A learned emotion space for emotion recognition and emotive speech synthesis

Zack Hodari

Master of Science by Research

Centre for Doctoral Training in Data Science

School of Informatics

University of Edinburgh

2017

Abstract

Emotion is a complex phenomenon that contributes heavily to human communication. Typically, human-computer interaction and text-to-speech systems do not account for emotion information, possibly due to lack of accurate emotion recognition and emotive speech synthesis methods. It seems likely that emotion recognition and synthesis has the ability to greatly improve how humans interface with machines.

Existing methods in speech recognition make use of a wide range of features and models. However, these methods make use of either categorical, or appraisal-based emotion descriptions. We believe these have flaws that limit their ability to describe emotion.

We present several neural network models for emotion recognition. In addition, we propose an abstract emotion space that avoids the flaws of existing emotion descriptions. We use stimulation to improve the interpretability of our emotion space, along with multi-task learning to improve its robustness. Finally, we investigate auxiliary features for style adaptation in statistical parametric speech synthesis, evaluating both our emotion space and other descriptions of emotion.

The results indicate that our recognition models are state-of-the-art. However, evaluation using speech synthesis shows that our emotion space is no more informative than existing emotion descriptions. Additionally, we investigate a convolutional recognition model using the spectrogram, which outperforms other spectrogram based methods.

Acknowledgements

I would like to thank my supervisor, Professor Simon King, for his excellent guidance and advice. In particular, I am indebted to his diligent teaching, without which I would not have grasped speech as I did with his direction.

I am also greatly thankful to Srikanth, who was always willing to help me with questions about Merlin. Without his help I would not have been prepared for my listening experiment. In addition, the organisers of the Blizzard challenge graciously provided me with material necessary to produce voices based on the Usborne data.

Finally, thank you to my officemates and my friends. Without such gifted and friendly people surrounding me, this year would not have been the same. Thank you to Kaitlyn, who endured my endless talking about my project and always provided me with a positive outlook. And thank you to my family, who I know are always there to support me, I would not be here without you.

This work was supported in part by the EPSRC Centre for Doctoral Training in Data Science, funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1) and the University of Edinburgh.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Zack Hodari*)

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Virtual agents are beginning to pervade our everyday lives, the human-computer interaction (HCI) challenges these systems pose are many. One particular aspect that is not taken into account in these systems is emotion.

Currently commercial text-to-speech (TTS) systems may seem to portray emotion, however this is for the purpose of naturalness, or due to a high quality speaker. Put simply, prosody variation in TTS does not explicitly model emotive information. Likewise, HCI systems such as Apple's Siri or Amazon's Alexa do not take into account the emotional state of users. When interacting with machines we may become impatient or angry, access to this information would allow these systems to adapt in order to resolve the underlying issue. Additionally, recognising the user is joking, or even making jokes through emotive TTS, would make interaction with these systems more natural.

Utilising emotion information, or choosing what emotion to portray are major agent-interaction challenges. We focus on the task of predicting and generating emotive speech. With the vision that this information can be utilised in many HCI systems.

Much of the information conveyed by humans is non-verbal, Mehrabian et al. (1971) present evidence that only 7% of communication is verbal in conversations where emotions are concerned. As such, it is only logical to utilise body language and paralinguistic elements of speech when working with emotion. In this thesis, we make use of speech alone due to the scope of our work, a logical next step would be to incorporate video and text, as discussed in Section 2.3. Nonetheless,

this task presents many challenges, we begin by discussing some issues related to predicting emotion from speech.

1.1 Difficulty of isolating emotion in speech

Emotion is a complex phenomenon expressed through many modalities, including; body language, speech, and discourse. This increases the difficulty of identifying relevant factors within the signals available. For example, it may not be appropriate to factor out the content of a sentence as the words chosen by the speaker are part of the emotion portrayed. For speech in particular, emotion manifests through changes in prosody, this effects speech production through three main aspects; fundamental frequency, speaking rate, and energy (Vinciarelli et al., 2009). These three aspects are well studied and research has shown that certain properties of speech capture the majority of emotional content within speech (Eyben et al., 2016).

1.1.1 Emotional state vs. sentic modulation

Most recognition tasks have a well defined target, for example, the subject of an image is a member of a class. However, this is not the case for emotion recognition; we cannot define emotion in such a simple manner.

> What you reveal to others... is your emotional *expression*. Expression via the motor system, or "sentic modulation" is usually involuntary, and is one clue which others observe to guess your emotional state. (Picard and Picard, 1997, pp.5)

Humans are complex social creatures, their internal emotion - "emotional state" - is influenced by their environment. This is externalised through conscious and unconscious actions; modifying behaviour through changes in prosody, body language, facial expression and communication. These actions are referred to as "sentic modulation". Emotion presented via sentic modulation may not be equivalent to true emotional state, e.g. a person may hide that they are nervous.

Fortunately, this is not relevant to our work; we are concerned with the information available from observing an individual, hence when we refer to the emotion of an individual, we refer to the emotion portrayed through sentic modulation.

1.1.2 Ambiguity of emotion

People convey their emotional state in unique ways according to their personality. This creates an ambiguity in what certain behaviours imply; this variability could be addressed through speaker dependent techniques from automatic speech recognition. Building upon the work of Efron (1941), Ekman et al. (1987) showed that emotion is interpreted consistently across cultures; this suggests that ambiguity within emotional expression will not be an issue. We will pursue a speaker independent approach, however, an investigation of improvements using speaker dependent techniques would be an interesting topic for further research.

Another source of ambiguity is encountered when a person interprets another speaker's emotion. The listener's interpretation is subject to: their own methods of portraying emotion; their current emotional state; and their empathy. In our domain of data-driven learning this issue is particularly noteworthy when considering annotation of data.

1.2 Popular emotion annotation schemes

The task of describing emotion presents many challenges, throughout the literature two distinct methods are used; categorical and dimensional annotation. These both have flaws, but this is unavoidable in designing a practical annotation scheme.

1.2.1 Pure emotions

Categorical labelling emerged from the psychological theory of pure emotions, i.e. a person can only portray one emotion at a time (Plutchik, 1984). This theory is too coarse to describe the nuances of human emotion. For example, it does not take into account the idea of varying intensities of emotion, or mixtures of emotions as presented by (Ekman et al., 1987). It also restricts the description of emotional expression to a limited set of basic emotions - though there is no agreement on what these basic emotions are (Douglas-Cowie et al., 2003).

Regardless of these downsides, the majority of current datasets label each utterance with a single emotion. This is mitigated in two main ways,

- Multiple annotators are used, the consensus of the annotators is taken as the label - this introduces a "no-consensus" label (Busso et al., 2008).
- The single emotion condition is relaxed, this can be further developed to allow annotators to weight the intensity of emotions (Mower et al., 2011).

Despite the availability of multiple annotations per utterance (allowing for the calculation of an average emotion prediction), the majority of publications throughout the literature perform classification on pure emotions. This trivialises the task of emotion recognition, in reality pure emotions are rarely expressed in natural speech (Cowie and Cornelius, 2003). For certain use cases this may be appropriate, such as for interest/disinterest prediction (Kapoor and Picard, 2005). This is not the case for our task - predicting unconstrained emotion.

As mentioned previously, there is no agreed set of basic emotions, individual papers and datasets choose their own emotion sets, making it difficult to present comparable results. This is addressed by more recent datasets which use a larger set of emotions, including; happiness, sadness, anger, disgust, fear, surprise, frustration, excitement, joy, and neutral. This allows for subsets of datasets to be used. In papers, we see a common trend to predict only four basic emotions; happiness, sadness, anger, and neutral (Lee and Tashev, 2015; Kim et al., 2013; Lee et al., 2011). Such a restricted set of emotions greatly simplifies the task of emotion recognition, but does have the benefit of enabling comparability while early research improves at this simplified task.

1.2.2 Appraisal-based emotions

There exist many theoretical and practical downsides to treating emotion as a categorical variable. An alternative description of emotion, adopted by the emotion recognition community, is given by cognitive theory. Lazarus (1991) introduced the idea of an appraisal-based interpretation of emotion, where traits relevant to emotional expression are rated. A common method in emotion recognition uses two real valued dimensions to represent emotion - arousal and valence. However, two dimensions was shown to be insufficient to fully represent emotion by

Fontaine et al. (2007). Fontaine et al. presented evidence for using four dimensions - arousal, valence, dominance, and expectancy. Where arousal is a measure of activeness, valence is a measure of positivity, dominance is a measure of control, and expectancy is a measure of predictability.

While we agree that appraisal-based annotation is more descriptive and accurate than categorical labelling, we also stress a practical flaw of dimensional annotation. To produce a dataset with dimensional labels, the annotators must interpret and follow a set of appraisal instructions. These instructions will be interpreted differently by every annotator. In the IEMOCAP dataset, dimensional annotators have an average coefficient alpha of 0.67 (Busso et al., 2008), this reliability is questionable (Cortina, 1993).

1.3 Learning an abstract emotion space

In order to resolve the issues presented by existing emotion descriptions, we present a method to learn a high dimensional representation of emotion. This emotion space is created with the aim of modelling relevant factors in speech, which generalise beyond the categorical and dimensional descriptions of emotion. One shortcoming of this thesis is our focus on supervised learning, an unsupervised representation would not be reliant on the flawed annotations. However, the challenges associated with unsupervised representation learning mean this task is out of scope for this thesis.

Due to the uninterpretable nature of our emotion space, there is an issue with evaluation. We address this is two ways: in Section 6.2 we use a regularisation technique to improve the interpretability of our representation; and in Section 6.3 we use speech synthesis to evaluate the performance of our representation using an unseen dataset.

1.3.1 Cross-corpora use-case

The abstract nature of our representation should make it suitable for cross-corpora prediction. While the emotion space is trained using one dataset, it is trivial to re-train the representation using new labels from an unseen dataset.

In theory, our emotion space can incorporate new emotion labels into a larger and more complete representation of emotion.

1.4 Contributions

In this thesis we present an emotion recognition architecture that outperforms state-of-the-art speech-only results classifying a set of 4 emotions using the IEMOCAP dataset. We investigate the design of an abstract representation of emotion, making use of stimulation and multi-task learning. Our application of stimulation is for a novel domain. Despite the availability of multiple labelling schemes in popular datasets, we are the first to apply multi-task learning to emotion recognition in an end-to-end architecture. In addition, we investigate the use of other neural network models; our time-distributed CNN is state-of-the-art for spectrogram based emotion recognition.

We use our emotion space as auxiliary features for emotive speech synthesis. In evaluating our representation alongside other emotion representations, we discovered that the eGeMAPS features are much more descriptive than any feature predicted using our state-of-the-art recognition model. This leads us to propose a new model for producing emotive speech using existing techniques in Section 7.1.1.

1.5 Thesis outline

In Chapter 2 we discuss prior work in the fields of, emotion recognition, and emotive speech synthesis. Following this, we present the datasets used in Chapter 3. In Chapters 4 & 5 we describe the methods implemented, and in Chapter 6 we present our experiments and results using these methods. Finally, we conclude what we have achieved in this thesis, and outline ideas for further research in Chapter 7.

Chapter 2

Prior Work

2.1 Feature representations

Throughout machine learning, handcrafted features are used to simplify the task of working with complex data. For example, canny edge detection is used in computer vision (Canny, 1986), and Mel-frequency cepstral coefficients (MFCCs) in automatic speech recognition (ASR) (Davis and Mermelstein, 1980).

Similarly, emotion recognition often makes use of handcrafted features (Eyben et al., 2016), however the best suited feature set for emotion is not agreed upon (El Ayadi et al., 2011). A common method is to use large feature sets; these brute-force features sets attempt to describe all traits of the signal that have some relevance to emotion recognition. In Table 2.1 we outline the number of features used to describe the speech signal for the two most popular emotion recognition challenges; the INTERSPEECH ComParE paralinguistics challenge (Schuller et al., 2016), and the AVEC emotion (and depression) challenge (Valstar et al., 2016). It is clear that there is a trend to use more and more features, in an effort to include enough information to completely describe the emotive content of speech.

Additionally, the emotion recognition in the wild (EmotiW) challenge (Dhall et al., 2016) began in 2013. However, EmotiW does not restrict what features participants may use. A full survey of submissions was not undertaken, so we cannot speculate on the trend of features used for this challenge.

Table 2.1: Size of feature sets used for emotion recognition challenges

Challenge	INTERSPEECH ComParE (Schuller et al., 2016)							
Year	2009	2010	2011	2012	2013	2014	2015	2016
No. features	284	1,582	4,368	6,125	6,373	6,373	6,373	6,373

Challenge	AVEC audio/visual (Valstar et al., 2016)					
Year	2011	2012	2013	2014	2015	2016
No. features	1,941	1,841	2,268	2,268	102	88

Beginning with AVEC 2015, Schuller et al. (2015) opted to use the extended Geneva minimalistic acoustic parameter set, known as eGeMAPS (Eyben et al., 2016), in addition to several extra features. This minimalistic set of 88 perceptually, empirically, and theoretically motivated features aims to capture all relevant information from the speech signal. We explain eGeMAPS in more detail in Section 4.1.4. Eyben et al., evaluate the performance of eGeMAPS against the brute-force feature sets used by the INTERSPEECH ComParE challenge, eGeMAPS performs competitively with the much larger feature sets, despite having 1.4% of the size of the best performing feature set (ComParE 2013-2016).

Improvements in machine learning techniques have shifted the focus of many domains, from pre-processing, towards more complex models in combination with unprocessed data. In computer vision, state-of-the-art methods operate on raw images using convolutional neural networks (CNNs) (Krizhevsky et al., 2012). In ASR, recent work has shown improved performance using the raw waveform and CNNs (Palaz et al., 2013; Hoshen et al., 2015). The shift away from pre-processing is also seen in emotion recognition; Trigeorgis et al. (2016) successfully demonstrated emotion recognition using the raw waveform. Trigeoris et al. influenced further work using the spectrogram (Ghosh et al., 2016).

While the original signal is superior to extracted features (in terms of information content), working with the waveform is challenging. As mentioned above, an alternative to the waveform is the spectrogram - a frequency-domain over time representation of the waveform. The spectrogram explicitly represents frequencies present in speech, this is important as humans perceive speech in the frequency domain.

The use of spectrograms for emotion recognition has been explored more than raw waveforms. Ghosh et al. (2015) investigate the representations learnt from spectrograms and their performance for classification. Whereas Mao et al. (2014) focus more on techniques to disentangle salient features using lower-level features learnt from the spectrogram, they utilise a sparse autoencoder to pre-train a CNN.

2.2 Datasets

Initially, research in emotion recognition made use of data collected specifically for each paper (Lee et al., 2004), this put limitations on the quality and size of datasets. The major limitation of these datasets is their use of actors performing emotions as opposed to recording natural interactions. This means earlier research did not account for more subtle articulations of emotion produced in natural speech. Cowie et al. (2001) discuss this issue in their seminal paper, stating that a necessary direction for research is to use more natural data.

Douglas-Cowie et al. (2003) progressed in this direction, reviewing the current state of available datasets and producing new naturalistic datasets (Douglas-Cowie et al., 2000, 2007). Along with these datasets, focus also shifted from using pure emotions, theorised by Ekman (1992), to using mixtures of emotions (Mower et al., 2011), or dimensional emotions (Fontaine et al., 2007).

These improvements fostered incremental progress in the quality of datasets. Several notable datasets that provide high quality spontaneous speech are; RECOLA (Ringeval et al., 2013), SEMAINE (McKeown et al., 2012), eNTERFACE (Martin et al., 2006), and IEMOCAP (Busso et al., 2008) - we discuss the latter in Chapter 3.

2.3 Recognition methods

Initially, much of the research into emotion recognition focussed on the use of hidden Markov models (HMMs), Gaussian mixture models (GMMs), and support vector machines (SVMs). These were popular due to their use in ASR (Gales

and Young, 2008; Solera-Ureña et al., 2007), along with the significant number of software packages available, such as the HMM toolkit (Young and Young, 1993).

A lot of research has focussed on input features; Lee et al. (2004) investigated the use of phoneme classes modelled using HMMs, while Neiberg et al. (2006) took influence from the feature representations of ASR and investigated the use of spectral features such as MFCCs. Metallinou et al. (2010) used audiovisual data to compare the use of GMMs and HMMs to model a variety of features. More recent work using SVMs makes use of ensemble models to boost performance of multi-modal feature sets (Rozgic et al., 2012).

While SVMs are still competitive with state-of-the-art techniques, the availability of computing resources and open source machine learning software, cultivated interest in new models, namely neural networks. Early applications of neural networks for emotion recognition include Wöllmer et al. (2010) and Kim et al. (2013). Wollmer et al., investigate prediction of arousal and valence using bidirectional long short-term memory (BLSTM) networks, comparing performance to typical HMM and SVM classification techniques. While Kim et al., compare traditional feature selection techniques with unsupervised representations learnt with deep belief networks (DBN).

Along with the shift towards more flexible machine learning techniques, much work has been done to combine modalities. Poria et al. (2017) details the progress in multi-modal analysis and demonstrates that improvements can be made using multi-modal models. This is clearly demonstrated by Metallinou et al. (2008), Rozgic et al. (2012), and Poria et al. (2016), who all present ensemble models and detail performance with/without certain modalities. In all cases using more modalities improved the performance; text often provided less improvement, while audio consistently increased performance the most. In Section 6.1.3 we provide a more detailed discussion of comparable results for uni-modal and multi-modal approaches.

Despite the availability of multiple labelling schemes in existing databases, there are few applications of multi-task learning (MTL) (Caruana, 1998). Using feature representations from a DBN, Xia and Liu (2015) investigate the importance of the second task in MTL using SVM classification on the learned features. To our knowledge there is no end-to-end demonstration of MTL in the literature.

2.4 Emotive Speech Synthesis

With current commercial systems utilising unit-selection synthesis (or hybrid synthesis), emotions generated are limited to the content of the database; the unit with that emotion must already exist. This is discussed by the creator of unit-selection (Black, 2003). Black suggests various methods to record additional material for the purpose of emotive synthesis using a unit-selection database.

This technique is limited due to the expensive nature of collecting more data, as well as practical limitations of increasing the size of the database. Other methods are reviewed by Schröder (2001), including formant synthesis and diphone synthesis. In formant synthesis signal-processing is used to create synthetic speech from scratch, modifications can be added during signal-processing. In diphone synthesis; F0, duration, and intensity of diphones can be modified to an extent using signal-processing. Diphone synthesis is the predecessor to unit-selection; a diphone is the two adjoining halves of consecutive phones.

Producing emotion using these systems is challenging, whereas, the more modern technique of statistical parametric speech synthesis (SPSS) is more capable of adapting the production of speech due to its parametric acoustic representation. Yamagishi et al. (2004) were the first to demonstrate expressive speech adaptation using HMM synthesis. A later review by Schroder discusses several techniques for emotive speech synthesis using SPSS (Schröder, 2009). Barra-Chicote et al. (2010) performed a comprehensive analysis on the use of unit-selection and HMM synthesis for the purpose of neutral and emotional speech synthesis.

Speaker adaptation is a much broader field of research that investigates methods to adapt speech recognition models to individual speakers. As demonstrated by Wu et al. (2015), these ASR techniques can be applied for speaker adaptation across genders in speech synthesis. This is possible as speech synthesis makes use of an acoustic model that performs the reverse task to the acoustic model in ASR. These techniques can be applied for style adaptation, allowing for explicit modelling of different emotional styles.

Adaptation techniques from ASR fall into three main areas; feature-space transformation, auxiliary features, and model-based adaptation. Feature-space

transformation aims to perform speaker normalisation on the acoustic parameters. The linear input network (LIN) is a popular method for use with neural network acoustic models (Neto et al., 1995). LIN adds a speaker dependent layer after the input features, this transforms the inputs into a normalised speaker independent space for use by the acoustic model.

Auxiliary feature methods define features for use as additional inputs to the acoustic model; the most notable auxiliary feature used in ASR are i-vectors (Dehak et al., 2011). i-vectors are designed to be a basis over speaker-variability, this includes both speaker and channel variability. Using these additional features allows the acoustic model to explicitly factor out variability unique to individual speakers.

Model adaptation techniques perform adjustments to the model behaviour to explicitly model different speakers. Swietojanski and Renals (2014) presented a method for learning hidden unit contributions (LHUC). Unlike the other methods described, LHUC learns adaptation parameters for each new speaker using a small amount of training data from the speaker. Similar to i-vectors, LHUC provides a method for the acoustic model to factor out speaker variability.

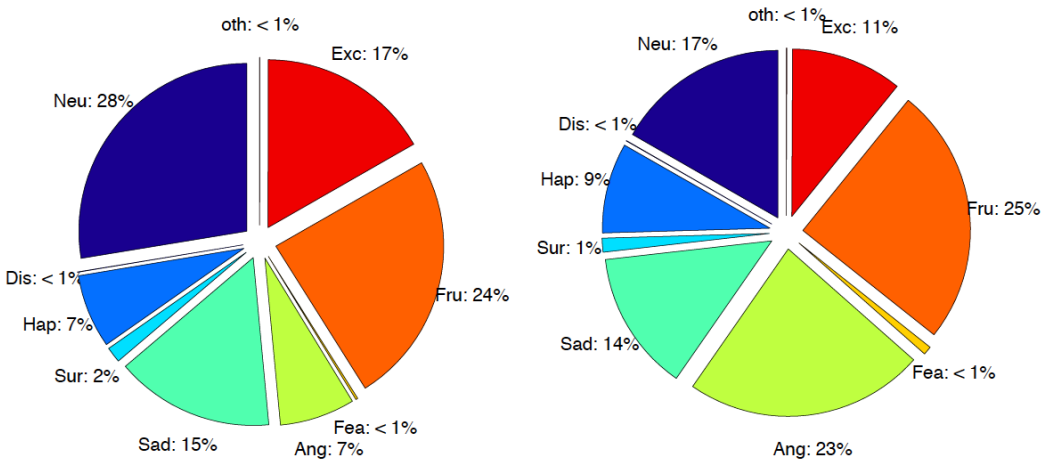
Chapter 3

Datasets

3.1 IEMOCAP

To perform our experiments on emotion recognition and representation learning, we use the interactive emotional dyadic motion capture database (IEMOCAP) (Busso et al., 2008). This dataset was created using ten actors (5 male and 5 female) split into five mixed-gender pairs, each pair is recorded for two sessions roughly one hour long, the dataset contains ten sessions and approximately 12 hours of data. Each session consists of scripted and spontaneous (improvised) conversations, with an average of 7 scripted and 8 spontaneous conversations per session. Design of the recording material focussed around production of 5 categorical emotions; anger, sadness, happiness, frustration, and neutral. Speech recordings and text transcriptions are available for both actors in all sessions. Additionally, there is motion capture data of facial expressions for one actor in each of the ten sessions, though in this thesis we focus on speech data alone.

IEMOCAP was labelled by 6 university students using the ANVIL annotation tool (Kipp, 2001), they were instructed to take into account the surrounding context when choosing an appropriate emotional label. Each utterance was assessed by 3 annotators, who were instructed to label utterances as angry, sad, happy, disgusted, fearful or surprised. If they thought one category was insufficient they were able to select multiple, or add their own additional emotion labels. After completing annotation, the authors added frustration, excitement, and "other" as categories, giving a total of 10 categorical emotions. The annota-



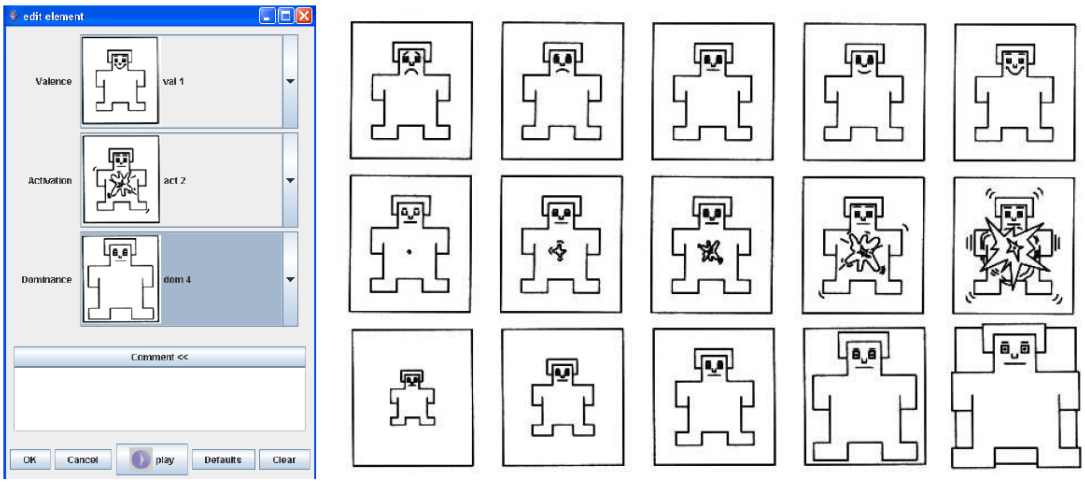
(a) Scripted session (b) Spontaneous sessions

Figure 3.1: Distribution of categorical emotion labels for (a) scripted and (b) spontaneous conversations from the IEMOCAP dataset. Figure credit, (Busso et al., 2008)

tors reached agreement in 74.6% of the utterances. Coverage across the categories used when recording the data is good (anger, sadness, happiness, frustration, and neutral), as seen in Figure 3.1.

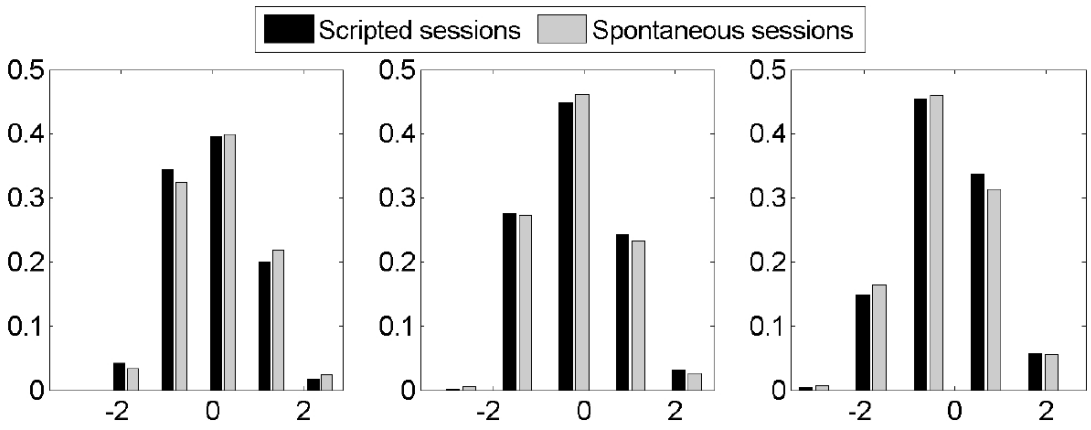
The annotators also performed appraisal-based annotation, along the arousal (activation), valence and dominance dimensions. The user-interface included *self-assessment manikins* (SAMs) (Figure 3.2) to help annotators accurately choose the correct value. The continuous axes are replaced with discrete measurements, as each manikin represents an integer value from 1 to 5. The distribution of dimensional annotations is shown in Figure 3.3. As discussed in Section 1.2.2, due to the nature of appraisal-based annotation, inter-annotator agreement might become an issue. The use of SAMs and discrete labels improve the reliability of the task greatly, the authors demonstrate this by calculating coefficient alpha between annotators. A coefficient alpha of 0.67 suggests the labels are reliable, but there are more factors to consider when interpreting this statistic. In particular, the number of items being compared can lead to low inter-correlation, but high coefficient alpha (Cortina, 1993), this was not discussed by Busso et al. (2008).

For the purpose of training machine learning models, discussed in Chapter 4, we perform 5-fold cross-validation. Each fold uses 4 pairs of speakers for the training data (i.e. 8 sessions), the validation and testing data each use one speaker from the remaining two speakers across their 2 sessions.



(a) (b)

Figure 3.2: Interface used for IEMOCAP dimensional emotion annotation along with pictoral descriptions of annotations. (a) ANVIL dimensional annotation UI. (b) Self-assessment manikins. Top row: valence, middle row: arousal (activation), bottom row: dominance. Figure credit, (Busso et al., 2008)



(a) Valence (b) Activation (c) Dominance

Figure 3.3: Distribution of dimensional emotion annotations for scripted and spontaneous conversations from the IEMOCAP dataset. Figure credit, (Busso et al., 2008)

3.2 Usborne children's audiobook dataset

In Chapter 5 we investigate emotive speech synthesis techniques. For this task we make use of Blizzard 2017 challenge[1] data, this is very similar to the Blizzard 2016 data (King and Karaiskos, 2016), but with several additional stories. This provides us with 6.5 hours of English audiobook data from a single female speaker. The recordings are collected in a studio by a professional voice actor, meaning the speech quality is very consistent. The data includes stories such as; Hansel and Gretel, Little Red Riding Hood, Macbeth, Robin Hood, Snow White, The Gingerbread Man, and The Ugly Duckling.

When training our SPSS models using the audiobook data, we follow the training, validation, test data split given with the 2017 Blizzard data. This reserves 3 stories for the test set, enabling evaluation on unseen data; a fraction of each remaining story (between 1 and 12 utterances per story) is used for the validation set, granting the validation set full coverage over the training material; and the training set uses the remaining utterances from non-test-set stories.

[1] https://www.synsig.org/index.php/Blizzard_Challenge

Chapter 4

Emotion Recognition

4.1 Input Features

To perform emotion recognition we train machine learning models, this requires data to learn from. The content and structure of a data point influences the effectiveness of the learning process, we discuss the relative benefits and drawbacks to different representations of speech.

4.1.1 Raw Waveform

Speech is encoded as a digital interpretation of an analogue signal, it is a time-domain signal with limitations caused by sampling rate and bit-depth. Using the raw waveform for recognition may be short-sighted as humans perceive sound as a combination of frequencies. Thus we require our model to learn long-range patterns, a task that machine learning models are notoriously ineffective at.

4.1.2 Spectrogram

The spectrogram is an alternate representation of the waveform, it is created using the Fourier transform on overlapping windows of the waveform - the Fourier transform gives a frequency-domain representation of the time-domain window. An example is shown in Figure 4.1, each column is the response of a Fourier

transform on one window, and each row shows how the a particular frequency of the signal varies with time.

While it is not possible to translate a spectrogram into text by eye, important details can be extracted through visual inspection of a spectrogram. In Figure 4.1 the vertically stacked horizontal lines are the harmonics, these are produced by the vocal cords. The lowest harmonic is the fundamental frequency (F_0), this is the lowest contributing frequency of the vocal folds when producing a voiced sound. F_0 is influenced by the length, size, and tension of the vocal folds. The harmonics and F_0 are disjointed across time due to pauses in speech and unvoiced sounds (such as "s" or "sh"), which do not involve vibration of the vocal folds. The shape of the vocal tract influences the sounds produced, these changes are shown in the formants. The formants are only usually visible in wideband spectrogram, however in our example it is possible to see part of a formant.

The tall boxes in our example contain sounds made by the speaker, such as a phone, syllable or short word. In one of these boxes we observe co-articulation, a phenomena where two consecutive sounds influence each other. This is an example of how context makes speech synthesis so difficult, depending on the surrounding sounds and the prosody of the speech, the production of one sound can be changed dramatically. By using the spectrogram, our model may be able to learn to generalise contextual effects such as this.

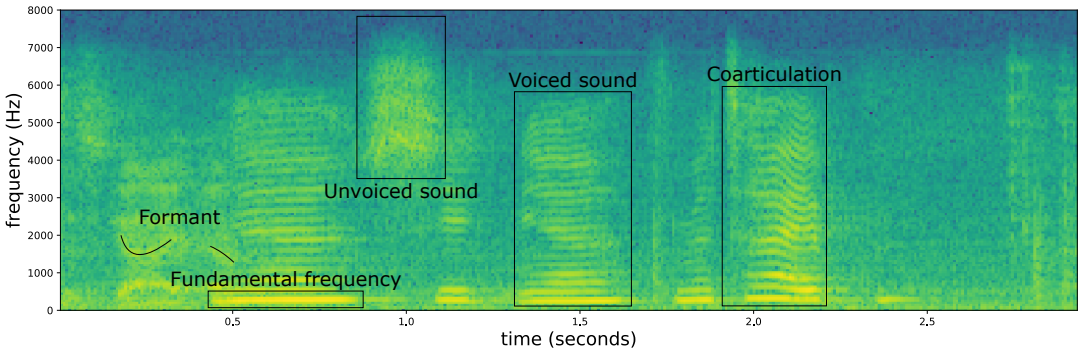


Figure 4.1: Narrowband spectrogram created from an utterance in the IEMOCAP dataset. For the utterance: "Yeah, she-yeah, that's my type."

4.1.3 Engineered features

There exist many relevant features derived from the waveform and spectrogram, possibly the most notable feature is the fundamental frequency, F_0; this represents the frequency of the source, i.e. vibration of the speaker's vocal cords.

A more complex feature, designed for automatic speech recognition using hidden Markov models (HMMs), are Mel-frequency cepstral coefficients (MFCCs) (Davis and Mermelstein, 1980), these are decorrelated Mel-filter bank coefficients (MFBs). MFBs are derived from the spectrogram using triangular filters scaled according to human auditory perception (using the Mel-scale); each filter bank represents the contribution of a band of frequencies perceived to be similar by humans. The discrete cosine transform (DCT) is used to convert MFBs into the time-domain, producing MFCCs. The DCT serves to decorrelate the features, a necessary step for HMMs which model un-correlated features using a diagonal covariance. For this reason, more recent research using neural networks has made use of MFBs which are a better representation of speech than MFCCs (Mohamed, 2014).

For other modalities there exist many different features, for example in the IEMOCAP dataset used in this thesis the video data is provided with facial animation parameters (FAPs). These features are created using motion capture markers placed on the actors faces, in post-processing these are converted into 3-dimensional co-ordinates over time. FAPs are a useful source of data as they allow for direct modelling of the actors facial movements, as opposed to using computer vision approaches to learn from raw video. This is similar to how MFCCs and MFBs aim to provide the most relevant information from the raw waveform.

4.1.4 GeMAPS

As discussed in Section 2.1, it is common to use handcrafted feature sets, this is due to their lower-dimensionality compared to the raw data. We make use of the extended Geneva minimalistic acoustic parameter set (eGeMAPS) (Eyben et al., 2016), this feature set aims to improve upon the preceding brute-force feature sets by only including relevant features. The choice of features is motivated by: their ability to model perceptual changes in voice production; their proven performance in empirical studies; and their theoretical significance.

Similarly to the brute-force feature sets, eGeMAPS makes use of low-level descriptors (LLDs). An LLD is a feature calculated on a window (frame) of speech; the energy, spectral and formant features use 20ms frames, while the remaining voicing related features use 60ms frames. The LLDs used by eGeMAPS are detailed in Table 4.1, various statistical functionals are applied to the LLDs to give a total of 88 utterance-level parameters.

Many of these features encode trivial concepts of speech, such as loudness. However, in practice it can be very difficult to extract these automatically. The solution to extracting loudness may appear to involve averaging over the waveform, however, the human auditory system does not perceive speech in this way. The method used by eGeMAPS begins by extracting MFBs using 26 triangular

Table 4.1: eGeMAPS low-level descriptor features

1 energy related LLD	**Group**
Loudness (signal intensity)	Prosodic

25 spectral LLD	**Group**
α ratio - 50-1000 Hz & 1000-1500 Hz	Spectral
Spectral slope - 0-500 Hz & 500-1500 Hz	Spectral
Hammarberg index	Spectral
MFCC 1-4	Cepstral
Spectral flux	Spectral

16 voicing related LLD	**Group**
Log F_0 on a semi-tone scale	Prosodic
Formant 1-3 frequency	Voice quality
Formant 1-3 bandwidth	Voice quality
Formant 1-3 amplitude	Voice quality
Harmonic difference - H1-H2 & H1-A3	Voice quality
Harmonics-to-noise ratio	Voice quality
Jitter of consecutive F_0 periods	Voice quality
Shimmer of consecutive F_0 periods	Voice quality

Mel-scale filters. Following this, each filter bank is weighted and scaled using; an equal loudness curve, and cubic root amplitude compression (Hermansky, 1990), to create the auditory spectrum. Finally the loudness LLD is extracted by summing over all bands of the auditory spectrum.

Formants are another feature of speech that are difficult to reliably extract from a waveform, for this reason many feature sets do not make use of them (Eyben et al., 2016). A common method is to make use of linear prediction (LP), a technique for source-filter separation (Makhoul, 1975). LP assumes that speech is fully described by a source (vocal cords vibration) and a filter (vocal tract shape). The algorithm calculates the coefficients representing the filter by solving a linear system - which can be created using a number of methods. These coefficients form the LP spectrum, the peaks in this spectrum are the formants. Therefore, the formants can be calculated by solving for the roots of the LP spectrum.

By designed a open-source minimalistic parameter set, Eyben et al. (2016) have avoided implementation specific issues relating to comparability. eGeMAPS also allows researchers to make use of difficult to extract or complex features that are important for recognising emotion.

4.2 Neural Networks

Machine learning is a well established field, recently it has seen a dramatic rise in popularity with the widespread use of neural networks (NN). Following the large success of backpropagation for training NNs, they have been applied to many areas of scientific interest. A NN is a complex function approximator, given high-dimensional inputs, it is able to learn a non-linear transformation to predict some target.

We focus on supervised learning, where we provide the correct answer for each example during the training stage, this allows the model to improve its performance using feedback. There is an adjacent field of work called unsupervised learning, where the model must learn what aspects of the signal are important without feedback on training examples.

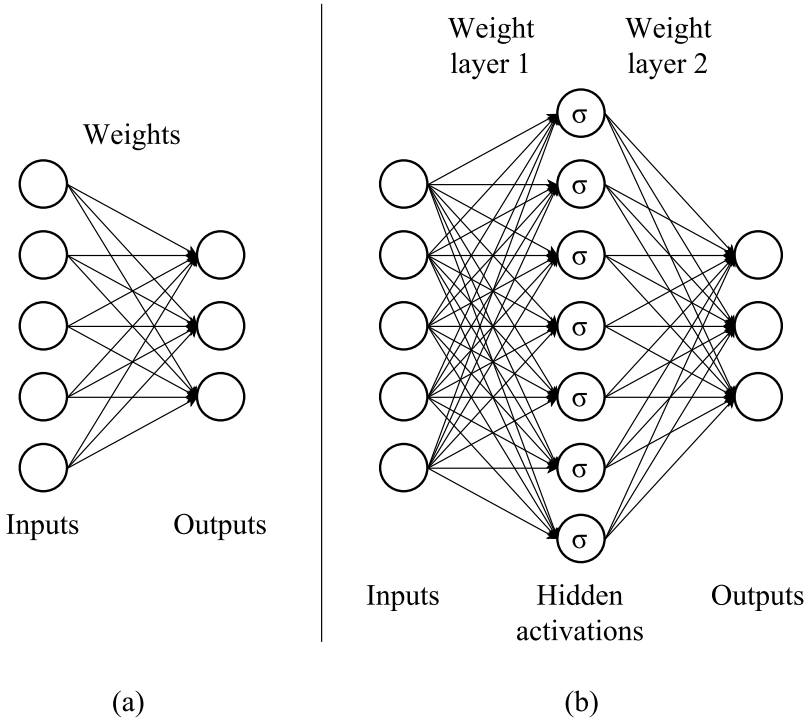


Figure 4.2: Simple machine learning models. (a) Linear regression. (b) Feed-forward neural network

Feed-forward NNs are the most basic model, at their simplest they are stacks of linear regression models with non-linear transformations between each layer, this is demonstrated in Figure 4.2.b for a 1 hidden layer feed-forward NN.

4.2.1 Recurrent Networks

Unlike classical feed-forward NN models, recurrent neural networks (RNNs) incorporate recurrency into each unit (cell) (Jordan, 1997). RNNs operate on temporal data; instead of a flat vector they use a list of vectors that are distributed over time. This means RNNs can explicitly learn temporal characteristics within the data using a cell's recurrent connection.

Computation of RNNs require them to be "un-rolled", this process is illustrated in Figure 4.3. All the weights are shared across time steps, however, each time step must be computed sequentially. This computation cannot be easily distributed using GPUs, therefore RNNs are generally much slower.

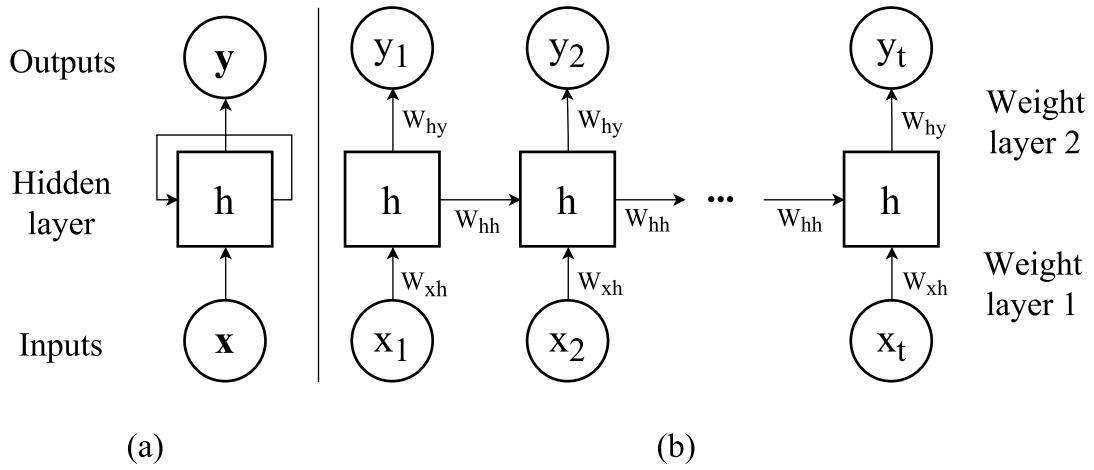


Figure 4.3: (a) Basic RNN model. (b) Un-rolled RNN model

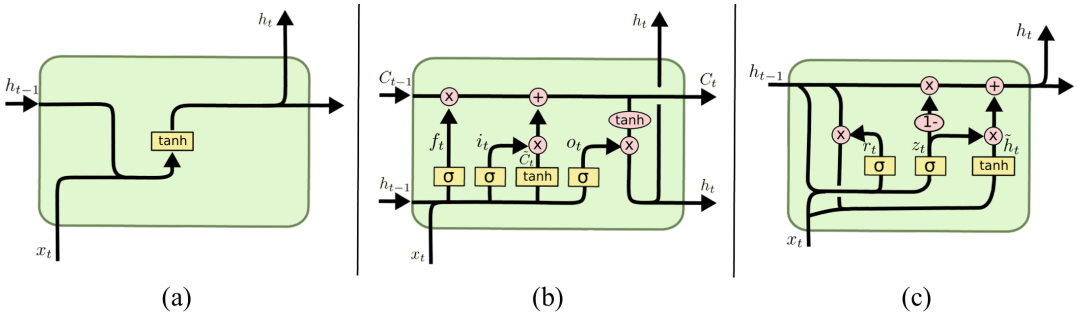


Figure 4.4: RNN cell diagrams. (a) Basic RNN cell. (b) LSTM cell. (c) GRU cell. Figure credit, (Olah, 2015)

A basic RNN cell, shown in Figure 4.4.a, has a single output h_t. The previous output h_{t-1} is combined with the input x_t and recursively passed to the next time step, as outlined in Equation 4.1. This simple architecture gives the model the ability to use previous information when calculating future outputs.

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t) \tag{4.1}$$

This basic cell encounters issues with backpropagation; for a large number of time steps the gradients become numerically unstable, this is known as vanishing/exploding gradients (Bengio et al., 1994). To solve this issue and improve learning of long-term dependencies, Hochreiter and Schmidhuber (1997) proposed the long short-term memory (LSTM) cell. The LSTM cell has a cell state, which allows the cell to remember previous information and incorporate it in any future output through different gating mechanisms. The cell state is not transformed with an activation between time steps, this avoids the vanishing gradient issue.

The LSTM cell is depicted in Figure 4.4.b, the three $\otimes$ nodes represent the gates; f_t controls the forget gate, i_t controls the input gate, and o_t controls the output gate. The gates allow the cell to choose what portion of the previous cell state C_{t-1} to forget, how much of the new proposed cell state $\tilde{C}_t$ to include, and how much of the new cell state C_t to add to the output h_t. The precise operations of the LSTM cell are as follows,

$$\begin{aligned}
\text{forget gate} \quad & f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
\text{input gate} \quad & i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
\text{output gate} \quad & o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
\text{proposed state} \quad & \tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
\text{new state} \quad & C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \\
\text{new output} \quad & h_t = o_t * \tanh(C_t)
\end{aligned} \tag{4.2}$$

Many variants of the LSTM cell have been proposed, however the gated recurrent unit (GRU) (Cho et al., 2014) is the most notable variant. GRUs simplify the LSTM design by combining the forget and input gates into a single update gate, controlled by z_t. It uses a reset gate r_t, to control how much of the previous output h_{t-1} contributes to the new proposed output $\tilde{h}_t$. Additionally, it merges the cell state and output, these changes can be seen in Figure 4.4.c. The details of the GRU operations are as follows,

$$\begin{aligned}
\text{update gate} \quad & z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \\
\text{reset gate} \quad & r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \\
\text{proposed output} \quad & \tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t]) \\
\text{new output} \quad & h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t
\end{aligned} \tag{4.3}$$

4.2.2 Convolutional Networks

In structured inputs such as a spectrogram or waveform, the local context is not explicitly modelled by feed-forward networks. We can take advantage of this contextual information using the concept of convolutions from image processing. Convolution is the process of sliding a window across the input and computing the sum of products with the window and a provided kernel matrix.

Convolutional neural networks (CNNs) (LeCun et al., 1998) perform convolutions on the input just as in image processing, however, it is not necessary to specify the kernels. As a machine learning technique, the model learns the kernels during training by minimising some objective function. CNNs can learn patterns and recognise them at any location within the image. They use multiple channels of kernels; for N input channels and M output channels, the CNN will have $N \times M$ kernels. An output channel is referred to as a feature map, as it contains a map of the responses of the kernels across the input space, this is illustrated in Figure 4.5. It is common for a CNN to finish using several feed-forward layers, however the necessity of this is questioned by Springenberg et al. (2014).

CNNs have a large number of hyperparameters, most notable are the kernel sizes and number of channels. We can use the spectrogram to visually predict a reasonable kernel size, attempting to cover a portion of the harmonics as demonstrated in Figure 4.5. Choosing the number of channels is more difficult, generally this must be done using a grid search. The same is true for the number of layers in the CNN model.

It is common for CNN models to use maxpooling layers. Maxpooling is a convolution operation, where the kernel is the max function. This is normally used to reduce the size of feature maps, however recent work has suggested that using maxpooling is unnecessary (Springenberg et al., 2014); using strided convolution to reduce the dimensionality allows the model to learn the maxpooling operation. Using maxpooling, or choosing the strides adds yet more choices when designing a convolutional architecture.

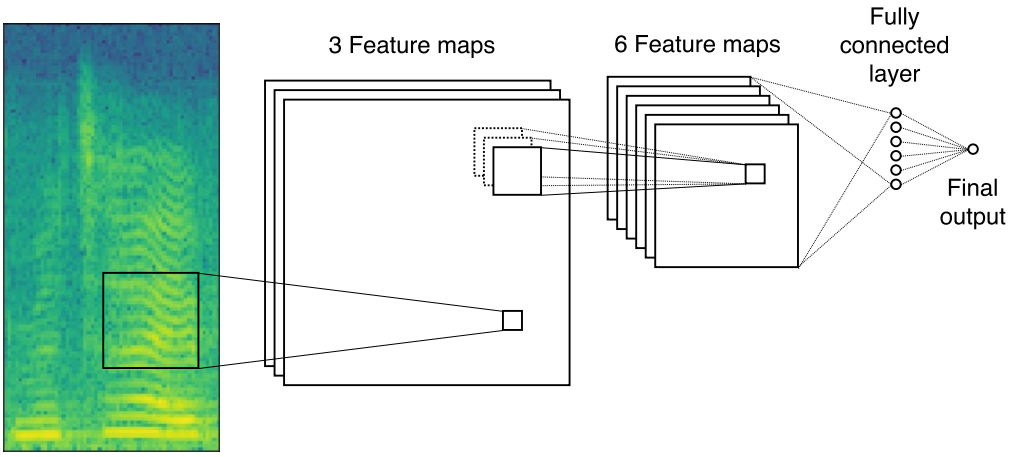


Figure 4.5: Convolutional neural network architecture

4.2.3 Time-Distributed CNN

One limitation of using CNNs is that all inputs must have the same dimensionality. This is an issue as spectrograms have a variable width - since they are created from a variable length waveform. An elegant solution for video is presented by Donahue et al. (2015); a CNN is applied to consecutive frames of video and each CNN response is fed into an LSTM, from which predictions can be made.

To apply this technique to speech we must vertical slices from the spectrogram, these can be used as inputs for our time-distributed CNN (TD-CNN). Figure 4.6 shows the structure of this model, note that as in Figure 4.3.b the TD-CNN is "un-rolled"; there is only one CNN and one LSTM, a slice of the spectrogram is considered to be one time-step. A similar architecture has been used for feature learning in emotion recognition (Mao et al., 2014), however our investigation focusses on end-to-end recognition, as opposed to feature learning.

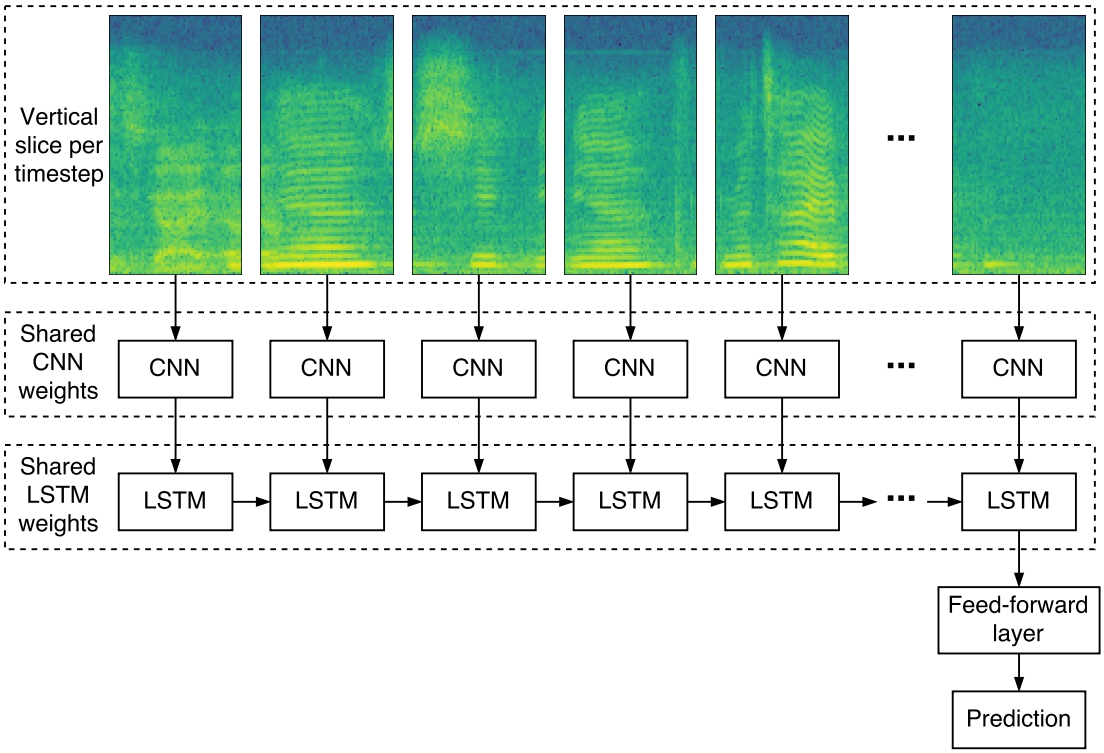


Figure 4.6: Time-distributed CNN architecture, operating on vertical slices of a spectrogram.

4.3 Regularisation

A common method to improve the generalisation capability of neural networks is to use regularisation techniques. Regularisation discourages unnecessary complexity, it does so by penalising the model for focussing on details of the training data not representative of the underlying problem being addressed.

4.3.1 Dropout

Dropout is arguably the most common form of regularisation, dropout randomly excludes individual features from the input (Srivastava et al., 2014). By removing features, dropout puts a penalty on fitting meaningless patterns in the data, thus forcing the model to learn more robust features. These robust features are much less susceptible to memoising the training data, which reduces the likelihood of overfitting.

4.3.2 Multi-task learning

Multi-task learning (MTL) makes use of multiple objective functions based on different attributes that can be predicted from the data (Caruana, 1998). When trained end-to-end, multiple channels of feedback can provide improved generalisation. This requires the tasks to be related, for example, learning speaker identity as a secondary task for automatic speech recognition (ASR) can enable the model to learn to factor out speaker identity when transcribing speech (Chen et al., 2015).

We propose a simple architecture using private hidden layers, this is outlined in Figure 4.7.c. This architecture provides us with a straightforward method for learning the abstract emotion space, the last shared layer of the model is the learnt representation.

4.3.3 Multi-modal learning

As presented in Figure 4.7.b, incorporating multiple modalities into a model requires only simple modification. Due to the scope of this thesis, we did not inves-

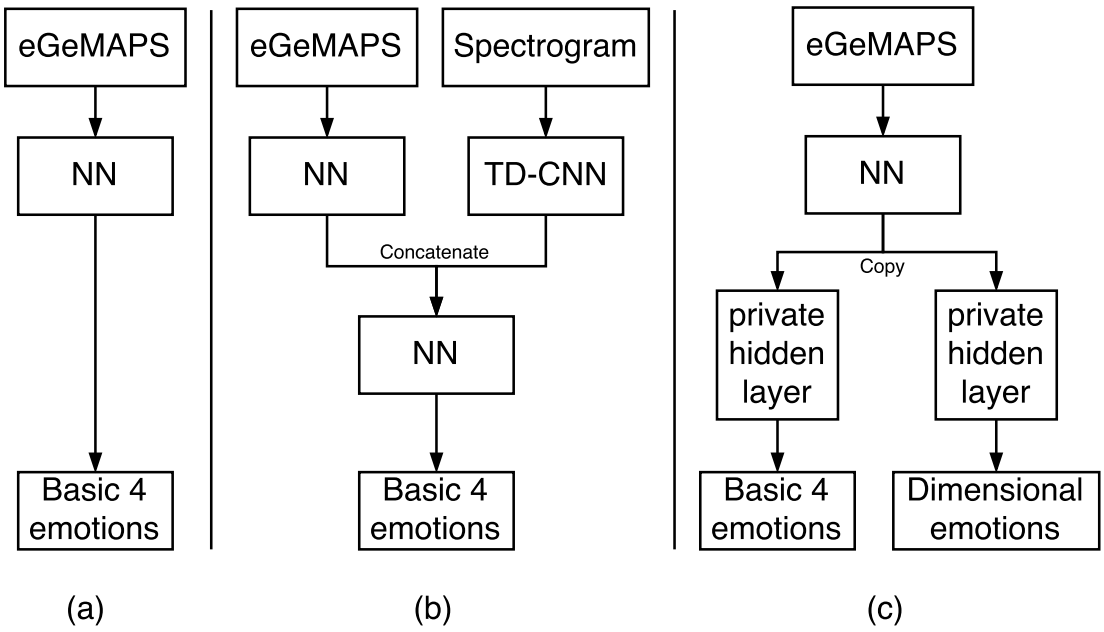


Figure 4.7: Machine learning architectures. (a) Basic feed-forward architecture. (b) Multi-modal architecture. (c) Multi-task learning architecture.

tigate this method, however, as discussed in Section 2.3 using multiple modalities is an important step to improving performance. This is further demonstrated in Section 6.1.3 where we present the performance of comparable models from the literature, clearly showing that multi-modal approaches are more accurate.

4.3.4 Stimulation

Tan et al. (2015) presented a novel method, called stimulation, for improving the interpretability and adaptability of neural networks, these are qualities we would like our abstract representation to have. An interpretable representation builds trust in the method by providing some explanation of its operation. An adaptable model is beneficial for the cross-corpora use-case discussed in Section 1.3.1. Stimulation has been demonstrated to provide performance gains for ASR (Wu et al., 2016a), it can be easily modified for emotion recognition, as outlined below.

Through the use of a prior distribution, stimulation imposes a penalty on hidden layers of a neural network. This penalty aims to improve the utility of the model - i.e. its interpretability and adaptability. The penalty is implemented as a regularisation term, encouraging the stimulated hidden layers to have high activations in certain locations according to the given prior map.

The prior is a function of the emotion category e_t for the current input x_t, each emotion is given a location in a unit square $\boldsymbol{s}_{e_t}$ by the prior map. Similarly, each unit i in the layer being stimulated is mapped to a location in a unit square $\boldsymbol{s}_i$. The prior $g(\boldsymbol{s}_i, \boldsymbol{s}_{e_t})$ is a normalised Gaussian kernel defined by the values of $\boldsymbol{s}_{e_t}$ and parametrised by σ_{st}, a hyperparameter which controls the sharpness of the prior.

$$\begin{aligned} \tilde{g}(\boldsymbol{s}_i, \boldsymbol{s}_{e_t}) &= \exp\left(-\frac{1}{2\sigma_{st}^2} \|\boldsymbol{s}_i - \boldsymbol{s}_{e_t}\|_2^2\right) \\ g(\boldsymbol{s}_i, \boldsymbol{s}_{e_t}) &= \frac{\tilde{g}(\boldsymbol{s}_i, \boldsymbol{s}_{e_t})}{\sum_j \tilde{g}(\boldsymbol{s}_j, \boldsymbol{s}_{e_t})} \end{aligned} \tag{4.4}$$

An approximating distribution $\bar{h}_i^{(l)}$ of the normalised hidden activation output is calculated using the learnable parameters of the model, i.e. the hidden activations $h_i^{(l)}$ of that layer and the weights of the following layer, $w_{i,k}^{(l+1)}$. The hidden activations are weighted by their impact on the following layer using $\beta_i^{(l)}$.

$$\begin{aligned} \beta_i^{(l)} &= \sqrt{\sum_k w_{i,k}^{(l+1)^2}} \\ \tilde{h}_i^{(l)} &= h_i^{(l)} \beta_i^{(l)} \\ \bar{h}_i^{(l)} &= \frac{\tilde{h}_i^{(l)}}{\sum_j \tilde{h}_j^{(l)}} \end{aligned} \tag{4.5}$$

Finally, the regularisation term $\mathcal{R}_{st}$ is calculated using the KL-divergence of the prior distribution $g(\boldsymbol{s}_i, \boldsymbol{s}_{e_t})$ and the approximating distribution $\bar{h}_i^{(l)}$. $\bar{h}_i^{(l)}$ is a function of the current input x_t and the outgoing weight matrices $\boldsymbol{W}^{(l+1)}$ of the stimulated layers. The KL-divergence is calculated for all units i in all stimulated layers l. This is weighted by the hyperparameter η_{st}, which controls the contribution of the stimulation penalty $\mathcal{R}_{st}$ to the loss function.

$$\begin{aligned} \boldsymbol{\theta} &= \left\{\boldsymbol{W}^{(l+1)}\right\}_{l \in \text{stimulated layers}} \\ \mathcal{R}_{st}(x_t; \boldsymbol{\theta}) &= \eta_{st} \sum_l \sum_i g(\boldsymbol{s}_i, \boldsymbol{s}_{e_t}) \log\left(\frac{g(\boldsymbol{s}_i, \boldsymbol{s}_{e_t})}{\bar{h}_i^{(l)}(x_t; \boldsymbol{\theta})}\right) \end{aligned} \tag{4.6}$$

Minimising $\mathcal{R}_{st}$ encourages the activations, for input x_t with emotion e_t, to be high surrounding the point $\boldsymbol{s}_{e_t}$. This means that we can visualise the activations as a grid, by inspection we can see what emotion the layer expects the input to contain.

4.4 Implementation

The implementation of the methods covered in this chapter made use of several excellent open-source projects. Except for feature extraction, all the code for emotion recognition was written in Python (Van Rossum and Drake, 2003) with the use of NumPy (Walt et al., 2011), Matplotlib (Hunter, 2007), TensorFlow (Abadi et al., 2016), Jupyter notebooks (Kluyver et al., 2016), SciPy (Jones et al., 2014), and Pandas (McKinney, 2011).

Feature extraction of the eGeMAPS was performed with the command line tool openSMILE (Eyben et al., 2010), this open-source project has helped the emotion recognition community greatly, without it there was no standardised method for feature extraction, meaning researchers may have used varying implementations of feature extraction algorithms.

4.4.1 Machine learning modular architecture

In order to test the architectures that we have described, we designed a modular NN software package. We define a model by its inputs, NN modules, outputs, and computation graph. The inputs, modules, and outputs inherit from abstract handler classes. We use the delegation design pattern for our generic *Model* class. *Model* takes *InputHandler*, *ModuleHandler*, and *OutputHandler* instances as input and uses them to delegate operations and information, as outlined in Figure 4.8. At initialisation, an adjacency list must also be provided, this defines the computation graph, and the *Model* instance uses this to build the TensorFlow model, implemented by individual *ModuleHandler* subclasses.

We can easily train, evaluate, save, and restore the model by interfacing with the *Model* class's functions. Using the abstract handlers we can define new input and output types easily, creating a new module such as a TD-CNN requires the implementation of one function, *build_graph*, using TensorFlow. Additionally, this architecture allows us to easily define new models with a simple configuration as demonstrated in Table 4.2, additional settings are available in the configuration, but we exclude these for brevity.

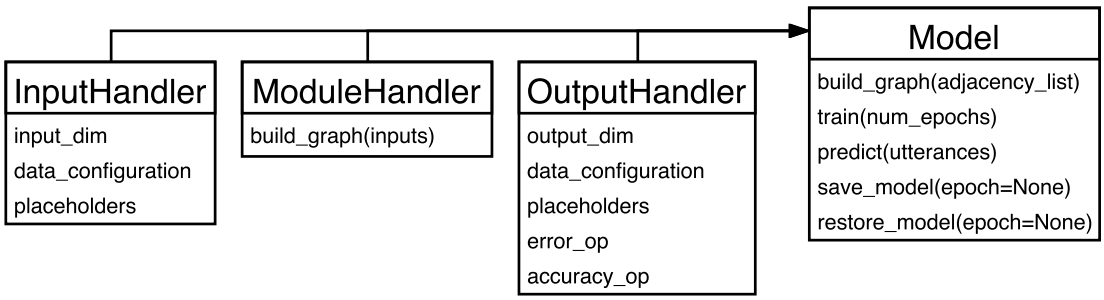


Figure 4.8: Class architecture of our modular neural network implementation

Table 4.2: Example model configuration using modular neural network implementation, there exist other configuration options that were omitted for brevity.

InputHandlers	
Name	**Type**
input 1	eGeMAPS

ModuleHandlers	
Name	**Type**
module 1	Fully connected 128 sigmoid
module 2-1	Fully connected 16 sigmoid
module 2-2	Fully connected 16 sigmoid

OutputHandlers	
Name	**Type**
output 1	Categorical
output 2	Dimensional

Graph	
Node	**Children**
input 1	module 1
module 1	module 2-1, module 2-2
module 2-1	output 1
module 2-2	output 2

Chapter 5

Speech Synthesis

The vast majority of speech synthesis techniques operate using a standard pipeline architecture; the frontend – linguistic processing – and the backend – acoustic regression, followed by waveform generation. Baidu's Deep Voice (Arik et al., 2017) and Montréal's char2wav (Sotelo et al., 2017) are the first fully end-to-end synthesis techniques, i.e. from text to waveform in one model.

In all other cases the first step required is linguistic processing, as discussed in Section 5.1 this converts the text into a linguistic representation. Following this, the linguistic parameters must be transformed into acoustic parameter space, we make use of SPSS as outlined in Section 5.2. Finally the acoustic parameters are used by a vocoder to generate the waveform, we discuss the limitations of using a vocoder in Section 5.2.3.

5.1 Linguistic processing frontend

The first step in any speech synthesis system is to process the input text into a linguistic description. This step is known as the frontend and is a collection of NLP operations such as part-of-speech (POS) tagging and word-sense disambiguation. Resources such as a pronunciation dictionary as well as handwritten letter-sound-rules enable the system to produce a sequence of phones from the processed sentence, along with metadata about the phones.

These processes (i.e. the frontend) serve to produce a linguistic representation of the input - a complete description of what sounds should be produced and how their production should vary. For the purpose of unit selection synthesis, the linguistic parameters are used as the target when selecting speech units from the database available.

We use the Festival toolkit (Taylor et al., 1998) as our frontend. Festival, was built to be a stable, easily maintainable synthesis system. For this reason it is still used today, for unit-selection synthesis, as a baseline in the Blizzard challenge (King and Karaiskos, 2016), as well as for research into hybrid synthesis systems (Merritt et al., 2016).

5.2 Statistical Parametric Speech Synthesis (SPSS)

An alternative to using a fixed database of speech, is to learn a statistical model of speech, this is known as statistical parametric speech synthesis (SPSS) (Zen et al., 2009). SPSS involves learning a mapping from linguistic parameters to acoustic parameters using an acoustic model, this is then converted into a waveform using a vocoder.

The use of any signal processing negatively effects the naturalness of produced speech, for this reason using a vocoder introduces an upper bound on the naturalness we can achieve. It is possible to generate the waveform directly from linguistic features, this was demonstrated by DeepMind using dilated convolutions (Oord et al., 2016). Due to the waveform's very high dimensionality, generating it directly is computationally expensive.

5.2.1 HMM synthesis

Following the success of hidden Markov models (HMMs) in automatic speech recognition (ASR), they were adopted for speech synthesis as an acoustic model. This was due to the availability of learning algorithms and search algorithms, in reality the HMM isn't an especially good model of speech.

An HMM uses probabilistic transitions, while this is suitable for ASR, using HMMs as a generative model for speech requires a more explicit duration model.

Typically a decision tree would be trained as the duration model, using this in place of the HMM's transitions makes them hidden semi-Markov models, however we simply refer to them as HMMs.

Each HMM models one phone's acoustic parameters, it learns the distribution of observations given for that phone. For generation, maximum likelihood parameter generation is used to choose the best sequence of HMM observations, taking into account velocity (Δ) and acceleration ($\Delta\Delta$) of the observations.

The most commonly used toolkit for HMM synthesis is HTK (Young and Young, 1993), this contains many useful operations for handling HMMs. HTK also includes HTS tools for creating full-context labels, these are the standard format for linguistic parameters used in SPSS (Zen et al., 2007). We utilise HTK for the preparation of HTS full-context labels.

5.2.2 DNN synthesis

The acoustic and duration models in SPSS perform a well-defined supervised learning task. The acoustic model must learn a function from linguistic labels to acoustic labels; and the duration model must learn a function from linguistic labels to discrete durations. This learning task is well suited to deep neural networks (DNNs), using DNNs as acoustic and duration models is called DNN synthesis.

In SPSS, the duration model is used to determine the number of frames that should be produced by the vocoder for each phone. This information is utilised by upsampling the linguistic description and adding a frame counter for each phone. By upsampling the linguistic inputs, the DNN acoustic model will predict acoustic parameters equal to the number of frames predicted by the duration model.

Learning the acoustic model requires an objective function, it is standard to use a combination of error metrics to calculate objective performance. The spectrum parameters; Mel-generalised cepstrum and band aperiodicity, are evaluated as an error in decibels (dB), i.e. log difference between predictions and targets. F_0 is evaluated using the root-mean-square error (RMSE), and the voiced/un-voiced prediction (VUV) is evaluated using percentage error as the target is binary.

DNN synthesis is facilitated by the sharp increase in available computing power, as well as the availability of high-level machine learning software, such as Theano (Al-Rfou et al., 2016) and TensorFlow (Abadi et al., 2016). We use the Merlin[1] toolkit for DNN synthesis (Wu et al., 2016b) in combination with the Blizzard 2017 data, this allows us to easily build SPSS voices based on the given recipes.

The inputs used for DNN synthesis have a great deal of impact on the performance of the model. For example, neural networks cannot understand categorical variables, therefore, such features are represented as "on-hot" vectors, this avoids encoding ordinal information. On the other-hand neural networks may learn slower given inputs with broad distributions; normalising real-valued features can improve DNN performance greatly. We follow the design choices made by the authors of Merlin in creating the DNN input features. Based on the performance of the DNN benchmark in the Blizzard challenge, created using Merlin, we are confident these design choices are well-informed.

5.2.3 The vocoder

In this thesis we use the WORLD vocoder (Morise et al., 2016), this provides state-of-the-art synthesis from acoustic parameters. The parameters used are $\log F_0$, Mel-generalised cepstrum (MGC), and band aperiodicity (BAP). These are transformed into a waveform by reconstructing the spectrum.

WORLD is based on the assumption that speech is fully described by these acoustic parameters. While this is mostly correct, there exist certain types of speech that WORLD cannot reproduce. Creaky and breathy voice cannot be produced by a vocoder, since these require modification of the source (vocal folds), something which is not modelled by traditional vocoders. In order to pursue a full range of emotive speech it is necessary to investigate more powerful waveform generation techniques. Neural network based waveform generation, such as sampleRNN (Mehri et al., 2016) may be a useful direction to investigate, as they remove the restrictions imposed by traditional vocoders.

[1] available at https://github.com/CSTR-Edinburgh/merlin

5.3 Emotive Speech Synthesis

To perform style adaptation for producing emotive speech, we use auxiliary features, a speaker adaptation technique from ASR. We incorporate additional features into the Merlin data preparation process. The acoustic and duration models are trained using a combination of the normal linguistic features, as well as our emotion features - these should indicate what emotion each utterance contains. We experiment with a variety of emotion features, these are discussed and evaluated in Section 6.3.

As discussed in Section 2.4, there are many other methods for performing style adaptation. One particular model adaptation technique that may be useful is learning hidden unit contributions (LHUC) (Swietojanski and Renals, 2014). LHUC adds additional scaling parameters to every unit in the neural network, these parameters are learnt for each style allowing the model to adapt. This can include new styles, where we use a small amount of training data to learn additional adaptation parameters. LHUC was developed for speech recognition, however, Wu et al. (2015) demonstrated its use for speech synthesis. While we did not have time to investigate this, using a model adaptation method such as LHUC would allow us to train a voice that more explicitly varies the acoustic model according to each emotional style. Alternatively, we could make use of LHUC for speaker adaptation, allowing us to perform speaker dependent synthesis with a multi-speaker dataset such as IEMOCAP.

Chapter 6

Experiments

6.1 Emotion recognition

We present our investigation of the methods outlined in Chapter 4. We begin by evaluating a basic model to act as our baseline neural network system, following this we provide results from progressive additions and changes to the model. Ultimately, we devise a single architecture in Section 6.2; this will be evaluated using emotive speech synthesis in Section 6.3.

Throughout our investigation we used the Adam optimizer (Kingma and Ba, 2014) with the default parameters of $\eta = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = e^{-8}$ to train our models by gradient descent. Through preliminary experiments we determined that training the models for 40 iterations (epochs) of the training data allowed for the majority of architectures to converge, with the exception of the TD-CNN model which did not fully converge in 40 epochs. All non-linearities for our recognition models use the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$.

6.1.1 Baselines

We begin by devising a baseline neural network model through a brief parameter search. Our best deep neural network (DNN) architecture was 1024–16, using eGeMAPS inputs. This architecture achieved an accuracy of 72.77% when classifying 4 basic emotions (angry, happy, sad, and neutral; referred to as Basic4). In addition, we create two lower-bound baselines; random, and most common.

Their accuracies are 24.14% and 33.00% respectively, the large margin between these lower-bounds and our baseline DNN indicates that the task is not trivial, and we are learning something useful.

The error and accuracy learning curves for our baseline DNN predicting Basic4 with varying levels of dropout can be seen in Figure 6.1. It is clear that dropout degrades performance. This suggests that the eGeMAPS inputs used are minimal, that is, if we do not have access to all 88 features, we are unable to learn certain relevant traits in the data. It is also possible that our model is underfitting, however we did not investigate the use of dropout for other, larger architectures.

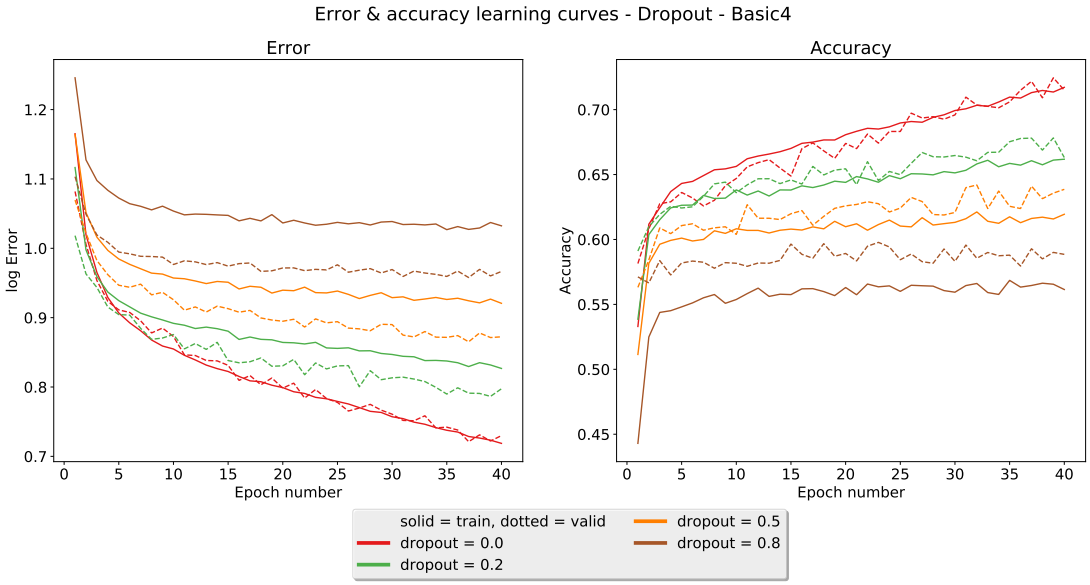


Figure 6.1: Sigmoid cross-entropy error and accuracy results for the feed-forward architecture 1024–16, using eGeMAPS inputs and predicting 4 basic emotions; angry, happy, sad, and neutral. Different colours indicate different levels of dropout, the dotted lines indicates validation performance.

6.1.2 RNN and TD-CNN models

We implemented a recurrent neural network (RNN), and a time-distributed convolutional neural network (TD-CNN), as outlined in Section 4.2. The grid search we performed to devise these architectures was limited, due to computational resource limitations. The best architectures for these two models, as well as our baseline DNN and lower-bounds are given in Table 6.1.

Table 6.1: Architecture of best models built, all classifying Basic4

Model	**Inputs**	**Architecture**	**Accuracy**
Random	N/A	choose random class	24.14%
Most common	N/A	choose neutral class	33.00%
DNN	eGeMAPS	DNN layers: 1024–16	**72.77%**
RNN	LLDs	GRU layer: 64–64–64–64	
		DNN layer: 16	43.17%
TD-CNN	Spectrogram	CNN kernels: 20 x 20, 4 channels	
		GRU layers: 128–128	
		DNN layer: 20	58.94%

We were surprised with the poor performance of our RNN model. In theory it should be able to perform at least as well as the DNN, since the LLDs are used to create the static eGeMAPS features, thus the RNN has access to more information than the DNN - though it is possible that modelling temporal LLDs is more challenging than expected.

We performed some investigation into this; we designed an RNN that took both LLDs and eGeMAPS static features as inputs for every time step. We would expect this model to find a solution equivalent to, or better than, the DNN. One such solution is to only use the eGeMAPS features from the final step, i.e. set all other weights to zero, restricting the model to the capacity of a DNN. Unfortunately, this was not observed, which leads us to believe we introduced a bug in our RNN code. We were unable to investigate this further since this thesis also focuses on the use of speech synthesis, leaving us with less time to investigate recognition architectures.

Similarly, our TD-CNN model performed worse than we predicted. This task is particularly sensitive to hyperparameters as it makes use of the raw spectrogram. However, we could not investigate their effect on performance, due to computational limitations and time constraints. We believe the TD-CNN model holds potential, and merits substantial further research to establish its capability.

6.1.3 Literature comparison

In Table 6.2, we present results from the literature for the same task; recognition of Basic4 using the IEMOCAP dataset. It is important to note, these results are not directly comparable, in particular because training-validation-test splits are not standardised. Our baseline DNN model performs much better than other uni-modal techniques, the large performance gap suggests that the discrepancies in data selection and training procedures have a significant effect. We would hope to publish our state-of-the-art work on speech emotion recognition, however, a necessary step would be to replicate a subset of these systems to provide a valid comparison. This may involve contacting the authors, in order to verify their data selection methods.

The table of results is split in half, the first half contains results using features derived from audio only, while the second half contains various multi-modal approaches. It is clear that multi-modal approaches provide better performance; Metallinou et al. (2008), Rozgic et al. (2012), and Poria et al. (2016) present results for uni-modal and multi-modal models that demonstrate this difference. Input features utilised varies greatly, with MFCCs and ComParE feature sets being used the most, this suggests that it is more important to focus on better models that use multi-modal data. Poria et al. (2016) is an excellent example of this approach, they use multiple kernel learning (MKL) to combine various modalities, including features learnt from a spectrogram using a CNN; their method is state-of-the-art on IEMOCAP.

As mentioned, our audio-only baseline appears to be state-of-the-art, despite the variety of audio-only methods. We believe this is due to most methods focussing on learning features and performing basic classification using the features. Our method uses a single flexible model (a neural network) to learn the task end-to-end, including classification. We believe it would be worthwhile to replicate some of these results to verify if performing the final classification using models such as SVMs is one cause of the performance difference. In addition, our TD-CNN model out-performs the only other spectrogram-based method we could find for this task by 9.85% (Ghosh et al., 2015).

Table 6.2: Emotion recognition results on IEMOCAP, all classifying Basic4 (angry, happy, sad, and neutral).

Paper	Method[1]	Audio	Video	Text	Input features[1]	Accuracy
(Ghosh et al., 2015)	autoencoder, DNN	A			Spectrogram	49.09%
(Han et al., 2014)	DNN, ELM	A			MFCCs, F_0, voice probability, zero-crossing rate	52.13%
(Metallinou et al., 2008)	GMM, SVM	A			12 MFCCs	54.34%
TD-CNN (our method)	TD-CNN, RNN	A			Spectrogram	**58.94%**
(Rozgic et al., 2012)	ensemble SVM	A			12 MFCCs, jitter, shimmer	60.9%
(Poria et al., 2016)	CNN, MKL	A			ComParE 2016	61.33%
(Xia and Liu, 2015)	DBN, SVM	A			ComParE 2010	62.5%
(Lee and Tashev, 2015)	RNN, ELM	A			MFCCs, F_0, voice probability, zero-crossing rate	63.89%
DNN (our method)	DNN	A			eGeMAPS	**72.77%**
(Metallinou et al., 2010)	GMM, HMM	A	V		13 MFBs, pitch, energy, FAPs	62.42%
(Mower et al., 2011)	SVM	A	V		MFBs, pitch, energy, FAPs	64.5%
(Kim et al., 2013)	DBN	A	V		MFBs, pitch, energy, FAPs	66.12%
(Kim and Provost, 2013)	SVM	A	V		MFBs, pitch, energy, FAPs	68.5%
(Jin et al., 2015)	GMM, SVM	A		T	ComParE 2010, lexical features	69.2%
(Rozgic et al., 2012)	ensemble SVM	A	V	T	12 MFCCs, jitter, shimmer, FAPs, lexical features	69.4%
(Metallinou et al., 2008)	GMM, SVM	A	V		12 MFCCs, FAPs	**75.45%**
(Poria et al., 2016)	CNN, MKL	A		T	ComParE 2016, raw video, word2vec, POS	**76.85%**

[1] extreme learning machine (ELM); multiple kernel learning (MKL); deep belief network (DBN); Mel-frequency cepstral coefficient (MFCC); Mel filter bank (MFB); facial animation parameters (FAPs); word2vec (Mikolov et al., 2013); part of speech (POS).

6.1.4 Multi-task learning

As discussed in Section 2.3, the use of multi-task learning (MTL) has the potential to improve generalisation performance. Moreover, by learning a single representation for both tasks, the model may avoid overfitting to either labelling scheme's flawed properties, partially addressing our concerns about the annotations used.

Using our modular architecture, we implement MTL by defining a computation graph where each task is given a private hidden layer before the output. We present an MTL architecture with shared layers of sizes 1024–256, followed by two private, 16 hidden unit layers which connect to each target. The two single-task architectures used for comparison have the same setup as the multi-task architecture, except with only one 16 unit layer.

In Table 6.3 we see that using MTL has no effect on categorical emotion prediction, but performance degrades for dimensional annotations. This suggests that the shared representation learnt during training captures the concept of categorical emotions more than dimensional emotions. Nonetheless, its performance on the dimensional labels is affected only slightly, meaning that we have captured a representation that can predict both types of labels with only a small performance reduction for one task.

Table 6.3: Performance with and without MTL using a 1024–256–16(x2) architecture, predicting both categorical and dimensional annotations. Performance metrics are accuracy for Basic4, and sigmoid cross-entropy error for Dimensional.

Target	**Single-task**	**Multi-task**
Basic4	72.62%	72.98%
Dimensional	0.64	0.65

6.2 Learning an abstract representation of emotion

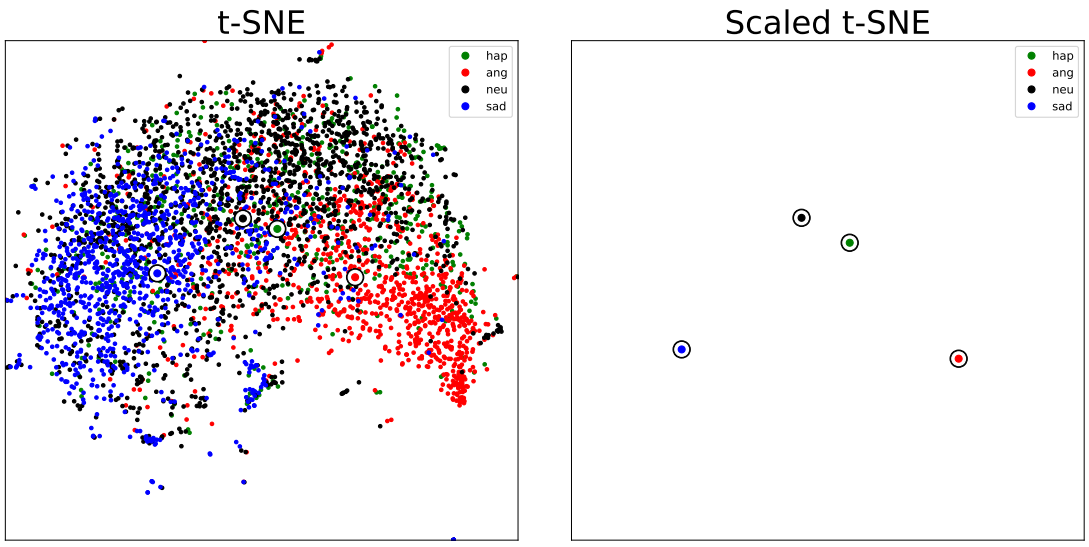
Using the models we have evaluated, we now investigate the use of stimulation, for the purpose of learning an abstract representation of emotion. Unlike Tan et al. (2015), we choose to stimulate one layer of the network, in doing so we aim to reduce the restrictions placed on the network. When stimulating a model, we interpret the stimulated layer as a grid, this is necessary for visualisation of the stimulation effect.

In order to demonstrate the effect of stimulation on the activation grid, we run the given model on an utterance and report the activations produced. We picked 16 utterances at random, 4 from each of our 4 basic emotions; angry, happy, sad, and neutral. These utterances are labelled from 1 to 16, any reference to an utterance will be consistent throughout this section. In the stimulated grid visualisations, the emotion label of each utterance is indicated by a red outline around that emotion's location in the grid. In addition, the full figures with all 16 utterances, along with a description of the utterances can be found in Appendix A.

6.2.1 The effect of emotion priors on stimulation

Stimulation makes use of a prior distribution, we define the prior by locations $\boldsymbol{s}_{e_t}$ of the categorical emotions e_t in a unit square. These locations can be derived in many ways, we make use of t-SNE; the state-of-the-art dimensionality reduction technique (Maaten and Hinton, 2008). We reduce the 88-dimensional eGeMAPS features into 2 dimensions in Figure 6.2a, and scale the means of each class to better cover the unit square in Figure 6.2b.

We demonstrate the effect of stimulation in Figure 6.3, both grids are from feed-forward models (1024–grid–16) with a 32 x 32 grid classifying Basic4, the single difference is the use of stimulation on the grid layer. As described in Section 4.3.4, the prior is Gaussian, this is evident in the example given, which has a 2-dimensional Gaussian shape around the stimulated location. The stimulated model performs slightly better with an accuracy of 72.78%, compared to 70.98% for the unstimulated model. This improvement suggests that restricting the model using stimulation guides it to a more accurate representation of emotion.



(a) t-SNE embedding and class means. Each point is an utterance, colour-coded by its emotion class.

(b) t-SNE class means, scaled to fill unit square

Figure 6.2: t-SNE embedding of 88-dimensional eGeMAPS speech features for IEMOCAP. Class means are used by stimulation as an emotion prior map to create interpretable neural network activations.

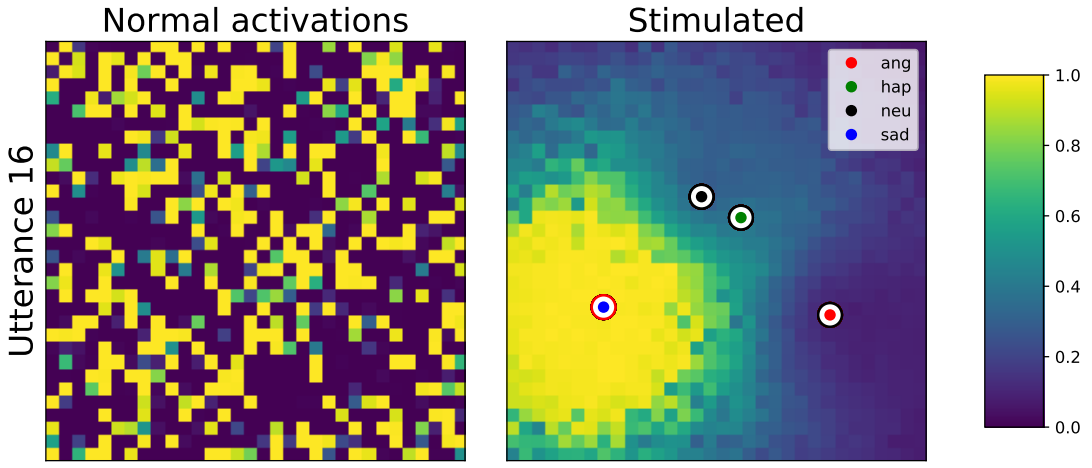
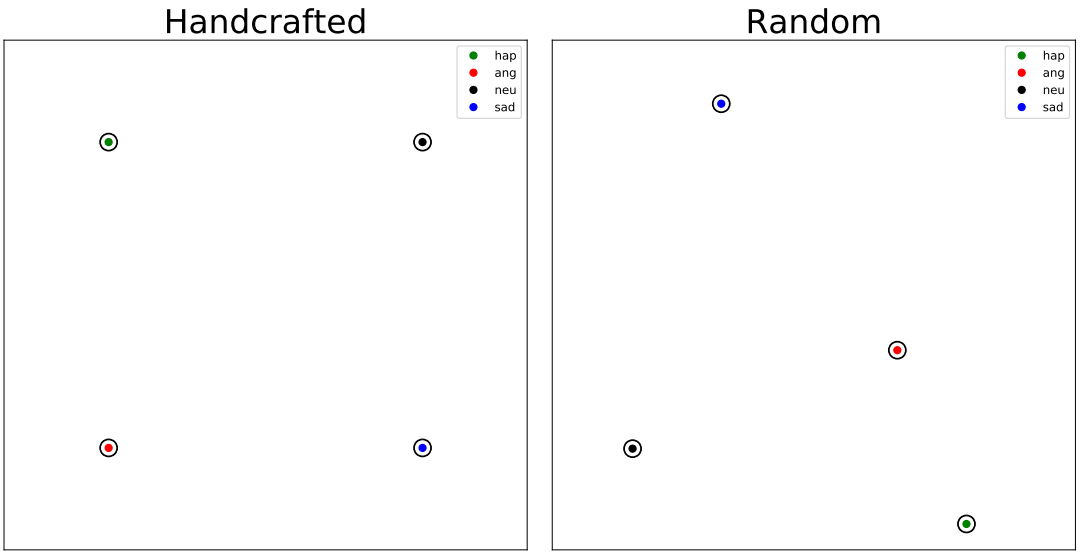


Figure 6.3: Visualisation of hidden activations represented as a grid. The architecture used was 1024–grid–16 where the grid is 32 x 32 units, the two images show the grid with and without stimulation. Stimulation is performed using the t-SNE embedding of the input features, scaled to fit in a unit square.

It is possible to choose any arbitrary layout for the prior. We believe using the 2-dimensional t-SNE embedding is sensible as similar emotions will be close to each other in the embedding, this should allow an utterance with multiple possible emotions to produce a contiguous mass of activations in the grid. To demonstrate the effect of the prior, we investigate the use of two other priors; a handcrafted layout, and a random layout. The two additional prior maps are shown in Figure 6.4

We built 4 stimulated models using the same 1024–grid–16 architecture with a 32 x 32 grid, each using a different prior map; handcrafted, random, t-SNE, and scaled t-SNE. The activations of the stimulated layer for these 4 models on two utterances are shown in Figure 6.5. The handcrafted and random priors struggle with the happy utterance, a class that is often misclassified as neutral. While the t-SNE priors are able to utilise areas in between emotion locations for similar emotions. One concern was with the spacing of the t-SNE prior maps, in the angry utterance we see that the model learns to use the space flexibly if other classes are located close by. We see little performance difference between the priors; the random prior outperforms the scaled t-SNE prior by under 2%. Since the performance difference is small, and we see good flexibility modelling similar emotions, we choose to use the scaled t-SNE prior in the remaining experiments.



(a) Stimulation prior map created by hand (b) Simulation prior map created randomly

Figure 6.4: Emotion prior maps used by stimulation to create interpretable neural network activations.

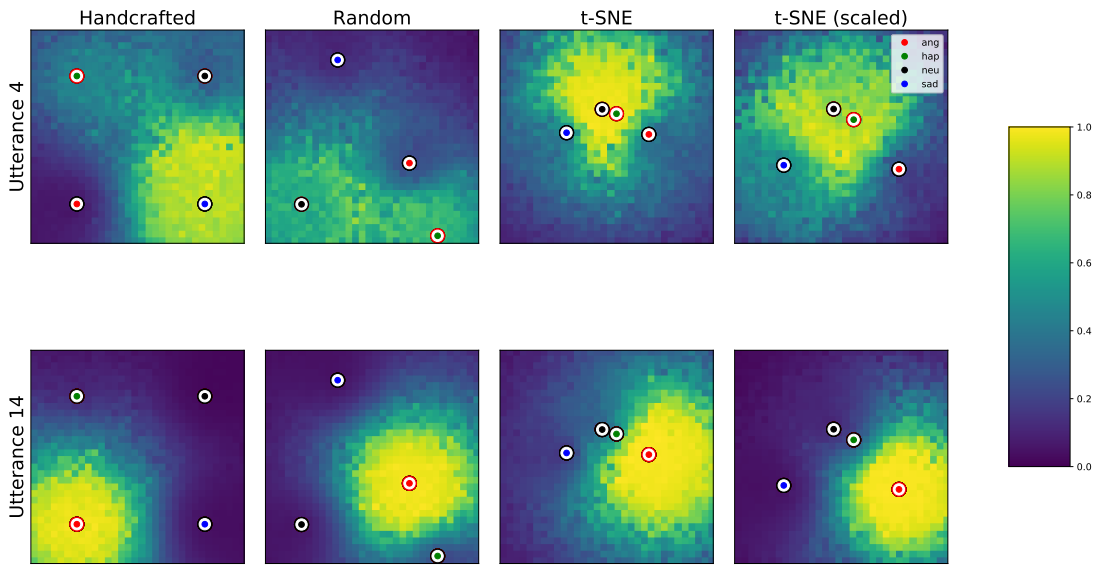


Figure 6.5: Visualisation of stimulated activations from a 1024–grid–16 model, with a 32 x 32 grid. The four columns show the use of different emotion prior maps

6.2.2 Grid size comparison

Next we investigate different grid sizes, using a 1024–grid–16 architecture with 8 x 8, 16 x 16, and 32 x 32 grids. We make use of the stimulation hyperparameters suggested by Wu et al. (2016a), $\sigma_{st} = \sqrt{0.1}$ and $\eta_{st} = 0.05$. The accuracies of the stimulated and non-stimulated versions of these architectures are outlined in Table 6.4. The stimulated architectures perform better, in addition the larger two grid sizes demonstrate the best accuracy. We continue by using 16 x 16 grids throughout our experiments. We choose to use this grid over the larger 32 x 32 grid (which performs similarly) as we do not want to maintain too large a representation for use with a speech synthesis system.

Table 6.4: Accuracy of stimulated and unstimulated models predicting Basic4 emotions, for varying grid sizes.

	Grid size		
	32 x 32	16 x 16	8 x 8
stimulated	72.78%	**73.27%**	71.84%
normal	70.98%	72.62%	72.72%

6.2.3 Stimulation parameter exploration

As discussed in Section 4.3.4, stimulation makes use of two hyperparameters; σ_{st} scales the Gaussian prior, while η_{st} weights the contribution of the stimulation penalty. In this section we use a 1024–grid–16 architecture with a 16 x 16 grid and scaled t-SNE prior.

We investigate the effect of the parameters independently, beginning with σ_{st}'s effect on the activations; we use $\eta_{st} = 0.05$ for all models and $\sigma_{st} \in \left\{\sqrt{0.01}, \sqrt{0.1}, \sqrt{0.5}\right\}$. σ_{st} affects the sharpness of the grid, this is indicated in the column headings in Figure 6.6. Smaller values of σ_{st} lead to hard boundaries in the grid, this severely limits the model's flexibility. On the other hand, we see that large σ_{st} causes saturation throughout the grid. In this case stimulation encourages activations to be high everywhere in the layer, with a slight preference towards $\boldsymbol{s}_{e_t}$. While there is more structure than without stimulation, it is difficult to interpret the contents visually.

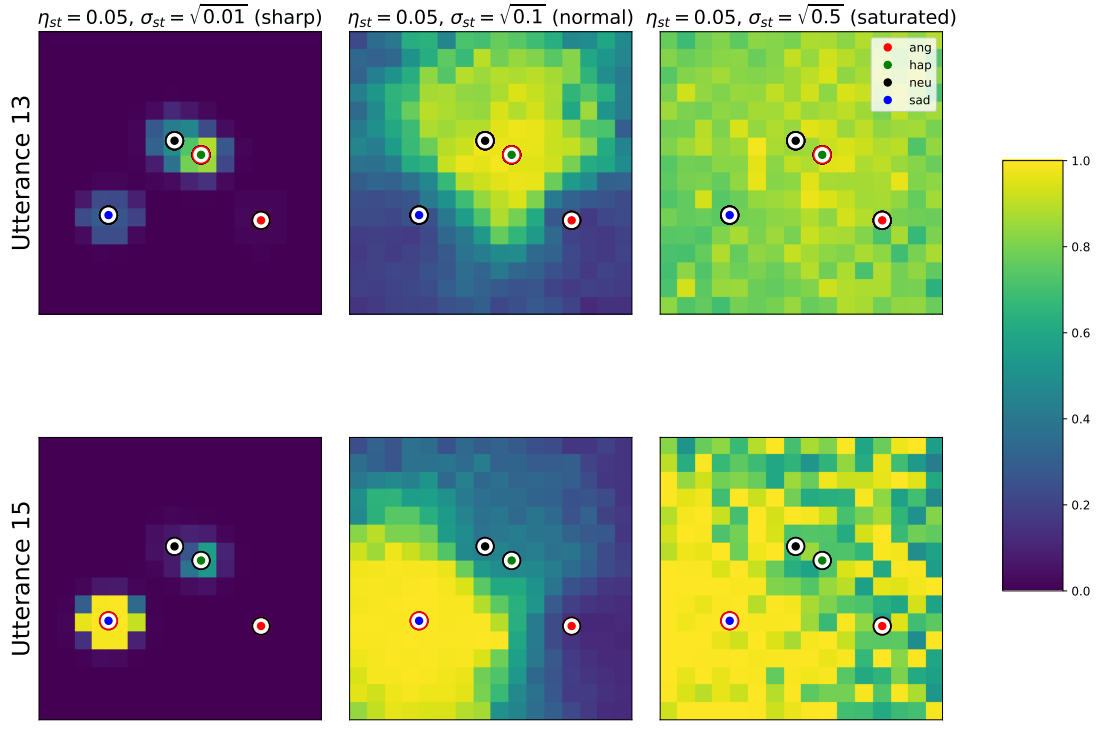


Figure 6.6: Visualisation of stimulated activations from a 1024–grid–16 model, with a 16 x 16 grid using the scaled t-SNE emotion prior and $\eta_{st} = 0.05$. The three columns show $\sigma_{st} \in \left\{\sqrt{0.01}, \sqrt{0.1}, \sqrt{0.5}\right\}$.

Our visualisation of η_{st}'s effect on the activations can be seen in Figure 6.7; we use $\sigma_{st} = \sqrt{0.1}$ for all models and $\eta_{st} \in \{0.01, 0.05, 0.2\}$. η_{st} controls the contribution of the penalty to the total loss function. As $\eta_{st} \to 0$ the model is able to ignore the stimulation term in favour of using units anywhere in the grid, this results in "rougher" activations. In contrast, as $\eta_{st} \to 1$ (or even $\eta_{st} \to \infty$) the penalty for not conforming to Gaussian will become large and the model will learn to replicate the prior. For this reason, we see a smoother surface for the larger value of η_{st}, as it is learning to become smoother like a Gaussian.

While using smaller η_{st} allows the model to utilise units more liberally, it also reduces the usefulness of stimulation as the prior constraint is too loose; activations can appear in distant locations. Similarly, using larger η_{st} reduces the usefulness of stimulation as our model will learn the prior distribution, which is equivalent to softmax emotion prediction. For lack of more computational resources needed to perform a more detailed parameter search, we use the suggested parameter values $\sigma_{st} = \sqrt{0.1}$ and $\eta_{st} = 0.05$ (Wu et al., 2016a) which showed good performance and produce interpretable visualisations in our experiments.

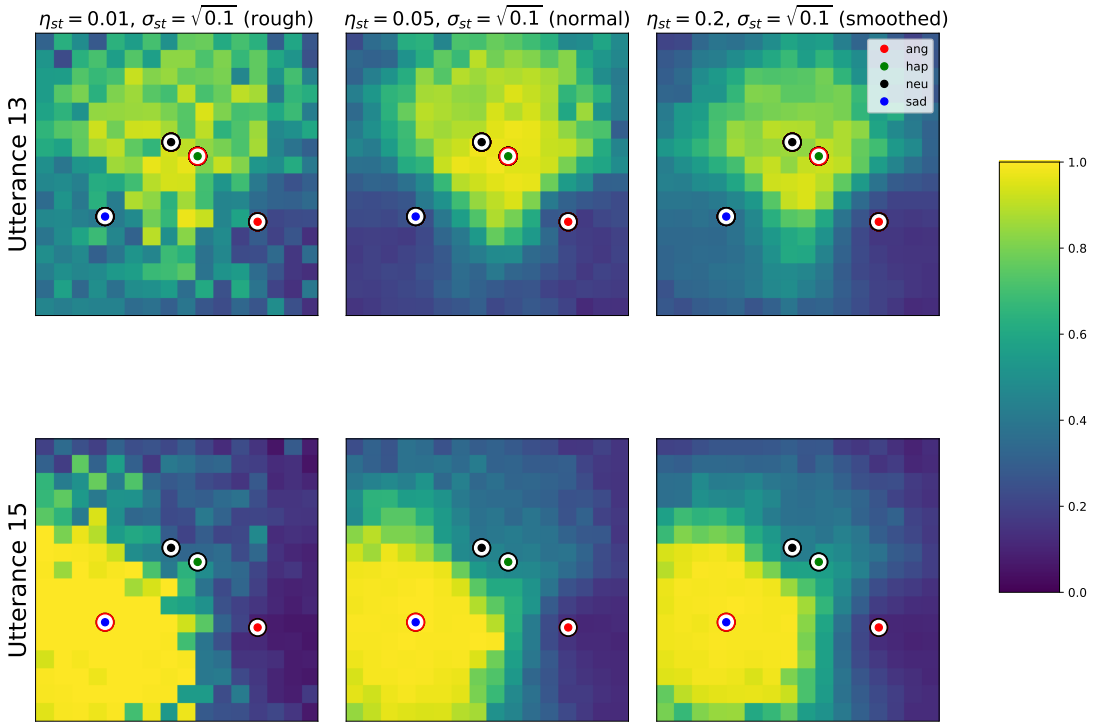


Figure 6.7: Visualisation of stimulated activations from a 1024–grid–16 model, with a 16 x 16 grid using the scaled t-SNE emotion prior and $\sigma_{st} = \sqrt{0.1}$. The three columns show $\eta_{st} \in \{0.01, 0.05, 0.2\}$.

6.2.4 Final system

Finally, we investigate the use of stimulation in conjunction with a multi-task architecture. We use identical architectures of 1024–grid–16, except for the multi-task models in which there are two private 16 unit layers. We train two single-task models for each of the tasks, these are similar to the baseline DNN model presented in Section 6.1.1 but with an extra layer for the grid. The comparison of the final single-task and multi-task models is presented in Table 6.5.

Table 6.5: Stimulated model results using chosen hyperparameters ($\sigma_{st} = \sqrt{0.1}$, $\eta_{st} = 0.05$), compared with our baseline and multi-task models. Performance metrics are accuracy for Basic4, and sigmoid cross-entropy error for Dimensional.

Target	**Non-stimulated**		**Stimulated**	
	Single-task	**Multi-task**	**Single-task**	**Multi-task**
Basic4	72.62%	72.98%	72.27%	71.92%
Dimensional	0.64	0.65	0.65	0.65

As in Section 6.1.4, we see that MTL has little effect on non-stimulated model performance; the same is true for stimulated models using MTL. While adding stimulation and MTL have very little effect on performance, we see that in combination adding stimulation to a multi-task architecture decreases performance by 1%. Fortunately this difference is negligible, especially when we consider the benefits of these methods; improved representation that captures both descriptions of emotion, and improved interpretability.

For the purpose of speech synthesis, which we discuss in the following section, we make use of the stimulated multi-task architecture presented in Table 6.5; 1024–grid–16(x2) using a 16 x 16 grid, scaled t-SNE prior, $\sigma_{st} = \sqrt{0.1}$, and $\eta_{st} = 0.05$.

6.3 Emotive speech synthesis

Throughout this thesis we have presented various features that provide descriptions of emotion; eGeMAPS, categorical labels, dimensional labels, and our abstract representation. Here we present an evaluation of these features, making use of speech synthesis to give perceptual results of their descriptive capability.

Using the merlin DNN synthesis toolkit (Wu et al., 2016b), we created 5 voices. All voices use the same architecture of 6 1024 unit feed-forward layers with *tanh* activations. We added additional features to the emotive voices using specific labels files generated by our recognition model. The first system is a baseline non-emotive voice, the remaining 4 voices are created by adding an additional feature vector to the vectorised linguistic parameters, these features are; eGeMAPS - a representation of the waveform; dimensional emotion predictions; categorical emotion predictions; and our abstract emotion space (eGrid).

6.3.1 Objective evaluation

Training an SPSS voice involves optimising an objective function, as described in Section 5.2, these metrics are calculated using the acoustic parameters. While the objective metrics may not correlate with listening test scores, they are useful for empirical evaluation during training. We present the objective results for the 5 trained voices in Table 6.6. Pearson's correlation refers to the correlation coefficient between the predicted and true acoustic parameters. The eGeMAPS voice has the highest correlation coefficient, this is unsurprising since eGeMAPS is derived from the waveform; eGeMAPS contains features such as MFCCs and $\log F_0$ that are directly related to MGC and $\log F_0$.

Table 6.6: Objective results of Merlin DNN synthesis voices on Usborne test data.

		Objective metric				
	Extra features	MCD (dB)	BAP (dB)	$\log F_0$ (RMSE)	VUV (error %)	Pearson's correlation
eGeMAPS	88	5.631	0.314	44.356	14.254	**0.700**
Dimensional	3	5.850	0.327	50.439	14.864	0.581
eGrid	256	5.825	0.327	51.420	15.211	0.562
Categorical	4	5.820	0.324	52.372	14.493	0.537
Non-emotive	0	5.845	0.329	52.846	14.768	0.525

6.3.2 Subjective evaluation by listening test

Listening experiments are the standard form of evaluation in speech synthesis. Our experiment is similar to a naturalness test, in which participants rate the similarity of examples to natural speech. We ask participants to rate the similarity of the emotional content, both in terms of how appropriate the emotion portrayed is and the quality of the emotion portrayal. We are not concerned with how natural the synthesis sounds, merely that the emotion chosen and its quality are good.

For subjective evaluation, we created an additional reference system using copy synthesis. Copy synthesis is a technique that takes real human speech, encodes it as acoustic parameters and re-synthesises the parameters using a vocoder. This adds artifacts to the waveform that are commonly associated with synthetic speech. The aim of this process is to avoid participants from rating the human speech as best merely because it is a human. Our analysis is comparing which acoustic model is better, not investigating the vocoder.

The Usborne dataset contains only text transcriptions, for this reason we used our model devised in the previous section to perform recognition on the Usborne speech. We collected dimensional and categorical predictions, as well as stimulated activations of the grid layer.

The distribution of each component of the categorical predictions is shown in Figure 6.8. We find that neutral is the largest component for the majority of predictions, followed by angry and sad. The majority of predictions for happy are below 0.2, thus happy is never the largest component in any prediction. In order to improve the performance of our dimensional, categorical, and eGrid features it would be worthwhile to investigate speaker adaptation techniques when training the recognition model. LHUC, discussed in Section 5.3 was originally created as a speech recognition technique for speaker adaptation, it would be useful in our situation of multiple speakers from multiple corpora.

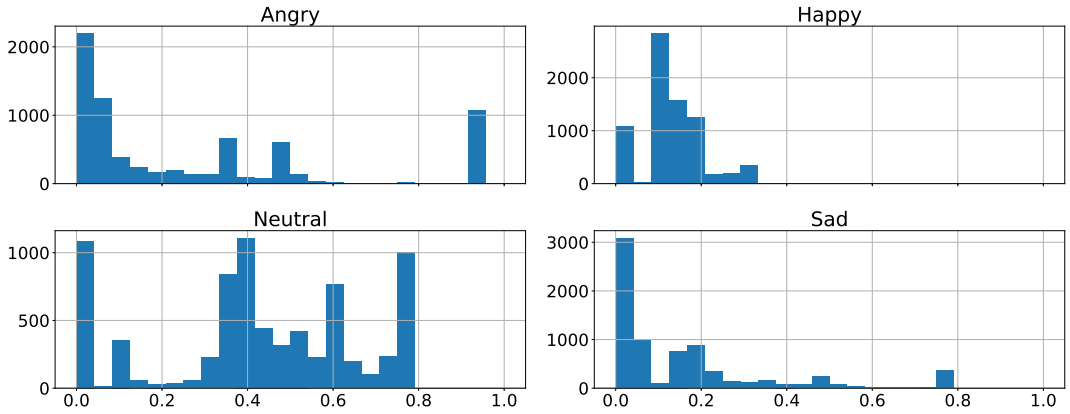


Figure 6.8: Histogram of 4-dimensional categorical softmax emotion predictions on Usborne data. Prediction made using a multi-task learning 1024–grid–16(x2) model with a 16 x 16 grid, scaled t-SNE emotion prior, $\sigma_{st} = \sqrt{0.1}$, and $\eta_{st} = 0.05$ trained on IEMOCAP multi-speaker data.

6.3.2.1 Experimental methodology

Our experiment is similar to a multiple stimuli with hidden reference and anchor test (MUSHRA), but without a visible reference (Series, 2014). We chose to remove the visible reference as it would suggest that it is the correct way to speak that sentence, in reality there are many correct ways to express a sentence. The reference was still included, hidden among the other voices.

We use the BeaqleJS[1] package for hosting listening tests through a web-browser (Kraft and Zölzer, 2014). Figure 6.9 shows the BeaqleJS interface used by participants. During the test, participants were sat in sound-deafened booths, using studio-grade headphones. We ensured the volume was consistent across all participants. The 21 participants were students from the University of Edinburgh; the experiment took between 30 and 45 minutes for which participants were paid £5.

6.3.2.2 Results

We calculated significance results using Wilcoxon's ranksums (Wilcoxon, 1945). We chose not to use the student t-test as the distribution of results was not normally distributed (Figure 6.10), making it unsuitable for a t-test which makes use

[1] https://github.com/HSU-ANT/beaqlejs

of a χ^2 distribution (a sum of squared normal distributions). Additionally, the ratings are very subjective, varying greatly between participants, this is demonstrated by the spread of the ratings in Figure 6.10.

The Wilcoxon test is a non-parametric test which uses the relative ranks of the scores (instead of the scores themselves), this is a more appropriate method for our experiment. Additionally, we perform Holm-Bonferroni correction to account for the number of pairwise significance tests performed (Holm, 1979). Finally we calculated 95% confidence intervals using Walsh averages (Geyer, 2003) to perform an inverse Wolcoxon. The boxes in Figure 6.11 represent the 95% confidence interval.

The hidden reference system (copy synthesis) was clearly recognised as the best, with the exception of two outliers. These two cases were by the same participant, supporting our assumption that participants interpret the ratings uniquely to other participants. Using the Wilcoxon test makes our system more robust to cases such as these. The remaining systems performed similarly, for clarity they are shown in more detail in Figure 6.11.b. The categorical voice is the only system not to perform significantly better than the baseline non-emotive voice, their pairwise Wilcoxon test has p-value of 0.058. The eGrid, dimensional, and categorical systems do not perform significantly different from each other.

The eGeMAPS system performs much better than the other 4 systems, indicating that this representation of the waveform contains the most relevant information. However, these features require a waveform to be created, making them inappropriate for novel emotion synthesis. The same is also true for our representation, since it makes use of the eGeMAPS features derived from the waveform, this provides a broad range of further work investigating methods to make this system end-to-end. One method we believe has promise, is to learn an emotion model that predicts eGeMAPS features from the linguistic description. This would function similarly to the duration model, inputting predicted features to the acoustic model for novel synthesis. Creating an emotion model such as this should allow for more explicit modelling of emotion within the voices.

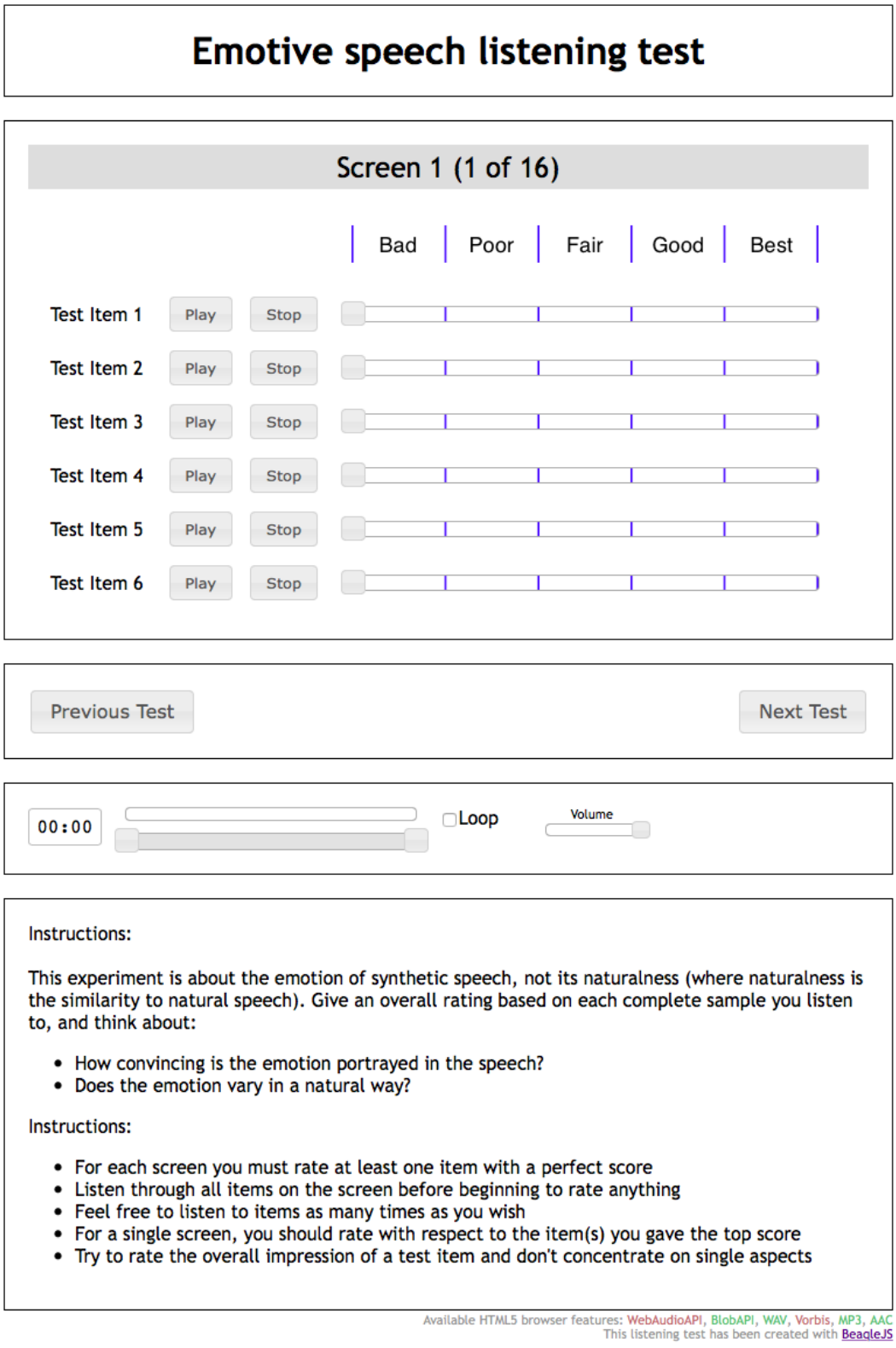


Figure 6.9: Web-interface viewed by participants for MUSHRA-like emotive quality listening test. Created using BeaqleJS

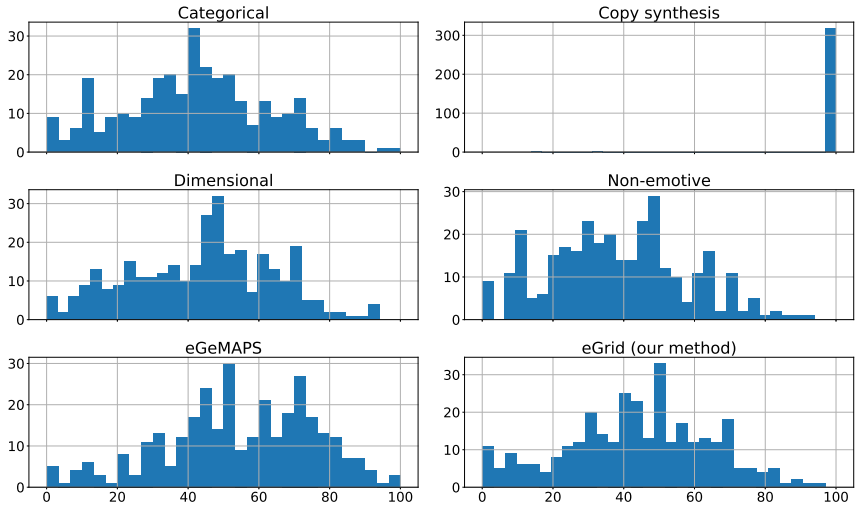


Figure 6.10: Distributions of ratings given by participants for the 4 emotive voices, the baseline non-emotive voice, and the reference copy synthesis voice.

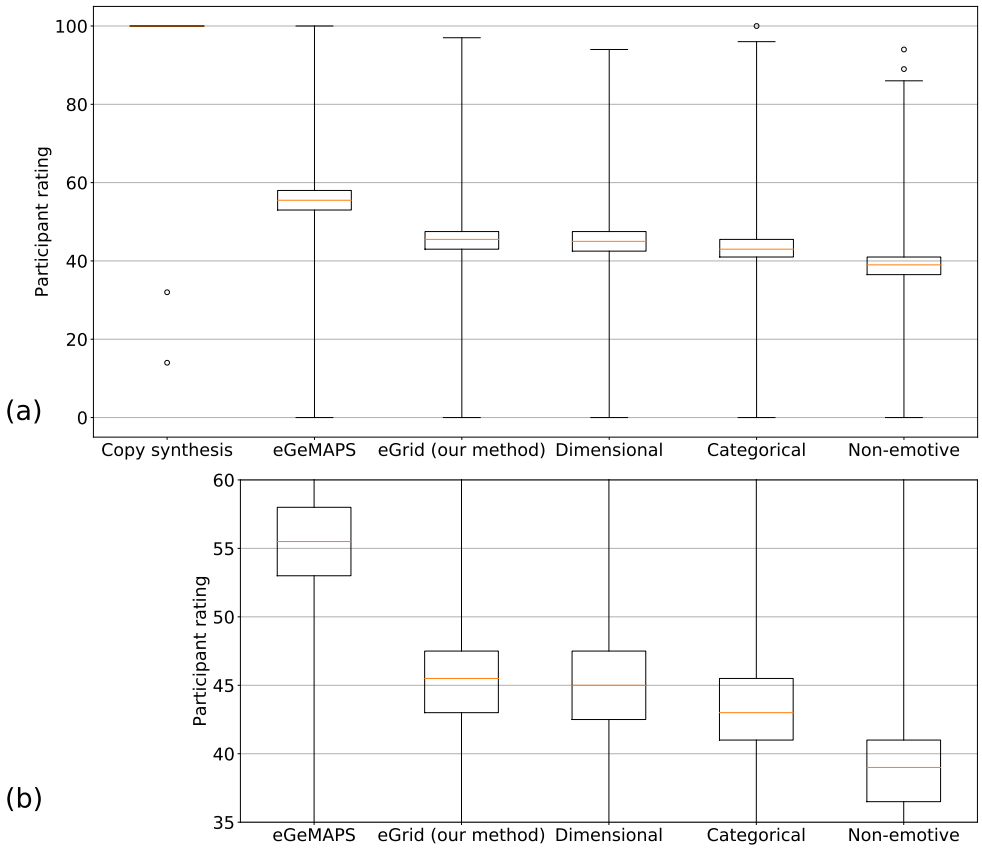


Figure 6.11: Box plot of listening test results, sorted by significance tests. Boxes show confidence intervals from Wilcoxon test, whiskers show range of results, and outliers are calculated using the standard deviation. (a) Full box-whisker plot. (b) Zoomed in on confidence interval, without copy synthesis.

Chapter 7

Conclusion

In this thesis we undertook two tasks; emotion recognition, and emotive speech synthesis. We presented an end-to-end DNN model for categorical emotion prediction with the IEMOCAP dataset, this basic technique provided excellent performance. Following replication of state-of-the-art methods we will be able to verify if our model outperforms existing methods for this task.

In addition we presented two, more complex architectures; RNN and TD-CNN. We encountered some difficulty with these methods, however, we hope to experiment with the TD-CNN model in the future. Since our TD-CNN model outperformed other spectrogram based methods.

In order to design an emotion space, we applied stimulated learning to a new domain. We made use of multi-task learning in an effort ensure our emotion space was robust to the shortcomings of categorical labels. Our emotion space has applications for novel emotions; it should be capable of representing unseen emotions, possibly on a spectrum across known emotions in the stimulated grid. However, the space not particularly interpretable for novel emotions.

In the second half of our work, we created various emotive speech synthesis voices and evaluated their portrayal of emotion using a MUSHRA-like listening test. This provided us with the means to quantify both; the quality of our emotion space, and the cross-corpus performance of our recognition model. Unsurprisingly we found that the eGeMAPS feature set, which aims to capture all emotion-relevant information from the waveform, produced the best emotive speech. The three voices built using predictions from our recognition model did

not perform significantly different, this suggests that our feature space does not encode significantly more information than the predicted labels.

7.1 Further work

In order to verify our emotion recognition model's high performance, it is important that this work is followed up by a more detailed comparison with existing methods on the same task. This would involve re-implementing other work from the literature in order to remove any differences in experimental setup. Following this it will be possible to make valid claims that work presented in this thesis is state-of-the-art.

We believe the issues encountered with RNNs and TD-CNNs are superficial. The use of both architectures merits further research, however, from our experience, we regard the TD-CNN model as a more promising technique.

Our aim for the abstract emotion space was to produce a feature space not susceptible to the flaws of existing emotion descriptors; categorical labels being too coarse to describe such a complex phenomena as emotion, and dimensional labels being too complex for annotators to interpret consistently. We chose to focus on supervised learning, making use of multi-task learning to alleviate these issues. It would have been more appropriate to make use of unsupervised learning, and to modelling the space in fewer dimensions. This is a complex but interesting area of work that was too time-consuming for us to consider in this thesis.

We did not investigate the use of any model-based speaker adaptation methods in this thesis. A technique such as LHUC could be applied for the purpose of explicit emotion modelling using style adaptation in either emotion recognition or emotive speech synthesis. Alternatively, LHUC could be used for speaker adaptation in speech synthesis, this would allow for the use of multi-speaker datasets such as IEMOCAP which also include emotion information.

In creating synthetic voices, we used the WORLD vocoder as a black box, ignoring the limitation it poses with respect to styles such as creaky and breathy speech. As discussed in Section 5.2.3, it would be worthwhile to make use of neural vocoders such as sampleRNN (Mehri et al., 2016), to alleviate such issues.

7.1.1 Improved evaluation of emotive speech

During the listening experiments we discovered a trend where participants would choose the copy synthesis voice as the best as it sounded like human speech. Our experiment was not intended to be a naturalness test, however by including the copy synthesis voice it instructed participants what the "correct" emotion should sound like. In reality there are many ways to express any given sentence. We propose an alternative methodology to perform listening tests for emotive speech.

In our listening test participants were presented with two combined tasks; the first being to rate the appropriateness of the emotion portrayed, and the second being to rate the quality of the emotion expressed. These two tasks model very different things; the former relates to the emotion chosen using the additional features, and the latter relates to the quality of the acoustic model. A more robust experiment would factor out these two attributes.

To perform a listening test to evaluate the quality of the emotion expressed, it is possible to select a full-blown emotion for each utterance, therefore voices would synthesise speech containing the same emotion. However, the emotion would be expressed according to the quality of the acoustic model. In addition, no human reference should be provided, this would avoid participants assuming this is the correct way to express the emotion. Thus, the test would focus solely on the quality of the emotion expressed by the voices.

Similarly, it is possible to design a listening test that evaluates the emotion chosen to be portrayed, while factoring out how the emotion is expressed. This can be accomplished through the use of an emotion model, given a sentence it would explicitly model what emotion to portray using emotion parameters. This would require the definition of emotion parameters, similar to acoustic parameters, these must be extracted from the waveform and sufficiently describe the emotion being portrayed. During training, the ground-truth emotion parameters - extracted from the waveform - would be used to create the acoustic model. Therefore, emotion models can be trained separately to the acoustic model and be evaluated using a listening test. A trivial option would be to use eGeMAPS, and design an emotion model that can predict these features from the linguistic parameters. However, it would be preferable for the emotion features to contain more interpretable values, allowing for control over the emotions produced.

Appendix A

Stimulation visualisations

We present the stimulated activations for all 16 selected utterances on the following pages. In addition, we outline the content of the 16 utterances in Table A.1.

Table A.1: Details of utterances used to demonstrate stimulation effects.

Utterance	**Emotion**	**Gender**	**IEMOCAP name**
Utterance 1	Neutral	Male	Ses02M_impro08_M011
Utterance 2	Happy	Female	Ses02F_script03_1_F023
Utterance 3	Neutral	Male	Ses01F_script02_1_M031
Utterance 4	Happy	Female	Ses01F_script01_3_F010
Utterance 5	Happy	Female	Ses01M_script03_1_F040
Utterance 6	Sad	Female	Ses04F_impro02_F005
Utterance 7	Angry	Female	Ses04M_script01_1_F036
Utterance 8	Angry	Female	Ses04F_script01_1_F042
Utterance 9	Neutral	Male	Ses04M_script02_2_M021
Utterance 10	Sad	Male	Ses05F_impro02_M034
Utterance 11	Neutral	Female	Ses05M_impro08_F015
Utterance 12	Angry	Male	Ses05F_script03_2_M019
Utterance 13	Happy	Female	Ses05M_impro03_F012
Utterance 14	Angry	Male	Ses03M_script03_2_M045
Utterance 15	Sad	Female	Ses03M_impro02_F031
Utterance 16	Sad	Female	Ses03F_impro06_F013

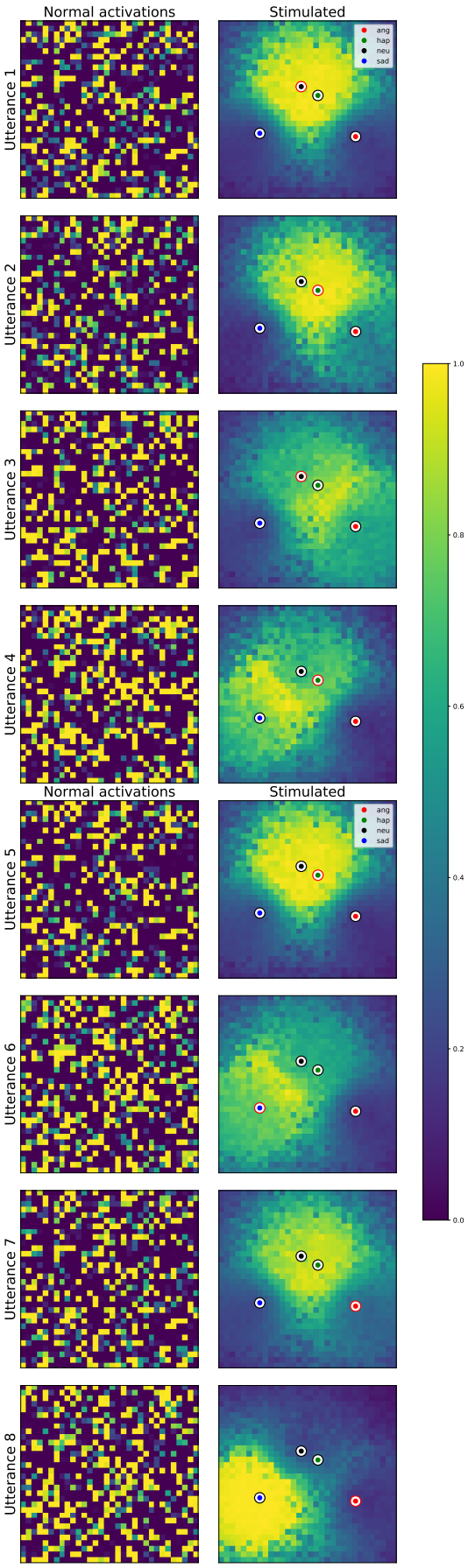


Figure A.1: Samples of activations with and without stimulation, using the scaled t-SNE emotion prior

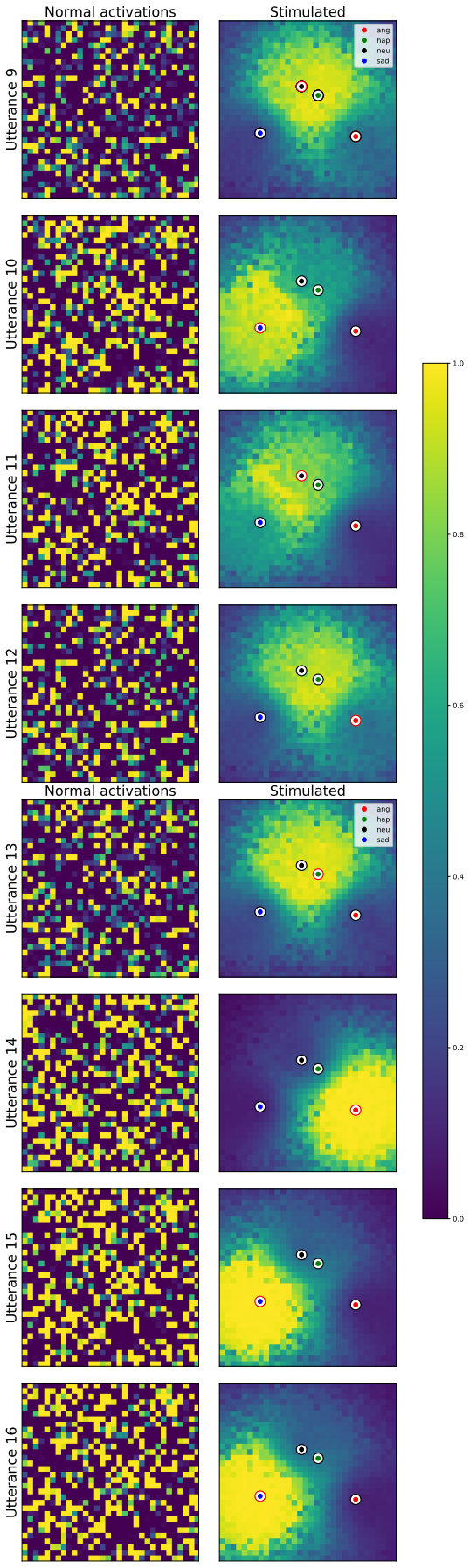


Figure A.2: Samples of activations with and without stimulation, using the scaled t-SNE emotion prior

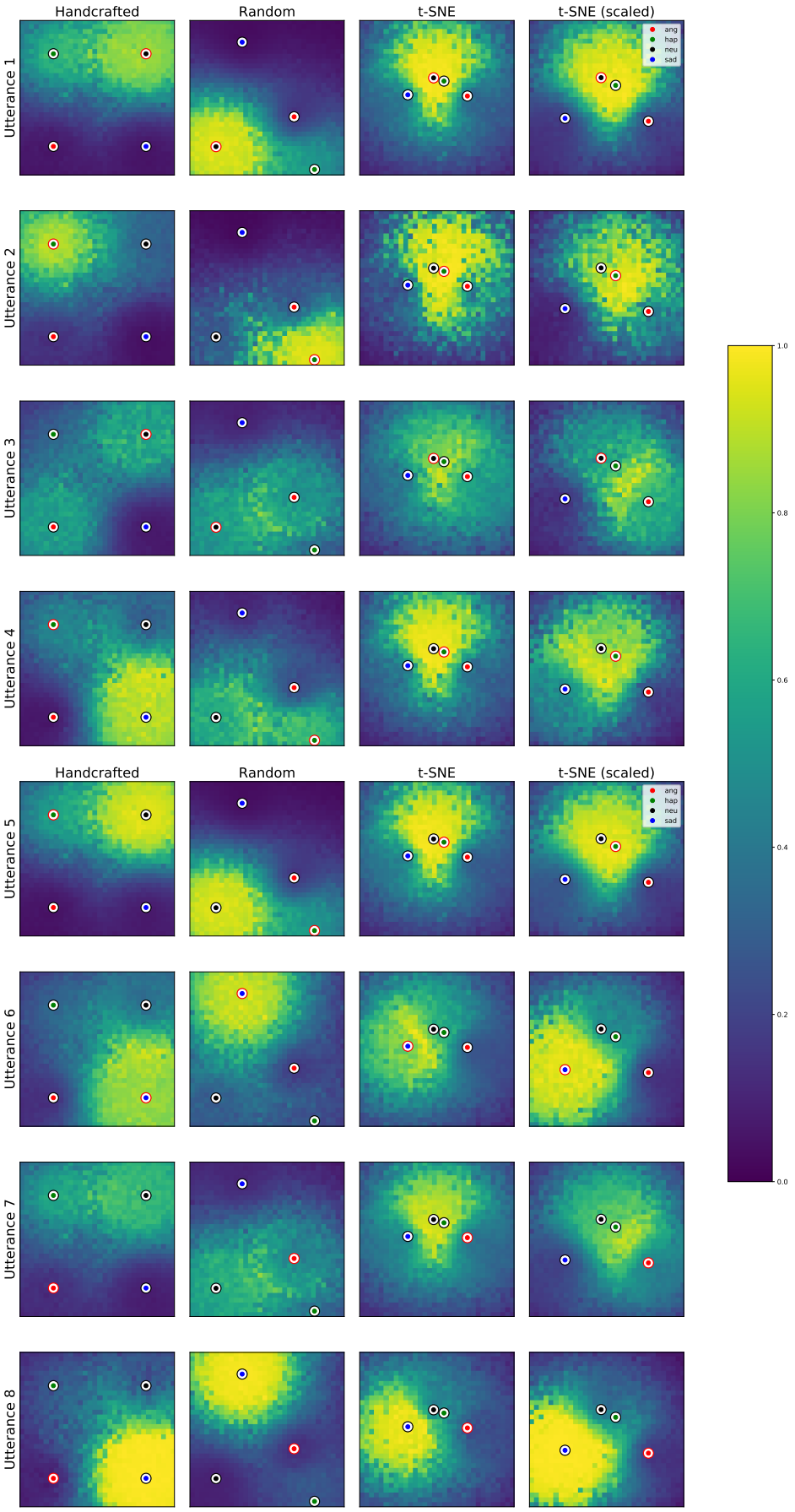


Figure A.3: Samples of stimulated activations for handcrafted, random, t-SNE, and scaled t-SNE emotion priors

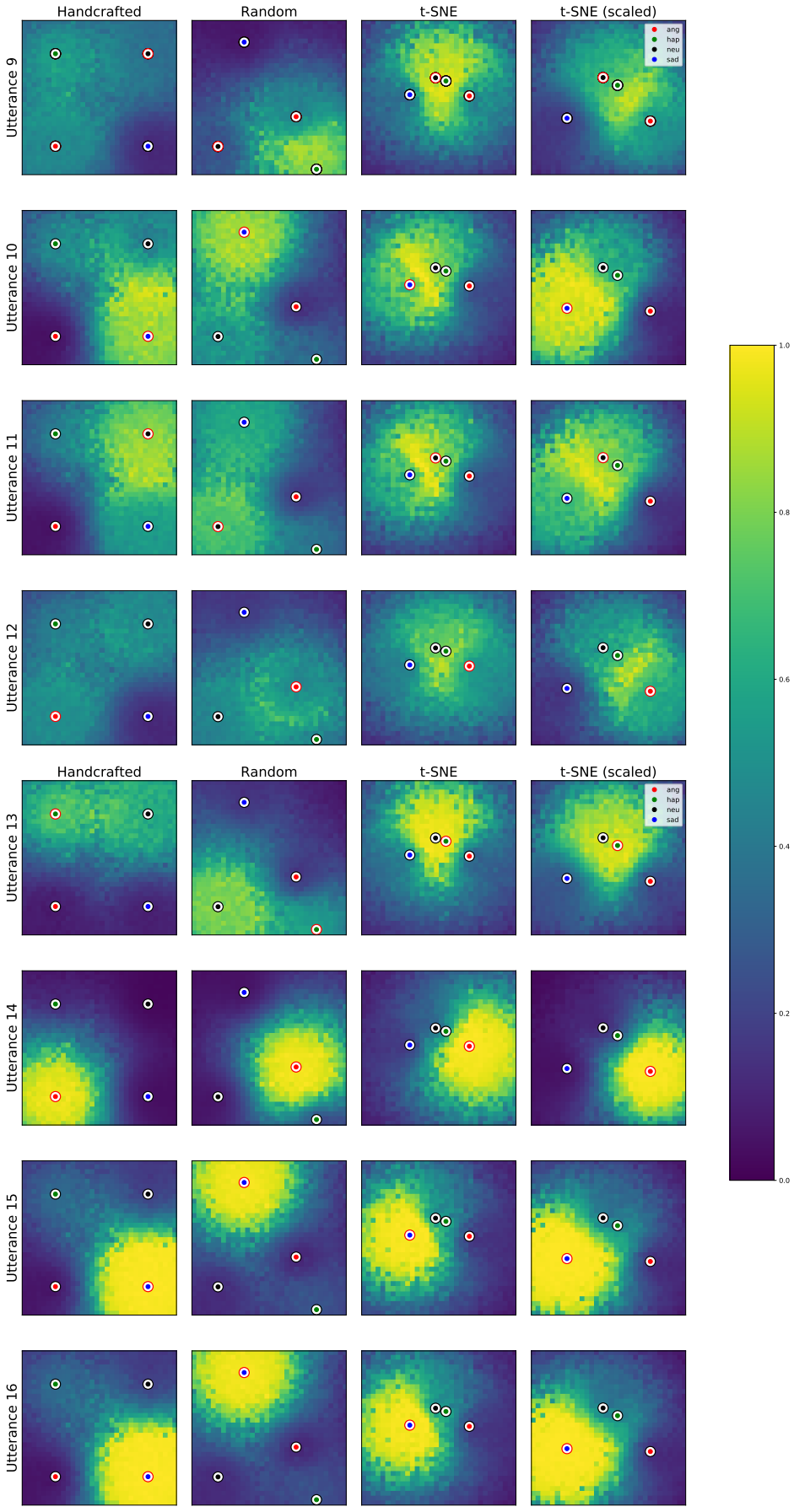


Figure A.4: Samples of stimulated activations for handcrafted, random, t-SNE, and scaled t-SNE emotion priors

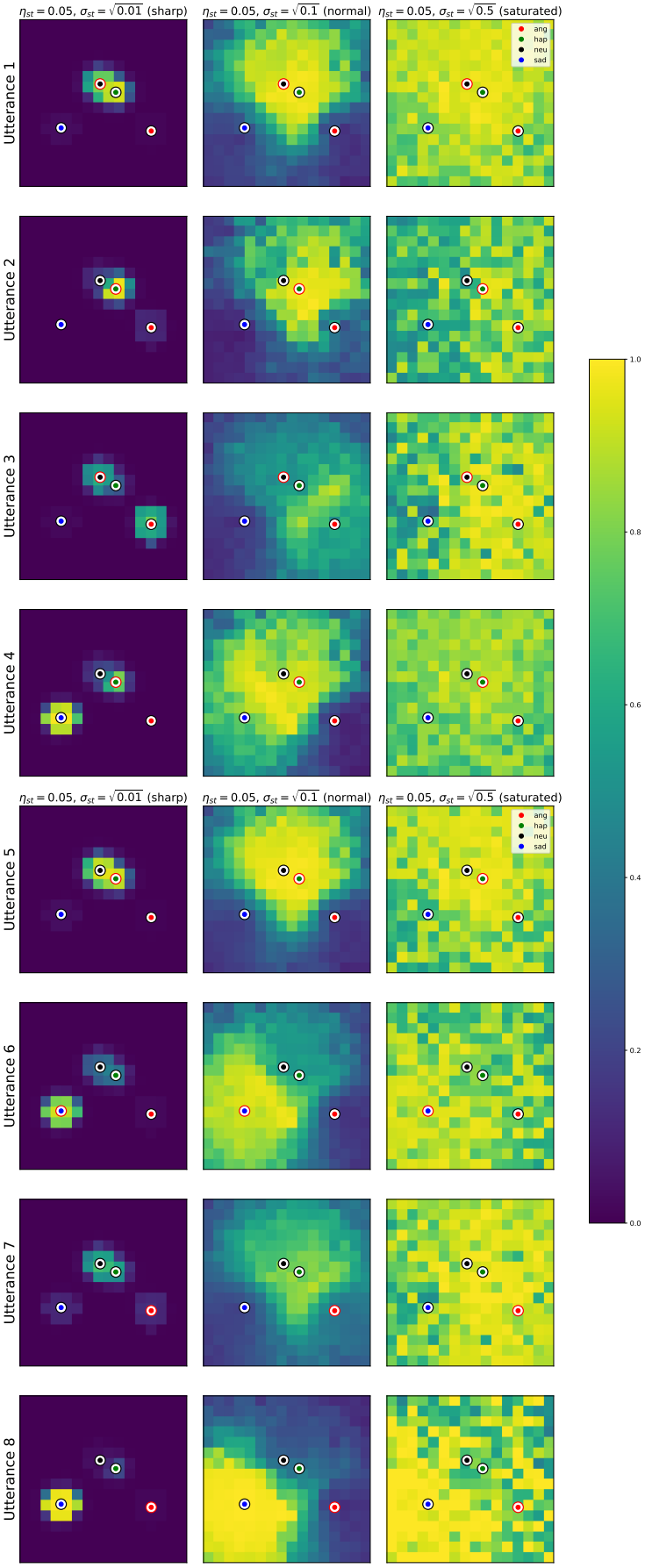


Figure A.5: Samples of stimulated activations for $\eta_{st} = 0.05$ and $\sigma_{st} \in \left\{\sqrt{0.01}, \sqrt{0.1}, \sqrt{0.5}\right\}$

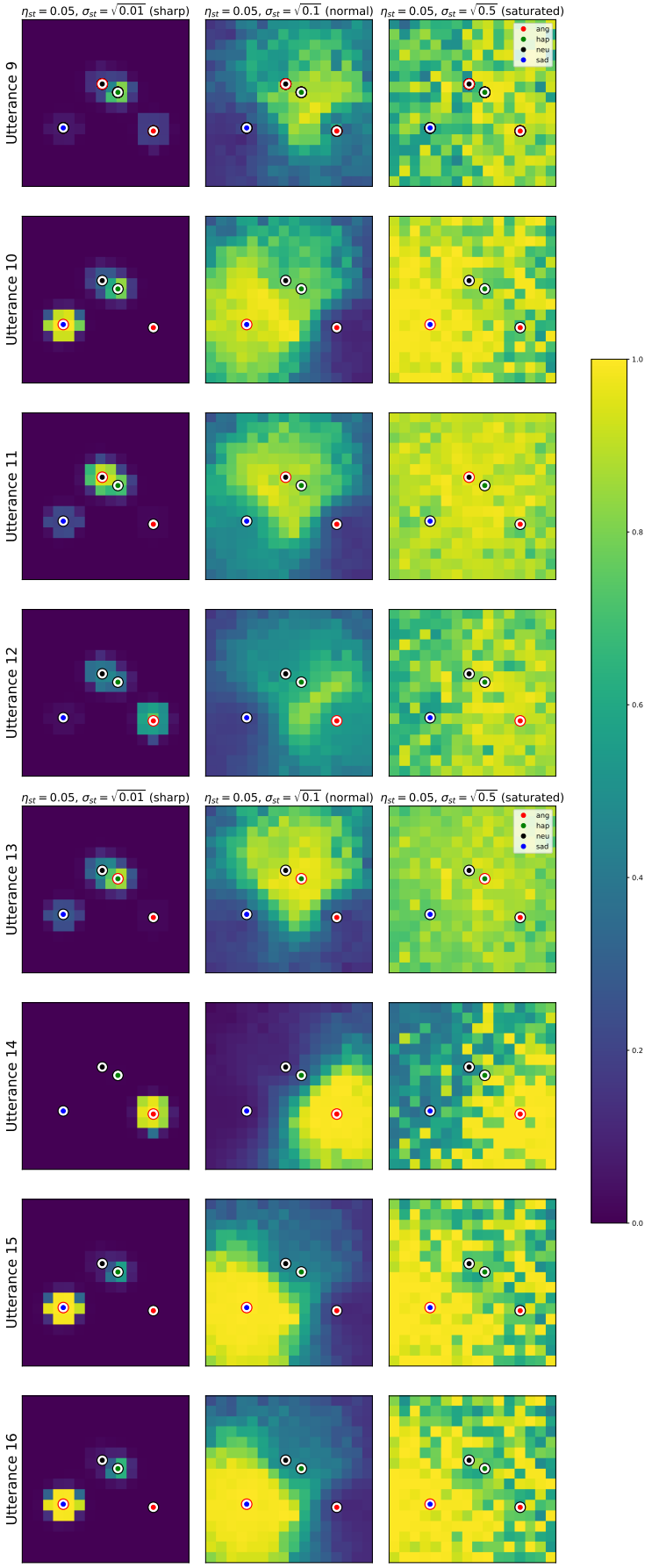


Figure A.6: Samples of stimulated activations for $\eta_{st} = 0.05$ and $\sigma_{st} \in \left\{\sqrt{0.01}, \sqrt{0.1}, \sqrt{0.5}\right\}$

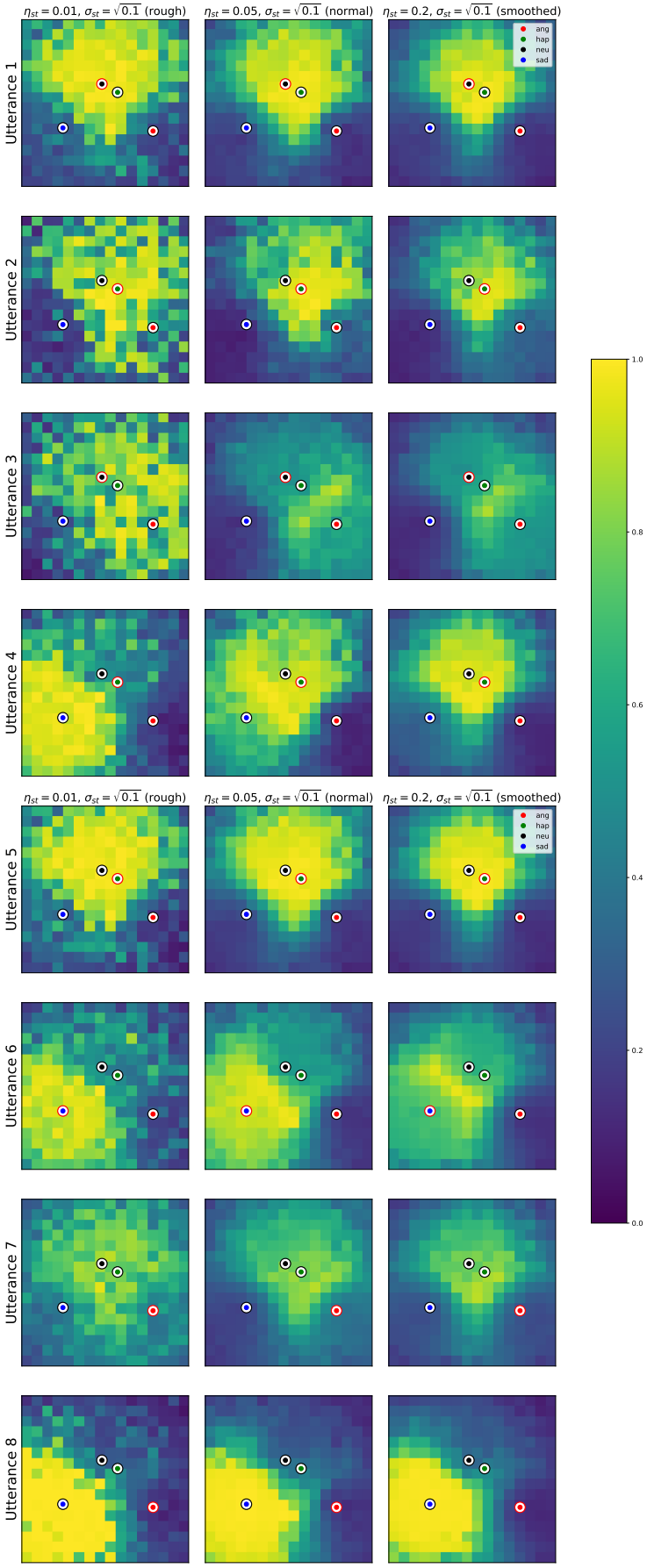


Figure A.7: Samples of stimulated activations for $\sigma_{st} = \sqrt{0.1}$ and $\eta_{st} \in \{0.01, 0.05, 0.2\}$

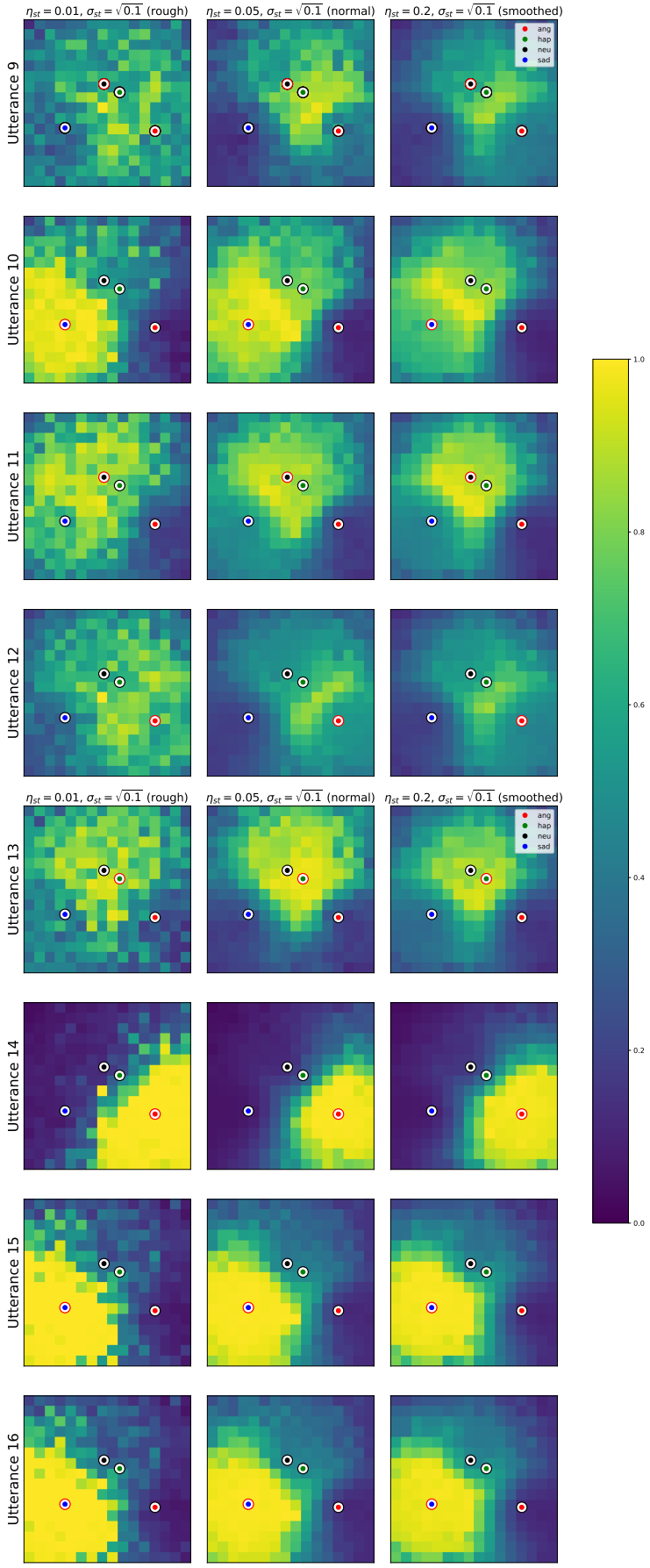


Figure A.8: Samples of stimulated activations for $\sigma_{st} = \sqrt{0.1}$ and $\eta_{st} \in \{0.01, 0.05, 0.2\}$

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*. [Cited in sections 4.4 and 5.2.2.]

Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., Bastien, F., Bayer, J., Belikov, A., Belopolsky, A., et al. (2016). Theano: A python framework for fast computation of mathematical expressions. *arXiv preprint*. [Cited in section 5.2.2.]

Arik, S. O., Chrzanowski, M., Coates, A., Diamos, G., Gibiansky, A., Kang, Y., Li, X., Miller, J., Raiman, J., Sengupta, S., et al. (2017). Deep voice: Real-time neural text-to-speech. *arXiv preprint arXiv:1702.07825*. [Cited in section 5.]

Barra-Chicote, R., Yamagishi, J., King, S., Montero, J. M., and Macias-Guarasa, J. (2010). Analysis of statistical parametric and unit selection speech synthesis systems applied to emotional speech. *Speech Communication*, 52(5):394–404. [Cited in section 2.4.]

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166. [Cited in section 4.2.1.]

Black, A. W. (2003). Unit selection and emotional speech. In *Interspeech*. [Cited in section 2.4.]

Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., and Narayanan, S. S. (2008). IEMOCAP: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335. [Cited in sections 1.2.1, 1.2.2, 2.2, 3.1, 3.1, 3.2, and 3.3.]

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698. [Cited in section 2.1.]

Caruana, R. (1998). Multitask learning. In *Learning to learn*, pages 95–133. Springer. [Cited in sections 2.3 and 4.3.2.]

Chen, N., Qian, Y., and Yu, K. (2015). Multi-task learning for text-dependent speaker verification. In *Sixteenth annual conference of the international speech communication association*. [Cited in section 4.3.2.]

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*. [Cited in section 4.2.1.]

Cortina, J. M. (1993). What is coefficient alpha? an examination of theory and applications. [Cited in sections 1.2.2 and 3.1.]

Cowie, R. and Cornelius, R. R. (2003). Describing the emotional states that are expressed in speech. *Speech communication*, 40(1):5–32. [Cited in section 1.2.1.]

Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., and Taylor, J. G. (2001). Emotion recognition in human-computer interaction. *IEEE Signal processing magazine*, 18(1):32–80. [Cited in section 2.2.]

Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4):357–366. [Cited in sections 2.1 and 4.1.3.]

Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., and Ouellet, P. (2011). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798. [Cited in section 2.4.]

Dhall, A., Goecke, R., Joshi, J., Hoey, J., and Gedeon, T. (2016). EmotiW 2016: Video and group-level emotion recognition challenges. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pages 427–432. ACM. [Cited in section 2.1.]

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. (2015). Long-term recurrent convolutional

networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634. [Cited in section 4.2.3.]

Douglas-Cowie, E., Campbell, N., Cowie, R., and Roach, P. (2003). Emotional speech: Towards a new generation of databases. *Speech communication*, 40(1):33–60. [Cited in sections 1.2.1 and 2.2.]

Douglas-Cowie, E., Cowie, R., and Schröder, M. (2000). A new emotion database: considerations, sources and scope. In *ISCA tutorial and research workshop (ITRW) on speech and emotion*. [Cited in section 2.2.]

Douglas-Cowie, E., Cowie, R., Sneddon, I., Cox, C., Lowry, O., Mcrorie, M., Martin, J.-C., Devillers, L., Abrilian, S., Batliner, A., et al. (2007). The humaine database: addressing the collection and annotation of naturalistic and induced emotional data. *Affective computing and intelligent interaction*, pages 488–500. [Cited in section 2.2.]

Efron, D. (1941). Gesture and environment. [Cited in section 1.1.2.]

Ekman, P. (1992). An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200. [Cited in section 2.2.]

Ekman, P., Friesen, W. V., O'sullivan, M., Chan, A., Diacoyanni-Tarlatzis, I., Heider, K., Krause, R., LeCompte, W. A., Pitcairn, T., Ricci-Bitti, P. E., et al. (1987). Universals and cultural differences in the judgments of facial expressions of emotion. *Journal of personality and social psychology*, 53(4):712. [Cited in sections 1.1.2 and 1.2.1.]

El Ayadi, M., Kamel, M. S., and Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572–587. [Cited in section 2.1.]

Eyben, F., Scherer, K. R., Schuller, B. W., Sundberg, J., André, E., Busso, C., Devillers, L. Y., Epps, J., Laukka, P., Narayanan, S. S., et al. (2016). The geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing. *IEEE Transactions on Affective Computing*, 7(2):190–202. [Cited in sections 1.1, 2.1, 2.1, 4.1.4, and 4.1.4.]

Eyben, F., Wöllmer, M., and Schuller, B. (2010). openSMILE: the Munich ver-

satile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1459–1462. ACM. [Cited in section 4.4.]

Fontaine, J. R., Scherer, K. R., Roesch, E. B., and Ellsworth, P. C. (2007). The world of emotions is not two-dimensional. *Psychological science*, 18(12):1050–1057. [Cited in sections 1.2.2 and 2.2.]

Gales, M. and Young, S. (2008). The application of hidden markov models in speech recognition. *Foundations and trends in signal processing*, 1(3):195–304. [Cited in section 2.3.]

Geyer, C. J. (2003). Stat 5102 notes: Nonparametric tests and confidence intervals. [Cited in section 6.3.2.2.]

Ghosh, S., Laksana, E., Morency, L.-P., and Scherer, S. (2015). Learning representations of affect from speech. *arXiv preprint arXiv:1511.04747*. [Cited in sections 2.1, 6.1.3, and 6.2.]

Ghosh, S., Laksana, E., Morency, L.-P., and Scherer, S. (2016). Representation learning for speech emotion recognition. In *INTERSPEECH*, pages 3603–3607. [Cited in section 2.1.]

Han, K., Yu, D., and Tashev, I. (2014). Speech emotion recognition using deep neural network and extreme learning machine. In *Fifteenth Annual Conference of the International Speech Communication Association*. [Cited in section 6.2.]

Hermansky, H. (1990). Perceptual linear predictive (plp) analysis of speech. *the Journal of the Acoustical Society of America*, 87(4):1738–1752. [Cited in section 4.1.4.]

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780. [Cited in section 4.2.1.]

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70. [Cited in section 6.3.2.2.]

Hoshen, Y., Weiss, R. J., and Wilson, K. W. (2015). Speech acoustic modeling from raw multichannel waveforms. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4624–4628. IEEE. [Cited in section 2.1.]

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95. [Cited in section 4.4.]

Jin, Q., Li, C., Chen, S., and Wu, H. (2015). Speech emotion recognition with acoustic and lexical features. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4749–4753. IEEE. [Cited in section 6.2.]

Jones, E., Oliphant, T., and Peterson, P. (2014). {SciPy}: open source scientific tools for {Python}. [Cited in section 4.4.]

Jordan, M. I. (1997). Serial order: A parallel distributed processing approach. *Advances in psychology*, 121:471–495. [Cited in section 4.2.1.]

Kapoor, A. and Picard, R. W. (2005). Multimodal affect recognition in learning environments. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 677–682. ACM. [Cited in section 1.2.1.]

Kim, Y., Lee, H., and Provost, E. M. (2013). Deep learning for robust feature generation in audiovisual emotion recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3687–3691. IEEE. [Cited in sections 1.2.1, 2.3, and 6.2.]

Kim, Y. and Provost, E. M. (2013). Emotion classification via utterance-level dynamics: A pattern-based approach to characterizing affective expressions. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3677–3681. IEEE. [Cited in section 6.2.]

King, S. and Karaiskos, V. (2016). The blizzard challenge 2016. [Cited in sections 3.2 and 5.1.]

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. [Cited in section 6.1.]

Kipp, M. (2001). Anvil-a generic annotation tool for multimodal dialogue. In *Seventh European Conference on Speech Communication and Technology*. [Cited in section 3.1.]

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J. B., Grout, J., Corlay, S., et al. (2016). Jupyter

Notebooks - a publishing format for reproducible computational workflows. In *ELPUB*, pages 87–90. [Cited in section 4.4.]

Kraft, S. and Zölzer, U. (2014). Beaqlejs: Html5 and javascript based framework for the subjective evaluation of audio quality. In *Linux Audio Conference, Karlsruhe, DE*. [Cited in section 6.3.2.1.]

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105. [Cited in section 2.1.]

Lazarus, R. S. (1991). *Emotion and adaptation*. Oxford University Press on Demand. [Cited in section 1.2.2.]

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324. [Cited in section 4.2.2.]

Lee, C.-C., Mower, E., Busso, C., Lee, S., and Narayanan, S. (2011). Emotion recognition using a hierarchical binary decision tree approach. *Speech Communication*, 53(9):1162–1171. [Cited in section 1.2.1.]

Lee, C. M., Yildirim, S., Bulut, M., Kazemzadeh, A., Busso, C., Deng, Z., Lee, S., and Narayanan, S. (2004). Emotion recognition based on phoneme classes. In *Interspeech*, pages 205–211. [Cited in sections 2.2 and 2.3.]

Lee, J. and Tashev, I. (2015). High-level feature representation using recurrent neural network for speech emotion recognition. In *INTERSPEECH*, pages 1537–1540. [Cited in sections 1.2.1 and 6.2.]

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605. [Cited in section 6.2.1.]

Makhoul, J. (1975). Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580. [Cited in section 4.1.4.]

Mao, Q., Dong, M., Huang, Z., and Zhan, Y. (2014). Learning salient features for speech emotion recognition using convolutional neural networks. *IEEE Transactions on Multimedia*, 16(8):2203–2213. [Cited in sections 2.1 and 4.2.3.]

Martin, O., Kotsia, I., Macq, B., and Pitas, I. (2006). The enterfaceâĂŹ05 audio-

visual emotion database. In *Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on*, pages 8–8. IEEE. [Cited in section 2.2.]

McKeown, G., Valstar, M., Cowie, R., Pantic, M., and Schroder, M. (2012). The semaine database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent. *IEEE Transactions on Affective Computing*, 3(1):5–17. [Cited in section 2.2.]

McKinney, W. (2011). Pandas: a foundational python library for data analysis and statistics. *Python for High Performance and Scientific Computing*, pages 1–9. [Cited in section 4.4.]

Mehrabian, A. et al. (1971). *Silent messages*, volume 8. Wadsworth Belmont, CA. [Cited in section 1.]

Mehri, S., Kumar, K., Gulrajani, I., Kumar, R., Jain, S., Sotelo, J., Courville, A., and Bengio, Y. (2016). Samplernn: An unconditional end-to-end neural audio generation model. *arXiv preprint arXiv:1612.07837*. [Cited in sections 5.2.3 and 7.1.]

Merritt, T., Clark, R. A., Wu, Z., Yamagishi, J., and King, S. (2016). Deep neural network-guided unit selection synthesis. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 5145–5149. IEEE. [Cited in section 5.1.]

Metallinou, A., Lee, S., and Narayanan, S. (2008). Audio-visual emotion recognition using gaussian mixture models for face and voice. In *Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium on*, pages 250–257. IEEE. [Cited in sections 2.3, 6.1.3, and 6.2.]

Metallinou, A., Lee, S., and Narayanan, S. (2010). Decision level combination of multiple modalities for recognition and analysis of emotional expression. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 2462–2465. IEEE. [Cited in sections 2.3 and 6.2.]

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. [Cited in section 6.2.]

Mohamed, A.-r. (2014). *Deep neural network acoustic models for asr*. PhD thesis. [Cited in section 4.1.3.]

Morise, M., Yokomori, F., and Ozawa, K. (2016). World: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE TRANSACTIONS on Information and Systems*, 99(7):1877–1884. [Cited in section 5.2.3.]

Mower, E., Mataric, M. J., and Narayanan, S. (2011). A framework for automatic human emotion classification using emotion profiles. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(5):1057–1070. [Cited in sections 1.2.1, 2.2, and 6.2.]

Neiberg, D., Elenius, K., and Laskowski, K. (2006). Emotion recognition in spontaneous speech using GMMs. In *Ninth International Conference on Spoken Language Processing*. [Cited in section 2.3.]

Neto, J., Almeida, L., Hochberg, M., Martins, C., Nunes, L., Renals, S., and Robinson, T. (1995). Speaker-adaptation for hybrid hmm-ann continuous speech recognition system. [Cited in section 2.4.]

Olah, C. (2015). Understanding LSTM networks. [Cited in section 4.4.]

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*. [Cited in section 5.2.]

Palaz, D., Collobert, R., and Doss, M. M. (2013). Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks. *arXiv preprint arXiv:1304.1018*. [Cited in section 2.1.]

Picard, R. W. and Picard, R. (1997). *Affective computing*, volume 252. MIT press Cambridge. [Cited in section 1.1.1.]

Plutchik, R. (1984). Emotions: A general psychoevolutionary theory. *Approaches to emotion*, 1984:197–219. [Cited in section 1.2.1.]

Poria, S., Cambria, E., Bajpai, R., and Hussain, A. (2017). A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion*, 37:98–125. [Cited in section 2.3.]

Poria, S., Chaturvedi, I., Cambria, E., and Hussain, A. (2016). Convolutional MKL based multimodal emotion recognition and sentiment analysis. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 439–448. IEEE. [Cited in sections 2.3, 6.1.3, and 6.2.]

Ringeval, F., Sonderegger, A., Sauer, J., and Lalanne, D. (2013). Introducing the RECOLA multimodal corpus of remote collaborative and affective interactions. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–8. IEEE. [Cited in section 2.2.]

Rozgic, V., Ananthakrishnan, S., Saleem, S., Kumar, R., and Prasad, R. (2012). Ensemble of SVM trees for multimodal emotion recognition. In *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, pages 1–4. IEEE. [Cited in sections 2.3, 6.1.3, and 6.2.]

Schröder, M. (2001). Emotional speech synthesis: A review. In *Seventh European Conference on Speech Communication and Technology*. Citeseer. [Cited in section 2.4.]

Schröder, M. (2009). Expressive speech synthesis: Past, present, and possible futures. *Affective information processing*, pages 111–126. [Cited in section 2.4.]

Schuller, B., Steidl, S., Batliner, A., Hantke, S., Hönig, F., Orozco-Arroyave, J. R., Nöth, E., Zhang, Y., and Weninger, F. (2015). The INTERSPEECH 2015 computational paralinguistics challenge: Nativeness, parkinson's & eating condition. In *Sixteenth Annual Conference of the International Speech Communication Association*. [Cited in section 2.1.]

Schuller, B. W., Steidl, S., Batliner, A., Hirschberg, J., Burgoon, J. K., Baird, A., Elkins, A. C., Zhang, Y., Coutinho, E., and Evanini, K. (2016). The INTERSPEECH 2016 computational paralinguistics challenge: Deception, sincerity & native language. In *INTERSPEECH*, pages 2001–2005. [Cited in sections 2.1 and 2.1.]

Series, B. (2014). Method for the subjective assessment of intermediate quality level of audio systems. *International Telecommunication Union Radiocommunication Assembly*. [Cited in section 6.3.2.1.]

Solera-Ureña, R., Padrell-Sendra, J., Martín-Iglesias, D., Gallardo-Antolín, A., Peláez-Moreno, C., and Díaz-de María, F. (2007). SVMs for automatic speech recognition: a survey. In *Progress in nonlinear speech processing*, pages 190–216. Springer. [Cited in section 2.3.]

Sotelo, J., Mehri, S., Kumar, K., Santos, J. F., Kastner, K., Courville, A., and Bengio, Y. (2017). Char2wav: End-to-end speech synthesis. [Cited in section 5.]

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2014). Striving

for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*. [Cited in section 4.2.2.]

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958. [Cited in section 4.3.1.]

Swietojanski, P. and Renals, S. (2014). Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 171–176. IEEE. [Cited in sections 2.4 and 5.3.]

Tan, S., Sim, K. C., and Gales, M. (2015). Improving the interpretability of deep neural networks with stimulated learning. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, pages 617–623. IEEE. [Cited in sections 4.3.4 and 6.2.]

Taylor, P., Black, A. W., and Caley, R. (1998). The architecture of the festival speech synthesis system. [Cited in section 5.1.]

Trigeorgis, G., Ringeval, F., Brueckner, R., Marchi, E., Nicolaou, M. A., Schuller, B., and Zafeiriou, S. (2016). Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 5200–5204. IEEE. [Cited in section 2.1.]

Valstar, M., Gratch, J., Schuller, B., Ringeval, F., Lalanne, D., Torres Torres, M., Scherer, S., Stratou, G., Cowie, R., and Pantic, M. (2016). AVEC 2016: Depression, mood, and emotion recognition workshop and challenge. In *Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge*, pages 3–10. ACM. [Cited in sections 2.1 and 2.1.]

Van Rossum, G. and Drake, F. L. (2003). *Python language reference manual*. Network Theory. [Cited in section 4.4.]

Vinciarelli, A., Pantic, M., and Bourlard, H. (2009). Social signal processing: Survey of an emerging domain. *Image and vision computing*, 27(12):1743–1759. [Cited in section 1.1.]

Walt, S. v. d., Colbert, S. C., and Varoquaux, G. (2011). The NumPy array: a

structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30. [Cited in section 4.4.]

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83. [Cited in section 6.3.2.2.]

Wöllmer, M., Metallinou, A., Eyben, F., Schuller, B., and Narayanan, S. S. (2010). Context-sensitive multimodal emotion recognition from speech and facial expression using bidirectional LSTM modeling. In *Eleventh Annual Conference of the International Speech Communication Association*. [Cited in section 2.3.]

Wu, C., Karanasou, P., Gales, M. J., and Sim, K. C. (2016a). Stimulated deep neural network for speech recognition. In *INTERSPEECH*, pages 400–404. [Cited in sections 4.3.4, 6.2.2, and 6.2.3.]

Wu, Z., Swietojanski, P., Veaux, C., Renals, S., and King, S. (2015). A study of speaker adaptation for dnn-based speech synthesis. In *INTERSPEECH*, pages 879–883. [Cited in sections 2.4 and 5.3.]

Wu, Z., Watts, O., and King, S. (2016b). Merlin: An open source neural network speech synthesis system. *Proc. SSW, Sunnyvale, USA*. [Cited in sections 5.2.2 and 6.3.]

Xia, R. and Liu, Y. (2015). Leveraging valence and activation information via multi-task learning for categorical emotion recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 5301–5305. IEEE. [Cited in sections 2.3 and 6.2.]

Yamagishi, J., Masuko, T., and Kobayashi, T. (2004). HMM-based expressive speech synthesis-towards TTS with arbitrary speaking styles and emotions. In *Proc. of Special Workshop in Maui (SWIM)*. [Cited in section 2.4.]

Young, S. J. and Young, S. (1993). *The HTK hidden Markov model toolkit: Design and philosophy*. University of Cambridge, Department of Engineering. [Cited in sections 2.3 and 5.2.1.]

Zen, H., Nose, T., Yamagishi, J., Sako, S., Masuko, T., Black, A. W., and Tokuda, K. (2007). The hmm-based speech synthesis system (hts) version 2.0. In *SSW*, pages 294–299. [Cited in section 5.2.1.]

Zen, H., Tokuda, K., and Black, A. W. (2009). Statistical parametric speech synthesis. *Speech Communication*, 51(11):1039–1064. [Cited in section 5.2.]