Multimodal Emotion Recognition

Wen Wu

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy in Machine Learning and Machine Intelligence

Trinity College August 2020

Declaration

I, Wen Wu of Trinity College, being a candidate for the MPhil in Machine Learning and Machine Intelligence, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

The main model in this project is built upon the script provided by Guangzhi Sun [1]. The existing code implements 2D self-attention structure for speaker diarisation and has been modified for this emotion recognition task. An additional time asynchronous branch has been added to the existing system. Details would be discussed in Section 4.3.1 and 4.4.2. The HTK Toolkit [2] was used to implement the model and the Kaldi ASR Toolkit [3] was used to compute pitch in Section 4.3. The PyTorch “transformers" library[1] was used to obtain BERT embeddings. The Google Cloud Speech API[2] was used to generate transcriptions. Section 4.4.2 and 4.6.2 rely on these codes, respectively. The open source “facenet-pytorch" library[3] was used for face detection and the open source pretrained VGG-19 model[4] was modified to extract face embeddings in Section 4.5.

[Word count of the thesis: 12929]

Wen Wu
August 2020

[1] https://pypi.org/project/transformers/

[2] https://cloud.google.com/speech-to-text/

[3] https://pypi.org/project/facenet-pytorch/

[4] https://github.com/WuJie1010/Facial-Expression-Recognition.Pytorch

Acknowledgements

I would like to acknowledge Prof. Phil Woodland and Dr. Chao Zhang for their patient guidance and constant support throughout the project. I sincerely appreciate that they introduced me to the field of machine learning and affective computing. I would also like to thank Xixin Wu, Qiujia Li, Guangzhi Sun, and Florian Kreyssig for their kind help. Working with them was both highly valuable and enjoyable.

This work was supported by Cambridge Trust and China Scholarship Council.

Abstract

One of the major tasks of intelligent human-machine interaction is to empower computers with the ability of “affective computing" [4] such that it can recognize a user’s emotional status and respond to the user in an affective way. This project develops a complete multimodal emotion recognition system that predicts the speaker’s emotion state based on speech, text, and video input.

The system consists of two branches. A time synchronous branch where audio, word embeddings, and video embeddings are coupled at frame level. And a time asynchronous branch where sentence embeddings are combined with their context. These two branches are then fused to make prediction. The system generates state-of-the-art multimodal emotion classification accuracy on IEMOCAP database. In-depth investigation of properties of different modalities and their combination is provided.

The emotion recognition problem is then re-examined. IEMOCAP database contains a large proportion of utterances that human annotators don’t completely agree on their emotion labels. These utterances are more common in reality but are usually ignored by traditional emotion classification problems. In that case, it is more reasonable to match the label distribution of the sentence rather than doing classification. “Soft" labels are then introduced, which improves label distribution matching by a significantly better KL divergence. Different ways of modelling the label distribution are discussed which includes the proposal of Dirichlet prior network.

Table of contents

List of figures

List of tables

Nomenclature

A-softmax	Angular softmax
AER	Automatic Emotion Recognition
ASR	Automatic Speech Recognition
CNN	Convolutional Neural Network
FC	Fully-Connected
GRU	Gated Recurrent Unit
KL	Kullback-Leibler
LLD	Low-level Descriptor
LSTM	Long Short Term Memory
MAP	Maximum a Posteriori
MFB	Mel-Frequency Filter Bank energy
MFCC	Mel-Frequency Cepstral Coefficient
MHA	Multi-Head Attention
MLB	Multimodal Low-rank Bilinear pooling
MLE	Maximum Likelihood Estimate
MLM	Mask Language Model
MTCNN	Multi-task Cascaded Convolutional Networks
NSP	Next Sentence Prediction

POV	Probability of Voicing
ResNet	Residue Network
RNN	Recurrent Neural Network
SER	Speech Emotion Recognition
TDNN	Time Delay Neural Network
UA	Unweighted Accuracy
WA	Weighted Accuracy

Chapter 1

Introduction

Emotions are intrinsically part of human mental activity and play a key role in human decision handling, interaction and cognitive processes [5]. Recognizing emotion is an essential step to have complete interaction between human and machine.

Automatic emotion recognition (AER) has attracted attention due to its wide potential application in environments where machines need to interact or monitor humans. For instance, emotional states can be used to monitor and predict the fatigue state of the driver [6] and intervene or issue an alarm. It can also be used in mental health analysis and health care to provide prescription and accompany depressed patients [7–9]. In addition, AER is important in human-machine interfaces such as chat-bots and voice-assistants. Tracking the user's emotional states can help the agent adapt its response to provide better service. And it can also be used to evaluate the quality of the service provided by the agent [4]. Other applications include online gaming, digital advertisement, hate speech detection in social media, affective learning systems, etc. However, the task of recognizing emotion is challenging because human emotion is very complex in nature. It lacks clear temporal boundaries. It can be easily affected by contextual information. And it is extremely personal. Different individuals express emotion in different ways. New trends in AER research includes transfer learning from automatic speech recognition (ASR) [10, 11] and speaker recognition [12], multi-task learning with, for example, gender recognition as an auxiliary task [13], developing new hierarchical encoder structures [14, 15], and the use of multimodal data.

Humans express emotion by various modalities such as facial expressions, voice characteristics, linguistic content of verbal communications, and body postures. It has been highlighted that an ideal AER system should be multimodal as this is closer to the human sensory system [16, 17]. Combining and collating information from multiple modalities to infer the perceived emotions is beneficial [18]: different modalities can augment or complement each other thus giving richer information. For instance, speech waveform provides voice characteristics such as pitch, text provides linguistic content, and videos provide facial expression such as smile and frown. These augmented and complementary information can make the system more robust to sensor noise if some of the modalities are corrupted and ineffectual, which is especially prevalent in in-the-wild datasets. However, multimodal emotion recognition comes with its own challenge: which modalities should be combined and how. Currently, there is still a lack of consensus on the most efficient mechanism for combining (also called "fusing") multiple modalities.

Emotions can be assessed either by discrete categorical based annotations (i.e., labels such as happiness, anger, and sadness) [19–21] or continuous attribute-based annotations (i.e., activation, valence and dominance) [22, 23]. The former is more intuitive and is more widely used in industry while the latter can track subtle changes in emotion and is usually used in psychological research. As emotional labels are often provided for the whole utterances in many databases [24], there can be a mismatch between short-term inputs at frame level (i.e., the 10 ms acoustic features) and long-term outputs at the utterance level. Common ways to deal with the mismatch can be classified into: i) sequence-to-sequence method: copying the label for the utterance into a frame-level label sequence; ii) sequence-to-one method: using various pooling methods to summarize emotion information in the whole utterance.

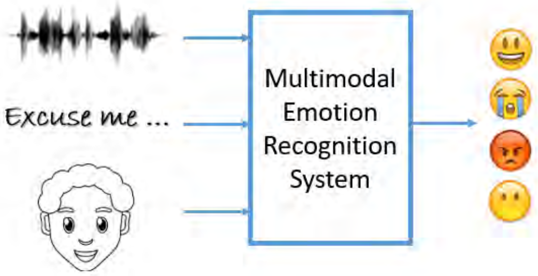


Fig. 1.1 Flow chart of the AER system. Input modalities include speech, text and facial expressions. The system predicts the emotional content of a speaker's utterance.

In this thesis, a multimodal emotion recognition system across audio, text and video is developed. As shown in Figure 1.1, the AER system uses multimodal inputs for determining the emotional content of a speaker's utterance. Features and embeddings are first extracted from the raw inputs and then fed into the system. The system uses sequence-to-one method and predicts categorical discrete label on each sentence.

1.1 Dataset

This thesis uses Interactive Emotional Dyadic Motion Capture (IEMOCAP) database [25], one of the biggest multimodal dyadic conversational dataset available in emotion detection. It consists of approximately 12 hours of audio-visual data, including speech, text transcriptions and facial recordings. IEMOCAP contains 5 sessions acted by 10 different professional actors. In each session one male and female actor either performed selected emotional scripts or improvised hypothetical scenarios. The recorded sessions were then segmented into utterances manually.

Each utterance was annotated by at least three human annotators for categorical labels (neutral, happiness, sadness, anger, etc) or for continuous labels (valence, activation, and dominance). Categorical labels are used in this thesis. An issue with labelling is that there are 25% utterances that don't have majority agreed label. In other words, annotators didn't agree on the emotion of the utterance. This issue will be discussed in detail in Section 4.1 and in Chapter 5. The system will first be trained using strong emotional utterances for cross comparison in Chapter 4 and all data will be taken into account in Chapter 5.

.

1.2 Contribution

- A complete multimodal emotion recognition system is developed which generates state-of-the-art classification results on IEMOCAP database.
- The system used a new model structure which contains a time synchronous branch that couples multiple modalities at frame level and a time asynchronous branch that incorporates context information at utterance level. Two branches are fused to predict the emotion of the speaker.
- The innovative use of BERT in context mode leads to notable increase in classification accuracy as well as more robustness of the system.

- In-depth investigation of the contribution of each modality and the correlation among modalities is provided.
- The innovative use of “soft" labels allows the use of all data (including those don’t have majority agreed labels) and significantly improves the label distribution matching.
- The interesting characteristics of uncertainty in emotion labels are investigated.

1.3 Thesis outline

The structure of the thesis is as follows:

- Chapter 2 reviews the methods of extracting features and embeddings from raw input of audio, text, and video modality.
- Chapter 3 reviews the encoder models and pooling methods that model structured data and combine features from different modalities.
- Chapter 4 presents the emotion classification results of the multimodal emotion recognition system using audio, text, and video inputs. Properties of different modalities and their combination is investigated.
- Chapter 5 re-examines the emotion recognition problem on IEMOCAP and proposes new approaches to matching emotion label distributions instead of doing classification.
- Chapter 6 concludes this thesis and discusses the possible future work.

Chapter 2

Feature representations and embeddings

Chapter 2 and Chapter 3 introduce the background about methods of processing raw inputs and modelling structured data. Given raw inputs such as speech waveform, text and video recordings, the first step is to transform them into more compact features and embeddings. These representations will then be treated as input to the AER system and fed into the model. This chapter discusses the approaches to extract features and generate embeddings for audio, text, and video modality. Models and approaches that further process and combine these representations to predict emotion will be discussed in Chapter 3.

2.1 Audio features

The input speech waveform first needs to be transformed into a sequence of parameter vectors. Audio features used in this project are Mel-Frequency Log Filerbank Energies (MFBs) and pitch.

Mel-Frequency Cepstral Coefficients (MFCCs) are very popular features in speech signal processing. However, MFBs have been shown to be better discriminative features than MFCCs in emotion recognition [25]. This project uses 40-channel log energies extracted using a 25ms window with a frame shift of 10ms.

Pitch often refers to the perception of fundamental frequency (f_0), the frequency at which vocal chords vibrate in voiced sounds. Many studies has established a link on prosody attributes that high pitch levels are related to emotions carrying a high level of activity, such as joy, anxiety, or fear; medium pitch levels account for more neutral attitudes; low pitch levels are related to sober emotions: sadness, calmness, or security [26]. And it has been suggested that acoustic parameters such as pitch plays a crucial role in obtaining better performance in speech emotion recognition (SER) [12]. In this project, log-pitch with Probability of Voicing (POV)-weighted mean subtraction over a 1.5s window computed by the Kaldi toolkit [3] were used.

Besides, there are other popular acoustic feature sets that are widely used in SER such as GeMAPs [27] which includes 88 parameters such as pitch, jitter, shimmer, formants, MFCC, plus the statistical functions (mean, variance, min, max, etc.) applied to these Low-level Descriptors (LLDs) over specified time sliding window. These are not used in this project because most of them should be able to be extracted from spectrum representations with a powerful neural network. There is no need to input these explicitly, otherwise it will cause a large increase in the number of parameters in the input layer. But f_0 information is a separate thing. Filterbank information describes vocal tract shape and vocal spectrum. It doesn't include information about excitation. It is difficult to extract pitch from filterbank spectrum. In order words, filterbank spectrum and pitch are meant to be complementary information.

2.2 Text embeddings

Text embeddings are semantically meaningful distributed representations of text in the form of numeric vectors. Much progress has been made in learning embeddings of individual words such as word2vec [28], GloVe [29] and of phrases and sentences such as doc2vec [30]. In the last few years, the new trend in large unsupervised pre-trained context-specific language models such as ELMo [31], GPT [32], BERT [33] have achieved excellent performance on a variety of language tasks using generic model architectures. Some of the widely used text embeddings are listed in Table 2.1. GloVe and BERT are used in this project.

	concurrency-based	context-specific
word-level	word2vec, GloVe	ELMo
sentence-level	doc2vec	GPT, BERT

Table 2.1 Summary of popular text embeddings

2.2.1 Word embedding: GloVe

GloVe (Global Vectors for Word Representation) is an unsupervised learning algorithm for obtaining vector representations for words trained on aggregated global word-word co-occurrence statistics from the corpus [29]. GloVe has been widely used in AER tasks [34–36].

In the algorithm, the co-occurrence probability is defined as:

$$p_{\mathrm{co}}\left(w_k \mid w_i\right)=\frac{C\left(w_i, w_k\right)}{C\left(w_i\right)} \tag{2.1}$$

where $C\left(w_i, w_k\right)$ counts the co-occurrence between words w_i and w_k. The key idea of GloVe is that the word meanings are captured by the ratios of co-occurrence probabilities rather than the probabilities themselves. The global vector models the relationship between two words regarding to the third context word as:

$$F\left(\left(w_i-w_j\right)^{\top} \tilde{w}_k\right)=\exp \left(\left(w_i-w_j\right)^{\top} \tilde{w}_k\right)=\frac{\exp \left(w_i^{\top} \tilde{w}_k\right)}{\exp \left(w_j^{\top} \tilde{w}_k\right)}=\frac{p_{\mathrm{co}}\left(\tilde{w}_k \mid w_i\right)}{p_{\mathrm{co}}\left(\tilde{w}_k \mid w_j\right)} \tag{2.2}$$

The loss function for the GloVe model is designed to preserve the above formula by minimizing the sum of the squared errors.

In comparison with word2vec which is based on local concurrency of a context window, GloVe takes the global context into account. They are both learned based on word concurrency but not sequential context. ELMo is context-specific which uses a language model based on a bi-directional LSTM to bring semantic information to the word vector.

2.2.2 Sentence embedding: BERT

BERT, short for Bidirectional Encoder Representations from Transformers, is a popular technique for pre-training contextualized universal sentence embeddings based on transformer model [37]. As shown in Figure 2.1, transformers use stacked self-attention and point-wise, fully-connected (FC) layers for both the encoder and decoder. Each layer has two sub-layers: a multi-head self-attention mechanism followed by a simple, position-wise fully-connected feed-forward network. Residual connections are implemented around each of the two sub-layers, followed by layer normalization.

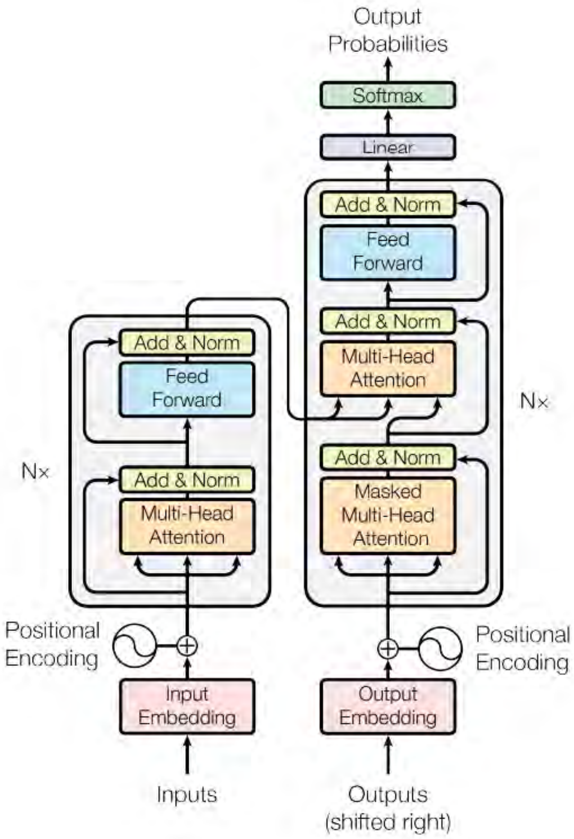


Fig. 2.1 Transformer model architecture. (Image source: [37])

Compared to GPT, the largest difference and improvement of BERT is to make training bi-directional. By obtaining both previous and subsequent context, it enables bi-directional prediction and sentence-level understanding. Being bi-directional means BERT can no longer be trained using basic language modelling task, namely predicting the next token given context, because the output is able to see the whole sentence. Instead, BERT is trained with two other tasks: Mask Language Model (MLM) which is similar to a cloze deletion test that predicts the missing word given the context and Next Sentence Prediction (NSP) for understanding the relationships between sentences and telling whether one sentence is the following sentence from the other.

2.3 Video embeddings

Based on the approaches of temporal emotion cues modelling in video, there are three main categories of facial emotion recognition methods [38]:

- Low level spatial-temporal feature based methods: treating video data as 3-d pixel volumes and applying image feature descriptors along all spatial and temporal dimensions [39–41].
- Image set based methods: treating video as a set of images and viewing video frames as representations of the same object captured under different conditions (pose, illumination, etc) [42–44] .
- Sequence model based methods: applying sequence models such as Recurrent Neural Networks (RNNs) to capture the temporal cues among video frames.

The latter two methods are more robust to the temporal variations of facial emotion expression [42, 45].

Recently image feature extraction is typically implemented with transfer learning methods [38, 46–50]. A Convolutional Neural Network (CNN) is trained with an external large dataset, e.g., FER2013 [51]. The output of the last CNN layer are used as feature vectors for the input image. After that, conventional classification techniques, e.g., Support Vector Machine (SVM) and FC layer with softmax activation function are applied to these features to recognize facial expressions.

Video in IEMOCAP is semi-front and half-length, as illustrated in Figure 2.2. Video signal processing can then be divided into two parts: i) detecting faces in the raw video; ii) extracting embeddings from faces. In this project, the first step is done by the Multi-task Cascaded Convolutional Networks (MTCNN) [52]. The second step is done by VGG-19 network [53] finetuned on FER2013.

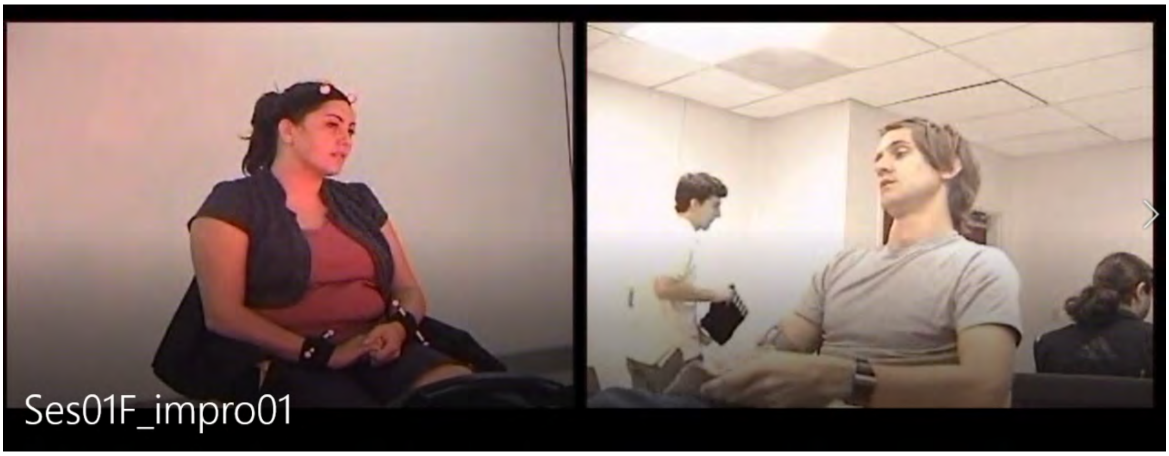


Fig. 2.2 Screenshot of the semi-front half-length video clip of IEMOCAP

2.3.1 Face detection: MTCNN

MTCNN [52] is a popular face detection technique that has achieved state-of-the-art results on a range of benchmark datasets. It is capable of also recognizing other facial features such as eyes and mouth, called landmark detection. The deep cascaded multi-task framework uses different features of "sub-models" to each boost their correlating strengths. The network consist of three stages: the input image is first re-scaled to a range of different sizes, then the first shallow CNN (Proposal Network) proposes candidate facial regions, the second more complex CNN (Refine Network) filters the bounding boxes to reject a large number of non-faces windows, and the third more powerful CNN (Output Network) refines the result and output facial landmarks positions.

In addition to deep learning methods, traditional feature-based face detection algorithms are also fast and effective such as Viola-Jones object-detection algorithm [54] which uses Haar basis feature filters and the AdaBoost algorithm.

2.3.2 Face models

Although most studies use the FER2013 emotion corpus for finetuning, the choice of pre-trained face models is quite flexible. Such models include the VGG-Face model [49, 50] pre-trained for face recognition using the large VGG face dataset [55] and AlexNet Deep CNN model [38] pre-trained on ImageNet [56], an object centric dataset that includes the category person. This thesis uses VGG-19, a CNN trained on the ImageNet. VGG-19 consists of 25 layers (16 convolution layers, 3 fully connected, 5 maxpool layers and 1 softmax layer). It improves over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another. Within a given receptive field, multiple stacked smaller kernels are better than one larger kernel because multiple non-linear layers increases the depth of the network which enables it to learn more complex features at a lower cost.

Chapter 3

Encoder models and pooling methods

Chapter 2 has described methods to process and extract information from the raw input of the various modalities such as speech waveform, text, and video. This chapter reviews the models and methods that process these sequence of parameter vectors and make prediction on the emotional content of the input utterance.

As discussed in Chapter 1, inputs have been represented as sequence of feature vectors but emotional labels are provided for the whole utterances in many databases including IEMOCAP. There can be a mismatch between short-term inputs at the frame level and long-term outputs at the utterance level. Common sequence-to-sequence method includes assigning the overall emotional label to each frame within the utterance, and training the system in a frame-wise manner [57] or excluding those emotionally irrelevant frames in an utterance and aligning the overall emotional label to each emotionally relevant one [58].

This project uses sequence-to-one methods. Typical sequence-to-one methods involve using a sequence model to encode context information from the feature sequence and using various pooling methods to summarize emotion information for the whole utterance.

Commonly used sequence models include RNN [59], Long Short Term Memory (LSTM) [60], bidirectional-LSTM (bi-LSTM) [61], Gated Recurrent Unit (GRU) [62], CNN, Time Delay Neural Network (TDNN) [63], etc. Tripathi et al. [34] used an LSTM to process text, Yoon et al. [36] used a bi-LSTM to process both text and audio in their 2018 work and GRU in their 2019 work [35]. CNNs have been widely used to process text [64–66].

Commonly used pooling methods include final pooling where only the final hidden representation at the last frame of an utterance is used as the representation of the whole sequence and the system makes a decision at the end of each utterance [19, 20, 67], mean pooling where the average of hidden representations of all inner frames of an utterance are used as the representation [19–21], and weighted pooling which computes a weighted sum as the representation, where the weights are normally determined based on an additional attention mechanism [20, 21, 68].

In this project, the structure used for speaker diarisation [1] has been adopted and adapted: using a TDNN model with residue connection (ResNet-TDNN) as the encoder and multi-head self-attention (MHA) for pooling.

3.1 Encoder: ResNet-TDNN

One effective architecture in modelling long range temporal dependencies is the Time Delay Neural Network, originally proposed in [63] and often used in its sub-sampled form [69] to reduce computation during training.

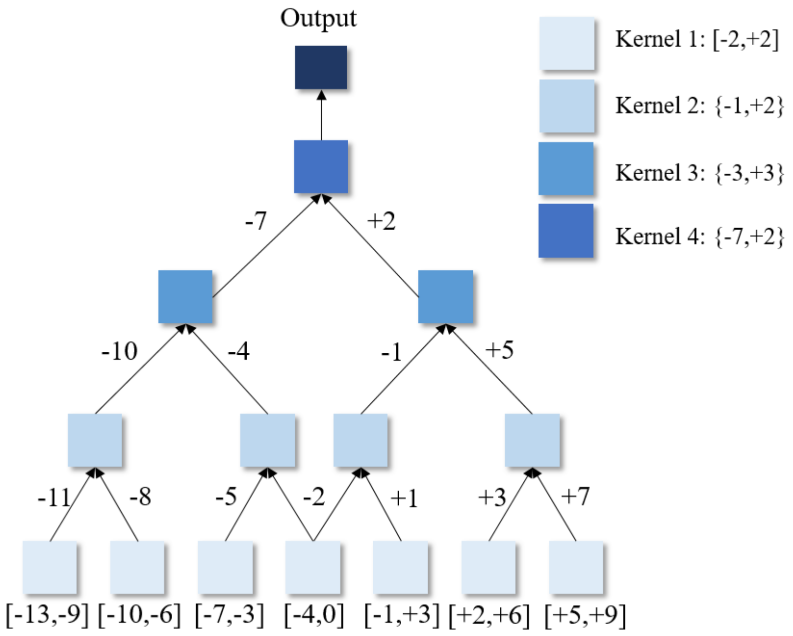


Fig. 3.1 Illustration of TDNN structure used in the project

A TDNN consists of identical fully-connected layers repeated at different time steps. When processing a wider temporal context, instead of learning the entire temporal context by the initial layer (which standard feed-forward deep neural networks do), each layer in a TDNN operates at a different temporal resolution and the higher layers in turn have the ability to learn wider temporal relationships. The basic TDNN structure used in the project is illustrated in Figure 3.1.

Kreyssig et al. [70] proposed a method of deepening the kernel used in the TDNN temporal convolutions by introducing residue connection to speed up training. Deeper networks generally improve the performance of neural network architectures but they are usually harder and slower to train and might raise issue such as gradient vanishing. The problem of training very deep networks has been alleviated with the introduction of the residual network (ResNet) [71]. Residual connections add the outputs from previous layers to the outputs of stacked layers, described by Equation 3.1 where x and $F(x,\theta)$ are the input and the output of the block of layers that is to be "skipped".

$$y = F(x,\theta) + x \tag{3.1}$$

This direct connection reduces the effective minimum depth of networks in terms of layers and results in the ability to train much deeper networks. Figure 3.2 shows the comparson of standard kernel and ResNet kernel.

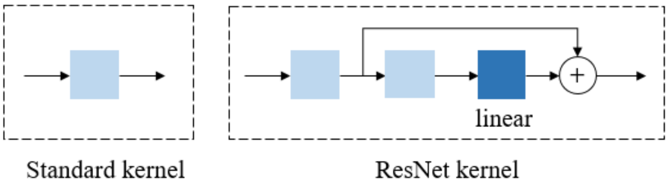


Fig. 3.2 Comparison of standard kernel and ResNet kernel. Lighter block are FC layers with ReLU activation function. The darker block denotes an FC layer with linear activation function.

3.2 Pooling: Multi-head self-attention

The structured self-attention mechanism, introduced in [72] to replace the max pooling or the averaging step, can be viewed as dynamic linear combination of input vectors with the combination weights provided by an annotation matrix A, which is computed from the inputs themselves. The annotation matrix can be calculated using Equation 3.2

$$\mathbf{A} = \text{softmax}\left(\tanh(\mathbf{H}\mathbf{W_1})\mathbf{W_2}\right) \tag{3.2}$$

where $\mathbf{H} = [\mathbf{h}(1)\ldots\mathbf{h}(T)]^T$ is the $T \times n$ input matrix corresponding to T input vectors of dimension n and $\mathbf{A}(T \times h)$ is the h-head annotation matrix. $\mathbf{A}$ is generated by passing the input matrix through two FC layers with weight matrices $\mathbf{W_1}$ and $\mathbf{W_2}$ respectively. Each column of the annotation matrix is an annotation vector which gives a set of scaling factors that weight the importance of each input. Softmax is performed column-wise to ensure each annotation vector sums to one. The outputs $\mathbf{E}(h \times n)$ is achieved by applying A to the inputs:

$$\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_h] = \mathbf{A}^T\mathbf{H} = \text{SelfAtten}(\mathbf{h}(1)\ldots\mathbf{h}(T)) \tag{3.3}$$

The multi-head self-attention structure is illustrated in Figure 3.3.

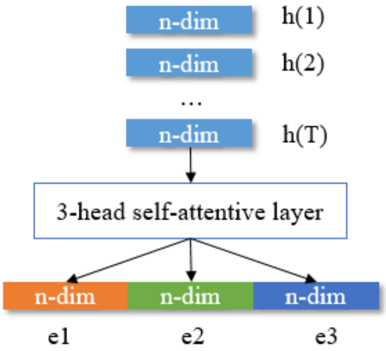


Fig. 3.3 Illustration of a three-head self-attentive layer

When a multi-head self-attentive layer is used (i.e. $h > 1$), to encourage different heads to extract dissimilar information, a penalty term in Equation 3.4 is added to the cross-entropy loss function during training.

$$P = \mu \left\|\mathbf{A}^T\mathbf{A} - \mathbf{I}\right\|_F^2 \tag{3.4}$$

where $\mathbf{I}$ is the identity matrix, $||\cdot||$ denotes the Frobenius norm, μ is the penalty coefficient. This penalty term was originally designed for sentence embeddings to focus on as few words as possible while encouraging different annotation vectors to be estimated. Sun et al. [1] modified the penalty term by replacing $\mathbf{I}$ with a diagonal matrix Λ:

$$P = \mu \left|\left|\mathbf{A}^T\mathbf{A} - \Lambda\right|\right|_F^2 \tag{3.5}$$

The diagonal values $\lambda_i = \Lambda_{ii}$ control the smoothness of the annotation vectors. By varying the value of λ between $1/h$ and 1, the annotation vectors can not only focus on a few key input vectors but also reflect the general trends of importance.

Different from final pooling and mean pooling, attention computation requires a fixed matrix dimension, which means that the attention computation needs to be performed on a fixed window length. This would lead to a window-level decision. There are two simple ways to combine window-level decisions to obtain predictions for the whole sentence: i) majority voting: picking the most probable class over each window and taking the dominant class among all windows; ii) averaging: averaging the probabilities of each class over all windows and taking the overall most probable one. Both two methods are investigated in the experiments.

3.3 Fusion

Fusion is a key research topic in multimodal studies, which integrates information extracted from different unimodal data into one compact multimodal representation. Fusion methods can be divided based on the stage that it appears in the procedure, e.g., early fusion (also "feature-level" fusion) and late fusion (also "decision-level" fusion). Early fusion combines the input modalities into a single feature vector on which a prediction is made (usually this happens before the neural network). In late fusion methods, each of the input modalities is used to make an individual prediction, which are then combined for the final classification (usually this happens after the neural network). This distinction has gradually become blurred with the development of deep neural networks. Features and embeddings can be extracted using deep neural networks and it is hard to determine whether fusion happens before the network or after. Besides, early and late fusion can suppress either intra- or inter-modality interactions. Therefore, recent studies focus on the intermediate methods that allows fusion to happen at multiple layers of a deep model [73].

Commonly used fusion methods can be classified into simple operation-based, attention-based, and bilinear-pooling-based methods. Simple operation-based methods includes concatenation and weighted sum with scalar weights. Concatenation can be applied to either low-level input features or high-level features extracted by pre-trained models [74]. Weighted sum fusion with scalar weights can be achieved by a FC layer with dimension control being implemented at the same time. An attention mechanism as described in Section 3.2 can also be used for fusion. The basic attention mechanism can be extended to use a more complex structure such as co-attention mechanism [75] and dual attention network [76]. Bilinear pooling is another method often used to combine vectors into a joint representation by computing their outer product [77]. Compared to concatenation and attention, bilinear pooling allows a multiplicative interaction between all elements in both vectors. The bilinear representation is often linearly transformed into an output vector using a two-dimensional weight matrix. As multiplication leads to high dimensionality, bilinear pooling often requires decomposing the weight tensor so that the associated model can be trained properly and efficiently. For instance, Multimodal Low-rank Bilinear pooling (MLB) enforces a low rank to the weight tensor [78].

Concatenation, weighted sum using FC layer and MLB are all investigated in this thesis.

3.4 Classification: Angular softmax

Classification can use traditional machine learning techniques such as Support Vector Machine (SVM) [64] and a FC layer with a softmax activation function. Instead of using the standard softmax function, angular softmax (A-softmax) [79] is used as activation function of the classification FC layer in this project. A-softmax was originally proposed to address the deep face recognition problem with an open-set protocol, where ideal face features are expected to have smaller maximal intra-class distance than minimal inter-class distance under a suitably chosen metric space. It has also been shown to improve the generalisation ability to unseen data which is useful in small datasets.

The posterior probability obtained by standard softmax loss can be described by Equation 3.6:

$$p_i = \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_j e^{W_j^T x_i + b_j}} \tag{3.6}$$

where x_i, W_j, W_{y_i} are the i-th training sample, the j-th and y_i-th column of fully connected layer W respectively. The standard softmax loss can be written as the negative log likelihood of the posterior probability, as shown in Equation 3.7.

$$L_i = -\log\left(\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_j e^{W_j^T x_i + b_j}}\right) \tag{3.7}$$

$W_{y_i}^T x_i + b_{y_i}$ can be re-written as $\left\|W_{y_i}\right\| \|x_i\| \cos(\theta_{y_i,i}) + b_{y_i}$ where $\theta_{j,i} (0 \leq \theta_{j,i} \leq \pi)$ is the angle between vector W_j and x_i. Equation 3.7 can then be re-written as:

$$L_i = -\log\left(\frac{e^{\left\|W_{y_i}\right\| \|x_i\| \cos\left(\theta_{y_i,i}\right) + b_{y_i}}}{\sum_j e^{\left\|W_j\right\| \|x_i\| \cos\left(\theta_{j,i}\right) + b_j}}\right) \tag{3.8}$$

Normalizing $\left\|W_j\right\|$ in each iteration and zeroing the biases, the posterior probability becomes $p_j = \|x_i\| \cos\left(\theta_{j,i}\right)$. Note that all p_j share the same x_i, the final result only depends on the angles $\theta_{j,i}$. The modified softmax function is given by Equation 3.9.

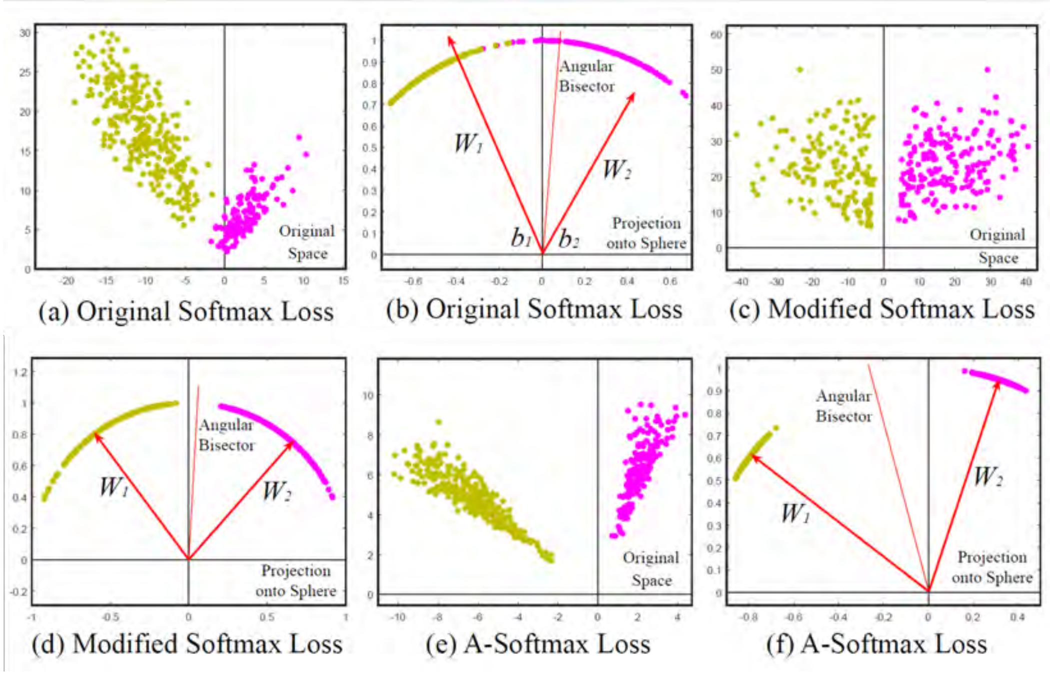
$$L_i = -\log\left(\frac{e^{\|x_i\| \cos\left(\theta_{y_i,i}\right)}}{\sum_j e^{\|x_i\| \cos\left(\theta_{j,i}\right)}}\right) \tag{3.9}$$

The idea of angular margin is to make the decision more stringent. Taking binary classification as example, the modified softmax loss requires $\cos(\theta_1) > \cos(\theta_2)$ to correctly classify x. A-softmax instead requires $\cos(m\theta_1) > \cos(\theta_2)$ $(m > 1)$ which requires $\theta + 1 < \frac{\theta_2}{m}$ in order to correctly classify x. This is more difficult than original $\theta_1 < \theta_2$ and would thus leads to more separable classes. Below is the equation of angular softmax loss:

$$L_{\text{ang}} = \frac{1}{N} \sum_i -\log\left(\frac{e^{\|x_i\| \cos\left(m\theta_{y_i,i}\right)}}{e^{\|x_i\| \cos\left(m\theta_{y_i,i}\right)} + \sum_{j \neq y_i} e^{\|x_i\| \cos\left(\theta_{j,i}\right)}}\right) \tag{3.10}$$

where $\theta_{j,i}$ has to be in the range of $[0, \frac{\pi}{m}]$, and the size of angular margin can be quantitatively adjusted by the parameter m.

Figure 3.4 compares among softmax loss, modified softmax loss and A-softmax loss.



(a) Original Softmax Loss (b) Original Softmax Loss (c) Modified Softmax Loss

(d) Modified Softmax Loss (e) A-Softmax Loss (f) A-Softmax Loss

Fig. 3.4 Comparison among softmax loss, modified softmax loss and A-Softmax loss. it can be seen that A-Softmax loss can further increase the angular margin of learned features. (Image source: [79])

Chapter 4

Multimodal Emotion Recognition System

In this chapter, a complete multimodal emotion recognition system is developed across audio, text, and video modalities. Section 4.1 and 4.2 provide details about the dataset used and the experiment setting. Section 4.3 examines basic model configurations and the combination of different audio features. Text and videos are then added to the system in Section 4.4 and 4.5 respectively. Section 4.6 investigates the contribution of different modalities and discusses the use of ASR transcriptions and long-term audio features. Section 4.7 summarizes the chapter and cross compares the system performance with the literature.

4.1 Dataset

This thesis uses IEMOCAP database [25]. In total, the corpus contains 10039 utterances with an average duration of 4.5 s. The average number of words per utterance was 11.4. Categorical labels were used in this thesis, which contains 10 categories (neutral, happiness, sadness, anger, surprise, fear, disgust, frustration, excited, other).

Most of the related publications on IEMOCAP only used part of the dataset. Only utterances satisfying the following two conditions were considered: i) majority of the three annotators agreed on one emotion label; ii) this label belongs to the following categories: angry, happy, excited, sad, and neutral. “happy” and “excited” were combined as “happy” to balance the number of samples in each emotion class. The task thus became a four-way classification problem. To keep consistent and cross compare with the related work, this setting is used in this Chapter. The label statistics are listed in Table 4.1 and the emotion distribution is shown in Figure 4.1. Chapter 5 will discuss approaches to deal with utterances that have been discarded in this setting, namely utterances that don’t have majority labels and don’t belong to these four target classes.

happy	angry	sad	neutral	total
1636	1103	1084	1708	5531

Table 4.1 Statistics of four emotion categories used in this chapter. Only utterances with ground truth label belonging to these four categories are used in this chapter.

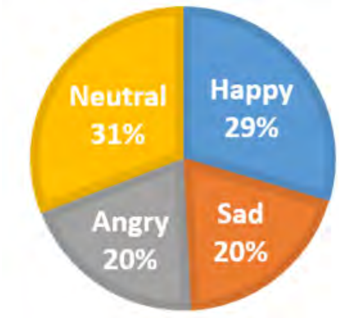


Fig. 4.1 Statistics of four emotion categories used in this chapter.

4.2 Experiment setup

Related work usually used leave-one-session-out 5-fold cross validation (5-cv) or speaker-independent 10-fold cross validation (10-cv). However, cross validation can be time-consuming. In order to save time, systems are trained on Session 1-4 and tested on Session 5 unless otherwise stated. 10% of training data is randomly chosen for validation. Key conclusions will be verified by 5-fold cross validation and 10-fold cross validation results will be provided for the final system for cross comparison. The number of utterances in each fold of 5-cv and 10-cv are shown in Table 4.2 and Table 4.3, respectively.

Test session	Ses1	Ses2	Ses3	Ses4	Ses5
Training	4002	4058	3942	4050	3861
Validation	444	450	438	450	429
Testing	1085	1023	1151	1031	1241

Table 4.2 Number of utterances in training, validation and testing set of each fold in 5-cv. “Ses1" denotes Session 1, etc.

Test speaker id	1F	1M	2F	2M	3F	3M	4F	4M	5F	5M
Training	4503	4477	4545	4491	4509	4412	4503	4526	4447	4032
Validation	500	497	505	498	500	490	500	502	494	488
Testing	528	557	481	542	522	629	528	503	590	651

Table 4.3 Number of utterances in training, validation and testing set of each fold in 10-cv. “1F" denotes the female speaker in Session 1. “1M" denotes the male speaker in Session 1, etc.

Since the test sets are slightly imbalanced between emotion categories, as shown in Figure 4.1, both the weighted accuracy (WA) and unweighted accuracy (UA) are reported. WA corresponds to the overall accuracy on test examples which equals the correctly detected samples divided by the total number of samples. UA corresponds to average recall over the different emotion categories.

4.3 AER using audio modality

4.3.1 Model structure and basic configuration

The model structure used is shown in Table 4.4. Specifically, the encoder consists of four 512-dimensional ResNet-TDNN layers, followed by a projection layer which reduces the outputs' dimension to 128-d before feeding into the self-attentive layer. All layers use ReLU as activation function except the last FC layer which uses A-softmax for classification. Models were implemented using an extended version of HTK Toolkit [2]. During training, the newbob learning rate scheduler with an initial learning rate of 5×10^{-5} was used, and batch size was set to 200.

Layer No.	Layer Type	Context width	Dimension
1-4	ResTDNN	[2,+2]	512
5-8	ResTDNN	{-1,+2}	512
9-12	ResTDNN	{-3,+3}	512
13-16	ResTDNN	{-7,+2}	512
17	FC	{0}	128
18	5-head self-atten	[-50,+49]	128*5
19	FC	{0}	128
20	FC	{0}	4

Table 4.4 Model structure of the audio-based AER system.

The basic configuration of the system such as the number of self-attention heads, the window length and window shift of the attention layer were selected by the following experiments. 40-d MFB features were computed using HTK and used as input with dialogue-level variance normalization and utterance-level mean normalization.

The number of heads was selected from the experiments shown in Table 4.5. Single head self-attention can be viewed as mean pooling with dynamic weights. Increasing the number of heads enables the attention to capture different characteristics. In the five-head system, by setting the diagonal value of the modified penalty term in Equation 3.5, two heads are responsible for capturing long-term characteristics over the whole window and the other three capture key frames. The utterance-level results can be obtained by combining window-level decisions using majority voting or by averaging. As shown in Table 4.5, averaging gives slightly better results and was used in later experiments.

# head	voting WA	voting UA	avg WA	avg UA
1	56.97	58.80	57.21	59.08
3	58.18	60.04	58.18	60.09
5	**58.90**	**61.47**	**59.39**	**62.15**

Table 4.5 Basic configuration test of the number of heads of attentive layer. 40-d MFB features used as input. Tested on Session 5. Results in percent. Bold: highest value in each column.

As discussed in Section 3.2, the attention mechanism requires a fixed window length. Another key configuration that affects system performance is the context of the attention layer, namely the window length and window shift, illustrated in Figure 4.2.

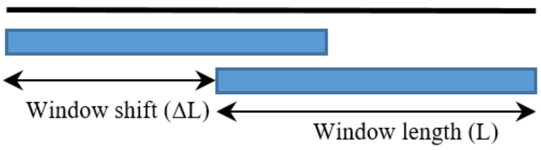


Fig. 4.2 Illustration of window shift and window length. Blue rectangle denotes the fixed-length window.

Five-head attention with different window lengths and window shift were tested. Results are shown in Table 4.6. The unit of window length is frame. “50*2” denotes a window of context [-50,+49]. Frame shift of audio features is 10ms, which means that a window length of 50*2 corresponds to a 1s window. As the average length of utterance in IEMOCAP is

4.5s, using window length of 300*2 would cover most of the sentences. Short sentences were padded by repeating the edge frame.

Row num.	L	ΔL	avg WA	avg UA
1	25*2	10	59.95	61.96
2	50*2	50	58.34	59.74
3	50*2	10	59.39	**62.15**
4	50*2	1	**60.52**	62.00
5	100*2	100	56.73	58.30
6	100*2	50	57.61	59.17
7	100*2	10	58.34	60.27
8	100*2	1	57.94	56.92
9	200*2	100	54.88	54.56
10	200*2	50	55.92	56.72
11	300*2	100	52.38	53.28

Table 4.6 Basic configuration test of window length and window shift. “50*2” denotes a window with context of [-50,+49]. 40-d MFB features used as input. Tested on Session 5. Results in percent. Bold: highest value in each column.

Comparing Row num.2,6,10 (Figure 4.3b) and Row num.5,9,11 (Figure 4.3c), with fixed window shift, although a longer window covers more frames at once, it doesn’t yield better performance. Comparing Row num.1,3,7 (Figure 4.3a), with fixed window shift of 10, a window size of 25*2 and 50*2 give close results. Comparing Row num.2-4 (Figure 4.3d) and Row num.9-10 (Figure 4.3f), with a fixed window size, smaller window shift gives better results but is meanwhile more expensive to train. Comparing Row num.5-8 (Figure 4.3e), when window shift decreases from 10 to 1, there’s no performance gain but it takes a longer time to train. Trading-off between system performance and training time, a window size of 50*2 and a window shift of 10 will be used in later experiments.

4.3.2 Combination of different audio features

POV-weighted pitch and first differentials were appended to the 40-d MFB. The 5-fold cross validation results of the system using different input audio features are shown in Table 4.7. Comparing rows 2 and 3, pitch information is useful. It improves the results by ~0.75% by only adding one more dimension. Although it can be argued that dynamic features can be redundant if they are combined with a linear layer in a subsequent neural network, appending the first differentials still improves the results by ~2.2%. The standard deviation

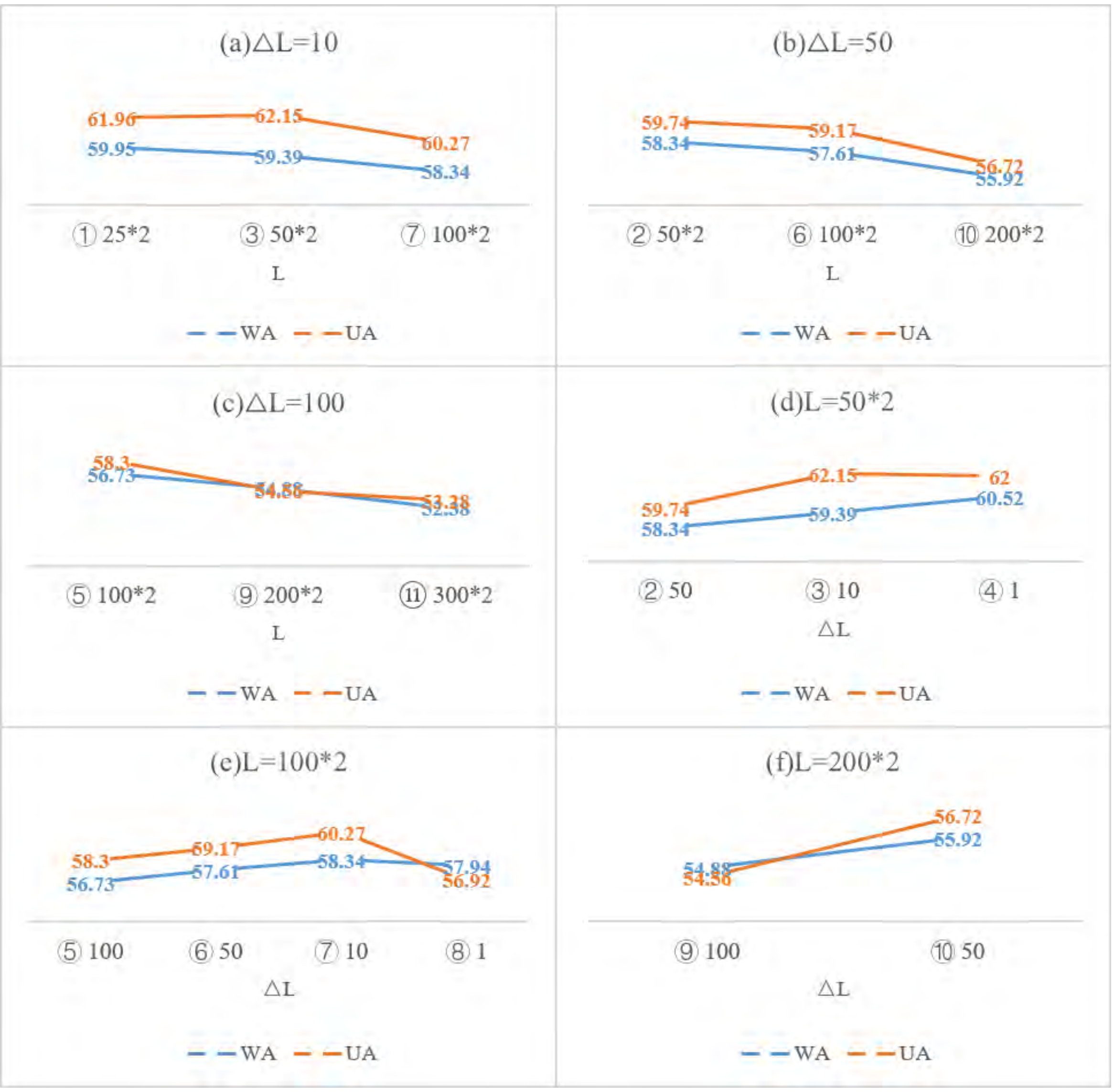


Fig. 4.3 Basic configuration test of window length and window shift.

also decreases when deltas are appended.

Feature	dim	avg WA	avg UA
MFB	40	57.20±1.84	58.78±3.21
MFB+pitch	41	57.91±2.17	59.59±3.40
MFB+pitch+Δ	82	60.64±1.96	61.32±2.26

Table 4.7 Average and standard deviation of 5-fold cross validation results on audio-based AER system using different combination of audio features as input. Results in percent.

In later experiments, unless otherwise stated, audio feature refers to the 82-d MFB+pitch+Δ. As the input has a context width of [-2,+2], the total input dimension is $82 * 5 = 410$.

4.4 AER using audio and text modalities

The released reference transcripts of the IEMOCAP dataset are used for the text modality in this section, as most related work with IEMOCAP did. The use of Automatic Speech Recognition (ASR) in obtaining transcription will be discussed in Section 4.6.2.

4.4.1 Word-level text information

GloVe vectors pre-trained on Twitter were used in this section. The Twitter database contains 2B tweets, 27B tokens, and a 1.2M vocabulary. GloVe vectors are available with dimensions of 25, 50, and 100. GloVe vectors were attached to audio features at the frame level to form a low-level concatenation fusion. As one word lasts for several frames, the same GloVe vector would be repeated for several frames. The input audio features have a context range from -2 to +2. Since the neighbouring frames are very likely to be the same, GloVe is only attached to the central frame, as illustrated in Figure 4.4. Although there's still temporal redundancy for the central GloVe embedding, word duration can also be cue for emotional content.

GloVe vectors of different dimensions were attached to audio features. The results are shown in Table 4.8. Although Tripathi et al. [34], Yoon et al. [35, 36] used 300-d GloVe in their experiments, Table 4.8 shows that 50-d GloVe performs best in this framework.

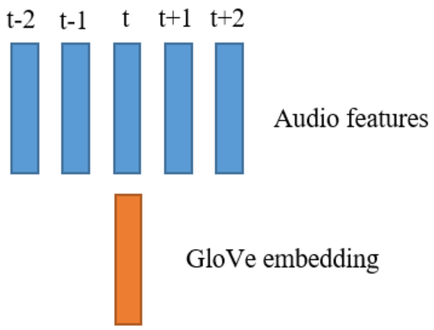


Fig. 4.4 Concatenation of audio features (blue) and GloVe vectors (orange). Input audio feature has a context of [-2, +2] and GloVe was attached only to the central frame.

GloVe dim	avg WA	avg UA
25	66.08	66.57
50	**67.2**	**68.11**
100	66.96	67.58

Table 4.8 Attaching GloVe of different dimensions to audio features. Tested on Session 5. Results in percent. Bold: highest value in each column.

With 50-d GloVe appended, the dimension of input features becomes $410+50=460$. 5-cv results using audio features, 50-d GloVe embeddings and their combination are shown in Table 4.9. Using GloVe alone gives slightly higher accuracy than audio alone but larger range across cv folds at the same time. Introducing GloVe to the audio features increases the results by ~5%. The standard deviation among five folds also decreases when these two features were combined, which indicates that the system becomes more robust to variation across datasets.

Feature	avg WA	avg UA
Audio	60.64±1.96	61.32±2.26
GloVe	61.27±3.73	62.67±3.55
Audio+GloVe	65.53±1.83	66.43±1.33

Table 4.9 Average and standard deviation of 5-fold cross validation results on AER system using audio features, 50-d GloVe embeddings and their combination. Reference transcripts used for the text modality. Results in percent.

4.4.2 Sentence-level text information and the time asynchronous branch

Previously, word-level GloVe embeddings are synchronized with audio features at the frame level. In order to incorporate BERT which is a sentence-level embedding, an additional branch was added to the system, termed the "time asynchronous" branch.

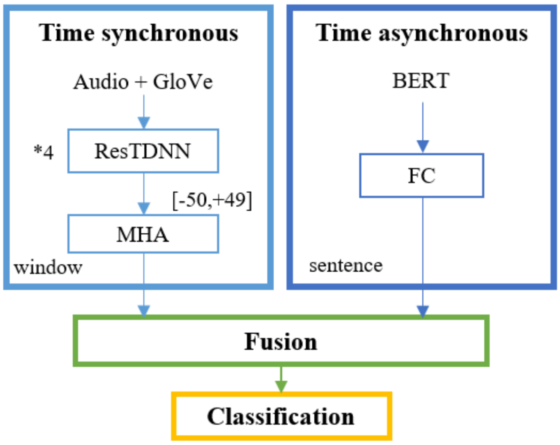


Fig. 4.5 Illustration of the modified system structure that contains a time synchronous branch and a time asynchronous branch.

The modified system structure is illustrated in Figure 4.5. Computation of the BERT embedding was implemented using PyTorch "transformers" library[1]. Again, reference transcripts were used as input. A 768-d BERT embedding of each sentence was extracted using "Bert-Model" with pretrained "bert-base-uncased" weights which behaves as an encoder. The 768-d BERT embedding first passes through an FC layer for dimension reduction (64-d) and then concatenates with the output of the multi-head self-attentive layer of the time synchronous branch (Layer 18 in Table 4.4) and together passes through an FC layer for fusion (Layer 19 in Table 4.4) before sending to classification.

Results are shown in Table 4.10. Comparing the 3rd and the 4th rows, BERT is more powerful than GloVe when added to the Audio-based system. Comparing the 4th and 5th rows, GloVe and BERT are complementary. Although BERT is powerful, adding Glove still improves the results, which indicates that GloVe can provide complementary information.

One of the main contributions of this thesis is the use of context BERT, namely the BERT embedding of previous and subsequent sentences. The model structure using context BERT

[1] https://pypi.org/project/transformers/

Feature	avg WA	avg UA
Audio	64.06	64.24
Audio+GloVe	67.20	68.11
Audio+BERT	69.46	69.32
Audio+GloVe+BERT	70.83	71.61

Table 4.10 Incorporating the BERT embedding to the system. Tested on Session 5. Reference transcripts used for text modality. Results in percent.

in the time asynchronous branch is shown in Figure 4.6. The dimension of each BERT embedding was reduced to 64-d by the FC layer, whose weights and biases were shared among all context BERT embeddings. Another five-head self-attentive layer was introduced to combine context BERT. In this case, as inputs are sentences, the unmodified penalty (Equation 3.4) was used, which reflects the extent to which sentences in the context affects the current emotion.

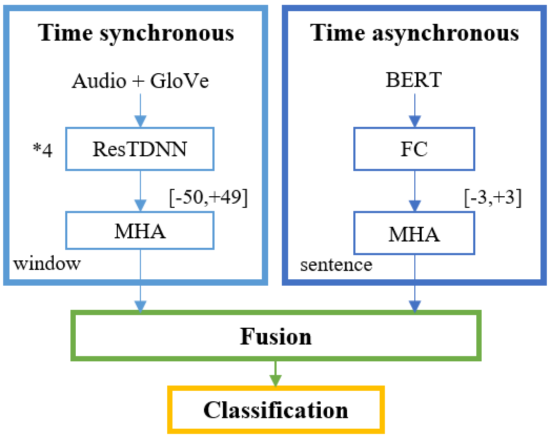


Fig. 4.6 Illustration of the model structure using context BERT in time asynchronous branch.

IEMOCAP is a dyadic database. The context can be chosen to either only include context of the current speaker or include context of both speakers in dialogue turn. Table 4.11 shows the results of using audio features, Glove, and BERT with different context range. Comparing Row num.1-5 in Table 4.11, increasing context width leads to increase in classification accuracies while Row num.4 and 5 give comparable results. Comparing Row num.3,6 and 4,7, with the same context width, including both speakers in context performs better. It can be argued that emotion is a long-term attribute that may last for several sentences. The use of context sentences of the current speaker helps capture this. And in dialogues, context information from the other speaker can carry reaction information, which can also be useful to infer the current speaker's emotion. What the other speaker says may affect the listener's emotion to a considerable extent and can trigger a change in the listener's emotion. The last two rows consider only previous sentences (in online learning manner). Comparing Row num.2 and Row num.8, using the same number of context utterances, the following sentence is more informative than that before the previous sentence. Comparing Row num.3,8 and Row num.4,9 shows the effectiveness of using subsequent sentences. And the use of both previous and following context is consistent with the bi-directional characteristic of BERT. Context of [-3,+3] was selected for later experiments. The detailed structure of the time asynchronous branch is shown in Figure 4.7.

Row num.	context	# of speaker	avg WA	avg UA
1	[0]	two	70.83	71.61
2	[-1,+1]	two	77.60	77.90
3	[-2,+2]	two	77.76	79.20
4	[-3,+3]	two	**81.22**	81.60
5	[-4,+4]	two	80.66	**81.87**
6	[-2,+2]	one	75.02	76.71
7	[-3,+3]	one	80.66	80.99
8	[-2,0]	two	74.94	75.74
9	[-3,0]	two	78.81	78.38

Table 4.11 Comparison of BERT with different context range. Audio features and Glove embedding used as inputs to the time synchronous branch. Number of speaker equals "one" means that only sentences of the current speaker were included in context. "Two" means that context of both speakers were included. Tested on Session 5. Reference transcripts used for the text modality. Results in percent. Bold: highest value in each column.

Table 4.12 shows the 5-fold cross validation results using single BERT ("BERT0"), context BERT ("BERT7"), and context BERT with audio features and GloVe embeddings as inputs. It can be seen that context BERT is much more powerful than single BERT. Comparing "BERT0" and "BERT7", context BERT has much higher average accuracy as well as lower standard deviation. Comparing with system using Audio+GloVe (last row in Table 4.9), introducing context BERT improves the result by ~10%. In the following content, unless otherwise stated, "BERT" refers to context BERT ("BERT7").

As BERT is a sentence-level embedding, if only the time asynchronous branch is considered, it makes a sentence-level decision, which is equivalent to a window containing one frame. Voting and averaging then have the same result.

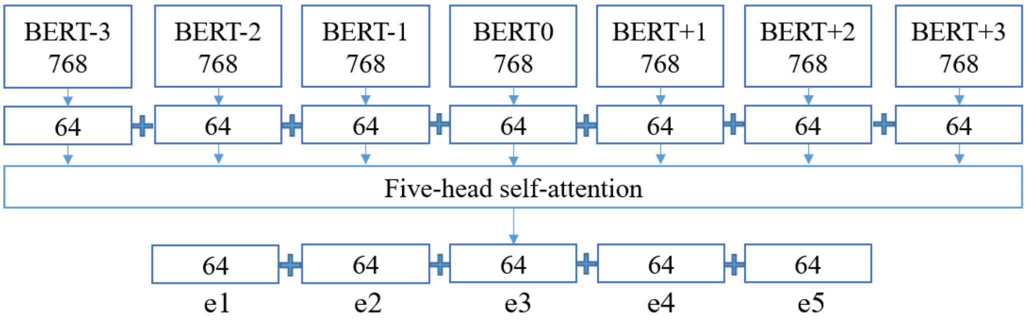


Fig. 4.7 Detailed structure of time asynchronous branch with context of [-3,+3]. “+" denotes concatenation operation. The dimension of each 768-d BERT embedding is reduced to 64-d by the FC layer. Weights and biases are shared among the seven 64-d FC layers. Context BERTs are then combined using the five-head self-attentive layer.

Feature	avg WA	avg UA
BERT0	58.53±4.41	59.20±5.57
BERT7	71.22±3.16	71.88±2.62
Audio+GloVe+BERT7	75.53±3.79	76.65±3.67

Table 4.12 Average and standard deviation of 5-fold cross validation results using single BERT (“BERT0"), context BERT (“BERT7"), and context BERT combined with audio features and GloVe embeddings. Reference transcripts used for the text modality. Results in percent.

4.4.3 Bilinear fusion

The previous experiments used an FC layer to fuse the time synchronous branch and the time asynchronous branch. This section examines the effect of bilinear pooling.

As shown in Table 4.13, surprisingly, although bilinear pooling has been shown to be more powerful in many cases, FC works much better in this framework. One possible reason is that bilinear pooling expects two systems to have close performance, in other words, to be more balanced. As shown in Table 4.14, time asynchronous branch performs much better than time synchronous branch (mainly due to the context information). In this situation, bilinear pooling is not effective while FC fusion increases the performance by a marked margin. FC fusion will continue to be used in later experiments.

Fusion	avg WA	avg UA
FC	81.22	81.60
Bilinear	70.91	74.91

Table 4.13 Comparison of two fusion methods: FC layer and bilinear pooling. Tested on Session 5. Audio features, GloVe embeddings, and context BERT used for the system. Reference transcripts used for the text modality. Results in percent. Results in the 2nd row is the same as Row num.4 in Table 4.11.

Branch	avg WA	avg UA
Time-sync	67.20	68.11
Time-async	74.21	73.31

Table 4.14 Results of time synchronous branch alone and time asynchronous branch alone. Tested on Session 5. Reference transcripts used for the text modality. Audio features and GloVe embeddings used for the time synchronous branch. Context BERT used for the time asynchronous branch.

4.5 AER using audio, text, and video modalities

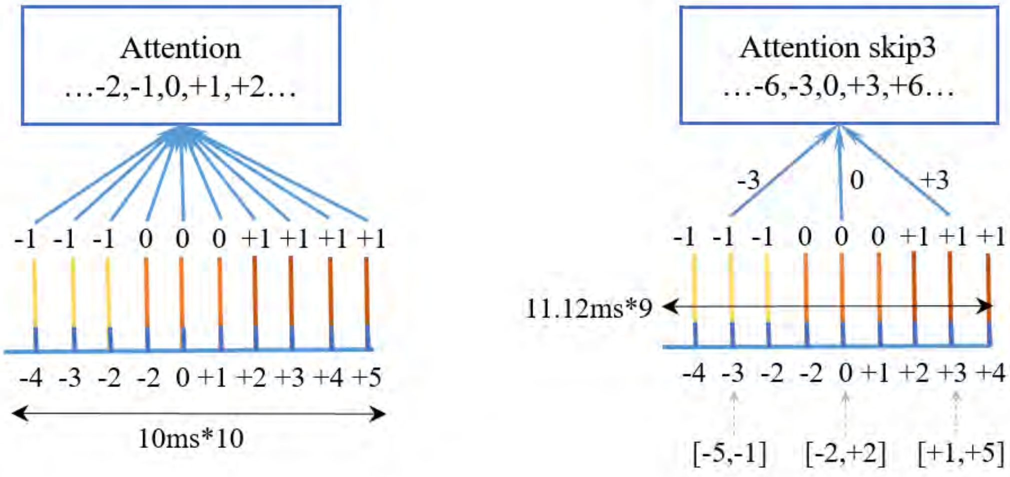
This section aims to add the video modality to the system. 160*160 face images were extracted by MTCNN[2] from each video frame and then resized into 48*48 to match the input size of the VGG-19 model finetuned on FER2013[3]. The pretrained VGG-19 behaves

[2] The implementation of MTCNN used an open source PyTorch library: https://pypi.org/project/facenet-pytorch/

[3] https://github.com/WuJie1010/Facial-Expression-Recognition.Pytorch

as an encoder. 512-d embedding for faces in each video frame can then be obtained from the outputs of the last layer before the final classification layer. Before feeding the embedding to the current AER system, the dimension of face embeddings are reduced to 32-d through an FC layer.

As video has frame shift of 33.37ms while audio features has frame shift of 10ms, the system structure needs to be modified before incorporating the video modality. One way to align video and audio is to up-sample video by repeating each frame by three or four times (Figure 4.8a). But this would lead to temporal redundancy as human can not detect such rapid changes. The other way is to down-sample audio features (Figure 4.8b). Instead of directly increasing frame shift of audio from 10ms to 33.37ms, which also requires to increase frame length and might lose some short-term information. The down-sample process was divided into two steps: i) increasing audio frame shift to 11.12 ms $(= 33.37/3)$; ii) in each attention window, selecting every three frame (termed skip3) to recover $(11.12 \ast 3 = 33.37)$. The window length was increased from 50*2 to 150*2 to keep the number of frames per window the same. Although the video features are repeated three times and attached to each audio frame, it would only be used once in each attention window due to the skip3 structure. As the audio features have a context width of [-2,+2] which still overlaps when skip3 was applied, overall, we will not lose any audio frames.

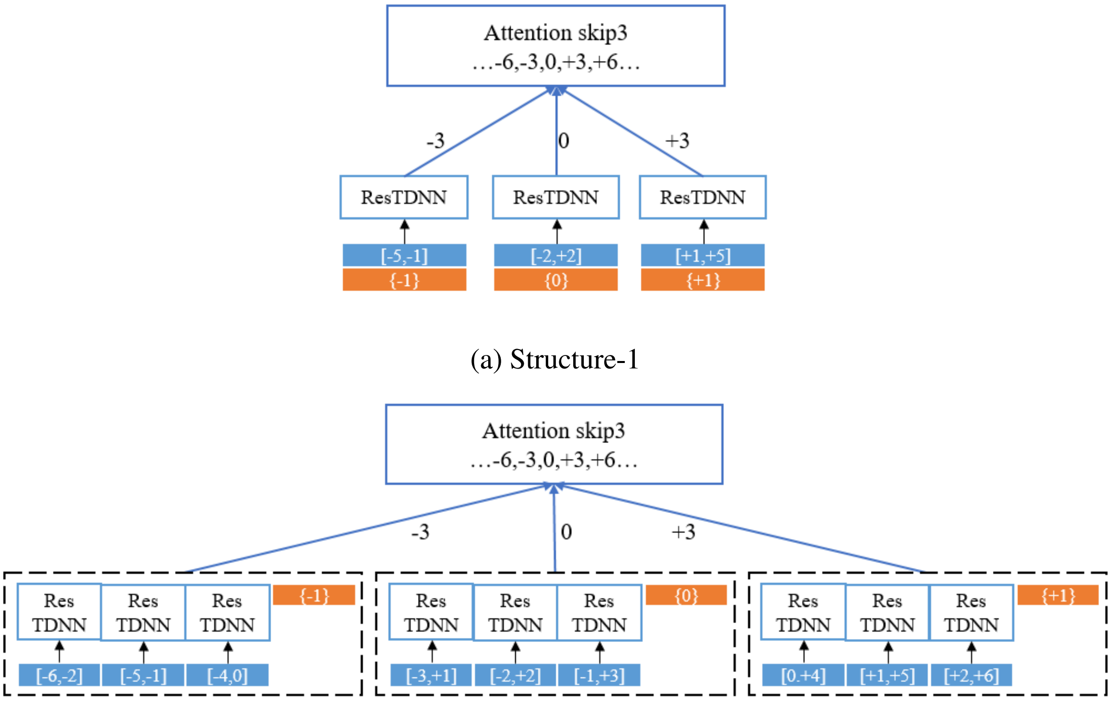


(a) Up-sample

(b) down-sample

Fig. 4.8 Illustration of two ways to align audio features (blue) and video features. Video features are coloured yellow, orange, and brown to show the repetition of frames.

Based on the skip3 idea, two structures have been investigated. The first structure combines the 32-d video features with audio and GloVe at the input of the ResNet-TDNN encoder (Figure 4.9a). Just the same way as how GloVe and audio features were combined. The second structure (Figure 4.9b) directly concatenates the 32-d video feature with the output of ResNet-TDNN encoder. The context of the encoder output was included to recover some context audio information lost due to down-sampling. In this case, video features didn't pass through the ResNet-TDNN encoder.



(a) Structure-1

(b) Structure-2

Fig. 4.9 Illustration of two skip3 structures. Audio features are shown by blue rectangles and video features are shown by orange rectangles.

Two structures were tested on Session 5 and the results are shown in Table 4.15. Row num.2-6 compare the effect of adding video to Audio+Text-based system. Row num.7-9 compare the effect of adding video to Audio-based system. Comparing Row num.2,3,5 and Row num.7,8, changes in structure due to down-sampling causes performance loss before video was added to the system. Audio+Text-based system has a smaller loss (~2.2%) compared to Audio-based system (~6.5%) mainly due to the sentence-level context information provided by the time asynchronous branch. Context mode provides more information and improves the robustness of the system.

As shown in Row num.1, the video modality in IEMOCAP is weak. The low UA of video alone system was due to the low accuracy in recognizing angry (19.34%) and sad (37.55%). The literature also shows a low accuracy for IEMOCAP video modality. Tripathi et al. [34] got 51.11% WA on Session 5. Poria et al. [66] and Majumder et al. [65] got 53.2% WA and 53.3% WA respectively for 10-fold cross validation.

Row num.	System.	down-samp.	avg WA	avg UA
1	Video only	Yes	53.18	43.61
2	Audio+GloVe+BERT	No	81.22	81.60
3	Audio+GloVe+BERT – Struc.1	Yes	79.45	78.96
4	Audio+GloVe+BERT+Video – Struc.1	Yes	79.53	79.03
5	Audio+GloVe+BERT – Struc.2	Yes	75.26	75.00
6	Audio+GloVe+BERT+Video – Struc.2	Yes	75.58	75.92
7	Audio	No	64.06	64.24
8	Audio – Struc.1	Yes	57.53	57.75
9	Audio+Video – Struc.1	Yes	57.21	59.35

Table 4.15 Results of two structures when frame shift of the system changed from 10ms to 11.12 ms and video was added. Tested on Session 5. Results in percent.

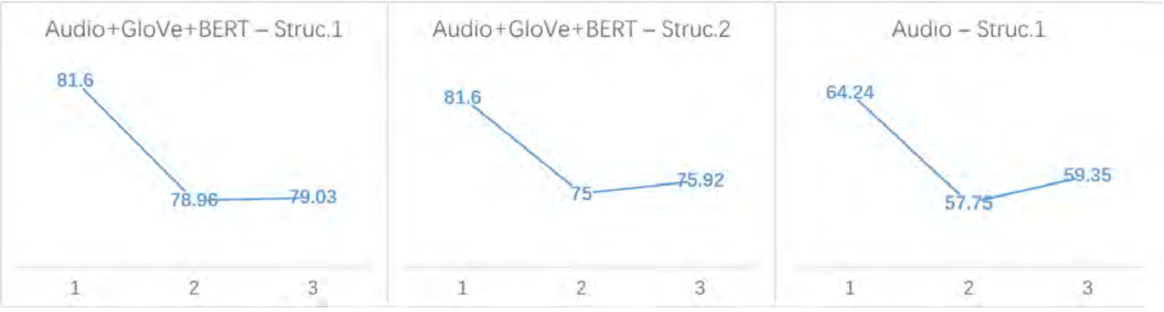


Fig. 4.10 Trends of WA of Audio+GloVe+BERT-based system and Audio-based system when the system was down-sampled and video was added. “1" denotes the original system. “2" denotes the down-sampled system. “3" denotes the down-sampled system with video being added. The slight improvement resulting from video modality cannot compensate for the performance loss resulting from down-sampling.

Comparing Row num.3,4, Row num.5,6, and Row num.8,9 of Tabel 4.15, adding video did improve the result slightly (can also due to noise as the dataset used is rather small). But, comparing Row num.2,4, Row num.2,6, and Row num.7,9, this slight improvement cannot compensate for the performance loss resulting from down-sampling. Taking WA as example (UA has the same trend), the trend when the original system was down-sampled and video was added is summarized in Figure 4.10.

The result of incorporating video modality is not satisfactory for the following possible reasons:

- The classification accuracy of video itself is rather low. Videos in IEMOCAP are primarily used for evaluation and labelling and are not intended to be used as features. As shown in Figure 2.2, it is semi-front half-length image and some expression information may be lost. IEMOCAP provides motion capture (MoCap) data which was captured by placing markers on the subject's face, head, and hand. But only one speaker in each dialog wore motion capture device, which means that half of the data is missing.

- Changing system structure to cope with the mismatch in frame rate of audio and video yields performance loss. Even if video works, the benefits may not be large enough to compensate for that performance loss result from lower frame rate.

- New modality can be informative but can also be misleading, especially when the system has already extracted "enough" information from the current modalities. As shown in Figure 4.10, the higher the accuracy of the system before adding the video ("2" in the figure), the smaller the improvement brought by the video modality. And, as will be shown in Section 4.6.1, adding new features does not necessarily yields better results.

It's worth mentioning that the video results are data-dependent. The results and conclusions will be different if other datasets are used. As incorporating video didn't improve the performance, further result for the system will only use audio and text modalities.

4.6 Discussion

4.6.1 Contribution of different features and modalities

Previous experiments focus on adding features and modalities to the system. This section removes some of the features and examines the contribution of each feature and modality to the system. Results are shown in Table 4.16.

Row num.	system	Audio	GloVe	BERT	avg WA	avg UA
1	Audio only	x			60.64	61.32
2	GloVe only		x		61.27	62.67
3	BERT only			x	71.22	71.88
4	Text only		x	x	70.40	71.88
5	Audio+GloVe	x	x		65.53	66.43
6	Audio+BERT	x		x	74.10	74.89
7	Audio+Text	x	x	x	75.53	76.65

Table 4.16 5-fold cross validation averages on combination of different features and modalities. Results in percent. Reference transcripts used for text modality.

Row num.1-3 of Table 4.16 show the performance using single feature as input. It can be seen that BERT is much more powerful than the others. Then one may ask: now that context BERT is so powerful, is GloVe still useful when BERT is introduced? Comparing Row num.3 and 4, context BERT itself is comparable and even slightly better than combining context BERT with GloVe. But the AER performance of “Audio+BERT" (Row num.6) is worse than that of Audio+GloVe+BERT (Row num.7), as summarized in Equation 4.1 (the comparison in the formula refers to the accuracy). One possible reason is the coupling effect of GloVe and audio features. GloVe and audio features are fused by concatenation at frame level which can help the model capture the correlation between two modalities.

$$\begin{aligned} \text{BERT} &\geq \text{BERT} + \text{GloVe} \\ \text{BERT} + \text{Audio} &< \text{BERT} + \text{GloVe} + \text{Audio} \end{aligned} \tag{4.1}$$

Comparing Audio+GloVe (Row num.5) and Audio+BERT (Row num.6), as GloVe alone system (Row num.2) gives lower results than BERT only system (Row num.3), Audio+GloVe also produces lower results than Audio+BERT system, as summarized in Equation 4.2.

$$\begin{aligned} \text{GloVe} &< \text{BERT} \\ \text{GloVe} + \text{Audio} &< \text{BERT} + \text{Audio} \end{aligned} \tag{4.2}$$

However, comparing GloVe+BERT (Row num.4) and Audio+BERT (Row num.6), although GloVe alone system (Row num.2) behaves better than audio alone system (Row num.1), Audio+BERT produces higher accuracies than GloVe+BERT, as summarized in Equation 4.3. This may due to the fact that audio features provide more complimentary information to BERT than GloVe vectors do.

$$\begin{aligned} &\text{GloVe} > \text{Audio} \\ &\text{GloVe} + \text{BERT} < \text{Audio} + \text{BERT} \end{aligned} \tag{4.3}$$

4.6.2 Transcription generated by ASR

The released transcripts of the IEMOCAP dataset was used for text modality in previous experiments. In practice, manual transcription of utterances is usually not available. This section investigates using transcriptions generated by ASR instead of the ground truth transcriptions so that the AER model only requires speech as input. The Google Cloud Speech API[4] was used to retrieve the transcripts. Among 5531 sentences, 9.7% of them can't be recognized by the API and the WER reaches 40.8% (with Levenshtein distance of 4.43). This can be explained by the relatively low quality of speech, the use of far-field microphones rather than close-talking microphones during recording, and the fact that recognizing emotional speech is indeed a difficult task.

Row num.	Feature	Text	avg WA	avg UA
1	BERT0	Ref	59.55	59.52
2	BERT7	Ref	74.21	73.31
3	Audio+BERT7	Ref	78.97	78.73
4	BERT0	ASR	50.60	46.89
5	BERT7	ASR	65.83	67.17
6	Audio+BERT7	ASR	73.65	73.26
7	Audio+BERT7	Mix	64.38	67.48

Table 4.17 Results using text generated by ASR for the text modality. “Ref" denotes reference transcripts provided in dataset. “Mix" means that the system was trained on reference transcripts and tested on ASR output. Tested on Session 5. Results in percent.

The experiments were implemented on sentence-level BERT embeddings. Results are shown in Table 4.17 and Figure 4.11. The use of ASR leads to a decrease of ~10% and ~7% for single BERT and context BERT, respectively. The decrease is reduced to ~5% when audio features were included. It's worth mentioning that despite the ~5% performance drop, the Audio+BERT7 system in Row 6 is no longer an Audio+Text-based system but an AER system that only uses audio as input. Comparing Row num.6 and 7 of Table 4.17, the system trained and tested both on ASR performs better than system trained on reference transcripts

[4] https://cloud.google.com/speech-to-text/

but tested on ASR outputs. One possible reason is that the system captured some error pattern or that it has learnt to be robust to various type of error. For example, the system might have learnt not to rely too much on a single modality. As discussed in Chapter 1, one major advantage of using multimodal data is that different modalities can augment or complement each other especially when certain modalities are susceptible to noise.

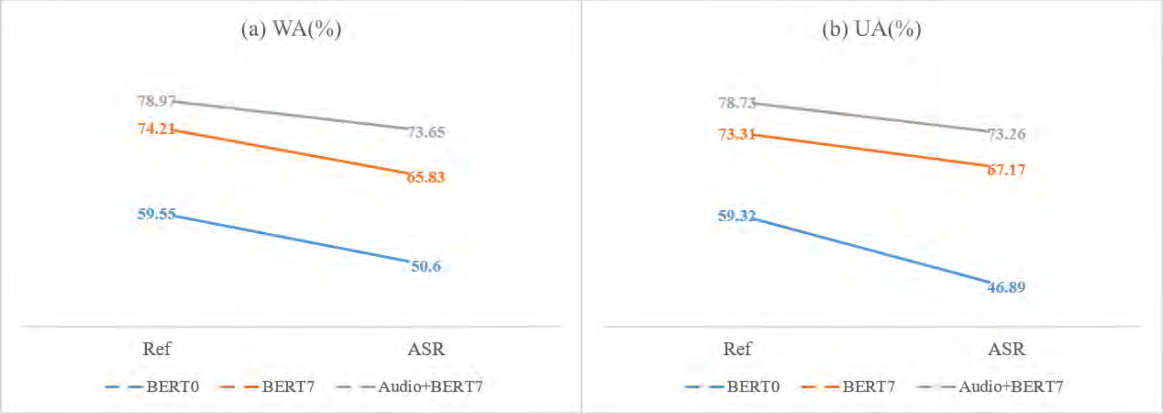


Fig. 4.11 Performance drop caused by using transcription generated by ASR instead of reference transcripts.

As shown in Figure 4.11, the performance loss due to the use of ASR is smaller when context information was added. Context can help in two ways: i) it provides more information; ii) it compensates for the situation that the utterance cannot be recognized by ASR. Splitting out the unrecognized utterances and testing only on those sentences, the results are shown in Table 4.18. Unrecognized utterances were treated as empty sentences and had identical BERT embedding. It can be seen that context mode produces much better results on unrecognized sentences as expected. UA of BERT0 is exactly equal to random guess of a four classification problem. WA of BERT0 relies mostly on prior information only. It shows that the lost information can be partly recovered by the context. Incorporating previous and subsequent sentences provides more information and makes the system more robust.

Feature	avg WA	avg UA
BERT0-ASR	40.80	25.00
BERT7-ASR	51.15	47.02

Table 4.18 Results on sentences that cannot be recognized by the ASR API. BERT embeddings of the unrecognized utterances are the output of an empty sentence, which are identical. Tested on Session 5. Results in percent.

4.6.3 Long-term audio features

Some early research viewed speech emotion recognition as a signal processing or feature engineering problem [80, 81] and shows that apart from short-term audio features such as cepstral and log energy that reflect local speech characteristics in a short time window, long term features such as time envelopes of pitch and energy which reflect voice characteristics over a whole utterance can help improve the performance of GMM-based SER systems. Although some of the long-term audio features can be implicitly captured with the help of neural networks, it is worth experimenting on introducing some explicit long-term audio features. In this section, additional log energies extracted using 75ms and 250ms window with a frame advance of 10ms were added to the system. As the extraction of MFB is based on the short-time transient hypothesis that the signal is assumed to be unchanged within the window, using long window would definitely lose some audio information. But in the context of emotion, it is hard to say whether losing these short-term characteristics is a good thing or not.

Feature	avg WA	avg UA
MFB_{25}	59.39	62.15
MFB_{75}	58.50	60.75
MFB_{250}	54.31	56.82
Audio+GloVe+BERT	81.22	81.60
Audio+GloVe+BERT+MFB_{75}	81.14	81.34
Audio+Glove+BERT+MFB_{250}	82.19	82.18

Table 4.19 Results on AER system with long-term MFB features being added. MFB_{25} denotes MFB using 25ms frame length and MFB_{250} denotes MFB using 250ms frame length. Both of them are 40-d without pitch appended. Test on Session 5. Results in percent. Reference transcripts used for the text modality.

Table 4.19 shows the experiment results on long-term MFB. Long-term MFBs were appended the same way as GloVe, attaching only to the central frame. Although using single long-term MFB yields decrease in performance (the longer frame length, the lower accuracy), appending MFB_{250} to the current system did improve the overall performance.

Feature	avg WA	avg UA
Audio+GloVe+BERT	75.53±3.79	76.65±3.67
Audio+Glove+BERT+MFB_{250}	76.12±4.12	77.36±3.25

Table 4.20 Average and standard deviation of 5-fold cross validation results on system with long-term features. Results in percent.

The results can be noisy as the dataset used in the experiments is rather small. To eliminate the possibility that the increase is caused by noise, 5-fold cross validation was performed and results are shown in Table 4.20. Adding long-term features leads to ~0.65% increase. Appending 250ms MFBs makes the dimension of the input of time synchronous branch 500-d ($82 \times 5 + 50 + 40$).

system	Audio	GloVe	BERT	MFB250	avg WA	avg UA
BERT only			x		71.22	71.88
Text only		x	x		70.4	71.88
Audio+BERT+MFB$_{250}$	x		x	x	74.74	75.60
Audio+Text+MFB$_{250}$	x	x	x	x	76.12	77.36

Table 4.21 5-fold cross validation averages on coupling effect when long-term features were added.

As shown in Table 4.21, the coupling equation (Equation 4.1) described in Section 4.6.1 still stands when long-term feature was introduced.

$$\begin{aligned} \text{BERT} &\geq \text{BERT+Glove} \\ \text{BERT+Audio} &< \text{BERT+Glove+Audio} \\ \text{BERT+Audio+MFB}_{250} &< \text{BERT+Glove+Audio+MFB}_{250} \end{aligned} \tag{4.4}$$

4.6.4 Dropout regularization

Larger feature sets might require more regularization. Dropout was added to the output of ResNet layers and self-attentive layers[5]. Table 4.22 shows the results of system with different dropout probabilities. It can be seen that introducing dropout is beneficial and dropout probability of 0.5 gives relatively higher result.

Feature	dropout	avg WA	avg UA
Audio+Glove+BERT+MFB$_{250}$	/	82.19	82.18
Audio+GloVe+BERT+MFB$_{250}$	0.2	82.67	82.65
Audio+GloVe+BERT+MFB$_{250}$	0.5	**83.45**	**82.92**

Table 4.22 Results on AER system with different dropout probability. Tested on Session 5. Results in percent. Bold: highest value in each column.

[5] Experiments of dropout layers were done with PyTorch in combination with HTK

To eliminate the possibility that the increase is caused by noise, 5-fold cross validation was performed and results are shown in Table 4.23. Adding dropout with 0.5 dropout probability increases the overall performance by ~1%.

Feature	dropout	avg WA	avg UA
Audio+Glove+BERT+MFB$_{250}$	/	76.12±4.12	77.36±3.25
Audio+GloVe+BERT+MFB$_{250}$	0.5	77.57±4.15	78.41±3.71

Table 4.23 Average and standard deviation of 5-fold cross validation results on system with and without dropout. Results in percent.

4.7 Summary

The structure of the final system is shown in Figure 4.12. As adding video didn't improve the result, the final system only uses audio and text modalities. The inputs to time synchronous branch is the 500-d combined feature listed in Table 4.24. The inputs to time asynchronous branch is the 768-d pretrained BERT sentence embedding with context range of [-3,+3].

Feature	dim	context
MFB$_{25}$+pitch+Δ	82	{-2,-1,0,+1,+2}
GloVe	50	{0}
MFB$_{250}$	40	{0}
Total:	$82 \times 5 + 50 + 40 = 500$	

Table 4.24 Inputs to the time synchronous branch.

In addition to the reference transcripts that most related work used for text modality, the use of transcriptions generated by ASR was also studied. Recognizing emotional speech is difficult. The ASR transcription which had WER above 40% led to ~10% decrease in results of single BERT system. This decrease was reduced when context BERT and audio were included. Reference transcripts will be used for cross comparison to the literature.

To compare to the related work, 10-fold cross validation was performed on the final system. Results are shown in Table 4.25. Tripathi et al. [34] and Majumder et al. [65] got 71.04% WA and 76.5% WA on Ses05, respectively. Yoon et al. [35] got 71.8% WA on 5-cv. Poria et al. [66] got 76.1% WA and Yoon et al. [36] got 76.5% WA and 77.6% UA on 10-cv. The detailed features used in related work are summarized in Table 4.26. Comparing to the literature, to

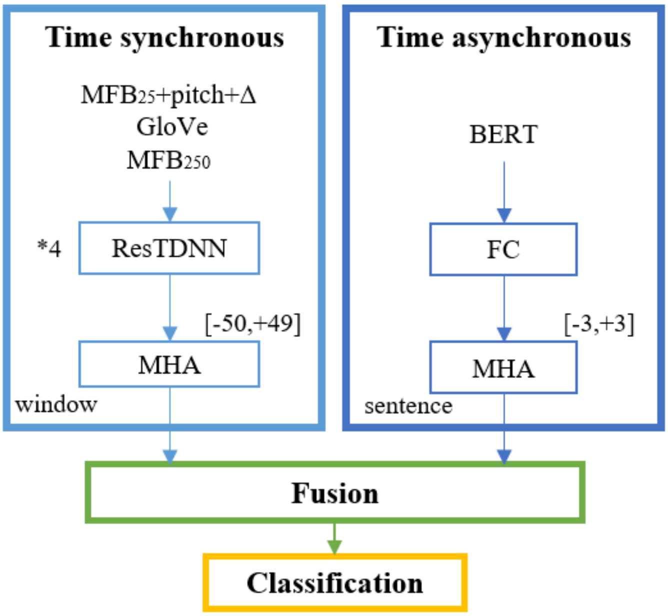


Fig. 4.12 Illustration of the structure of the final system

the best of my knowledge, the final system results are better than previously reported ones.

Test	avg WA	avg UA
Ses05	83.08	83.22
5-cv	77.57	78.41
10-cv	77.76	78.30

Table 4.25 Emotion classification results using different training and testing setting. “Ses05" denotes training on Session 1-4 and testing on Session 5. “5-cv" denotes 5-fold leave-one-session-out cross validation. “10-cv" denotes 10-fold leave-one-speaker-out cross validation. Averages across folds reported. Results in percent.

Paper	Audio	Text	Visual	Test	Result
Tripathi et al. [34]	MFCC	GloVe	MoCap	Ses05	71.04% WA
Majumder et al. [65]	LLD+HSF	word2vec	video	Ses05	76.5% WA
Yoon et al. [35]	MFCC	GloVe	/	5-cv	71.8% UA
Poria et al. [66]	ComParE	n-gram	video	10-cv	76.1% WA
Yoon et al. [36]	MFCC	GloVe	/	10-cv	76.5% WA, 77.6% UA

Table 4.26 Summary of literature with feature used and results.

Chapter 5

Emotion recognition with soft labels

In Chapter 4 and in most of the related work on IEMOCAP, AER system was evaluated by classification accuracy and only strongly emotional utterances were used. However, there's a large proportion of utterances in IEMOCAP that human annotators don't agree on their emotion labels. In other words, these utterances cannot be classified into a specific emotion category. Previous methods developed based on emotion classification are not able to cope with these utterances. This chapter re-examines the emotion recognition problem. “Soft" labels are introduced and the system is trained to match the label distribution instead of doing classification. This approach allows all data to be used and can better reflect the uncertainty in emotion labels. Based on soft labels, other approaches to model the label distribution are discussed in Section 5.3 and Dirichlet Prior Network (DPN) is proposed as a candidate. Experiments in this chapter were implemented in PyTorch in combination with HTK.

5.1 Re-examination of emotion recognition with IEMOCAP

In previous experiments, in order to be consistent with related work, only utterances satisfied the following two conditions were used: i) the majority of annotators agreed on an emotion label; ii) this emotion label belongs to the following four categories: angry, happy (merged with excited), sad, and neutral. This setting leads to nearly half of the data being discarded and is questionable.

The IEMOCAP database contains 10039 utterances in total. Each utterance was evaluated by three annotators. Since the annotators were allowed to tag more than one label for each utterance, some of the utterances may have more than three labels. As shown in Table 5.1, 1186 out of 10039 utterances have more than three labels.

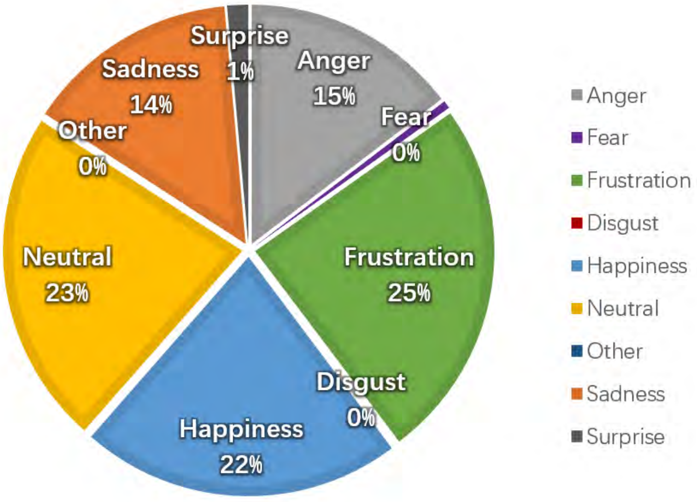
Total utterances	10039
Total evaluations	30117
Evaluation with more than one label	1272
Utterance with more than three labels	1186

Table 5.1 Multi-label evaluations in IEMOCAP

It is necessary to define the concept of “ground truth" and “majority unique". Table 5.2 lists several typical situation that may happen during labelling. If the utterance has majority emotion label, in other words, the emotion category with the highest votes was unique and there’s no tie, then we say this utterance has ground truth or is majority unique. Among the 10039 utterances, 7532 of them are majority unique. The ground truth emotion distribution of these majority unique utterances is shown in Figure 5.1.

e1	e2	e3	Majority	Term	
A	A	A	A	Majority unique-agree3	have “ground truth"
A	A	B	A	Majority unique-agree2	
A	AB	C	A	Majority unique-agree2	
A	B	C	/	No majority	no “ground truth"
A	AB	BC	AB	Majority non-unique	

Table 5.2 Typical situations during labelling. “e1” denotes evaluator 1, etc. ‘A’ ‘B’ ‘C’ denotes different emotion categories.



Anger	1103
Fear	40
Frustration	1849
Disgust	2
Happiness	1636
Neutral	1708
Other	3
Sadness	1084
Surprise	107
Total	7532

Fig. 5.1 Statistics of the ground truth emotion of the majority unique utterances. Excited merged with happy.

The previous work in Chapter 4 (and in most publications on IEMOCAP) only considers majority unique utterances belonging to four emotion category (termed “target-4"): happy (merged with excited), sad, neutral, angry. But ignores the biggest emotion class “frustration" which accounts for 1/4 of the majority unique data, as shown in Figure 5.1.

Another problem concerns the utterances that don’t have ground truth. Emotion is complex, subjective, hard to describe, and hard to label. As shown in Figure 5.2, among all 10039 utterances, only 24% of them gets 100% agreement from all three evaluators. Among the 7532 majority unique utterances, 5149 of them (68.4%) only get agreement from two annotators. Just to clarify, although some utterances may have more than three labels, the denominator in the chart is still three as one evaluator will not tag two identical labels for one sentence.

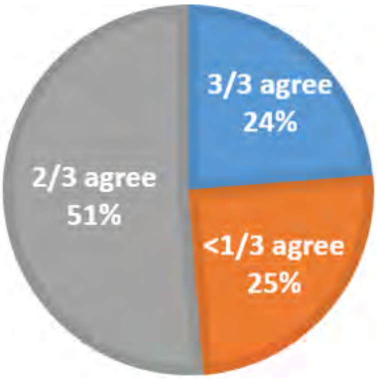


Fig. 5.2 Statistics of the number of evaluators that agreed on the emotion label. “3/3" denotes complete agreement. “2/3" denotes that two out of the three evaluators agreed on the emotion label. “1/3" denotes that the evaluators didn’t reach agreement.

The utterances that don’t have unique majority, which were totally discarded by the previous setting, is exactly the most interesting, meaningful, and useful part of the data. In reality, one cannot expect the user to have strong and unique emotion all the time.

After clarifying the above points, the utterances in IEMOCAP can be divided into different groups according to their label condition, as shown in Figure 5.3. Total utterances are divided into two groups: *utt_ground_truth* and *utt_*1 based on whether the utterance has unique majority or not. *utt_ground_truth* are further divided into *utt_*3 and *utt_*2 based on the number of evaluators agreeing on the majority emotion label. These two groups are further divided according to whether the majority emotion label belongs to the target-4 emotion categories and whether it contains any label that doesn’t belong to target-4. The utterances used in previous setting correspond to $utt\_3\_t4\_ + utt\_2\_t4\_$, which is majority unique utterances with majority label belonging to target-4. And the utterances that at least have one label belonging to target-4 correspond to $Total - utt\_3\_o\_o - utt\_2\_o\_o - utt\_1\_o$.

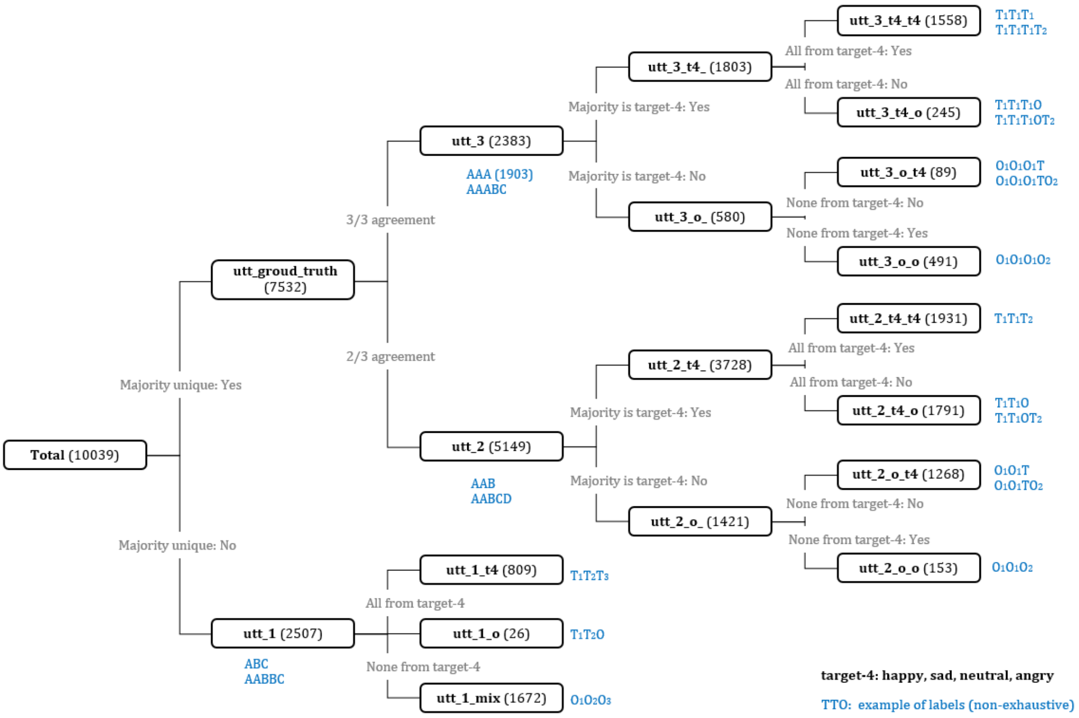
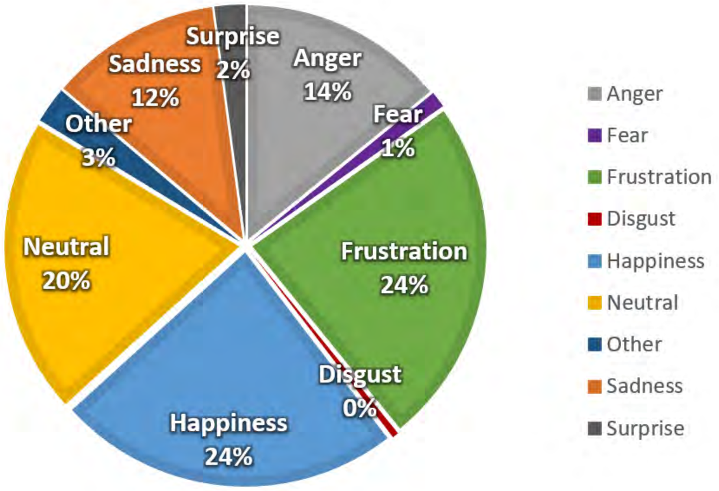


Fig. 5.3 Summary of utterances in IEMOCAP according to their label setting. Number of utterances in each data group is shown in the bracket.



Anger	4433
Fear	348
Frustration	7514
Disgust	137
Happiness	7441
Neutral	6406
Other	783
Sadness	3694
Surprise	658
Total	31414

Fig. 5.4 Statistics of all labels. Excited merged with happy.

Statistics of all labels are shown in Figure 5.4. There is an interesting founding that if we remove labels that don't belong to target-4 and only keep utterances that have at least one label belonging to target-4, as shown in Table 5.3, each utterance lose nearly one label on average but only 670 (6.7%) utterances have been removed. This indicates that although frustration is the largest emotion group, it seldom appears alone. In other words, frustration often appears in combination with other emotions. It can be argued that frustration is a relatively weak emotion comparing to those strong emotions such as happy and angry. In order to capture weak emotion and cope with sentences that don't have ground truth, the emotion problem task was re-defined.

	before	after
label	31313	21974
utt	10039	9369
label/utt	3.12	2.34

Table 5.3 Number of label and utterances before and after removing labels that don't belong to target-4.

As discussed above, emotion is complex and has a large degree of uncertainty. In this case, it is more reasonable to match the distribution of emotion labels than doing classification. The concept of “soft” label is then introduced.

Changes includes: i) changing the problem from a four-way classification to a five-way classification and using all data. Emotion labels that do not fall into target-4 categories are all re-labelled as “other"; ii) changing “hard” label into “soft” label. Previously, as only majority unique utterances were considered, each utterance was tagged with its ground truth label which is referred to as a “hard” label. If each label is treated as a one-hot vector, this operation is equivalent to summing over the one-hot vectors produced by three evaluators and setting the maximum value to one and others to zero, thus producing a new one-hot vector. In the new situation, instead of only keeping the max, the sum vector is normalized. This is equivalent to averaging over the one-hot label vectors and the value of each dimension roughly corresponds to the proportion of that emotion. A three-dimensional example is shown for explanation purpose in Table 5.4. It can be seen that under “hard” label condition, the utterance with “2/3” labeller agreement is forced to become a one-hot vector, thus losing some important information of emotion uncertainty. In order words, utterances with “2/3" agreement are treated in the same way as utterances with “3/3" agreement under “hard" label condition, which is unreasonable.

Five-dimensional soft labels are used in experiments in this chapter and all 10039 utterances are used to train and test the system.

Input:	A A B
Sum:	[2,1,0]
Hard:	[1,0,0]
Soft:	[0.67, 0.33, 0]

Table 5.4 Example of the operation of hard and soft label where A=[1,0,0], B=[0,1,0], C=[0,0,1].

5.2 Experiments on 5-d soft labels

The 5-d soft label system was trained to learn the label distribution of an utterance by minimizing the Kullback-Leibler (KL) divergence between the target soft label and the prediction. The KL divergence is a measure of the similarity between two distributions. Smaller KL divergence indicates more similar distributions and the minimum (zero) is reached when two contributions are identical. A hard label system was also built for comparison, which was trained on five-way classification. In this section, systems were trained on Session 1-4 and tested on Session 5. Soft system was trained using all utterances in Session 1-4 (7083 training and 786 validation). As hard label system requires utterance to have ground truth, it was trained using all majority unique utterances in Session 1-4 (5291 training and 663 validation). Both systems were tested on the whole Session 5. As listed in Table 5.5, Session 5 can be split into three groups according to the degree of emotion uncertainty shown in Figure 5.3.

Data group	Total	utt_3	utt_2	utt_1
Number of sentences	2170	479	1171	520

Table 5.5 Number of utterances in different data group of Session 5.

Systems were first evaluated by five-way classification accuracy (Table 5.6) then by KL divergence (Table 5.7). Classification was only done on majority unique utterances in Session 5 ($utt\_3 + utt\_2$) while the KL divergence was computed on all utterances in Session 5 and was averaged over all windows.

majority_unique	avg WA	avg UA
Hard	83.45	82.92
Soft	74.30	71.28

Table 5.6 Five-way classification results of hard label system and soft label system. Tested on majority unique utterances in Session 5. Results in percent.

Total	KL Divergence	Entropy
Hard	0.7695	0.6850
Soft	0.5000	1.0460

Table 5.7 KL divergence and entropy of hard label system and soft label system. Tested on all utterances in Session 5. Averaged over windows. Natural logarithm base used.

Accuracies are expected to decrease when changing a four-way classification into a five-way classification. However, comparing the five-way results in the 2nd row of Table 5.6 and the four-way results in the 2nd row of Table 4.25, the five-classification system even had slightly better performance. This may due to the increase in the number of training samples used by five-way classification. As discussed in the previous section, frustration is a relatively weak emotion. Since related work excluded it in training and testing, one may assume that frustration is difficult to detect. However, the even better five-classification results indicate that the system proposed in this thesis is able to classify frustration (although the fifth class “other" contains frustration and all other emotions that don’t belong to target-4, it is dominated by frustration as shown in Figure 5.4). This system works well with frustration.

Table 5.6 shows that the system trained using soft labels have lower classification accuracies than the system trained using hard labels. This is as expected. Training a hard system was learning a 0/1 distribution. Uncertainty was introduced when training a soft system. The confidence went down and the system was more prone to classification error. However, as shown in Table 5.7, the soft system has much lower KL divergence, which indicates that the soft system can match the target label distribution better.

Session 5 was then split into different data groups as listed in Table 5.5 and the test results of each group were reported separately, as shown in Table 5.8 – 5.10. As utterances in *utt*_1 don’t have unique majority, only KL divergence and entropy are reported for that group.

Trends are summarized in Figure 5.5.

*utt*_3	KL Divergence	Entropy	avg WA(%)	avg UA(%)
Hard	0.4502	0.6030	87.89	87.93
Soft	0.5640	0.9670	80.17	78.94

Table 5.8 Comparison of hard-label system and soft-label system. Tested on *utt*_3 utterances in Session 5.

*utt*_2	KL Divergence	Entropy	avg WA(%)	avg UA(%)
Hard	0.7122	0.7449	81.64	80.59
Soft	0.4687	1.0489	71.90	67.50

Table 5.9 Comparison of hard-label system and soft-label system. Tested on *utt*_2 utterances in Session 5.

*utt*_1	KL Divergence	Entropy
Hard	1.2958	0.6355
Soft	0.5008	1.1359

Table 5.10 Comparison of hard-label system and soft-label system. Tested on *utt*_1 utterances in Session 5. Natural logarithm base used.

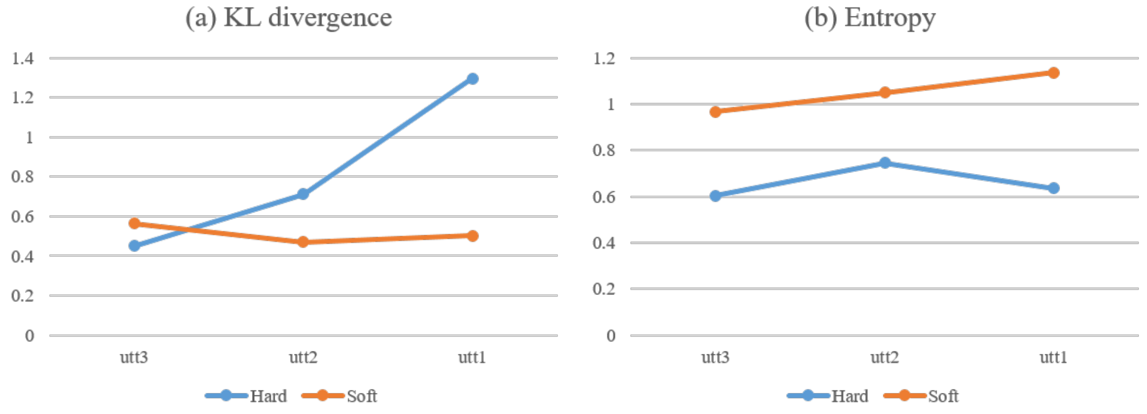


Fig. 5.5 Comparison of hard-label system and soft-label system in terms of KL divergence (a) and entropy (b) of three data groups in Session 5.

As shown in Figure 5.5b, the soft system has higher entropy than hard system in all cases because soft labels have flatter label distribution and larger uncertainty. Despite the larger uncertainty, soft labels better reflect the true uncertainty of the emotion of the sentence. But this uncertainty also leads to less confidence in classification and lower classification accuracy.

Also due to the larger uncertainty, as shown in Figure 5.5a, the soft system has larger KL divergence than the hard system on utt_3 utterances which are strongly emotional utterances that all three evaluators agree on the emotion label. But the soft label system has smaller KL divergence on utt_2 and significantly smaller KL on utt_1. The hard system nearly can't learn the label distribution of utt_1 utterances as it can only be trained on majority unique data. The soft system improves distribution matching of utt_1 sentences by significantly better KL divergence. Note that utt_2 utterances were treated as one-hot vectors in hard systems but they are actually not. Matching distribution is more reasonable than doing classification for these data. In sum, for those sentences that human annotators don't completely agree, which account for 76% of total data as shown in Figure 5.2, soft labels can better match their label distribution. This is one of the major benefits of using soft labels.

Besides, as shown in Figure 5.5b, the entropy of soft system increases when fewer evaluators reach agreement. This is because the label distribution becomes flatter and the uncertainty in emotion of the utterance increases. If three evaluators all give different labels, these emotions are equally likely and the entropy is the largest[1]. Human evaluators are uncertain about the emotion of the utterance, so does the machine.

Overall, the hard system produces better classification results on strong emotional utterances while the soft system can cope with all labelling situation and yields smaller KL divergence. High accuracy and low KL divergence cannot be achieved at the same time. The following section discusses approaches to better modelling the label distribution and can probably solve this problem.

5.3 Label distribution modelling approaches

Consider the label of an utterance as a five-dimensional vector, then it can be viewed as a distribution, specifically, a categorical distribution over five emotions. Each one-hot label from each annotator can be regarded as a sample drawn from this categorical distribution. As

[1] In the extreme case, all five emotion categories have the same probability. The maximum entropy value is then 1.6094. Natural logarithm base is used.

mentioned before, using hard labels corresponds to a maximization operation. If the distribution has more than one maximal value, the utterance cannot be used and was discarded. Taking the average then corresponds to implementing Maximum Likelihood Estimate (MLE) given the observation (label samples). Without a prior, the MLE of a categorical distribution is equivalent to the relative frequency. Taking average is simple and easy to implement, but this approach is not flawless. Taking coin toss as an example. Even if three heads have been observed out of four tosses, it is still very unlikely that the probability of head is 0.75 due to the prior knowledge that a coin is more likely to be fair. Obtaining soft labels by simply taking average doesn't take prior information into account. Therefore, a better way to model label distribution is to include a prior, in other words, using Bayesian approach.

Maximum A Posteriori (MAP) is a common method of Bayesian learning, which can be seen as adding a pseudo count. However, MAP corresponds to a global prior. Prior probability is computed by counting the relative frequency of each emotion in the whole dataset, as in Figure 5.4. This setting doesn't apply to the emotion recognition problem here. As mentioned in Chapter 1, emotion can be easily affected by contextual information and is extremely personal. Utterances produced by different people under different situation should not carry the same prior. Instead, a local prior for each sentence is more suitable in this situation.

The set of emotion labels is a categorical distribution. A conjugate prior of a categorical distribution can be a Dirichlet distribution. The proposal is to train a Dirichlet Prior Network (DPN) [82] which generates local prior for each utterance. A Dirichlet distribution (Equation 5.1) is parameterized by its concentration parameters α, which are exactly the pseudo count. However, in this case, instead of having a global pseudo count, DPN generates a pseudo count for each utterance individually.

$$\mathrm{Dir}(\mu \mid \alpha) = \frac{\Gamma(\alpha_0)}{\prod_{c=1}^{K}\Gamma(\alpha_c)}\prod_{c=1}^{K}\mu_c^{\alpha_c-1}, \quad \alpha_c > 0, \alpha_0 = \sum_{c=1}^{K}\alpha_c \tag{5.1}$$

The process is summarized in Figure 5.6. The DPN generates the concentration parameters α of the Dirichlet prior distribution. The categorical label distribution μ is a sample drawn from the Dirichlet distribution parameterized by those concentration parameters. The predicted emotion class ω_c is a sample drawn from that categorical distribution. The posterior probability over the emotion class can be easily obtained from the concentration parameters: $p(\omega_c|x^*, \mathscr{D}) = \frac{\alpha_c}{\alpha_0}$.

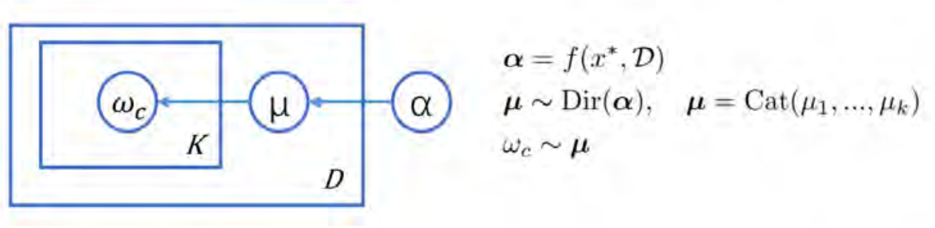

ω_c
μ
α
K
D
$\alpha = f(x^*, \mathcal{D})$
$\mu \sim \mathrm{Dir}(\alpha), \quad \mu = \mathrm{Cat}(\mu_1, ..., \mu_k)$
$\omega_c \sim \mu$

Fig. 5.6 Illustration of the DPN process. $\mathscr{D}$ is the training set and K is the number of emotion classes.

Training an AER system with DPN can be easily implemented by replacing the current training criterion of minimizing KL divergence loss by maximizing the log likelihood of the prior distribution. Assuming each utterance x_i has m_i one-hot labels from three evaluators $\mu^{(i_1)}, ..., \mu^{(i_{m_i})}$. Given data $\mathscr{D} = \{x^{(n)}, \mu^{(n_1)}, ..., \mu^{(n_{m_n})}\}_{n=1}^{N}$, the optimization target of the DPN $f(x, \theta)$ is to maximize the log likelihood $\log p(\mu|x, \theta)$:

$$\mathscr{L}(\theta) = \sum_{i=1}^{n} \log p(\mu^{(i)}|x^{(i)}, \theta) \tag{5.2}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{m_i} \log \mathrm{Dir}(\mu^{(i_j)}|\alpha^{(i)}) \tag{5.3}$$

The training process is to obtain $\theta^* = \arg\max_\theta \mathscr{L}(\theta)$. The detailed theory and explanation about DPN can be found in Appendix A.

Chapter 6

Conclusion and future work

6.1 Conclusion

In this thesis, a complete multimodal emotion recognition system was build to predict the emotion of a speaker given speech, text and video information. The system achieves state-of-the-art classification accuracy of 77.76% WA and 78.30% UA. The innovative use of context BERT and long-term audio features has been shown to be beneficial. Apart from reference transcripts, transcriptions generated by ASR was also investigated. The high WER shows that recognizing emotional speech is still a difficult task for ASR and using multimodal inputs and context information can make the system more robust to various type of error. The contribution of each modality and the correlation between features such as the coupling effect and the complimentary effect have been analyzed.

The thesis also re-defined the emotion recognition problem. Given the fact that large proportion of data doesn't get complete agreement from the annotators, matching the label distribution is more reasonable than doing classification. This has two main benefits: i) all data in the database can be used; ii) it better matches the label distribution of the utterance and can better reflect the uncertainty in emotion labels. The thesis shows that using soft labels improves distribution matching by a significantly better KL divergence. DPN was proposed as a candidate to better model label distribution which generates local prior for each label distribution.

6.2 Future work

There are several aspects of future work. First, as video data in IEMOCAP dataset doesn't work well due to several reasons, other dataset such as CMU-MOSEI Dataset[1] could be used as supplement. It includes more data, more speakers, as well as videos of better quality. Second, it has been shown that criteria used to define the cross-validation folds has large effect on results [83]. Currently, the validation set was chosen by random. The results may be improved by carefully choosing validation set. Third and the most important future work are the experiments on DPN.

Emotion recognition tasks are challenging as emotion is extremely personal but personal data is hard to obtain. Researchers are working on balancing the personalization and generalization of emotion. Some propose to use demographic information of people such as gender, age, occupation and then use transfer learning or adaptation methods to adapt the general system trained on large number of speakers to the individual user. DPN can also be a possible approach to this problem, which can be used to generate prior based on speaker's demographic information. Theses are all interesting directions for future research.

[1] https://github.com/A2Zadeh/CMU-MultimodalSDK

References

[1] G. Sun, C. Zhang, and P. C. Woodland. Speaker diarisation using 2d self-attentive combination of embeddings. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5801–5805, 2019.

[2] Steve Young, Gunnar Evermann, M.J.F. Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, James Odell, Dave Ollason, Daniel Povey, Anton Ragni, Valtcho Valtchev, Philip Woodland, and Chao Zhang. *The HTK Book (version 3.5a)*. 12 2015.

[3] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur. A pitch extraction algorithm tuned for automatic speech recognition. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2494–2498, 2014.

[4] R. W. Picard. Affective computing, 1995.

[5] M. Sreeshakthy and J. Preethi. Classification of human emotion from deap eeg signal using hybrid improved neural networks with cuckoo search. 2016.

[6] Qiang Ji, Zhiwei Zhu, and P. Lan. Real-time nonintrusive monitoring and prediction of driver fatigue. *IEEE Transactions on Vehicular Technology*, 53(4):1052–1068, 2004.

[7] M. Chen, Y. Zhang, M. Qiu, N. Guizani, and Y. Hao. Spha: Smart personal health advisor based on deep analytics. *IEEE Communications Magazine*, 56(3):164–169, 2018.

[8] F. Doctor, C. Karyotis, R. Iqbal, and A. James. An intelligent framework for emotion aware e-healthcare support systems. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, 2016.

[9] K. Lin, F. Xia, W. Wang, D. Tian, and J. Song. System design for big data application in emotion-aware healthcare. *IEEE Access*, 4:6901–6909, 2016.

[10] S. Bhosale, R. Chakraborty, and S. K. Kopparapu. Deep encoded linguistic and acoustic cues for attention based end to end speech emotion recognition. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7189–7193, 2020.

[11] Z. Lu, L. Cao, Y. Zhang, C. Chiu, and J. Fan. Speech sentiment analysis via pre-trained features from end-to-end asr models. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7149–7153, 2020.

[12] Raghavendra Pappagari, Tianzi Wang, Jesus Villalba, Nanxin Chen, and Najim Dehak. X-vectors meet emotions: A study on dependencies between emotion and speaker recognition. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020. doi: 10.1109/icassp40776.2020.9054317. URL http://dx.doi.org/10.1109/ICASSP40776.2020.9054317.

[13] A. Nediyanchath, P. Paramasivam, and P. Yenigalla. Multi-head attention for speech emotion recognition with auxiliary learning of gender recognition. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7179–7183, 2020.

[14] Jianyou Wang, Michael Xue, Ryan Culhane, Enmao Diao, Jie Ding, and Vahid Tarokh. Speech emotion recognition with dual-sequence lstm architecture. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020. doi: 10.1109/icassp40776.2020.9054629. URL http://dx.doi.org/10.1109/ICASSP40776.2020.9054629.

[15] Y. Xu, H. Xu, and J. Zou. Hgfm : A hierarchical grained and feature model for acoustic emotion recognition. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6499–6503, 2020.

[16] Maja Pantic, Nicu Sebe, Jeffrey Cohn, and Thomas Huang. Affective multimodal human-computer interaction. pages 669–676, 01 2005. doi: 10.1145/1101149.1101299.

[17] M. Soleymani, M. Pantic, and T. Pun. Multimodal emotion recognition in response to videos. *IEEE Transactions on Affective Computing*, 3(2):211–223, 2012.

[18] Trisha Mittal, Uttaran Bhattacharya, Rohan Chandra, Aniket Bera, and Dinesh Manocha. M3er: Multiplicative multimodal emotion recognition using facial, textual, and speech cues. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(02):1359–1367, Apr 2020. ISSN 2159-5399. doi: 10.1609/aaai.v34i02.5492. URL http://dx.doi.org/10.1609/aaai.v34i02.5492.

[19] Linlin Chao, Jianhua Tao, Minghao Yang, Ya Li, and Zhengqi Wen. Long short term memory recurrent neural network based multimodal dimensional emotion recognition. pages 65–72, 10 2015. doi: 10.1145/2808196.2811634.

[20] Linlin Chao, Jianhua Tao, Minghao Yang, Ya Li, and Zhengqi Wen. Audio visual emotion recognition with temporal alignment and perception attention, 2016.

[21] C. Huang and S. S. Narayanan. Deep convolutional recurrent neural network with attention mechanism for robust speech emotion recognition. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 583–588, 2017.

[22] Martin Wöllmer, Florian Eyben, Stephan Reiter, Björn Schuller, Cate Cox, Ellen Douglas-Cowie, and Roddy Cowie. Abandoning emotion classes – towards continuous emotion recognition with modelling of long-range dependencies. pages 597–600, 01 2008.

[23] George Trigeorgis, Fabien Ringeval, Raymond Brueckner, Erik Marchi, Mihalis Nicolaou, Björn Schuller, and Stefanos Zafeiriou. Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. pages 5200–5204, 03 2016. doi: 10.1109/ICASSP.2016.7472669.

[24] Moataz Ayadi, Mohamed S. Kamel, and Fakhri Karray. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44: 572–587, 03 2011. doi: 10.1016/j.patcog.2010.09.020.

[25] Carlos Busso, Sungbok Lee, and Shrikanth Narayanan. Using neutral speech models for emotional speech analysis. volume 4, pages 2225–2228, 01 2007.

[26] Emma Rodero. Intonation and emotion: Influence of pitch levels and contour type on creating emotions. *Journal of voice : official journal of the Voice Foundation*, 25: e25–34, 01 2011. doi: 10.1016/j.jvoice.2010.02.002.

[27] F. Eyben, K. R. Scherer, B. W. Schuller, J. Sundberg, E. André, C. Busso, L. Y. Devillers, J. Epps, P. Laukka, S. S. Narayanan, and K. P. Truong. The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing. *IEEE Transactions on Affective Computing*, 7(2):190–202, 2016.

[28] Tomas Mikolov, Ilya Sutskever, Kai Chen, G.s Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26, 10 2013.

[29] Jeffrey Pennington, Richard Socher, and Christoper Manning. Glove: Global vectors for word representation. volume 14, pages 1532–1543, 01 2014. doi: 10.3115/v1/D14-1162.

[30] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. *31st International Conference on Machine Learning, ICML 2014*, 4, 05 2014.

[31] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018. doi: 10.18653/v1/n18-1202. URL http://dx.doi.org/10.18653/v1/N18-1202.

[32] Alec Radford. Improving language understanding by generative pre-training. 2018.

[33] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

[34] Samarth Tripathi, Sarthak Tripathi, and Homayoon Beigi. Multi-modal emotion recognition on iemocap dataset using deep learning, 2018.

[35] Seunghyun Yoon, Seokhyun Byun, and Kyomin Jung. Multimodal speech emotion recognition using audio and text. *2018 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2018. doi: 10.1109/slt.2018.8639583. URL http://dx.doi.org/10.1109/SLT.2018.8639583.

[36] Seunghyun Yoon, Seokhyun Byun, Subhadeep Dey, and Kyomin Jung. Speech emotion recognition using multi-hop attention mechanism. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019. doi: 10.1109/icassp.2019.8683483. URL http://dx.doi.org/10.1109/ICASSP.2019.8683483.

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

[38] Wan Ding, Mingyu Xu, Dongyan Huang, Weisi Lin, Minghui Dong, Xinguo Yu, and Haizhou Li. Audio and face video emotion recognition in the wild using deep neural networks and small datasets. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ICMI '16, page 506–513, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450345569. doi: 10.1145/2993148.2997637. URL https://doi.org/10.1145/2993148.2997637.

[39] T. R. Almaev and M. F. Valstar. Local gabor binary patterns from three orthogonal planes for automatic facial expression recognition. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pages 356–361, 2013.

[40] A. Dhall, A. Asthana, R. Goecke, and T. Gedeon. Emotion recognition using phog and lpq features. In *Face and Gesture 2011*, pages 878–883, 2011.

[41] G. Muhammad, M. Alsulaiman, S. U. Amin, A. Ghoneim, and M. F. Alhamid. A facial-expression monitoring system for improved healthcare in smart cities. *IEEE Access*, 5:10871–10881, 2017.

[42] Mengyi Liu, Ruiping Wang, Shaoxin Li, Shiguang Shan, Zhiwu Huang, and Xilin Chen. Combining multiple kernel methods on riemannian manifold for emotion recognition in the wild. In *Proceedings of the 16th International Conference on Multimodal Interaction*, ICMI '14, page 494–501, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450328852. doi: 10.1145/2663204.2666274. URL https://doi.org/10.1145/2663204.2666274.

[43] Nianyin Zeng, Hong Zhang, Baoye Song, Weibo Liu, Yurong Li, and Dobaie Abdullah. Facial expression recognition via learning deep sparse autoencoders. *Neurocomputing*, 09 2017. doi: 10.1016/j.neucom.2017.08.043.

[44] Anbang Yao, Junchao Shao, Ningning Ma, and Yurong Chen. Capturing au-aware facial features and their latent relations for emotion recognition in the wild. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, page 451–458, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450339124. doi: 10.1145/2818346.2830585. URL https://doi.org/10.1145/2818346.2830585.

[45] Bo-Kyeong Kim, Hwaran Lee, Jihyeon Roh, and Soo-Young Lee. Hierarchical committee of deep cnns with exponentially-weighted decision fusion for static facial expression recognition. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, page 427–434, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450339124. doi: 10.1145/2818346.2830590. URL https://doi.org/10.1145/2818346.2830590.

[46] Andrey Savchenko. Deep convolutional neural networks and maximum-likelihood principle in approximate nearest neighbor search. pages 42–49, 05 2017. ISBN 978-3-319-58837-7. doi: 10.1007/978-3-319-58838-4_5.

[47] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks, 2013.

[48] Hong-Wei Ng, Viet Dung Nguyen, Vassilios Vonikakis, and Stefan Winkler. Deep learning for emotion recognition on small datasets using transfer learning. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, page 443–449, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450339124. doi: 10.1145/2818346.2830593. URL https://doi.org/10.1145/2818346.2830593.

[49] Heysem Kaya, Furkan Gürpınar, and Albert Salah. Video-based emotion recognition in the wild using deep transfer learning and score fusion. *Image and Vision Computing*, 02 2017. doi: 10.1016/j.imavis.2017.01.012.

[50] Alexandr Rassadin, Alexey Gruzdev, and Andrey Savchenko. Group-level emotion recognition using transfer learning from face identification. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, ICMI '17, page 544–548, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450355438. doi: 10.1145/3136755.3143007. URL https://doi.org/10.1145/3136755.3143007.

[51] Ian Goodfellow, Dumitru Erhan, Pierre Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov, John Park, and Y. Bengio. Challenges in representation learning: A report on three machine learning contests. *Neural Networks*, 64, 07 2013. doi: 10.1016/j.neunet.2014.09.005.

[52] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10): 1499–1503, 2016.

[53] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 09 2014.

[54] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001.

[55] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

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

[57] Seyedmahdad Mirsamadi and Emad Barsoum. Automatic speech emotion recognition using recurrent neural networks with local attention. 03 2017. doi: 10.1109/ICASSP.2017.7952552.

[58] Wenjing Han, Huabin Ruan, Xiaomin Chen, Zhixiang Wang, Haifeng Li, and Björn Schuller. Towards temporal modelling of categorical speech emotion recognition. In *Proc. Interspeech 2018*, pages 932–936, 2018. doi: 10.21437/Interspeech.2018-1858. URL http://dx.doi.org/10.21437/Interspeech.2018-1858.

[59] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning Representations by Back-Propagating Errors*, page 696–699. MIT Press, Cambridge, MA, USA, 1988. ISBN 0262010976.

[60] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

[61] Mike Schuster and Kuldip Paliwal. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45:2673 – 2681, 12 1997. doi: 10.1109/78.650093.

[62] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. 06 2014. doi: 10.3115/v1/D14-1179.

[63] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3):328–339, 1989.

[64] Jaejin Cho, Raghavendra Pappagari, Purva Kulkarni, Jesús Villalba, Yishay Carmiel, and Najim Dehak. Deep neural networks for emotion recognition combining audio and transcripts. *Interspeech 2018*, Sep 2018. doi: 10.21437/interspeech.2018-2466. URL http://dx.doi.org/10.21437/Interspeech.2018-2466.

[65] N. Majumder, D. Hazarika, A. Gelbukh, E. Cambria, and S. Poria. Multimodal sentiment analysis using hierarchical fusion with context modeling. *Knowledge-Based Systems*, 161:124–133, Dec 2018. ISSN 0950-7051. doi: 10.1016/j.knosys.2018.07.041. URL http://dx.doi.org/10.1016/j.knosys.2018.07.041.

[66] Soujanya Poria, Navonil Majumder, Devamanyu Hazarika, Erik Cambria, Alexander Gelbukh, Amir Hussain, and Erik Cambria. Multimodal sentiment analysis: Addressing key issues and setting up the baselines. *IEEE Intelligent Systems*, 33(6):17–25, Nov 2018. ISSN 1941-1294. doi: 10.1109/mis.2018.2882362. URL http://dx.doi.org/10.1109/MIS.2018.2882362.

[67] Efthymios Tzinis and Alexandros Potamianos. Segment-based speech emotion recognition using recurrent neural networks. 07 2017. doi: 10.1109/ACII.2017.8273599.

[68] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015. doi: 10.18653/v1/d15-1166. URL http://dx.doi.org/10.18653/v1/D15-1166.

[69] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. A time delay neural network architecture for efficient modeling of long temporal contexts. In *INTERSPEECH*, 2015.

[70] Florian L. Kreyssig, C. Zhang, and Philip C. Woodland. Improved tdnns using deep kernels and frequency dependent grid-rnns. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4864–4868, 2018.

[71] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. doi: 10.1109/cvpr.2016.90. URL http://dx.doi.org/10.1109/cvpr.2016.90.

[72] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *ArXiv*, abs/1703.03130, 2017.

[73] Chao Zhang, Zichao Yang, Xiaodong He, and Li Deng. Multimodal intelligence: Representation learning, information fusion, and applications. *IEEE Journal of Selected Topics in Signal Processing*, 14:478–493, 2020.

[74] Antonios Anastasopoulos, Shankar Kumar, and Hank Liao. Neural language modeling with visual features, 2019.

[75] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering, 2016.

[76] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. Dual attention networks for multimodal reasoning and matching. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. doi: 10.1109/cvpr.2017.232. URL http://dx.doi.org/10.1109/CVPR.2017.232.

[77] J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000.

[78] Jin-Hwa Kim, Kyoung On, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard product for low-rank bilinear pooling. 10 2016.

[79] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6738–6746, 2017.

[80] S. Wu, T. H. Falk, and W. Chan. Automatic recognition of speech emotion using long-term spectro-temporal features. In *2009 16th International Conference on Digital Signal Processing*, pages 1–6, 2009.

[81] Pierre Dumouchel, Najim Dehak, Yazid Attabi, Réda Dehak, and Narjès Boufaden. Cepstral and long-term features for emotion recognition. In *INTERSPEECH*, 2009.

[82] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks, 2018.

[83] L. Pepino, P. Riera, L. Ferrer, and A. Gravano. Fusion approaches for emotion recognition from speech using acoustic and text-based features. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6484–6488, 2020.

Appendix A

Dirichlet Prior Network

A Dirichlet distribution (Equation A.1) is a prior distribution over a categorical distribution, which is parameterized by its concentration parameters α.

$$\mathrm{Dir}(\mu \mid \alpha) = \frac{\Gamma(\alpha_0)}{\prod_{c=1}^{K}\Gamma(\alpha_c)} \prod_{c=1}^{K} \mu_c^{\alpha_c - 1}, \quad \alpha_c > 0, \alpha_0 = \sum_{c=1}^{K} \alpha_c \tag{A.1}$$

The objective of the AER system is now to predict the expected categorical distribution over emotion labels of the utterance under a Dirichlet prior, as described in Equation A.2

$$p(\omega_c|x^*, \mathscr{D}) = \int p(\omega_c|\mu)\ p(\mu|x^*, \mathscr{D}) d\mu \tag{A.2}$$

where μ is a categorical distribution which is a vector of probabilities: $[\mu_1, ..., \mu_k]^T = [P(y = \omega_1), ..., P(y = \omega_k)]^T$. $p(\mu|x^*, \mathscr{D})$ is the distribution over predictive categoricals, a distribution over distribution. In this case, it is Dirichlet.

$$p(\mu|x^*, \mathscr{D}) = \mathrm{Dir}(\mu \mid \alpha)$$

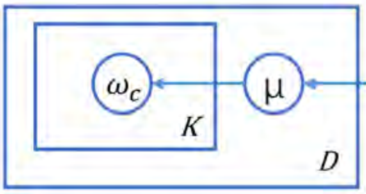
In other words, μ is a sample drawn from the Dirichlet distribution $\mathrm{Dir}(\alpha)$. The predicted class label ω_c is a sample drawn from the categorical distribution $\mathrm{Cat}(\mu)$.

$$\begin{aligned} &\mu \sim \mathrm{Dir}(\alpha) \\ &\omega_c \sim \mu \end{aligned}$$

A DPN generates the concentration parameters α of the Dirichlet distribution.

$$\alpha = f(x^*, \mathscr{D}) \tag{A.3}$$

The process can be summarized in the following graph:



$$\boldsymbol{\alpha} = f(x^*, \mathcal{D})$$
$$\boldsymbol{\mu} \sim \mathrm{Dir}(\boldsymbol{\alpha}), \quad \boldsymbol{\mu} = \mathrm{Cat}(\mu_1, ..., \mu_k)$$
$$\omega_c \sim \boldsymbol{\mu}$$

The posterior over class labels are given by the mean of the Dirichlet:

$$p(\omega_c|x^*, \mathscr{D}) = \int p(\omega_c|\mu)\ p(\mu|x^*, \mathscr{D})d\mu = \frac{\alpha_c}{\alpha_0} \tag{A.4}$$

If an exponential output function is used for the DPN: $\alpha_c = \exp(z_c)$, then the expected posterior probability of a label ω_c recovers the standard softmax function:

$$p(\omega_c|x^*, \mathscr{D}) = \frac{\exp(z_c)}{\sum_{k=1}^{K} \exp(z_k)} \tag{A.5}$$

Assuming each utterance x_i has m_i one-hot labels from three evaluators $\mu^{(i_1)}, ..., \mu^{(i_{m_i})}$. Given training data $\mathscr{D} = \{x^{(n)}, \mu^{(n_1)}, ..., \mu^{(n_{m_n})}\}_{n=1}^{N}$, the optimization target of the DPN $f(x, \theta)$ is to maximize the log likelihood $\log p(\mu|x, \theta)$:

$$\begin{aligned}
\mathscr{L}(\theta) &= \log p(\mu|x, \theta) && \text{(A.6)}\\
&= \sum_{i=1}^{n} \log p(\mu^{(i)}|x^{(i)}, \theta) && \text{(A.7)}\\
&= \sum_{i=1}^{n} \log \prod_{j=1}^{m_i} p(\mu^{(i_j)}|x^{(i)}, \theta) && \text{(A.8)}\\
&= \sum_{i=1}^{n} \sum_{j=1}^{m_i} \log p(\mu^{(i_j)}|x^{(i)}, \theta) && \text{(A.9)}\\
&= \sum_{i=1}^{n} \sum_{j=1}^{m_i} \log \mathrm{Dir}(\mu^{(i_j)}|\alpha^{(i)}) && \text{(A.10)}\\
&= \sum_{i=1}^{n} \sum_{j=1}^{m_i} \log \left[\frac{\Gamma(\alpha_0^{(i)})}{\prod_{c=1}^{K} \Gamma(\alpha_c^{(i)})} \prod_{c=1}^{K} (\mu_c^{(i_j)})^{\alpha_c^{(i)}-1} \right] && \text{(A.11)}
\end{aligned}$$

The training target is to obtain $\theta^* = \arg\max_\theta \mathscr{L}(\theta)$. In sum, DPN can easily fit into the current framework by simply changing the output activation function to exponential and the loss function to log likelihood.