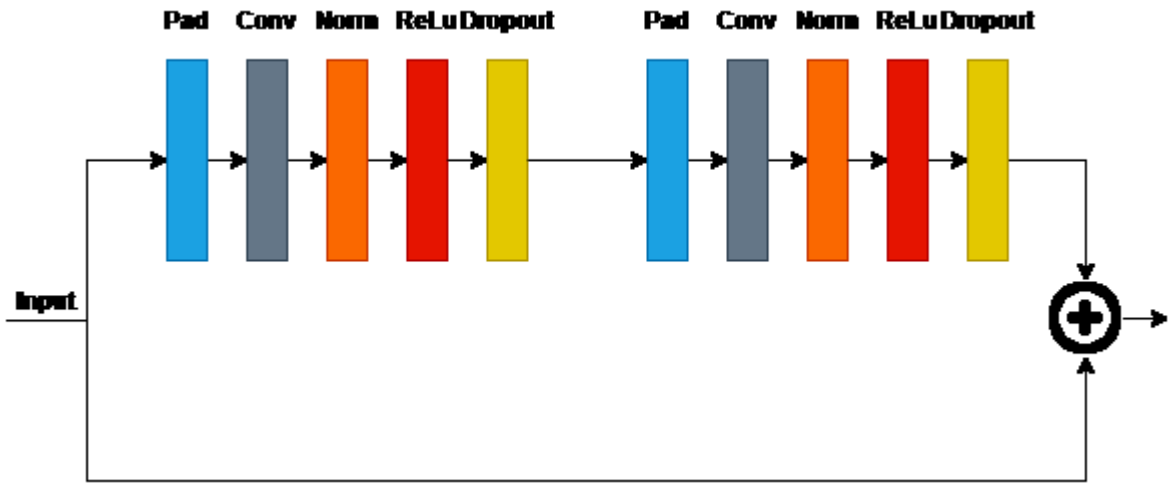

Figure 5: TCN diagram.

2.1.3 Hyperparameter Optimization

Numerous hyperparameters must be set for deep neural networks. However, performing this task manually is extremely difficult and may result in suboptimal performance. Hence, hyperparameter optimization algorithms can be used. Given a search space containing the possible values, a hyperparameter optimization algorithm aims to search for the values that produce the best results.

Grid search is a very simple algorithm for hyperparameter optimization. The algorithm iterates over several data points in the hyperparameter space and finds the values that produce the best results. Since grid search iterates over a discrete number of points, it may miss the optimal values that produce the best results. Moreover, decreasing the step size and increasing the number of samples leads to a very large search space, which can take an enormous amount of time to search. Bergstra et al. [20] demonstrated that random search is superior to grid search on this basis. They demonstrated that randomly searching the hyperparameter space can rapidly identify model configurations that are comparable to or better than grid search models.

More recently, hyperparameter optimization based on Gaussian Processes [21] has gained popularity. This approach aims to find the maximum of the posterior distribution or objective function, which represents the validation accuracy as a function of the hyperparameters. This is best suited for optimization over continuous variables with less than 20 dimensions [22].

For an objective function f and a finite collection of points in the hyperparameter space $x_1, \dots, x_n$, we assume that the f values on those points are unknown. Thus, we start by assuming that the vector $[f(x_1), \dots, f(x_n)]$ follows a prior normal multivariate distribution, with a mean function μ_0 computed at each point x_i and a covariance function or kernel Σ_0 evaluated between each pair of points x_i, x_j. Given that close points are expected to produce similar outputs, the kernel is chosen to generate a high degree of positive correlation between them.

The power exponential is a common and simple kernel function, defined as

$$\Sigma_0 = e^{-\alpha\|x,x'\|} \tag{1}$$

Where α is a hyperparameter of the function.

Another common kernel is the Matern kernel, defined as

$$\Sigma_0 = \alpha_0 \frac{2^{1-\upsilon}}{\Gamma(\nu)} \left(\sqrt{2\nu}\|x - x'\|\right)^{\upsilon} K_{\upsilon}\left(\sqrt{2\nu}\|x - x'\|\right) \tag{2}$$

Where α_0 and υ are hyperparameters and K_{υ} is the modified Bessel function.

This configuration results in a tractable posterior [21]. Hence, the variable distribution is updated after each iteration when a network with the sampled hyperparameters is trained over the input data.

An acquisition function is used to determine where to sample from the data space. The most frequently used acquisition function is Expected Improvement, which is defined as $EI_n(x) = E_n[[f(x) - f_n^*]^+]$

This function computes the expected improvement of f at the point x given its current best value f_n^*. Since we do not know beforehand the improvement, if any, a point x will have on f, we select the point with the maximum expected improvement. Hence, after evaluating f at n points, the next point to evaluate f on is x_{n+1}, which can be computed as

$$x_{n+1} = argmax\ EI_n(x) \tag{3}$$

2.2 The SEWA Database

The SEWA database [23] is an emotion recognition dataset consisting of subjects from different cultural backgrounds who engage in a conversation about advertisements they were shown. The subjects are British, Greek, Serbian, German, Hungarian, and Chinese. Parts of the database have been used in the AVEC challenges [5][6][24] since 2017. For the AVEC 2017 challenge [6], the dataset contained only German subjects with emotions labeled for arousal, valence, and liking, where liking refers to the subject's feelings towards the product. Hungarian and later Chinese subjects were added in later iterations of the challenge [5][24] to facilitate the study of cross-cultural emotion recognition. We elaborate more on the SEWA database in section 3.1.

2.3 Related Work

This section discusses the various methods for continuous emotion recognition that have been proposed. We overview the existing models, features, and multimodal fusion techniques. Numerous features and models have been proposed in the literature, including RNNs and Kalman

filters, and a variety of different types of features have been used, ranging from handcrafted to deep extracted.

Amirian et al. [25] proposed using echo state networks (ESNs) on the modalities of the RECOLA dataset [26], which consists of 46 French-speaking subjects participating in dyadic interactions while performing a task. For the audio modality, they used the Log Frequency Power Coefficient (LFPC) [27] with a log filter bank between 200 Hz and 4kHz in addition to the features provided in the dataset. For the visual modality, they used a combination of the LGBP-TOP appearance features and geometric landmark features. In total, their system has a total of nine input features. They trained a combination of random forests and ESNs on the input data. They compared 3 types of fusion: Early, mid-level, and late echo-state fusion. Early-level fusion involves training nine random forests on different input features and then fusing the results using an ESN. Late fusion uses the outputs of the nine random forests to train individual ESNs. The results of the nine ESNs are then linearly combined to produce the final predictions. A single ESN is used for mid-level fusion, with one reservoir for each feature set, for a total of nine reservoirs. After that, the output is calculated linearly by combining the reservoir states and the inputs. They were able to achieve a CCC of 0.776 for arousal and 0.634 for valence.

Huang et al. [28], on the other hand, investigated the effects of annotation delay compensation using a variety of models. They used relevance vector regression and relevance vector machines to combine the results. They obtained a CCC of 0.740 for arousal and 0.588 for valence.

Povolny et al. [29] proposed supplementing the dataset with additional features. Most notably, for the audio modality, they extracted bottleneck features from a neural network that was trained on phonetic targets. For the visual modality, they extracted features from the final two layers of a CNN trained to localize facial landmarks. These features contain appearance and geometric

information about the inputs. Finally, since the dataset does not provide textual data, they used automatic speech recognition software to extract the transcriptions. They extracted French Word2Vec embeddings with 200 dimensions and a lexicon of emotional words. They used linear regression on each modality, followed by another linear regression to fuse the prediction results. As a result, they obtained a CCC of 0.855 for arousal and 0.713 for valence.

Somandepalli et al. [30] proposed using Kalman filters as a decision-level fusion scheme. They extract several features such as a binary face detection status, which represents whether a face can be detected or not, and voicing probability features, which represent the probability of the subject speaking. Each feature is then trained on an SVR model. The predictions of each input feature are then treated as noisy observations. They are then combined using four Kalman filters: one for audio, one for video, one for both, and one using only physiological features. The filter used at a particular moment is selected based on the availability of data from each modality, which is represented using the face status and voicing probability. They designed separate Kalman filters for arousal and valence, and since arousal and valence are correlated [31], they used the arousal predictions as additional noisy observations to enhance valence prediction. Their system scored a CCC of 0.703 for arousal and 0.681 for valence.

Sun et al. [13] studied several types of visual features and compared their performances on the RECOLA dataset [26]. For the visual modality, they extracted reduced 168 appearance LGBP-TOP features [32] using PCA on a bigger number of features. They also extracted geometric features in the form of facial landmark points and the angles and distances between them. Multi-scale Dense SIFT features (MSDF) [33] were also proposed. Finally, they extracted deep visual features from several networks such as an AlexNet [34] based network pre-trained on the FER dataset [35]. Their results showed that the best arousal results are achieved using the audio

modality, while visual features achieve the highest valence scores. The outputs are then fused using SVR, and the model achieves a CCC of 0.683 on arousal and 0.642 on valence on the test dataset.

Chao et al. [36] used a ε-insensitive loss to ignore small errors and achieved superior results compared to the absolute loss and squared loss. They utilized an LSTM model with temporal pooling. They compared several types of pooling and achieved the best results using mean pooling as opposed to max pooling. They achieved the best CCC on arousal using the eGeMAPS features [37] and the best valence CCC using deep features extracted from the final layer of the FACE-CNN, a deep network based on the Caffe implementation [38]. Their results also showed that appearance features perform better than geometric features and that physiological features produce the least accurate results.

The sparsity of data is a significant issue with affective computing and sentiment analysis systems. To address this, some systems use generative adversarial networks (GANs) [39]. This has two advantages: GANs are capable of learning robust high-level representations of data and assisting in the resolution of the sparse data issue by producing their own realistic emotional data [39]. On the audio modality of the RECOLA dataset, Han et al. [40] proposed using a conditional adversarial training architecture. They used a framework that is similar to conditional GANs, in which two networks are trained to predict emotions collaboratively. Specifically, the first network is trained to predict arousal and valence using the input data. The second network, on the other hand, is trained to predict whether the input labels are real or not. The true labels come from the RECOLA dataset, while the false labels come from the first network's outputs. The results of the second network are then incorporated into the first model's loss function. Both the generator and discriminator were implemented using an LSTM network. The results demonstrate significant

improvement, particularly in the valence dimension, where the test set's valence CCC score increased from 0.390 to 0.455.

In the winning paper of AVEC 2016, Brady et al. [41] utilized different methods for each modality. For audio, they used sparse coding to extract higher-level features, which were used as input for an SVR model. For the visual modality, they used a CNN model as a feature extractor followed by an RNN. Finally, they employed LSTMs for the physiological signals. The three modalities are fused using a Kalman filter-based approach. This method achieved a CCC of 0.770 for arousal and 0.687 for valence.

Huang et al. [42] used a deep convolutional neural network (DCNN) followed by a hypergraph for emotion recognition using facial expressions. A hypergraph is a graph with edges that can be connected to more than two vertices. The DCNN is pre-trained and the output of its last hidden layer is exponentially normalized and used as attributes for the hypergraph. Emotion recognition follows by formulating and solving a partition problem on the hypergraph.

The Aff-Wild dataset [43][44], introduced in the Affect-in-the-wide challenge, contains videos of people reacting to emotion-inducing clips, such as an unexpected development in a movie, a trailer, a prank, or a disturbing video clip. The subjects are ethnically diverse and are recorded in various emotional states, head poses, illuminations, and occlusions. However, since the recorded emotions are induced by the clips the subjects are watching, they can be less subtle and more pronounced. Kollias et al. [45] proposed a deep learning architecture called AffWildNet, which includes CNN and RNN layers, and achieved state-of-the-art results on arousal and valence using only the visual modality of the dataset. Their network consists of a CNN, which is either a VGG-Face [46] or a ResNet-50 [47], followed by a fully connected layer and two GRU [48] layers. The inputs of the fully connected layer consist of the output of the final CNN pooling layer as well as facial

landmarks. Their results outperform other architectures proposed on the Aff-Wild dataset, such as MM-NET [49], FATAUVA-NET [50], and DRC-net [51].

2.3.1 Related Work on the SEWA Database

Gamage et al. [52] suggested that emotions can be predicted based on certain extremely sparse information comprising salient events and gestures embedded in speech. The salient events are characterized by being followed by a relatively consistent change in arousal and valence. This is consistent with the appraisal theory of emotions, which suggests that emotions are directly related to the way a person appraises internal or external stimuli [53][54]. Specifically, certain easily identifiable events can trigger non-verbal expressions of affect. In human psychology, these expressions are known as affect bursts [55]. Gamage et al. [52] identified salient events in the SEWA dataset by examining the nonverbal vocal events present in the SEWA transcriptions. Previously, such events were used infrequently for emotion recognition tasks [56]. Their findings indicate that laughter and slight laughter events can be extremely useful in predicting arousal and valence on their own.

Recent research has been increasingly utilizing LSTM models for continuous emotion recognition. For example, Huang et al. [57] used separate LSTM models on each input feature. They use several features such as LLDs for the audio modality and deep features for video modality. For the text modality, they used bag-of-text-words (BoTW) with a dictionary of 521 words. Finally, SVR is used to combine the predictions from the trained LSTMs. Typically, there is a delay in labeling, which stems from the observers' natural response time to the videos [58]. This is remedied by rearranging the annotations prior to training. Moreover, noise may be introduced by human annotators during the labeling process given the continuous and extended nature of the task. Huang et al. [57] use mean temporal pooling to smooth and reduce noise in the labels.

Dang et al. [59] extracted relevant textual features using The Suite of Linguistic Analysis Tools (SALAT) [60]. Additionally, they used Phone Log-Likelihood features (PLLR), a collection of phonetic features that have been shown to outperform eGeMAPS features [61]. They assumed that the predicted emotions are normal distributions with a mean and standard deviation to incorporate uncertainty into the fusion process. In this case, the mean represents the label's most likely value, while the standard deviation indicates the level of confidence in that value [62]. They use Output-Associative RVMs (OA-RVMs) to perform the fusion [28][63][64]. This technique is particularly useful for capturing temporal information and multidimensional affect dependencies. In addition to the input features, the OA matrices can include the arousal and valence predictions of all regressors.

One of the issues with the SEWA dataset is its small size, with only 34 training subjects. Moreover, LSTMs have difficulty learning very long-term patterns. Huang et al. [65] address these issues using data augmentation. This technique consists of dividing the time series inputs into smaller subsequences that may or may not overlap, thus increasing the dataset size. In both cases, the results showed an improvement over using the complete input sequences. In particular, it was demonstrated that using non-overlapping segments outperformed overlapping segments. However, this method results in a reduction of the time steps that the network is able to process. Chen et al. [66] employed a variety of deep features to represent the audio, visual, and text modality. They extracted audio features from SoundnNet's Conv5 layer [67], a CNN-based model. Chen et al. [[66]] used a variety of strategies to address the issue that the audio contains speech from both the subject and the interlocutor. Given the availability of turn information, they considered using only the subject's audio features and omitting the interlocutor features, which resulted in enhanced network performance. They used features extracted from DenseNet [68] and a custom VGGFace

network [46] pre-trained on the FER+ dataset [8] for the visual modality. The FER+ dataset consists of thousands of greyscale pictures of faces displaying various emotions. Each image is labeled by 10 crowd-sourced taggers to provide a less subjective classification, as well as a distribution of a range of potential emotions. The labels represent discrete emotional categories: neutral, happiness, surprise, sadness, anger, disgust, fear, and contempt. Although the dataset provides a discrete and limited emotion model, the pretrained models contain relevant information which is very helpful when predicting emotions. Finally, for the text modality, they extracted Word2Vec features from an unofficial German word embedding model [69]. Additionally, they translated the transcriptions to English and used the Google English word embedding model [70]. They compared the performance of an LSTM model to that of an SVR model and found that the LSTM model outperforms the SVR model. Their findings indicate that text is the most effective modality for predicting the liking dimension.

Similar to [66], Zhao et al. [71] extracted deep features and used them in conjunction with LSTMs to create a time series model, followed by a regressor for fusion. They used the VGGish network [72], which can learn complex audio representations, to extract audio features. They used similar features to [66] for the other modalities. They proposed several interaction strategies to take into account both the speaker's and the interlocutor's features. When the interlocutor speaks, they make use of both the interlocutor's audio and facial features. When the speaker speaks, they make use of both the interlocutor's and the speaker's audio and facial features. Textual features are used in the same way as audio features.

Conventional CNN techniques previously applied to the visual modality do not account for the time dimension. To address this issue, three-dimensional convolutional neural networks (3D-CNNs) have been proposed [73], which compute features in both spatial and temporal dimensions.

On this basis, Du et al. [74] decomposed the 3D-CNN into (2+1) D-CNNs. This configuration benefits from additional non-linear rectification and contains fewer parameters, which makes optimization easier and results in lower training and testing loss. Chen et al. [16] proposed extracting frame-level features for both audio and video using pre-trained 2D CNNs, followed by a 1D causal convolution similar to TCNs [15], which generates spatio-temporal features. They use a Resnet-50 [47] pre-trained on the FER+ dataset [8] to extract frame-level appearance features for the visual modality, and a spatial-temporal graph convolution network (ST-GCN) [75] to extract geometric features for the visual modality. The audio modality is implemented using the VGGish network [72]. Similar to [66], the dimensions of the output features are doubled, with one half representing the features from the target speaker, and the other half representing features from the interlocutor. Finally, both Glove [76] embeddings and a word embedding model trained on the 45-dimensional dataset are used for the text modality. The text modality is used exclusively to predict liking. The audio and visual features are then fed into separate deep bidirectional LSTMs to predict arousal and valence (DBLSTM). Chen et al. [16] proposed a fusion scheme that incorporates both feature and decision-level fusion. Along with the aforementioned DBLSTMs, the two best feature sets are concatenated and used to train another DBLSTM, as are the visual appearance and audio features. A final DBLSTM is used to combine the features predicted by the previous models.

The AVEC 2019 challenge [41][24] augmented the dataset used in AVEC 2018 [5] by adding Chinese test subjects. This aimed to study the cross-cultural transfer of emotions from Western European to Chinese culture. Based on this, Zhao et al. [77] proposed an unsupervised adversarial cross-cultural adaptation method. They used similar features and LSTM model to [71], in addition to Chinese Word2Vec embeddings [78]. The goal of adversarial cross-cultural adaptation is to

learn an emotion representation that is culture transferrable. The proposed method consists of three modules: a feature encoder E that encodes the features f into an emotion-salient and culture-robust representation z, an emotion regressor R that predicts emotions from z, and a culture classifier C that classifies the culture of its input z. R and C are treated as adversarial targets for E. Hence, E is trained to improve the classification performance of R while making it as hard as possible for C to differentiate between the encodings z of different cultures. The result is a training process that trains E to output features z that contain mostly non-culture specific emotional information. Their results show a marked improvement on the Chinese test dataset, with CCC scores of 0.400, 0.471, and 0.257 for arousal, valence, and liking respectively.

In general, the dominant trend for multimodal emotion recognition is to use deep features extracted from pre-trained networks with recurrent network architectures such as LSTMs. Using such methods has produced state-of-the-art results on the SEWA database as well as other emotion recognition datasets.

To the best of our knowledge, there has been only one work on TCNs for multimodal emotion recognition. Ayoub et al. [79] proposed using TCNs for the SEWA dataset. They utilized features from the audio and video modalities for emotion recognition. For audio, they utilized deep features extracted from a pretrained VGGish network. For the visual modality, they utilized deep features extracted from a VGGFace network pretrained on the FER+ dataset. Since they did not have access to the testing portion of the dataset, they utilized cross-validation on the training data and used the validation portion for testing. They achieved CCC scores of 0.7440 on arousal and 0.7557 on valence.

However, this work does not account for the text modality, nor does it predict emotions on the liking dimension. Moreover, it does not compare results using a multitude of visual and text

features such as DenseNet and BERT. In our work, we address all of those issues. We compare multiple features on the visual and text modalities and report our results on the liking dimension in addition to the arousal and valence dimensions.

2.4 Conclusion

In this section, we briefly discussed some of the state-of-the-art recurrent neural network architectures. We then discussed our proposed TCN, and finally we overviewed the related work on continuous emotion recognition.

Chapter 3 Dataset and Features

3.1 SEWA Database

The SEWA database [23] is a large dataset that is notable for being composed of recordings taken in the wild (i.e., during real-life interactions). It features recordings of several subjects engaged in dyadic human interactions, with each video featuring a single subject being interviewed by an interviewer that does not appear in the video. The subjects view four advertisements aimed at eliciting strong emotional responses and are then asked to describe their feelings during an interview. The videos last for a maximum of three minutes and are recorded using personal devices connected to the subjects' computers such as webcams.

The database features subjects from five distinct cultural backgrounds: British, German, Hungarian, Greek, Serbian, and Chinese. The objective is to assess the generalizability and transferability of machine learning models for affect estimation across cultures. We adopt the portion of the database which was used in the AVEC 2018 challenge [5]. It consists of 64 German subjects divided into three groups: 34 for training, 14 for validation, and 16 for testing, in addition to 66 Hungarian test subjects. However, we were unable to obtain the labels for the testing sets from the dataset provider. Hence, we consider the validation set consisting of 14 subjects as our testing set. We use cross-validation to use the training set for training and validation.

Each interaction is represented by audio and video data, as well as a text transcription of what the interviewees say during the video. However, only the interviewees' transcriptions are provided, along with any nonverbal vocalizations such as laughter. Given that two people speak in turn, turn information is provided for each person, and a native speaker provides the duration and time

stamps for the transcriptions. Several annotators from the subject's culture (six for German and five for Hungarian) label the videos with arousal, valence, and liking using a joystick, which they push or pull in real-time as they watch the videos. The annotation is then represented by the pitch value of the joystick, which is sampled at a rate of 66 Hz. To avoid mental overload, the annotations are done separately for each of the three dimensions. Moreover, for each dimension, the videos were annotated three times: first using audio only, then using video only, and finally using both audio and video together. The labels from each annotator are resampled using Hermitian resampling to a rate of 10 Hz and are thus provided every 100 ms. The resulting labels are then normalized to a continuous scale ranging from +1 to -1 based on the peak amplitude and median of the joysticks. The gold standard is then computed using canonical time warping, a time alignment methods based on canonical correlation analysis. Additionally, several features associated with the various modalities are included, such as facial landmarks and low-level descriptors (LLDs).

3.1.1 Audio Features

The AVEC challenges include several sets of audio features extracted from the recordings of the subjects. One of these is a set of supervised features consisting of summarizing low-level descriptors (LLDs) with a set of statistical measures computed over a 100ms sliding window. The LLDs include information such as spectral, cepstral, prosodic, and voice quality information. The dataset includes 23 LLDs extracted based on the extended Geneva Minimalistic Acoustic Parameter set (eGeMAPS) [37]. It also includes another set of LLDs consisting of Mel-frequency cepstral coefficients (MFCCs) 1-13 and their first and second-order derivatives. All the LLDs are extracted using the openSMILE toolkit [80]. Finally, the dataset includes semi-supervised bag-of-words (BoW) features extracted from the MFCCs using the openXBOW toolkit [81].

3.1.2 Visual Features

Visual features can be classified into two types [82]. The first type is based on appearance, while the second type is based on geometry. Appearance-based features are those that are associated with changes in the texture of an image, such as wrinkles and bulges. Geometric features, on the other hand, are related to the relative positions, distances, and velocities of various facial points.

Examples of appearance features include Local Gabor Binary Patterns from Three Orthogonal Planes (LGBP-TOP) [32] and Histogram of Gradients (HOG) features [83]. LGBP-TOP features [32] are extracted from facial videos by splitting them into several volumes and convolving them with a bank of Gabor filters, followed by applying local binary patterns. HOG features [83] are calculated for localized portions of an image and represent the gradients found in the given portions as well as their orientations.

The dataset also includes supervised visual features, similar to audio features. It contains the intensities of 17 Facial Action Units (FAUs) and a confidence measure for each frame, which were extracted using the openFACE toolkit [84]. Additionally, it includes visual BoW features extracted from FAU intensities using the openXBOW toolkit [81].

3.1.3 Textual Features

These features include BoW features that represent the frequency of words as well as more compact and semantically accurate word vectors [85].

3.1.4 Physiological Features

Although the SEWA database does not include physiological features, they are utilized in other datasets such as the RECOLA dataset [26]. Those features reflect biological patterns underlying the emotional behavior a person is experiencing. They include electrocardiogram (ECG),

electrodermal activity (EDA) or skin conductance, electroencephalograph (EEG), respiration rate, and skin temperature [86].

There are many types of features present in the literature. Handcrafted features have been the most commonly used features in the past. However, with modern advancements in deep learning, deep features have been shown to be better suited for many tasks [65][66].

3.2 Conclusion

In this chapter, we discussed the SEWA database. We overviewed the features that it provides for all three modalities. We also gave a brief overview of several types of features present in the literature, including physiological features which are not present in the dataset.

Chapter 4 Proposed Method

In this chapter, we describe the features we explore for our emotion recognition solution. Moreover, we describe the proposed unimodal and multimodal TCN models.

4.1 Features

Inspired by the previous work on emotion recognition and seeing that TCNs have not been sufficiently explored for this task, we studied TCN performance on several commonly used. We explore features from the audio, visual, and textual modalities that are provided with the SEWA dataset. Previous work has demonstrated the capabilities of deep audio and visual features in recognizing emotions [66][71]. Based on this, we also extract our deep features and compare their performance. Moreover, the text modality has not been sufficiently explored for the SEWA dataset. Hence, we will leverage word features that may be conducive to emotion prediction. In the following sub-sections, we describe in detail our proposed deep features and the extraction process, followed by our implementation and training process.

4.1.1 Word Features

Word vectors are representations learned from massive textual data [85]. They are more compact than other word representations and more closely associated with the word's meaning, making them suitable for emotion recognition. We translate the transcriptions to English using Google Translate (as the original language is German), and extract several word embeddings from both the original German transcriptions and English translations. There are no transcripts available for the interviewer. As a result, we use zero vectors for the time steps that are not associated with transcriptions.

4.1.1.1 Word2Vec Embeddings

One of the most used embeddings is Word2Vec, which extracts embeddings on a word basis. Hence, we divide both our original transcriptions and their English translations into words and remove any punctuation and other characters that are not letters. We then extract German word embeddings using an unofficial German Word2Vec model [69], which was created and trained by a student, and English word embeddings using the Google English Word2Vec model [70]. Since we do not know what word is being said at every time step, we take the average of the embeddings of all words in a transcription and use it across its duration. Both embeddings have 300 dimensions. We refer to them as word2vec.de and word2vec.en features, respectively.

4.1.1.2 BERT Embeddings

Attention mechanisms have become a very important tool for sequence modeling. Using attention allows a model to understand dependencies between different parts of a sequence, regardless of the distance between them [87]. Hence, attention can be very useful for linguistic tasks, such as machine translation.

Vaswani et al. [26] proposed a network architecture for sequence modeling based entirely on attention mechanisms, which they called the transformer. They created a multi-head attention mechanism, where the query, key, and value matrices are linearly projected to several dimensions. The attention function is then applied in parallel to the projected versions. This allows the model to learn connections through multiple representations, hence enabling the model to learn a richer and more diverse context for its inputs.

Based on the transformer architecture, Google's BERT [88] was proposed by researchers at Google. Since it utilizes multi-head attention, it takes contextual information into account when extracting embeddings. Sentences can be very complex with many interrelated phrases and words.

A word that comes at the end of a sentence can affect the meaning of another at its beginning. Hence, using a monodirectional architecture is not sufficient. As a result, the transformer is a bidirectional model, utilizing the context of the input from both directions, in contrast to many other RNN architectures.

There are two BERT models: one for English and one for multiple languages. We extract 768-dimensional embeddings from both models for English and German. We refer to the English model's embeddings as "bert.en" and to the multilingual model's embeddings as "bert.multi".

4.1.1.3 ALBERT Embeddings

ALBERT [89] is a simpler yet more powerful version of BERT. ALBERT utilizes three techniques to minimize the number of its parameters: factorized embedding parameterization, cross-layer parameter sharing, and inter-sentence coherence loss. First, the model separates the vocabulary embedding size from the hidden layer size. The embeddings learn context-independent representations, and the hidden layers learn context-dependent representations. Hence, separating the two results in a much smaller embedding size compared to BERT. Second, ALBERT shares all its parameters across its layers, which not only greatly reduces the number of parameters, but also results in more stable parameters. Finally, ALBERT uses masked language modeling (MLM) loss and next-sentence-prediction (NSP) loss. The second loss is a binary classification loss that aims to predict whether two given sentences are consecutive or not in the original text they were taken from. Positive examples are sampled consecutively from the same text, while negative samples are taken from different documents. However, this task was later found to be unreliable, and Lan et al. [89] proposed that this is due to it being a very simple task, as it combines topic prediction with coherence prediction. While coherence prediction is hard to learn, topic prediction is rather easy to learn, and is close to what the network learns using the MLM loss. Hence, Lan et

al. [89] proposed using a sentence-order prediction (SOP) loss instead. This loss is based on coherence and still takes into account inter-sentence modeling but does not include a topic prediction. Positive examples of coherence loss are created using the same technique as the NSP loss, but negative examples are simply swapped versions of the positive ones. This forces the network to understand fine-grained details about sentence coherence and relationship.

Since the only available ALBERT models are English, we extract ALBERT features using the English translations of the transcription. The embeddings have 768 dimensions, and we refer to them as albert.en features.

4.1.1.4 ELMo Embeddings

ELMo [90] is another recent development in NLP models. Unlike BERT and ALBERT, it does not rely on transformers or attention mechanisms. Instead, its architecture consists of two layers of bidirectional LSTMs, each of which is composed of two layers of different types of language models: Given a sequence $(t_1 \dots t_N)$ of N tokens, a forward language model predicts a token t_k based on the history $(t_1 \dots t_{k-1})$, while a backward language model predicts t_k based on future tokens $(t_{k+1} \dots t_N)$. A bidirectional language model (biLM) combines both forward and backward language models. As a result, using biLMs gives the network the capability to contextualize word embeddings. However, unlike transformers, it does not produce a deep bidirectional representation by simultaneously capturing both previous and subsequent contexts, but rather concatenates them later.

ELMo forms embeddings by combining the intermediate layer representations from each biLM layer and the character-based representation in the token layer. In the simplest case, ELMo extracts embeddings from the top layer only. As the weighted sum of the token and intermediate layer

representations, we extract English ELMo embeddings with 1024 dimensions. They are referred to as elmo.en.

4.1.2 Audio Features

Numerous audio features have been suggested in the literature. However, the best results were obtained by extracting features from the VGGish network [72], which is trained on a very large dataset and thus can learn rich audio representations. Hence, we extract short-term audio features from the pre-trained VGGish network. First, the recordings are divided into 0.98-second segments with a 100-millisecond overlap. Afterwards, each segment is converted to a log-mel spectrogram and finally fed into the network. We extract audio features from the 128-dimensional final fully connected layer. Finally, as with [66][71], we employ the turn information to adjust the influence of the interlocutor, as this was shown to improve performance. In the first approach, we replace the interlocutor's speech features with zeros, as reducing the influence of the interlocutor's audio was shown to improve model performance. These features are referred to as vggish.100ms.empty. Another approach aims to separate the interlocutor and the subject's features without removing any information. This is done by creating two sets of features, each of which has the features of one speaker removed. Thus, one of is identical to vggish.100ms.empty, and the other has zeroes instead of the speaker's features, and complements the first set. The features are then concatenated, creating the new features, which have 256 dimensions and we refer to as vggish.100ms.doubled.

4.1.3 Visual Features

Visual features convey information about subjects' facial expressions, which can aid in emotion recognition. Thus, to extract visual features, we first extract video frames. We then detect faces using the dlib library [91]. Specifically, we use the CNN-based detector instead of the HOG-based detector to improve facial recognition. We then align the detected faces and resize the images to

64 by 64 pixels. Finally, since there are frames in which no face is detected, we use black images instead. To improve the accuracy and robustness of our models, we apply data augmentation by randomly rotating the images by up to 15 degrees. The images are also randomly flipped horizontally, and the brightness and contrast are randomly varied by 12.5% and 50%, respectively. Finally, the images are normalized by dividing the pixel values by 255.

4.1.3.1 VGG-Face Features

Several visual features were previously utilized in the literature. On the SEWA database, the best results were achieved using VGG-Face and DenseNet features [46][71]. Hence, we pre-train a VGG-Face network [46] on the FER+ dataset [8]. We then fine-tune the network on arousal and valence using the detected faces from the SEWA dataset. The network consists of a series of convolutional layers with small filter sizes interleaved with max-pooling and dropout layers. The architecture is shown in Figure 6. Even though the network is trained on recognizing 8 discrete emotions, deep visual features contain useful information that can be used to predict arousal and valence. We extract visual features from the conv5 layer and apply global average pooling, as it has been shown to contain more generalizable information compared to the conv6 layer [[66]]. We refer to these features as vgg.conv5.finetune.av.

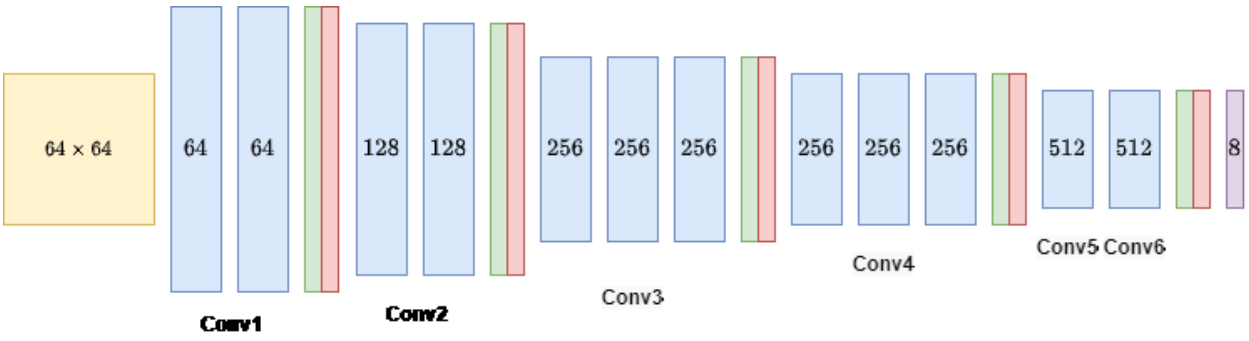

Figure 6: Our custom VGG-Face network. The yellow box represents the input. The blue layers are 2D convolutional layers followed by a ReLu activation. Each set of convolutions is followed by a green max-

pooling layer and a red dropout layer. Finally, the outputs are connected to the purple fully connected layer for classification.

4.1.3.2 DenseNet Features

Similar to [66], we extract visual features from a DenseNet [68] network pre-trained on the FER+ dataset [8] and fine-tuned on the extracted faces from the SEWA dataset. It is composed of a series of blocks, each of which is connected to all the layers preceding it in the same block. This strengthens feature propagation and helps in the solution of the vanishing gradient problem. Moreover, as a result of feature reuse, each layer only needs to learn a small set of features. Hence, the architecture has a much smaller number of parameters compared to VGGFace. The general DenseNet architecture is shown in Figure 7. Our DenseNet is implemented with a growth rate of 12 and a depth of 100 equally divided into three blocks with bottleneck layers and a compression factor $\theta = 0.5$. We extract features from the last maxpooling layer and refer them as denenset.finetune.av.

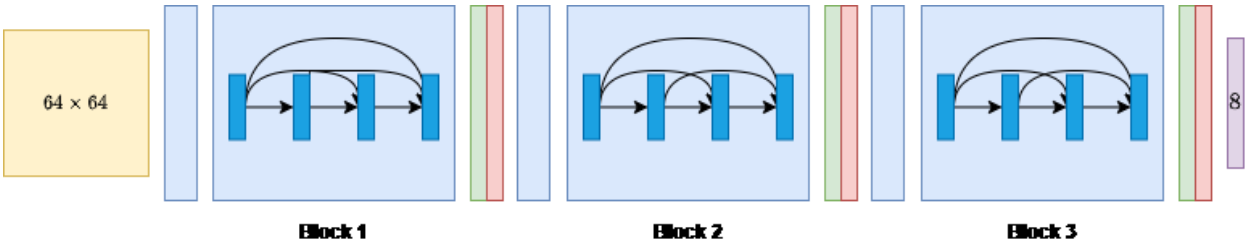

Figure 7: A general DenseNet architecture with three blocks. The yellow box represents the input image, and the light and dark blue layers represent convolutions. Each block is followed by a green max-pooling layer and a red dropout layer. The output of the final dropout layer is fed into the purple fully connected classification layer.

4.2 Unimodal TCN Implementation

All networks are implemented using TensorFlow 2 and Python 3.7. Each TCN is trained for a maximum of 120 epochs using the Adam optimizer with a learning rate of 0.01 which is halved every 50 epochs. We employ Gaussian process optimization to determine the optimal network configuration for each type of feature. We optimize the number of layers, the number of filters, the filter size, and the dropout, starting with a TCN that has five layers, 125 filters, a filter size of five, and a dropout of 0.25. Additionally, we perform smoothing on both the inputs and outputs with a smoothing window of size 5. As discussed in Section 3.1, we were unable to obtain the labels for the testing set. Hence, we use cross-validation to employ the training set for training and validation. We use the validation set for testing. This allows us to obtain a fair assessment of our models. However, to compare our results to existing work, we train our networks on the entire training set and use the validation results for comparison since, to the best of our knowledge, all existing studies on the SEWA database report validation results (the ones that have access to the testing labels also report testing set results).

The output sequences are evaluated on each output sequence using the concordance correlation coefficient (CCC), which is defined as the correlation coefficient between two sequences x and y:

$$\rho_c = \frac{2\rho}{{\sigma_x}^2 + {\sigma_y}^2 + \left(\mu_x - \mu_y\right)^2} \tag{4}$$

Where ρ is the Pearson Correlation Coefficient (PCC), μ_x and μ_y are the means of x and y respectively, and σ_x and σ_y are their corresponding standard deviations. In this case, sequence x would be the predicted sequence, and y would be the true label sequence. As opposed to the MSE loss, which has been used before for similar tasks, we use $1 - \rho_c$ as the loss function during

training, as proposed by Trigeorgis et al. [92]. For unimodal training, we implement the TCN model shown in Figure 8.

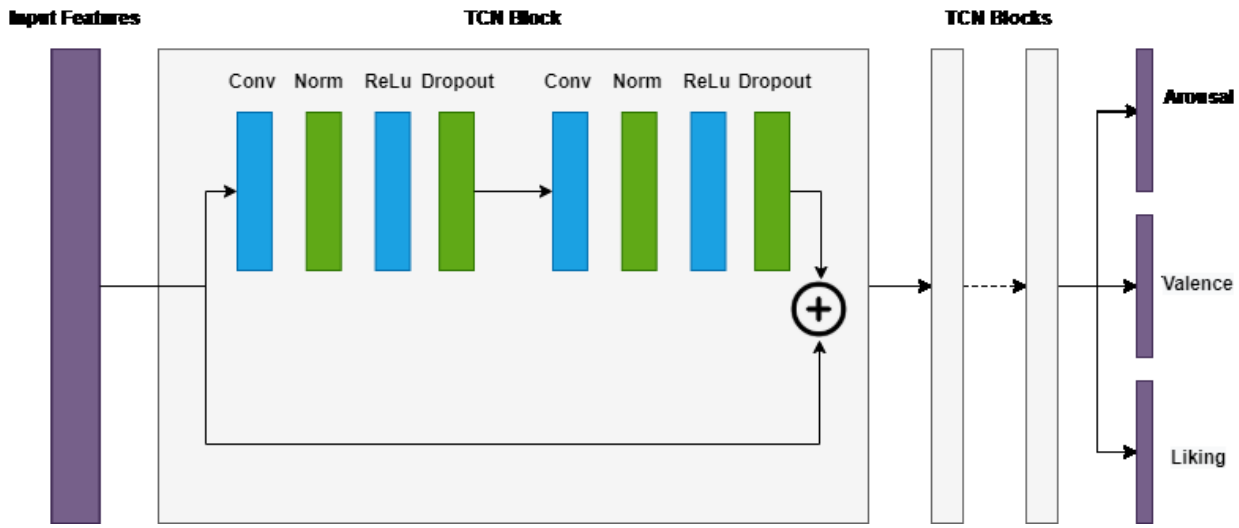

Figure 8: Our proposed TCN model

4.3 Multimodal TCN Implementation

After comparing the performance of different features from each modality (which we perform in Chapter 5), we will select the features with the best performances to use for multimodal fusion.

We will explore three multimodal fusion mechanisms, feature-, decision-, and model-level fusion.

4.3.1 Feature-Level Fusion Implementation

For our feature-level fusion implementation, we will train a TCN model on the concatenated set of our best-performing input features. This solution is similar to the unimodal TCN, except that the input features pertain to multiple modalities instead of only one.

4.3.2 Decision-level Fusion Implementation

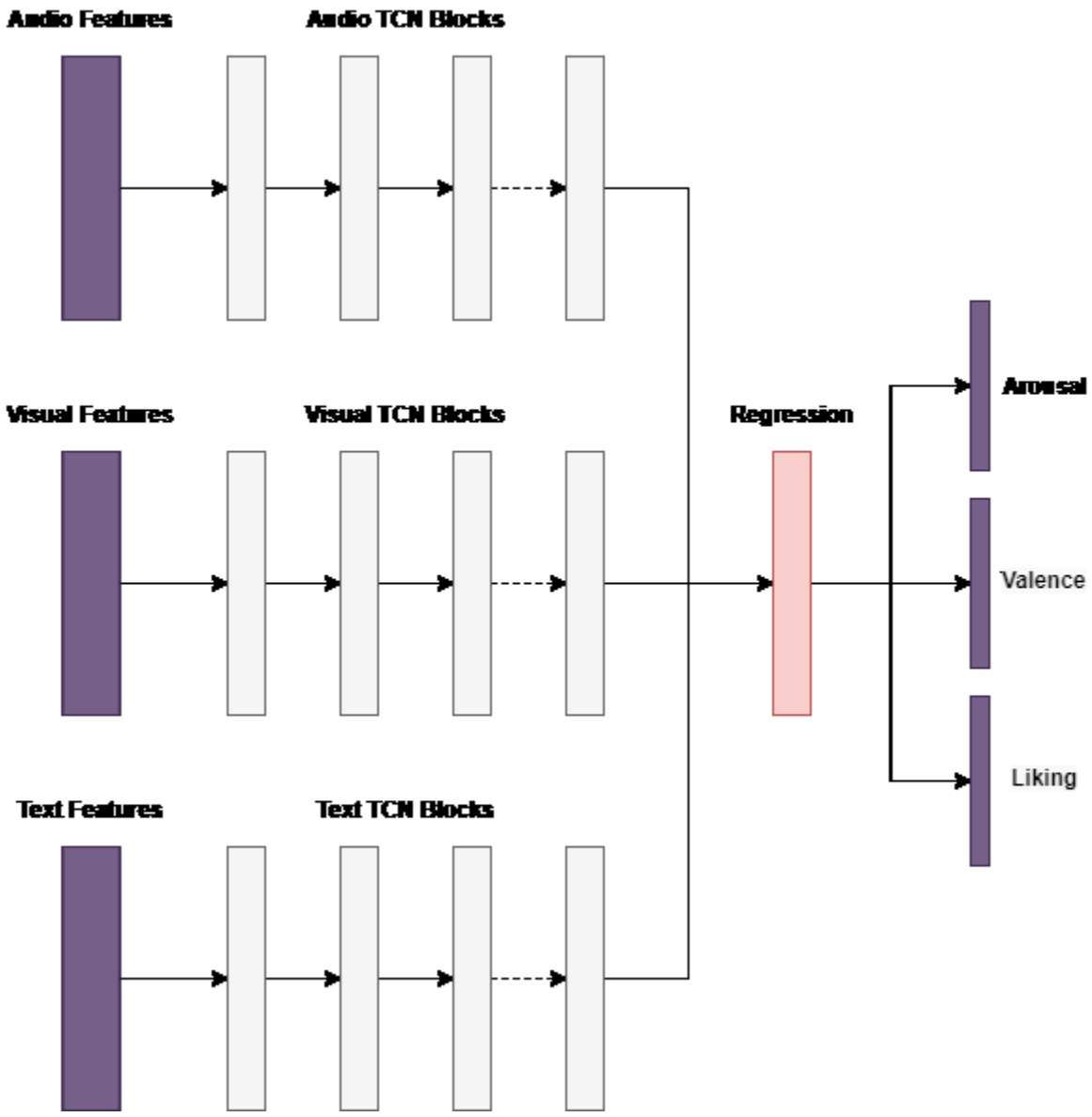

Figure 9: Decision-level fusion architecture

For our decision-level fusion, we train a separate TCN model for each modality. The predictions of each modality are then fused using a regression model. Our architecture is shown in Figure 9 below.

4.3.3 Model-level Fusion Implementation

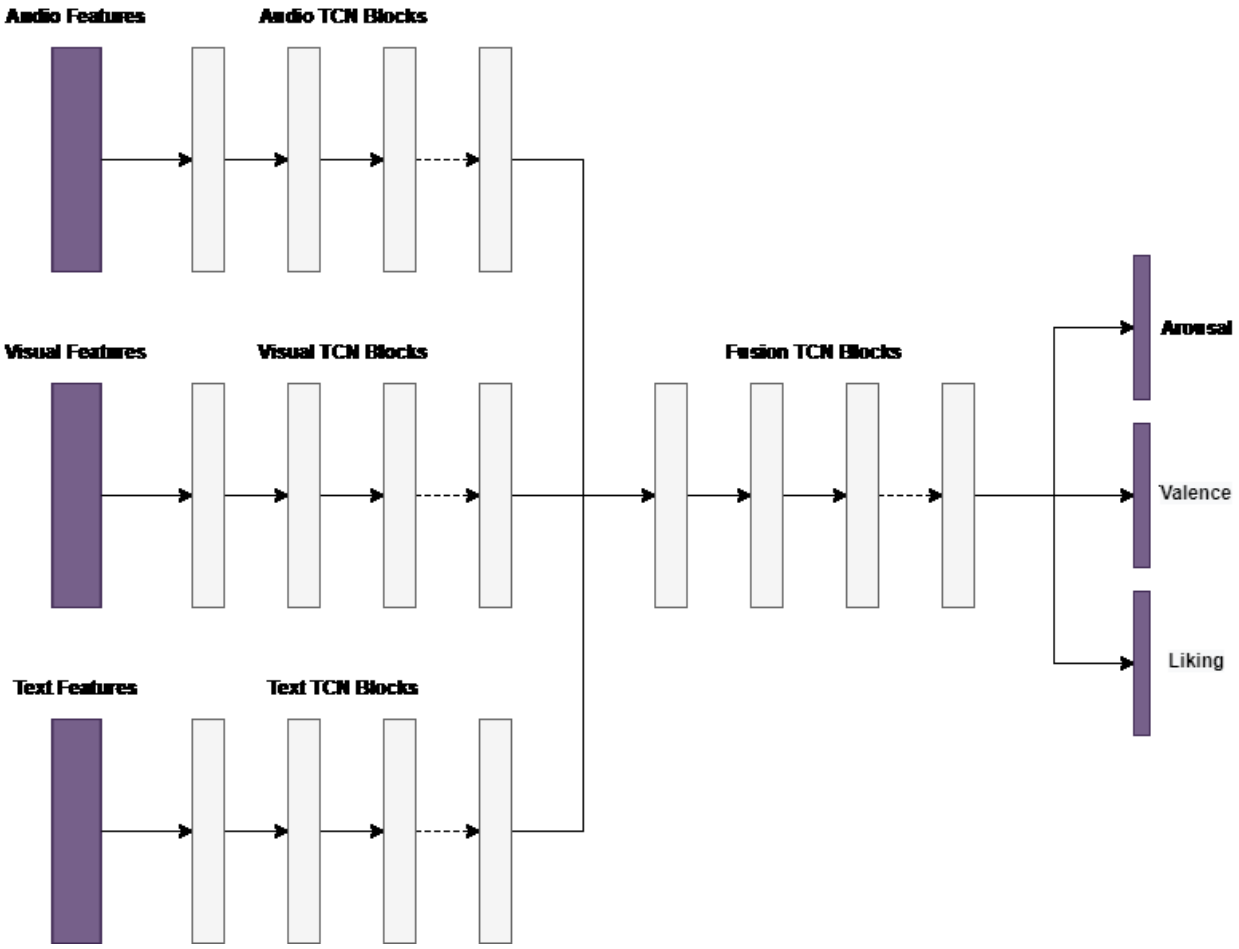

Figure 10: Model-level fusion architecture

Finally, we implement a model-level fusion model. For this architecture, we utilize separate TCNs for each feature or modality. We concatenate intermediate representations from each modality and then feed them to another TCN. We treat the model as one entity and perform training and validation on the whole system. Our network architecture is shown below.

4.4 Conclusion

In this chapter, we discussed our proposed method for continuous emotion recognition. We proposed using features from the audio, visual, and text modalities. For the audio modality, we

utilized the vggish network to extract deep features. For the visual modality, we utilized both the VGGFace network and DenseNet to extract deep visual features. For text, we utilized a variety of deep features extracted from several models.

We proposed multimodal fusion in order to combine the results of different modalities. The three methods we compare are feature-level, decision-level, and model-level fusion.

Chapter 5 Results and Discussion

In this chapter, we present the results of training the TCN on the audio, visual, and textual modalities. For each modality, we train the networks on several types of features and compare their performances. Since we do not have access to the testing portion of the dataset, we divide the training data using cross-validation into training and validation portions. We use Gaussian processes for hyperparameter optimization. After finding the network configuration with the best cross-validation results, we use the validation portion of the data for testing. However, when comparing our results with previous work, we use the full training data for training and compare the results on the validation data. We utilize the CCC for measuring our performance.

We report on our unimodal results in section 5.1, and then compare them with previous work in section 5.2. We follow this with our multimodal fusion results in 5.2. We then compare the performances of our different fusion methods in section 5.3, and we finally compare our best performing networks with previous work and other types of networks.

5.1 Results on Unimodal Features

5.1.1 Results on the Text Modality

We first train different models on our extracted textual features. The results of training TCNs on the Word2Vec embeddings for the original German (word2vec.de) and English translated (word2vec.en) text is shown in Table 1.

Table 1: Testing results on the English and German Word2Vec models.

Embedding	Hyperparameters				Results			
	Number of Layers	Number of Filters	Filter Size	Dropout	Arousal	Valence	Liking	Average
word2vec.de	5	69	8	0.01	0.5446	0.5500	0.3318	0.4755
word2vec.en	5	95	6	0.3397	**0.5930**	**0.5629**	**0.3823**	**0.5128**

The results for the English embeddings outperform those for the German embeddings. Thus, despite the inherent limitations of the translation, the English embeddings captured more pertinent information about the emotions expressed in the textual transcriptions. This is perhaps because the word2vec.en was trained on a far larger corpus of text than the word2vec.de. The English Word2Vec model was trained on part of the Google News dataset with 100 billion words [93] and contains word vectors for 3 million words and phrases. On the other hand, the German Word2Vec model was trained on about 651 million words from German Wikipedia [69], which is a much smaller dataset. Thus, the English Word2Vec model was trained on a dataset over 100 times larger than the German model.

Table 2 summarizes the findings from training a TCN on the BERT, ALBERT, and ELMo embeddings. Please note that BERT.multi refers to the multi-lingual version of the model while bert.en refers to the English version of the model.

Table 2: Testing results on the BERT, ALBERT, and ELMo embeddings.

Embedding	Hyperparameters				Results			
	Number of Layers	Number of Filters	Filter Size	Dropout	Arousal	Valence	Liking	Average

bert.multi	9	16	6	0.01	0.3976	0.3868	0.3021	0.3622
bert.en	12	20	3	0.0103	0.5325	0.5014	0.3529	0.4623
albert.en	10	16	6	0.01	0.5216	0.4835	0.4212	0.4754
elmo.en	6	68	7	0.0396	0.5313	0.4980	0.3142	0.4478

The bert.multi features exhibit the worst performance, possibly due to the fact that the BERT model is trained on 102 languages, which makes it a highly generalized model and less suitable for specialized tasks such as continuous emotion recognition. Even though the ELMo embeddings have 1024 dimensions, they perform worse than the BERT and ALBERT embeddings. This suggests that the biLSTM approach used in the ELMo model is not as effective as transformers in capturing contextual semantic meanings for this emotion recognition task. Finally, the BERT.en and ALBERT.en features have similar performance, with bert.en performing better on arousal and ALBERT.en performing better on valence, and bert.en performing slightly better on average. Notably, neither embedding outperformed the word2vec.en features across all dimensions. On average, the Word2Vec embeddings perform best on the testing data, and thus we utilize both for multimodal fusion. This could be because those embeddings have more dimensions are thus less suited for training.

5.1.2 Results on the Audio Modality

We train a TCN on our deep audio features. The results are shown in Table 3 below.

Table 3: Performance Results on the Audio Modality

Features	Hyperparameters				Results		
	Number of Layers	Number of Filters	Filter Size	Dropout	Arousal	Valence	Liking

vggish.100ms.empty	5	114	8	0.01	0.6163	0.5050	0.2315
vggish.100ms.doubled	5	16	6	0.0145	**0.6314**	**0.5201**	**0.2613**

Contrary to the text modality, the audio features do not provide a lot of information about the liking dimension. They do, however, provide better results on arousal and valence. Since the double features produce better results, we select them for multimodal fusion.

5.1.3 Results on the Visual Modality

We present the results on the features of the visual modality in Table 4 below.

Table 4: Results for different features the Visual Modality

Features	Hyperparameters				Results		
	Number of Layers	Number of Filters	Filter Size	Dropout	Arousal	Valence	Liking
vgg.conv5.finetune.av	11	153	2	0.0253	0.6136	**0.7007**	**0.1811**
denenset.finetune.av	5	235	7	0.0171	**0.6214**	0.6818	0.1509

The two features produce similar results, with the DenseNet features performing better on arousal and the VGGFace features performing better on valence and liking. The results for both are nevertheless very similar. Hence, we utilize both features for multimodal fusion.

5.2 Performance Comparison with Previous Work

In this section, we compare the performance results of our extracted features with those of previous work. Since we do not have access to the testing data, we train our model the training data and compare our validation results.

5.2.1 Performance Comparison on the Text Modality

We compare our findings to those of previous studies [65][71] in Table 5 below. To ensure that we make a fair comparison, we train the network on the training portion of the dataset and compare the validation results.

Table 5: Performance Comparison of the Word2Vec features with Previous Work

Embedding	Proposed TCN Architecture			Huang et al. [65]			Zhao et al. [71]		
	Arousal	Valence	Liking	Arousal	Valence	Liking	Arousal	Valence	Liking
word2vec.en	**0.5918**	**0.5560**	0.3770	NA	NA	NA	0.4691	0.517	**0.3942**
word2vec.de	**0.5982**	0.5473	**0.4593**	0.562	0.522	0.398	0.4866	**0.5603**	0.4356

Our results outperform the LSTM models used in the current state-of-art. Even though the model in [71] slightly outperforms ours for the valence, our model performs better on the arousal and liking dimension, the latter being the most important for the text modality

5.2.2 Performance Comparison on the Audio Modality

We compare our results with previous work below. We train our model on the training portion of the data and compare the validation results.

Table 6: Performance comparison of the Results on the Audio Modality with Previous Work

Features	Proposed Method			Zhao et al. [71]		
	Arousal	Valence	Liking	Arousal	Valence	Liking
vggish.100ms.empty	**0.6408**	**0.5231**	**0.3063**	0.6041	0.5107	0.1559
vggish.100ms.double	0.6473	0.5452	0.3034	N/A	N/A	N/A

The vggish.100ms.empty features outperform Zhao et al. [71] on all three dimensions. The difference is especially noticeable on the liking dimension, where the CCC difference is about 0.15.

5.2.3 Performance Comparison on the Visual Modality

We then compare our results to the state-of-art. To do that, we train our network on the training data and compare our validation results. The results are shown below.

Table 7: Performance comparison on our proposed method and previous work

Features	Proposed Method				Zhao et al. [71]			
	Arousal	Valence	Liking	Average	Arousal	Valence	Liking	Average
vgg.conv5.finetune.av	**0.6547**	0.6883	0.2205	**0.5212**	0.6224	**0.7006**	0.1658	0.4962
denenset.finetune.av	0.6404	0.6911	**0.2653**	**0.5323**	**0.6897**	0.6913	0.2107	0.5305

Our features outperform the current state-of-the art on average. However, we were unable to achieve better results on all three dimensions simultaneously. Since both features have comparable performance, we utilize them both for multimodal fusion.

5.3 Multimodal Fusion Results

In multimodal fusion, we construct a model that is trained on the best features from each modality, namely word2vec.en and word2vec.de for text, vggish.100ms.doubled for audio, and densenet.conv.finetune.av and vgg.conv5.finetune.av features for video.

5.3.1 Performance Results Using Feature Level Fusion

In this section, we report our performance results on arousal, valence, and liking when utilizing feature-level fusion on the audio, visual, and textual features.

In feature level fusion, we concatenate the features from the three modalities and then train a TCN on the concatenated features. The results are shown below.

Table 8: Performance results of feature-level fusion using features from the three modalities.

Features	Hyperparameters				Results		
	Number of Layers	Number of Filters	Filter Size	Dropout	Arousal	Valence	Liking
vggish.100ms.doubled + vgg.conv5.finetune.av + denenset.finetune.av + word2vec.en +word2vec.de	5	59	8	0.01	0.7204	0.7041	0.3809

The combination of different features significantly improves performance on arousal and valence. However, the performance on the liking dimension is comparable to the unimodal predictions from the Word2Vec features, even though we utilized both features together. This could be the result of the high dimensionality of the combined input data, which prevents the network from extracting liking information from the relevant input data.

5.3.2 Performance Results Using Decision Level Fusion

In this section, we report our results on arousal, valence, and liking when using decision-level fusion to combine our results.

For each modality, we train the TCN on the features using cross-validation. After finding the optimal network configuration using hyperparameter optimization, we train a linear regression model on the outputs of all three networks to predict the final results. We report the results in the table below.

Table 9: Performance results using decision-level fusion on all three modalities

Features	Hyperparameters For each modality: (number of layers, number of filters, filter size, dropout)	Results		
		Arousal	Valence	Liking
Audio: vggish.100ms.doubled Visual: vgg.conv5.finetune.av + denenset.finetune.av Text: word2vec.en + word2vec.de	Audio: (5, 16, 6, 0.0145) Visual: (11, 64, 2, 0.01) Text: (5, 101, 8, 0.0448)	0.7276	0.7483	0.4164

The performance improves on the three modalities compared to the unimodal networks. Since the modalities are separate, the inputs do not have a high number of dimensions.

5.3.3 Performance Results Using Model Level Fusion

In this section, we report on our results on arousal, valence, and liking when utilizing a model-level fusion architecture on the audio, visual, and textual features.

In model-level fusion, we test different combinations of data and compare the results. We report our best performing TCN in the table below.

Table 10: Performance results of our proposed model-level fusion TCN architecture

Features	Hyperparameters For each modality: (number of layers, number of filters, filter size)	Results		
		Arousal	Valence	Liking
Audio: vggish.100ms.doubled Visual: vgg.conv5.finetune.av + denenset.finetune.av Text: word2vec.en + word2vec.de	Audio: (1, 17, 2) Visual: (2, 256, 8) Text: (6, 122, 8) Rest of the net: (5, 35, 8) Dropout: 0.0102	0.7774	0.7435	0.4360

The model improves the performance over the individual modalities on all three dimensions.

5.4 Comparison of Different Fusion Methods

We compare the results of our three fusion methods in the table below.

Table 11: Performance comparison of our three proposed TCN architectures

Feature-level Fusion			Decision-level Fusion			Model-level Fusion		
Arousal	Valence	Liking	Arousal	Valence	Liking	Arousal	Valence	Liking
0.7204	0.7041	0.3809	0.7276	**0.7483**	0.4164	**0.7774**	0.7435	**0.4360**

Our model-level fusion architecture significantly outperforms the other methods on arousal. On the valence dimension, the decision-level fusion architecture slightly outperforms the other two. On liking, model-level fusion outperforms the other two networks. Taking into account all three dimensions, model-level fusion outperforms the other methods. The use of intermediate features instead of the final results for fusion appears to be more helpful for predicting emotions. Thus, we select it as our best-performing model to compare with the state-of-the-art.

5.5 Comparison with Other Methods

Since we do not have access to the testing data, we train our networks on the training data and compare the performances on the validation data. We compare our best performing network results with the state-of-the-art results [71].

Table 12: Performance comparison of our proposed method with the state-of-the-art

Proposed TCN Architecture				Zhao et al. [71]			
Arousal	Valence	Liking	Average	Arousal	Valence	Liking	Average
0.8063	0.7443	**0.5392**	**0.6967**	0.7914	**0.7823**	0.5098	0.6945

Our network outperforms the state of the art on arousal and liking, but falls short on valence. However, our proposed architecture has on average better predictions compared to Zhao et al. [71].

However, because we do not have access to the testing dataset, we are unable to make a direct comparison with previous work. Therefore, we implement a peephole LSTM similar to the one used by Zhao et al. [71] and we train it using cross-validation, similar to what we did with our TCN models. Additionally, we implement a 1-layer LSTM and a 1-layer GRU in order to perform a wide comparison. The results are shown below.

Table 13: Performance comparison of our TCN with several types of RNN networks

	Arousal	Valence	Liking	Average
Proposed TCN Architecture	**0.7774**	0.7435	**0.4360**	**0.6523**
Peephole LSTM	0.6623	0.6793	0.21834	0.5200
1-layer LSTM	0.7511	**0.7608**	0.2460	0.6300
1-layer GRU	0.7474	0.7394	0.4033	0.5860

Our proposed architecture outperforms all other methods on arousal and liking but not on valence. The LSTM slightly outperforms our network on valence. However, the difference on liking is massive, and our network performs much better on average.

5.6 Conclusion

In this chapter, we discussed the performance results of our proposed features as well as our proposed fusion strategies. For the audio modality, we selected the vggish.100ms.double features for fusion as they performed better than the vggish.100ms.empty features. For the visual modality, we selected both the DenseNet and VGGFace features for fusion, as they performed similarly well. For the text modality, we selected the Word2vec.en and Word2vec.de features for fusion. All of our unimodal networks outperformed similar methods in previous work.

We then utilized the selected features for multimodal fusion. Our model-level fusion architecture outperformed our decision-level and feature-level architectures. It also outperformed previous work on those modalities and our implementations of similar networks.

Chapter 6 Conclusion and Future Work

In this chapter, we summarize our work and propose future work.

6.1 Conclusion

The goal of this thesis was to utilize temporal convolutional networks for multimodal emotion recognition. The complexity of this problem comes from the fact that the emotions are spontaneous and continuously changing.

We utilized audio, visual, and textual features for our proposed method. Specifically, we utilized deep features extracted from pre-trained networks. We used temporal convolutional networks instead of LSTMs, which are used in the state-of-the-art emotion recognition methods. To accommodate the different modalities, we use different multimodal fusion strategies. We summarize our contributions as follows:

1. We utilized temporal convolutional networks for multimodal emotion recognition. Generally, the state-of-the-art methods utilize LSTMs or other RNNs. Our network instead relies on dilated convolutional layers in order to capture temporal dependencies.
2. We extracted several types of deep features for each modality and compared their performance.
3. We perform multimodal fusion to combine the different features and compare our results with the state-of-the-art results for our data.

Our proposed method achieves superior results compared to the state-of-the-art. Furthermore, given with the architectural advantages TCNs provide, we recommend utilizing TCNs for emotion recognition in the future.

6.2 Future Work

We summarize our future work as follows:

1. Using deep textual features extracted from modified BERT models.
2. Fine-tuning BERT and ALBERT for emotion recognition before extracting textual features. This has been shown to improve performance for deep textual features.
3. Utilizing multilingual data for training and testing. The dataset we use for this work is exclusively in German. However, the SEWA database includes data for other languages such as Hungarian and Chinese. This data can be used to test the generalizability of our model on data from different languages. Additionally, it can be used for training to improve cross-cultural emotion recognition.

References

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

[2] R. W. Picard, “Affective computing for HCI,” in HCI, vol. 1, pp. 829–833, 1999.

[3] T. Partala and V. Surakka, “The effects of affective interventions in human–computer interaction,” Interacting with Computers, vol. 16, pp. 295–309, 2004.

[4] M. Pantic, N. Sebe, J. F. Cohn, and T. Huang, “Affective multimodal human-computer interaction,” in Proceedings of the 13th annual ACM international conference on multimedia, 2005, pp. 669–676.

[5] Ringeval, F., Schuller, B., Valstar, M., Cowie, R., Kaya, H., Schmitt, M., 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 (AVEC'18) pp. 3–13. Association for Computing Machinery, New York, NY, USA (2018). doi: 10.1145/3266302.32663164

[6] 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.

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

[8] 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.

[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] 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.

[11] 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

[12] 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

[13] 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

[14] 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.

[15] 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.

[16] H. Chen et al., “Efficient Spatial Temporal Convolutional Features for Audiovisual Continuous Affect Recognition”. In Proceedings of the 9th International on Audio/Visual Emotion Challenge and Workshop, 2019, pp. 19–26.

[17] 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

[18] 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.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

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

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

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

[23] Jean Kossaifi, Robert Walecki, Yannis Panagakis, Jie Shen, Maximilian Schmitt, Fabien Ringeval, Jing Han, Vedhas Pandit, Antoine Toisoul, Bjorn Schuller, and et al. Sewa db: A rich database for audio-visual emotion and sentiment research in the wild. IEEE Transactions on Pattern Analysis andMachine Intelligence, 43(3):1022–1040, Mar 2021.

[24] Fabien Ringeval, Björn Schuller, Michel Valstar, Nicholas Cummins, Roddy Cowie, Leili Tavabi, Maximilian Schmitt, Sina Alisamir, Shahin Amiriparian, Eva-Maria Messner, Siyang Song, Shuo Liu, Ziping Zhao, Adria Mallol-Ragolta, Zhao Ren, Mohammad Soleymani, and Maja Pantic. 2019. AVEC 2019 Workshop and Challenge: State-of-Mind, Detecting Depression with AI, and Cross-Cultural Affect Recognition. In Proceedings of the 9th International on Audio/Visual Emotion Challenge and Workshop (AVEC '19).

Association for Computing Machinery, New York, NY, USA, 3–12. DOI:https://doi.org/10.1145/3347320.3357688

[25] Mohammadreza Amirian, Markus K¨achele, Patrick Thiam, Viktor Kessler, and Friedhelm Schwenker. Continuous multimodal human affect estimation using echo state networks. In Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge, AVEC '16, page 67–74, New York, NY, USA, 2016. Association for Computing Machinery.

[26] 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), 71 2013, pp. 1–8.

[27] T. L. Nwe, S. W. Foo, and L. C. De Silva. Classification of stress in speech using linear and nonlinear features. In Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP'03). 2003 IEEE International Conference on, volume 2, pages II-9. IEEE, 2003.

[28] 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

[29] 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

[30] 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.

[31] 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.

[32] T. R. Almaev and M. F. Valstar, "Local Gabor Binary Patterns from Three Orthogonal Planes for Automatic Facial Expression Recognition," 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, 2013, pp. 356-361, doi: 10.1109/ACII.2013.65.

[33] Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., & Gong, Y. (2010, June). Locality-constrained linear coding for image classification. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on (pp. 3360-3367). IEEE.

[34] 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.

[35] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, et al. Challenges in representation learning: A report on three machine learning contests. In Neural Information Pro-cessing, pages 117–124. Springer, 2013.

[36] Linlin Chao, Jianhua Tao, Minghao Yang, Ya Li, and Zhengqi Wen. 2015. Long Short Term Memory Recurrent Neural Network based Multimodal Dimensional Emotion Recognition. In Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge (AVEC '15). Association for Computing Machinery, New York, NY, USA, 65–72. DOI:https://doi.org/10.1145/2808196.2811634

[37] 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.

[38] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T., Caffe: Convolutional Architecture for Fast Feature Embedding, arXiv: 1408. 5093, 2015.

[39] Jing Han, Zixing Zhang, Nicholas Cummins, and Björn Schuller. Adversarial training in affective computing and sentiment analysis: Recent advances and perspectives, 2018.

[40] J. Han, Z. Zhang, Z. Ren, F. Ringeval and B. Schuller, "Towards Conditional Adversarial Training for Predicting Emotions from Speech," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 6822-6826, doi: 10.1109/ICASSP.2018.8462579.

[41] 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

[42] 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.

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

[44] 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.

[45] 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

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

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. CoRR abs/1512.03385 (2015). arXiv:1512.03385 http://arxiv.org/abs/1512.03385

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

[49] 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.

[50] Chang, W.Y., Hsu, S.H., Chien, J.H.: Fatauva-net : An integrated deep learning framework for facial attribute recognition, action unit (au) detection, and valence-arousal estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (2017)

[51] 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

[52] Kalani Wataraka Gamage, Ting Dang, Vidhyasaharan Sethu, Julien Epps, and Eliathamby 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, AVEC'18, page 47–55, New York, NY, USA, 2018. Association for Computing Machinery.

[53] P. C. Ellsworth and K. R. Scherer, "APPRAISAL PROCESSES IN EMOTION," Handb. Affect. Sci., pp. 572–595, 2003.

[54] A. Moors, P. C. Ellsworth, K. R. Scherer, and N. H. Frijda, "Appraisal Theories of Emotion: State of the Art and Future Development," Emot. Rev., vol. 5, no.2, pp. 119–124, Apr. 2013.

[55] 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.

[56] 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.

[57] Jian Huang, Ya Li, Jianhua Tao, Zheng Lian, Zhengqi Wen, Minghao Yang, and Jiangyan Yi. 2017. 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 (AVEC '17). Association for Computing Machinery, New York, NY, USA, 11–18. DOI:https://doi.org/10.1145/3133944.3133946

[58] S. Marioryad 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.

[59] 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.

[60] Kyle, K. NLP TOOLS FOR THE SOCIAL SCIENCES. https://www.linguisticanalysistools.org. Last accessed 26 Jul 2021.

[61] 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

[62] 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.

[63] 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.

[64] 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.

[65] 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.

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

[67] 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.

[68] 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.

[69] German Word Embeddings. http://devmount.github.io/GermanWordEmbeddings/, 2017. [Online; accessed 18-July-2021].

[70] English Word Embeddings. https://drive.google.com/le/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing, 2017. [Online; accessed 18-July-2021].

[71] 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.

[72] 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.

[73] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV) (ICCV ’15). IEEE Computer Society, 4489–4497. https://doi.org/10.1109/ICCV. 2015.510

[74] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann Lecun, and Manohar Paluri. 2017. A Closer Look at Spatiotemporal Convolutions for Action Recognition. (2017).

[75] Chaolong Li, Zhen Cui, Wenming Zheng, Chunyan Xu, and Jian Yang. 2018. Spatio-Temporal Graph Convolution for Skeleton Based Action Recognition. CoRR abs/1802.09834 (2018). arXiv:1802.09834 http://arxiv.org/abs/1802.09834

[76] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In Conference on Empirical Methods in Natural Language Processing.

[77] Jinming Zhao, Ruichen Li, Jingjun Liang, Shizhe Chen, and Qin Jin. 2019. Adversarial Domain Adaption for Multi-Cultural Dimensional Emotion Recognition in Dyadic Interactions. In Proceedings of the 9th International on Audio/Visual Emotion Challenge and Workshop (AVEC '19). Association for Computing Machinery, New York, NY, USA, 37–45. DOI:https://doi.org/10.1145/3347320.3357692

[78] https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.zh.300.bin.gzt

[79] I. Ayoub, V. Heiries and H. A. Osman, "Multimodal Affect Recognition Using Temporal Convolutional Neural Networks," 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 2022, pp. 1-8, doi: 10.1109/IJCNN55064.2022.9892145.

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

[81] 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.

[82] B. Jiang, M. F. Valstar and M. Pantic, "Action unit detection using sparse appearance descriptors in space-time video volumes," 2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG), 2011, pp. 314-321, doi: 10.1109/FG.2011.5771416.

[83] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 886 – 893, June 2005.

[84] 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.

[85] 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.

[86] 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.

[87] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones,Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[88] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert:Pre-training of deep bidirectional transformers for language understanding, 2018.

[89] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R.: ALBERT: A lite BERT for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942 (2019)

[90] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Proc. of NAACL, 2018

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

[92] Trigeorgis, G., Ringeval, F., Brueckner, R., Marchi, E., Nicolaou, M.A., Schuller, B., et al.: Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network, In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai (2016), pp. 5200-5204, doi: 10.1109/ICASSP.2016.7472669.

[93] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.