Automatic Detection of Laughter in Spontaneous Conversations

Gordon Rennie

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Engineering
College of Science and Engineering
University of Glasgow

September 2024

Abstract

Laughter is an important expression used to communicate in a variety of important ways. It is used to signal enjoyment and humour, to control and maintain the flow of conversations, to help mediate the discussion of controversial conversation topics and is used to help speakers bond. Given laughter's wide range of uses, if they are to engage in effective human-computer interactions, it is vital for computers to be able to detect laughter. However, laughter is not homogeneous. There are two broad types of laughter: voiced and unvoiced. In addition, many individuals have different and unique ways of laughing. The pitch, volume, length and frequency of laughter has a wide divergence across speakers. Furthermore, it is used infrequently. These factors make the application of machine learning approaches, to the automatic detection of laughter, difficult.

This thesis initially shows, through a literature review, that the task of laughter detection has been widely addressed previously. However, the field has placed constraints upon the task of laughter detection. These constraints are shown to split the field into three broad tasks. Type 1 classification tasks involve short clips of between 1 and 3 seconds and contain only one kind of speech event (i.e., laughter, speech, sighs or fillers) being classified. Type 2 tasks make use of medium length clips of between 3 and 11 seconds. Each clip of this type contains multiple speech events: however, laughter can constitute a large amount of the total audio of each clip, i.e., between 10-30%. Finally, type 3 tasks employ long form conversations that are between 10 minutes to an hour. Laughter makes up less than 10% of the audio in this case: there is no guarantee of any laughter being present. Initially, it is shown that these three types of tasks vary in difficulty. Evidence of this is given by examining the F1 score achieved by the same methodology when applied to the three tasks. Scores vary from 80-100% in type 1 tasks and 50% in type 2 tasks to 25% in type 3 tasks. Furthermore, a disparity in the effectiveness of laughter detection methods, as estimated by different evaluation metrics, is found. This is shown to lead to an over-estimation of the effectiveness of state-of-the-art methods in types 2 and 3 laughter detection tasks.

This thesis replicates the state-of-the-art research on a publicly available type 2 corpus, achieving a frame level F1 of 40% and an event level F1 of 52%. It then applies these methods to the SSPNet Mobile Corpus, a private type 3 dataset, and shows the same methods achieve a frame level F1 of 15% and an event level F1 of 26%. An extensive performance analysis illus-

trates that the longer length of the audio introduces a large number of false laughter detections that are centred on speech. It is then demonstrated that methods that specifically target the removal of these false detections, by leveraging automatic speech recognition, are able to achieve a frame level F1 of 30% and an event level F1 of 45%. This enables an almost two-fold increase in performance over the state-of-the-art approaches for type 3 tasks.

Transformers are then applied to the task. It is demonstrated that these transformers, pre-trained on audio tasks such as automatic speech recognition, can be used to extract attention embeddings in terms of low level descriptors of the audio data. Such embeddings are shown to be more effective than hand-crafted features for training laughter detectors. This method is then demonstrated to achieve a frame level F1 of 60% and an event level F1 of 80%, i.e., the best results achieved in type 3 laughter detection. The effectiveness of this approach is then replicated on the SSPNet Vocalisation corpus, which achieves a frame level F1 of 77% and an event level F1 of 88%. Furthermore, it is shown to be as effective at the task of automatic filler detection by achieving a frame level F1 of 70% and an event level of 80%. The final section applies a selection of the laughter detection systems to detect differences in laughter behaviour due to the gender composition of the speakers in a conversation. This demonstrates an initial use-case of automatic speaker information extraction. Overall, this thesis accomplishes effective laughter detection in a type 3 task.

Contents

List of Tables

List of Figures

Acronyms

ALISP Automatic Language Independent Speech Processing Model.

AMI AMI meeting (subset of 7 meetings).

ASR Automatic Speech Recognition.

AUC Area Under the (receiver operator) Curve.

BEA BEA Hungarian spoken language database (conversation sub-set).

Bi-LSTM Bi-directional Long Short-Term Memory Neural Network.

C Confidence Based Alteration.

CBA Confidence Based Alteration System.

CNN Convolutional Neural Network.

CTC Connectionist Temporal Classification.

CW Class Weight.

D Delta.

E Feature Vector Extension.

ESM Echo State Network.

FF Female-Female Conversation Pairing.

FFN Feedforward Neural Network.

FRR False Rejection Rate.

GMM Gaussian Mixture Model.

HMM Hidden Markov Model.

HNR Harmonic Noise Ratio.

IEMOCAP Interactive Emotional Dyadic Motion Capture Database.

IFA IFA corpus (sub-set).

LSTM Long Short-Term Memory Neural Network.

MBR Minimum Bayes Risk.

MF Male-Female Conversation Pairing.

MFCC Mel-Frequency Cepstral Coefficient.

MFSB Mel-Scale Filter Bank.

MM Male-Male Conversation Pairing.

MOD Modulation Spectrum.

NDC-ME Nonverbal Dyadic Conversation on Moral Emotions (NDC-ME) (sub-set).

NE No Effect.

NR Not Reported.

NVC Non-Verbal Communication.

PLP Perceptual Linear Prediction.

RF Random Forest.

ROC Receiver Operator Characteristic.

S Smoothing.

SMC SSPNet Mobile Corpus.

SOTA State-Of-The-Art.

STD Standard Deviation.

SVC SSPNet Vocalisation Corpus.

SVCFS baseline SVC feature set from [1] 12 MFCCs, energy (1st and 2nd order derivatives), voicing probability, HNR, F0 and zero crossing rate (deltas, mean and std).

SVM Support Vector Machine.

U Undersampling.

VAD Voice Activity Detection.

VC Verbal Communication.

Acknowledgements

I would not have been able to complete this Ph.D. without the help of a great many people. I would first like to thank my supervisors Alessandro Vinciarelli and Olga Perepelkina. Their guidance, advice and insight has been invaluable throughout my studies.

I owe thanks to my colleagues in the Social AI CDT for their support and camaraderie. Special thanks to Tobias Thejll-Madsen, Morgan Bailey, Andreas Drakopoulos, Thomas Goodge, Casper Hyllested and Mary Roth. Without their help I would not have been able to complete this Ph.D. With their help the process has often been a joy.

To my family and friends for their unwavering belief in me. They have always encouraged me to become the best version of myself. To my mother in particular for her help and care.

Finally, I owe an unending debt of gratitude to my partner Ellen, who kept my passion for science burning when the days were long.

For Ellen

Declaration

All work in this thesis was carried out by the author unless otherwise explicitly stated. Parts of the contents of Chapter 4 were published in:

G. Rennie, O. Perepelkina, and A. Vinciarelli, “Which Model is Best: Comparing Methods and Metrics for Automatic Laughter Detection in a Naturalistic Conversational Dataset," in *Proc. Interspeech 2022, 2022,* pp. 4008-4012.

Chapter 1

Introduction

1.1 Motivation

The words used during person-to-person communication represent only a part of what is being said [2]. The way the words are said, the speaker's body language and use of paralanguage all contribute to the listener's understanding. Laughter specifically has been shown to serve various purposes such as nullifying a previous statement and helping control the flow of conversation [3], manage delicate conversation topics such as divergences in opinion [4] and can be used in meetings to create collegiality or end a discussion on a topic [5]. For computational agents to effectively communicate with users, it is vital that they have the ability to understand and interpret these other channels of communication [2]. Furthermore, laughter can be an important marker for health and well-being. It has already been demonstrated that there are detectable differences in the laughter of healthy control subjects and those with depression [6, 7], autism [8], brain damage [9] and Parkinson's disease [10]. Meaning that laughter detection and analysis could aid in the detection of these disorders.

If computers are to fully engage in effective communication with humans, it is necessary for them to have an understanding of paralanguage due to its central role in human-to-human communication. This issue has long been recognised in the field of human-computer interaction and strides towards developing tools for computers to understand paralanguage have already been made. The INTERSPEECH challenge exemplifies this quickly growing field of research with tasks involving recognising age, gender and affect in 2010 [11]; social signals, conflict, emotion and autism in 2013 [1]; and atypical and self-assessed affect, crying and heart beats in 2018 [12], to mention only a few. This thesis intends to contribute to this field by developing methods to improve the efficacy of laughter detection systems. Moreover, it is planned to test and extend these methods into other forms of paralanguage; namely, filler and back-channel cues. Finally, laughter detection systems will be tested for their ability to identify common speaker traits, as an early validation of their ability to extract meaningful information from speech.

1.2 Thesis Statement

This thesis states that current State-Of-The-Art (SOTA) laughter detection has focused on constrained tasks where the ratio of laughter to non-laughter is artificially high and the total amount of audio is low compared to natural spontaneous conversations. It is demonstrated that the methods developed in the field are ineffective when those experimental constraints are removed. It further shows that the standard metrics used in the field are not fit for purpose in this less constrained environment. It then investigates and states that pre/post processing, especially using linguistic information, can be leveraged to improve laughter detection but that these methods are limited in their effectiveness. Finally, it is demonstrated that pre-trained transformer attention embeddings are effective representations of audio for the task of laughter detection. It is shown that this methodology quadruples the event level F1 performance of the best-performing SOTA laughter detection systems.

1.3 Research Questions

The thesis is organised a set of research questions. These are initially created by gaps in the literature. Each question then follows from issues discovered with each solution investigated. In Section 2.3 the literature review shows clearly that the field of laughter detection can be split into three different task types of increasing difficulty. Further, it is shown that the most difficult task has been under-addressed in the field although initial attempts show poor experimental results. This led to the creation of the first research question:

RQ1: **Are State-Of-The-Art laughter detectors effective when common experimental constraints are removed?**

It is then shown in Chapter 4 that when applied to this difficult task the methods developed in the field show relatively poor performance. A performance analysis demonstrates that the largest issue facing laughter detectors is that of false positives caused by speech. A research question was then created to address these issues:

RQ2: **Can the incorporation of linguistic data lead to improvements in laughter detection?**

Further to the above a second research question was created from the results garnered from RQ1. It was hypothesised that it would be easier for detectors to differentiate between verbal and non-verbal communication and then differentiate different types of non-verbal communication (such as laughter, fillers, and back-channel) leading to overall better laughter detection results. The following research question was therefore created to address this idea:

RQ3: **What is the effect of broadening the scope of laughter detectors to include multiple cues?**

Finally, the results from RQ2 and RQ3 were shown to improve laughter detection but to

only achieve F1 scores of around 50%. To further improve these results a new neural network architecture is investigated. Transformer based architectures have been shown to be highly effective in various machine learning tasks. They were therefore selected as a potential route for improving laughter detection and thus the final research question:

RQ4: **Are transformers effective when applied to the task of laughter detection?**

1.4 Contributions

The main contributions of this thesis are as follows. Initially, a literature review (Chapter 2) is carried out, which determines key issues in relation to the field of paralanguage detection. Firstly, that common constraints are placed upon the task, such as clip length and ratio of laughter to non-laughter and that when these constraints are removed the task becomes more difficult. Secondly, that Area Under the (receiver operator) Curve (AUC) is a widely used metric, which overestimates the effectiveness of detection systems due to the data imbalance inherit in laughter detection tasks. Thirdly, that laughter detection is generally carried out in isolation of other speech events. Fourth, the field has not utilised transformers as an architecture to create laughter detectors. The literature review demonstrated that SOTA methods achieve an F1 of 30% on laughter detection in spontaneous, naturalistic conversations. This thesis develops approaches to address this task. It then extends these methods to other forms of paralanguage, namely fillers and back-channel, to test the generalisability of the methods. Finally, it presents an initial use-case for laughter detectors demonstrating their utility.

Chapter 4 replicates SOTA detection methods on a publicly available corpus. These methods are then applied to a less constrained version of the task and are shown to reach performance ceilings of 15-30% in F1. A performance analysis demonstrates that the cause for these performance issues is centred on the following issues. Firstly, that speech is the most common cause of mistaken detections, even when controlling for its proportion of the data. Secondly, that the SOTA pre/post-processing methods are effective at improving precision or recall but not both.

In Chapter 5, methods are developed that directly address the issues identified in the previous chapter's performance analysis. The false detections caused by speech are addressed by using ASR to filter the output of the laughter detectors. Specifically, the estimated minimum Bayes risk (MBR) of a word is used to filter laughter detection results. This method is shown to double the performance of the SOTA laughter detection systems, which demonstrates that by including data about other speech events laughter detection can be improved. These methods currently achieve a F1 performance of 50%. This presents a promising route for future research. However, to further improve laughter detection using this approach would require improving the ASR systems, which is outside the scope of this thesis.

Instead, in Chapter 6, transformers are explored as an underlying architecture for laughter detection. Transformers were investigated as they have been effective at other difficult detection

tasks but have only been applied to laughter classification to date. It is shown that transformers, which are pre-trained on other related tasks, can be used as feature extractors. That the attention embeddings created by these transformers are effective representations of the audio for the task of laughter detection. Using these embeddings leads to an almost four-fold increase in performance over the SOTA methods. This approach is also validated on a publicly available dataset to enable comparison with the rest of the field.

The limits of this approach are then explored by extending the detectors to include other forms of paralanguage. The approach is demonstrated to also be effective at filler detection but not back-channel. Finally, an initial use case for laughter detection is presented. The laughter detection systems are shown to be able to reliably detect the differences, in terms of the total amount and frequency of laughter, in laughter behaviour due to the gender composition of the conversation.

1.5 Organisation of Thesis

The remainder of the thesis is organised as follows:

Chapter 2 provides an overview of SOTA in the laughter detection field. The detection methods used in the literature to date are explored. Moreover, the constraints placed upon the task are explained, leading to the development of a three-way split taxonomy. The datasets that have been developed are also explored before an examination of the evaluation methods used, and their relative strengths and weaknesses.

Chapter 3 details the two datasets used throughout this work. Their relative size and quality are compared with others in the field. Chapter 4 presents a replication of the SOTA detection systems in the laughter detection field. Initial results are given for performance on a publicly available corpus, showing that they achieve SOTA performance. Following this task, the same systems are applied to a less constrained task. It is shown that, on this more difficult task, the SOTA methods do not reliably detect laughter. A performance analysis is additionally presented that investigates why these methods fail and possible routes for improvement.

In Chapter 5, a series of novel detection systems are presented that address the issues identified in the performance analysis from the previous chapter. These systems focus on using the output from an ASR system to remove false positives. Furthermore, a set of systems are presented that carry out a multi-class classification. The goal of these systems is to address the class imbalance issue. The best of the resulting methods, which doubles the effectiveness of the SOTA methods, is again analysed using a performance analysis to understand its shortcomings. It is demonstrated that improvements in the ASR system would further improve laughter detection, however, this is outside the scope of this thesis. Instead of continuing to develop pre/post processing methods, different underlying neural networks architectures were tested.

In Chapter 6, the possibility of using transformers as an underlying architecture for laughter

detection is explored. Novel neural network architectures, such as long short-term memory networks, are effective at laughter detection, particularly when those systems included context of how the audio changes across time. Transformers have only been used in laughter classification tasks to date but have seen success in other detection tasks. Laughter detection transformers are trained but due to limitations in the amount of data available, this approach did not improve on SOTA methods. It is then demonstrated that transformers, pre-trained on other related audio tasks, can be leveraged through transfer learning to reduce the amount of data needed. This transfer learning system is then shown to be the most effective system found to date. This system is then extended to filler and back-channel detection. This work tests the method's generalisability to other paralanguage cues. Furthermore, its ability to detect commonly found differences in laughter behaviour present in gender pairings is shown. This demonstrates an initial use-case where laughter detection can be used to automatically extract speaker information. Finally, Chapter 7 presents the conclusions of this work and potential routes for future work.

1.6 Publications

The research presented in Chapter 4 was first presented in the following publication:

G. Rennie, O. Perepelkina, and A. Vinciarelli, "Which Model is Best: Comparing Methods and Metrics for Automatic Laughter Detection in a Naturalistic Conversational Dataset," in *Proc. Interspeech 2022, 2022,* pp. 4008-4012.

Chapter 2

State-of-the-Art Laughter Detection

This section describes previous work undertaken in the field of laughter detection. Section 2.1 defines laughter. Section 2.2 describes the detection methods used including feature extraction, classification and post-processing methods. Section 2.3 outlines the three main task definitions. Section 2.4 provides an overview of the datasets used. Finally, Section 2.5 explores the evaluation metrics and methodologies used.

2.1 What is Laughter

Laughter can be broadly split into two types: voiced and unvoiced. Voiced laughter is typified as having a 'tonal, song-like quality' and 'evident periodicity' of distinct bouts of sound [8]. Voiced laughter can be thought of as classical laughter. Unvoiced laughter, however, is characterised as 'noisy exhalation through nose or mouth and the vocal folds are not involved in laughter production' [13]. Furthermore, unvoiced laughter can be split into two groups, i.e., whether the sound is emitted through the nose (unvoiced snort-laughter) or mouth (unvoiced grunt-like laughter) [14]. These distinctions need to be emphasised due to the different role each type of laughter plays in communication. It has been demonstrated that voiced laughter elicits positive reciprocal emotions in listeners, whereas unvoiced laughter does not [15, 16]. These differing definitions and characteristics offer evidence as to why laughter is such a challenging phenomenon to detect. Laughter's variability in use, as described above, coupled with this multi-faceted characterisation shows why it is an important part of human communication and why its detection is a useful ability for computers to have.

2.2 Detection Approaches

Independent of the type of task, laughter detection is generally approached by carrying out the three steps that are displayed in Figure 2.1. First, the raw audio signal is converted into a series of frames S_f in the feature extraction step. Second, these frames are then classified using a frame

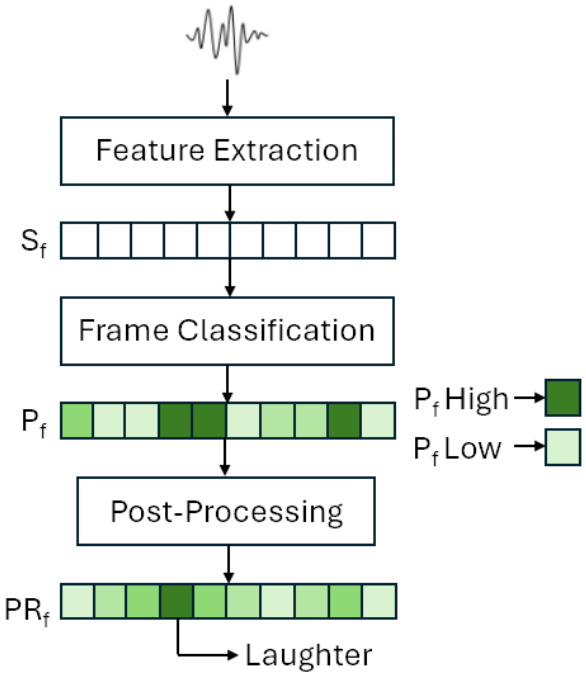

Figure 2.1: General Form of a Laughter Detection System

classifier to produce a series of estimates P_f that a frame belongs to the 'class laughter'. Third, these classifications can then be altered using a post-processing step to produce a sequence of refined estimations PR_f before a final classification is produced.

There are three ways in which features can be extracted from the audio signal: hand-crafted features, spectrograms and transformer embeddings. In all cases, feature extraction begins with the splitting of the audio input into a series of analysis windows. As can be seen from Tables 2.1 and 2.2, these windows have time lengths of 20 to 1220 ms with a hop of 10 to 400 ms. Each of these analysis windows then has a series of features extracted from it. In the case of hand-crafted feature extraction, pre-defined descriptors of sound are used to represent the audio segment. Tables 2.1 and 2.2 show the features used by each laughter detection study; common choices include MFCCs, pitch, intensity, F1 etc. These features each represent an aspect of the audio signal and enable analysis of the importance of each feature to the classification of a particular event. For example, ref. [17] carried out a principal component analysis and found that laughter events can be classified as polite or sincere based on the relative power of the laughter.

A further process by which features can be extracted from audio is through the use of spectrograms. This process can be used to visually represent the amplitude of the audio signals at variable frequencies over time. This type of representation enables the application of machine learning techniques from the field of image recognition [18]. In previous work, it has been shown that spectrograms can be effectively employed in the field of laughter detection [19], however, they do not see any improvement over the hand-crafted features described above.

The third and final approach is that of transformer embeddings extraction. Transformer

models have been used effectively in other audio tasks such as speech recognition [20], emotion recognition [21] and speaker diarization [22]. Transformers are, in the general case, made up of an encoder and a decoder. For the purpose of embedding extraction, only the encoder side is required. Transformers take as input windows either the raw audio signal or its spectrogram. These windows then go through a process of input embedding, followed by positional encoding and then multi-head attention embedding (for full details of transformer operations, see Section 6.1).

To use transformers for laughter detection, a novel transformer could be initialised and trained to classify individual laughter frames. However, transformer networks are generally larger than other machine learning approaches. This means they have more trainable parameters and, consequently, often need large datasets to be trained effectively [23]. Due to the size of the dataset and the number of trainable parameters, they also require large amounts of computational power. A different option, which eliminates these issues, is to apply transfer learning [24]. The latter is a process where pre-trained transformers are leveraged to create feature vectors. In this case, the embeddings output (after the attention step) are extracted from the transformers trained on similar tasks. These are then used as input into a new neural network, which can be trained on the target task - in this case, laughter detection.

Following the feature extraction step, a series of frames S_f are produced. In the case of hand-crafted features and spectrograms, each frame is represented without taking temporal information into account. This presents an issue in the case of laughter detection since even the shortest laughter event spans multiple frames and can contain many different audio markers. For example, it has been shown that laughs can be composed of loud vocalisations separated by periods of silence [25]. As such, extending the representation of frames to include information from those frames surrounding the target frame is often included. This can be undertaken through methods such as frame concatenation, where multiple frames are stacked. These frames can originate from before or after the target frame. There is no accepted number of frames that should be used in such a case; some studies have shown as few as 10 frames can be effective [26, 27], while others have tested over 20 [26, 28]. A second method for including this information could be the calculation of feature values over multiple frames [29].

Table 2.1: Part 1. Summation of Laughter Detection Studies. MFSB: mel-scale filter bank. PLP: perceptual linear prediction. MOD: modulation spectrum. MFCC: mel-frequency cepstral coefficients. HNR: harmonic noise ratio. HMM: hidden Markov model. FFN: feedforward neural network. SVM: support vector machine. CNN: convolutional neural nets. LSTM: long short-term memory recurrent neural network. Bi-LSTM: bidirectional LSTM. GMM: Gaussian mixture models. ESN: echo state network. RF: random forest. ALISP: automatic language independent speech processing models. BEA: BEA Hungarian spoken language database (conversation sub-set). AMI: AMI meeting (subset of 7 meetings). NDC-ME: nonverbal dyadic conversation on moral emotions (NDC-ME) (sub-set). IFADV: IFA corpus (sub-set). IEMOCAP: interactive emotional dyadic motion capture database. SVCFS: baseline SVC feature set from [1] 12 MFCCs, energy (1st and 2nd order derivatives), voicing probability, HNR, F0 and zero crossing rate (deltas, mean and std)

Study	Window Size (Time Step) ms	Input Vector Size	Audio Markers	Dataset	Architecture	Pre/Post-Processing	Task Type	Precision	Recall	F1	AUC (%)
[30]	25 (12.5)	16	8 MFCCs, Energy, ZCR	Unnamed [30]	HMM	Deltas	2	79.1	95.9	86.700	NR
[31]	40 (20)	18	7 PLP, Pitch, RMS	AMI	FFN	Delta	2	NR	NR	53-68	76-88
[32]	40 (20)	14	7 PLP	AMI	FFN	Delta	1			68-76	
[33]	NR	41	13 MFCCs, Log-Energy	ICSI	HMM	Derivative	3	36-55	25-27	31-34	NR
[34]	200 (20)	8	Mel Filter Bank	FreeTalk	ESN	Exclusion of Speech Laugh Overlap	2	Average Error Rate 13%			
[25]	100 (100)		13 MFCCs, Log Energy	ICSI	HMM	Removal of Unvoiced Laughter	3	NR	NR	26-32	NR
[35]	NR	NR	PLP, MOD	Freetalk	HMM	None	1	NR	NR	24.9 - 62.6	NR
[36]	200 (180)	8	Mel Filter Bank	FreeTalk, AVIC	SVM + ESN + HMM	Delta	2	64 ± 10	80 ± 18	49 - 72 $\pm$ 18	NR

Table 2.2: Part 2. Summation of Laughter Detection Studies

Study	Window Size (Time Step) ms	Input Vector Size	Audio Markers	Dataset	Architecture	Pre/Post-Processing	Task Type	Precision	Recall	F1	AUC (%)
[37]	NR	NR	MFCCs	SEMAINE-DB, MAHNOB	ALISP	Transfer Learning	1	90	64	75	NR
[38]	20 (10)	141	SVCFS	SVC	FFN	Two-Stage Stacked FFNs	2	NR	NR	NR	97
[1]	20 (10)	141	SVCFS	SVC	SVM	Deltas, Derivatives	2	NR	NR	NR	83
[39]	25 (10)		12 MFCC, PLP, F0, Jitter, RMS, HNR, ZCR, Spectral Slope, LPC-CoF (Center of Gravity)	BEA	GMM + SVM	Derivative	1	72-100	93-96	81-98	NR
[40]	40 (10)		6 MFCCs, ZCR	MAHNOB	FFN	Majority Voting, Undersampling	1	NR	NR	84.7	NR
[29]	25-40 (10)	NR	39 MFCCs, 8 Pitch, Intensity, 8 Formants, 28 Voice Quality	SVC	GMM	Median Filter, Delta, Mean	2	Equal Error Rate 9.3%			
[41]	25 (10)	39	13 MFCCs, 26 MSFB	SVC	HMM	None	2	35-67	61-80	48-63	NR
[42]	20 (10)	141	SVCFS	SVC	FFN	Multiple Post Processing Filters	2	NR	NR	NR	95
[43]	25 (10)	141	SVCFS	SVC	FFN + Bi-LSTM	Derivative	2	NR	NR	NR	94

Table 2.3: Part 3. Summation of Laughter Detection Studies

Study	Window Size (Time Step) ms	Input Vector Size	Audio Markers	Dataset	Architecture	Pre/Post-Processing	Task Type	Precision	Recall	F1	AUC (%)
[27]	NR	141-182	SVCFS	Switchboard Fisher, UT-Opinion	FFN and CNN	Low Pass Filter, Frame Concatination	3	NR	NR	NR	83- 97
[44]	30 (10)	168	Energy Contour	MAHNOB	RF	Derivative, Median Filter Over Input Features	1	92	93	NR	NR
[45]	NR	301	ZCR, Spectral Slope + Flatness, Specific Loudness, Pitch, Formants, Prosody	ARMEN, ROMEO2, OFFICE, JEMO	SVM	Delta, Mean, SD	1	NR	59.2-87.2	NR	NR
[46]	20 (10)		13 MFCCs	ROMEO2	SVM	Majority Voting	1	54	80.1	NR	NR
[47]	40-100 (10)	10	MFCCs, Pitch, Jitter	MAHNOB	SVM	Mean, SD	2	NR	NR	90.1	NR
[48]	20 (10)	141	SVCFS	SVC	FFN	Smoothing, Frame Concatenation	2	NR	NR	NR	95
[26]	NR	65	12 MFCC, 12 PLP, 40 MSFB, RMS Energy	SVC and BEA	FFN	Deltas, Derivatives, Frame Concatenation, Undersampling	2	64-100	89-100	73-100	NR
[49]	25 (10)	32	12 MFCC, Energy, Speech Intensity, Pitch	IEMOCAP	FFN + SVM	Undersampling, Bagging	3	15	41	22	88

Table 2.4: Part 4. Summation of Laughter Detection Studies

Study	Window Size (Time Step) ms	Input Vector Size	Audio Markers	Dataset	Architecture	Pre/Post-Processing	Task Type	Precision	Recall	F1	AUC (%)
[50]	20 (10)	140	50 Mel-Spectra, 20 MFCCs	SVC	LSTM	Delta	2	NR	NR	54.31	92
[51]	NR (10)	123	40 MSFB, Log-Energy	SVC, ERATO	Bi-LSTM-CTC	Delta and Derivative	2	65-89	35-66	50-66	NR
[28]	NR (10)	120	40 Mel Filter Bank Energies	BEA	GMM + FFN	Upsampling, Derivatives, Frame Concatenation	1 + 2	NR	NR	15.3	NR
[19]	1000 (NR)	128	Spectrograms, MFCC	Switchboard and Audioset	FFN + ResNet	Delta	2	30-67	70-84	43-75	NR
[52]	1220 (400)	NR	NR	NDC-ME, IFADV	CNN	Transfer Learning	2	34-49	36-47	35.55-48.45	NR
[53]	20	NR	Wav2Vec2 and Whisper Embeddings	Hume-VB	Transformer	Transfer Learning	1	NR	NR	78-91	94

2.3 Task Definition

Laughter detection work can be split into three sub-tasks with each task representing an increase in difficulty. Type 1 tasks take audio clips of stand-alone audio events (such as laughter, speech, filler and pause). Depending on the overall goal of the process, they then either classify those clips into multiple classes [53] or distinguish one type of audio event from all the others [40]. The length of clips used in type 1 tasks can vary from 1.5 s [32] to 2.88 s [44] and contain either 100% laughter or not. Type 2 tasks use longer audio clips (for example, 11 s [41] or 3-10 s [30]), which can contain multiple possible audio events. These clips are guaranteed to contain at least one target audio event. The laughter can comprise 10% [41] to 30% [31] or as high as 39% [19] of the audio of each clip. In type 2 tasks, the goal is to identify where the audio event occurs in the clip via a detection type task. This can be done by identifying a point in time in which the audio event is expected to occur, the start and end points of the audio event or by classifying the event using a frame-by-frame approach in which each frame is assigned a class. The analysis for Type 3 case is similar to type 2, with the difference being a further relaxing of the constraints placed on the task. Type 3 tasks use recordings that can last minutes or hours. There is no guarantee of the frequency or distribution of the target audio events. In terms of the percentage of data that is composed of laughter, it can vary from 8% in the BEA Hungarian spoken language database (conversation sub-set) [26] to 1.3% in IEMOCAP [49]. The three goals that apply to type 3 tasks are identical to the ones for type 2. It is important to distinguish the different task types since results found in one task do not generalise to others. The type of task a particular study addresses can be determined by the underlying dataset, the ratio of laughter to non-laughter and the pre/post-processing used. A summary of laughter detection studies, alongside their associated task definition, are shown in Tables 2.1, 2.2, 2.3 and 2.4.

Previous results obtained from Type 1 tasks suggest that the detection of laughter has already been successfully achieved. One study used the MANHOB database, a collection of posed and elicited laughter clips [40]. In this study, the events were pre-segmented, meaning that there is a defined start and end point for each audio event. This enables a majority rule classification approach to be used. As outlined in Section 2.2, each audio clip was split into a series of frames, each one receiving a classification of laughter or not. The majority rule then takes an average of all the frames classifications to create one final class decision for each clip. The final results are reported in regard to this final overarching classification decision, meaning that, if 49% of an audio clip's frames were misclassified, the results would still report that laugh had been correctly detected. In this study, an F1 of 84.7% was reported. However, as stated, using the majority rule could lead to an inflation of these results. A similar approach was used in conjunction with transformers, where pre-trained audio classification transformers were fine-tuned to classify clips of different types of paralanguage such as sighs, gasps and laughs [53]. Each clip was between 2 and 10 s and was classified by passing the entire clip to the transformer, which then generated one label per clip. As with the previous study, the results here were

between 78-91% F1 and 94% AUC across all classes. However, it is doubtful that these results from a type 1 task could be maintained when addressing either a type 2 or 3 task. There are consistent and detectable morphology differences between posed laughter (used in ref. [53]) and spontaneous laughter [54]. Furthermore, neither of the above studies have datasets with distributions of laughter (and other paralinguistic vocalisations) typical of spontaneous speech [55]. The effect of this class distribution issue is clearly shown by three studies [26, 28, 39]. In these studies, the goal was to detect laughter segments. All the studies used the BEA Hungarian spoken language database (conversation sub-set) dataset. In the first and second study, the non-target class was down-sampled to achieve balanced datasets with 332 laughter clips and 331 speech clips. Using this balanced dataset, F1 scores from 91 to 98% [39] and 70-100% [26] were achieved depending on the feature extracting methods used. However, in the third study where there was no down-sampling laughter, laughter made up only 8% of the dataset. In this study, F1 results of 15% were achieved [28]. This difference of between 50-80% in F1 clearly shows the advantages of simpler type 1 tasks with respect to the previous studies and the non-generalisability of these results.

Type 2 tasks broaden the scope of possibilities compared with the above type 1 tasks. Type 2 tasks can be distinguished from type 1 tasks by the underlying data. The data is made up of longer clips of audio with multiple different audio events present. However, type 2 clips remain short (between 2 and 11 s, as outlined above) compared with the length of average conversations (for instance, 12 min and 42 s [55] and $\sim$1 h [33]). An example of this kind of type 2 task dataset is the SSPNet vocalisation corpus (SVC), which comprises 2763 audio clips that are 11 s long, each of which contains at least one laugh or filled pause event. Type 2 tasks represent less constrained tasks compared with type 1 due to the fact that the event boundaries and the number of audio events in the clip are both unknown. However, type 2 tasks still have the advantage of a more balanced class distribution than type 3 tasks. Type 2 tasks were popularised due to the 2013 INTERSPEECH paralinguistic challenge [1]. This challenge introduced the SVC to the field and provided baseline detectors. As can be seen from Tables 2.1, 2.2 2.3 and 2.4, multiple studies have been carried out on the SVC. Results vary with area under the (receiver operator characteristic) curve (AUC) from 83% to 97%, with a variety of deep learning architectures being tested. These results would suggest that although type 2 tasks may be more difficult than type 1, research produced in this field is still able to effectively detect laughter. However, these results contrast with the F1 scores of 15% reported above [28]. This disagreement can be found within the same work, with one study reporting an AUC of 92% alongside an F1 score of 54% [50] when detecting laughter in the SVC. The discrepancy between these results is explained in Section 4.2.8. In spite of the disagreement, the F1 scores show a clear difference in performance from type 1 to type 2.

Finally, type 3 tasks involve detecting laughter in long recordings. In type 3 tasks, there are no known event boundaries and the percentage of audio that is laughter is lower than in type 2

tasks, as described above. Furthermore, there may or may not be laughter, or other cues, present in the data. Type 3 tasks have been previously addressed in the literature [25, 27, 49]. However, these studies have some limitations. In ref. [49], the corpus used is the interactive emotional dyadic motion capture database (IEMOCAP) (for full details, see Tables 2.5, 2.6 and 2.7), which is comprised of 5 conversations. These conversations are both scripted and spontaneous interactions between 2 actors, leading to a total of 10 speakers. This is a relatively small number of speakers, leading to a lack of generalisability in the results. Moreover, the scripted/posed parts of the interaction may lower the quality of the laughter audio within the corpus, as discussed above. In addition, the results showed mixed performance with an AUC of 88% and a precision, recall and F1 of 15.8%, 40.9% and 22.8%, respectively. Again, the discrepancy in AUC and F1 is fully explained in Section 4.2.8. Similar AUC scores were seen elsewhere for type 3 tasks, with one study reporting AUC scores between 85-95% [27]. These AUC results would suggest a similar level of performance from type 1, 2 and 3 tasks. Furthermore, when examining F1, similar levels of performance for type 3 tasks are seen in other studies. In two related studies, F1 performance was shown to vary from 21% [33] to between 25-30% [25]. In the latter study, the authors reclassified unvoiced laughter so that it was not included in the overarching laughter class, which may have contributed to the increase in F1. However, taken together, all 3 studies suggest a maximum F1 of 30% for type 3 laughter detection. This demonstrates that this is an incomplete task that needs further investigation and, as such, this is what is addressed in this body of work.

In summary, type 1 tasks classify audio clips of <3 s into laughter or not laughter. Each clip is composed entirely of laughter or not. In type 2 tasks, the audio clips are between 3 and 11 s in length. Multiple audio events may occur, though generally it is known at least one target event will occur. In type 3 tasks, the audio clips can vary from 10 min to 1 h. There is no guarantee that any target audio events will occur. Each task represents an increase in difficulty, as demonstrated by the difference in performance, with F1 scores of >95% in type 1 tasks, 50-86% in type 2 tasks, and 21-30% in type 3 tasks. Given the clearly insufficient performance ceiling seen for type 3 tasks, this thesis addresses this type of task with the goal of achieving much more effective laughter detection.

2.4 Dataset Review

The type of task addressed is an important lens through which the widely variable results reported in the field of laughter detection can be understood. A further important consideration are the datasets that have been used. This is because some offer advantages or unique difficulties. A summary of the datasets used in the studies shown above are displayed in Tables 2.5, 2.6 and 2.7.

Table 2.5: Part 1. Summation of Datasets Used in the Laughter Detection Field

Name	Elicitation of Laughter	Number Of Speakers	Gender Split Male/Female	Age	Laughs	Laugh Duration Total (mean±std) Seconds	Total Duration (all)
MANHOB Laughter Dataset	Participants Viewed Funny Video Clips, Spoke for 90 s, Tried to Pose a Laugh	22	12/10	27-28	563	930.72 (1.65±2.32)	61 m
ARMEN	Interacted With Wizard-of-Oz Virtual Agent Making Small Talk	77	NR	18-90	253	NR	68 m
ROMEO2	Interacted With Wizard-of-Oz Nao Robot on Four Tasks: Take Medicine, Call a Relative, Greeting and Song Recoginition	27	3/24	75-99	205	NR	98 m
OFFICE	Two Tasks Where a Nao Robot 1) Told Jokes and 2) Asked the Participants to Act Out Named Emotions	7	NR	18-50	123	NR	10 m
JEMO	Participants Acted an Emotion in an Attempt to Have a Computer Recognise their Emotion	59	NR	16-48	73	NR	29 m
BEA Hungarian Spoken Language Database (Conversation Sub-Set)	Meetings Between Participants in Sound Proofed Rooms	NR	NR	20-90	775	720 (0.91±0.61)	148 m
AMI Meeting ([31] Sub-Set)	Participants Engaged in Spontaneous Speech in Meetings	8	6/24	NR	40	58.4 (1.46±1.09)	2 m 56 s
FreeTalk	Participants Engaged in Spontaneous Speech in Meetings	4	NR	NR	300	NR ($\sim$1)	180 m

Table 2.6: Part 2. Summation of Datasets Used in the Laughter Detection Field

Name	Elicitation of Laughter	Number Of Speakers	Gender Split Male/Female	Age	Laughs	Laugh Duration Total (mean±std) Seconds	Total Duration (all)
Iranian Laughter Database (With Supplementary Data)	Taken From Iranian TV Shows	NR	NR	NR	3953	NR	NR
Teleoperated ERICA	Human-Robot (Wizard-of-Oz) Interactions by Phone in a Speed Dating Free Talk Scenario	61	61/0	NR	1206	NR	NR
TUM AVIC	Experimenter Presents Products to Participants in a Posed 'Commercial Presentation'	21	11/10	29.9	261	NR	10 h 22 m 30 s
Nonverbal Dyadic Conversation on Moral Emotions (NDC-ME) (Sub-Set)	Dydactic Conversations Elicited by Open Questions about Emotional Topics	14	10/4	NR	446	NR	90 m
IFA Corpus (IFADV) (Sub-Set)	Participants Were Interviewed and Asked to Read Aloud from Scripts	18	9/9	15-66	NR	NR	46 m
JOKER Project Human Robot Humorous Interactions	Participants Interacted with the JOKER Robot System (Nao) Which Told Jokes, Riddles and Questioned the Participant	37	23/14	21-62	NR	NR	7 h 58 m 50 s
Interactive Emotional Dyadic Motion Capture Database (IEMOCAP)	Professional Actors Engaged in Dydactic Conversations in Both Spontaneous and Scripted Conversations	10	NR	NR	248	382 (1.54±NR)	8 h

Table 2.7: Part 3. Summation of Datasets Used in the Laughter Detection Field

Name	Elicitation of Laughter	Number Of Speakers	Gender Split Male/Female	Age	Laughs	Laugh Duration Total (mean±std) Seconds	Total Duration (all)
ICSI Meeting Corpus	Participants Engaged in Spontaneous Speech in Meetings, Each Meeting Had Specific Focuses Such as Natural Language Theories	53	40/13	20-60	11515	24282 (NR)	72 h
Switchboard Corpus	Spontaneous Telephone Conversations Between Two Participants	543	302/221	NR	24485	18600 (NR)	260 h
AudioSet	Clips of YouTube Videos From a Variety of 'In-The-Wild' Settings	NR	NR	NR	1492	3480 (NR)	2 h 28 m
Hume Vocal Burst (Hume-VB)	Participants Were Played Clips of Paralinguistic Events and Asked to Replicated Them	4080	NR	18-92	NR	NR	194 h 26 m 35 s
ERATO Human-Robot Interaction Corpus	Participants Spoke With a Human Operated Robot	91	NR	NR	984	NR	16.8 h
SEMAINE-DB	Participants Spoke With an Operator Playing a Role	20	NR	NR	NR	389 (NR)	6 h 30 m 41 s
Unnamed [30]	Samples Extracted from TV Recordings	NR	NR	NR	100	NR	30 m
SSPNet Vocalisation Corpus	Clips Extracted from SSPNet Mobile Corpus	120	57/63	18-64	1158	1091 (0.94±0.70)	8 h 25 m
SSPNet Mobile Corpus	Participants Engaged in Dydactic Conversations Around the Winter Survival Task	120	57/63	18-64	1009	1456 (0.72±0.54)	12 h 41 m 55 s

The manner in which audio and audio events, such as laughter, are elicited is a central distinction between different corpora. Audio elicitation methods create three broad categories of audio: spontaneous, posed and mixed. In spontaneous datasets, participants can be told to engage in small talk [31], complete a task that requires discussion [55] or participate in an interview [52, 56]. In all of these cases, the participants speak naturally. During the course of these conversations, paralanguage events, such as laughter and fillers, occur spontaneously. In the posed scenarios, the participants may be asked to mimic a sound played to them [40], act out a script [49] or make a sound described to them [45]. In these cases, the targeted audio events are guaranteed to occur. However, as mentioned above, there are consistent detectable differences between posed and spontaneous laughter [54]. This creates a weakness in the datasets, which utilize elicitation methods that produce posed vocalisations.

The elicitation method impacts another important dimension, which is the ratio of laughter to speech. Tables 2.5, 2.6 and 2.7 display descriptive statistics for each corpus's laughter content. The total amount of laughter varies from as little as 73 laughs in the JEMO corpus [45] to 24485 in the Switchboard corpus [57]. However, more important than the total amount is the percentage of audio that is comprised of laughter. This can vary from 25% in the MANHOB laughter dataset [40] and 8% in the BEA Hungarian spoken language database (conversation sub-set) to 1.3% in the IEMOCAP [49]. This displays large differences in the amount of laughter present. It is important to note that the reporting of the laughter descriptive statistics is sporadic and non-standardised; the above statistics are not calculable for all the datasets.

However, the elicitation method and laughter percentage are not the only important dimensions. The number of participants, their nationalities, age and gender can also have an impact on the quality of the dataset and the tasks it can be used for. The number of participants in a dataset can vary from 4 in the FreeTalk corpus [36] to 4080 in the Hume-VB [53], with the median number reported in the studies in Tables 2.5, 2.6 and 2.7 being 32. A smaller number of participants creates multiple issues. Aside from the obvious issue of there being uncertainty in the generalisability of the results, the fewer participants mean that it is more difficult to split the dataset into independent folds for training and testing purposes. In the case of the FreeTalk dataset, both the training and testing folds contain audio and laughter from the same speakers [36]. This means that the developed systems could be considered as recognition rather than detection systems, as a result, when applied to other speakers, a performance drop would be expected. This issue is not limited to cases in which there are very few participants. In the ISCI meeting corpus, 75 multi-party meetings were recorded [25] and, in total, there were 53 speakers. Each speaker could appear in any number of the meetings and contribute any amount of audio. This makes it difficult to create training and testing sets that are speaker independent. Finally, for 'in-the-wild data' capture such as clips extracted from TV [58] or internet streaming sites [19], there is often no reported information on the participants. This means that it is impossible to know how many speakers are present and whether they appear in more than one clip. These corpora, therefore,

also encounter the issue of whether they are addressing a recognition or detection task.

A final important issue is the definition of laughter for each dataset/study. This is not purely a dataset issue since some studies alter datasets to better align with a particular definition of laughter. A particular point of disagreement in the field is how voiced/unvoiced laughter and laughter overlapping with other audio events should be treated. In some studies, it is argued that voiced/unvoiced laughter should be considered as distinct classes to enable better detection, due to breathing being often mistaken for unvoiced laughter [25, 33]. In other studies, laughter overlapping with speech was excluded [34] while, in the creation of some datasets, only clear laughter (that does not overlap with anything) is included [41]. These varied definitions of what is considered laughter change the difficulty of the task and the usefulness of the detectors created. In the SSPNet Mobile Corpus (SMC) both voiced and unvoiced laughter is labelled as laughter. Both overlapping and clear laughter events are also included ensuring that the work in this thesis addresses all types of laughter.

2.5 Evaluation Methods

Laughter detection presents unique challenges for evaluating deep learning approaches. The central issue is that of class imbalance. In type 2 and 3 tasks, as explored in the previous section, the percentage of target class in the total audio can be as low as 3% and is generally less than 10%. This large class imbalance means some evaluation metrics are unsuitable, as high scores may not correspond to effective detectors. Accuracy displays this issue. If a detector marked all audio as not laughter, the accuracy score, in the case of 3% of the data being laughter, would be 97%. This figure is deceptively high given that the model is not detecting laughter whatsoever. Although the average is not directly used in the literature, the average error rate has been employed [34], which is calculated as follows:

$$\text{AverageErrorRate} = 1 - \text{Accuracy}. \tag{2.1}$$

This metric suffers the same issue as accuracy by giving deflated error rate results; hence, the accuracy can be overly inflated by a certain extent. Another evaluation method, which suffers from similar issues, utilised in the literature is the equal error rate [29]. This is calculated by varying the cut-off value for the classification. The cut-off value refers to what value a frame's posterior must be equal to or greater than to be classed as the target class. For each cut-off value, the specificity (also termed false positive rate or true negative rate) and the false rejection rate (FRR) are calculated. These rates can be expressed as follows, respectively:

$$\text{Specificity} = \frac{\text{FalsePositive}}{\text{FalsePositive} + \text{TrueNegative}}. \tag{2.2}$$

$$\text{FRR} = \frac{\text{FalseNegaive}}{\text{FalseNegaive} + \text{TruePositive}}. \tag{2.3}$$

As the threshold is increased, the FRR must remain constant or decrease. While the specificity will remain constant or decrease. This means that for a certain threshold value the FRR and specificity will be equal. The equal error rate is then the percentage of misclassified frames at that threshold value. This essentially means that equal error rate uses the same equation as the average error rate above but with a optimised accuracy. This means that equal error rate suffers from the same class imbalance issue which exaggerates the efficacy as estimated by the metric.

AUC is a commonly used metric [1, 27, 31, 38, 42, 43, 48–50, 53]. This form of metric is calculated as the area under the receiver operating characteristic (ROC) curve (AUC). The ROC curve expresses how effectively a detector can discriminate between two classes. To plot it, the sensitivity (also termed recall or true positive rate) and specificity of the detector are calculated for every cut-off value for the classification. The sensitivity is calculated in the following manner:

$$\text{Sensitivity} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}}, \tag{2.4}$$

These values are plotted against one and other to create the ROC curve, then the area under the curve is calculated, giving the AUC. A full examination of the strengths and weaknesses of this method is given in Section 4.2.8. Finally, there are the measures of precision, recall and F1, which are calculated in the following manner, respectively:

$$\text{Precision} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}}, \tag{2.5}$$

$$\text{Recall} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}}, \tag{2.6}$$

$$\text{F1} = \frac{2(\text{precision} \times \text{recall})}{\text{precision} + \text{recall}}. \tag{2.7}$$

Since F1 is the harmonic mean of precision and recall, achieving high scores in either precision or recall does not lead to a high score in F1. This makes F1 a more conservative estimate of a detector's efficacy relative to AUC.

The above metrics can operate at either a frame or event level. Type 1 task results can be viewed as event level metrics when any form of majority voting rule is used, such as when the performance at the individual frame level is ignored in favour of a label applied to the audio clip as a whole. As outlined above, this can lead to a situation where the type 1 system can be wrong in 49% of the cases and still achieve a perfect score. This example illustrates the need to carefully select appropriate metrics and the appropriate level to apply these metrics. Which metric and level to chose centres on the data used and the proposed end goal of the

laughter detection system. For example, in an ASR system, the event level metrics are more important since the placement of laughter in relation to other audio events is more important than the exact start, end and length of the laugh. However, when using laughter detection for pathology detection, the 'makeup' of the laugh is important [7, 59]. For example, how long is it and how many vocal bursts were present are all relevant questions for which the start and end of the laughter may be needed. In this case, the frame level metrics are more important since the detection system must capture as much of the laughter as accurately as possible. In the present work, an event level evaluation system is presented (see Section 4.2.4 for full details) alongside frame level results.

AUC is provided throughout the thesis to enable comparison with previous studies in the field as it is one of the most commonly used metrics (see Table 2.1 - 2.4). However, due to the class imbalance issue inherent in type 3 tasks and AUCs inclusion of true negatives in its calculation, as outlined above, it was not selected as a central metric for the estimation of model effectiveness. Instead, F1 was selected as the central metric by which models would be compared. This was chosen as F1 does not include true negatives in its calculation and so offers an estimation of a models ability to detect where laughter is rather than where laughter is not. Further, as it is the harmonic mean of precision and recall, only models which are effective at both aspects of a detection task will be evaluated as effective. Both precision and recall are equally important for laughter detection and F1 gives equal weight to both in model evaluation.

Chapter 3

Datasets

Two datasets are used in the present work. The SSPNet vocalisation corpus (SVC) [41] and the SSPNet mobile corpus (SMC) [55]. The two datasets are closely linked, with the SVC being composed of extracts from the SMC. The SVC is a type 2 dataset and was used to enable comparison with previous work in the field. This work was carried out to ensure that State-Of-The-Art (SOTA) laughter detection was achieved before attempts were made to address type 3 tasks. The SMC, as shown below, is a type 3 dataset and is the central focus of the thesis. It was selected because type 3 tasks represent the least constrained version of laughter detection currently addressed in the field. Further, type 3 tasks do not yet have an effective solution. The following initially describes the SMC in detail before describing the SVC.

3.1 SSPNet Mobile Corpus

The SMC is composed of 60 two-person phone conversations. Each phone conversation had different speakers, meaning 120 participants partook in the data creation. Audio was elicited using 'The winter survival task' [60]. In this scenario, participants debate on which items from a predefined list can increase the chances of survival after a plane crash in a polar environment. The creators of the corpus selected the winter survival task since few people have knowledge of how to survive an extreme scenario, such as the one presented to participants in the task. Indeed, only one participant involved in the creation of the corpus had any sort of experience in the task. This makes it likely that the outcomes of the conversations tended to depend on social and psychological aspects rather than on the experience/knowledge of either of the two speakers.

The participant gender divide was 57 male and 63 female, and all participants were native English speakers. Participants' age ranged between 18-64. Most were current students or staff at university, with only 16 being former students. To avoid the effects that seniority of position may have on the conversation, participants were unaware of their conversation partner's background.

The total duration of the audio is 12 h, 41 min and 55 s, meaning that the average per call

was 12 min and 42 s. The total amount of laughter events in the corpus is 1009. Previous studies that utilise the SMC have reported a total laughter count of 1805 [55]. The difference in laughter amount is due to the definition used in the present study and the merging step carried out. In this work, a laughter event is defined as a sequence of audio in which at least one speaker laughs. This includes voiced and unvoiced laughter and overlapping laughter. Furthermore, it is important to note that, if both speakers laugh concurrently, this is counted as a single laughter event. In other reports on the SMC, when two speakers laugh concurrently, this is counted as two laughs rather than a one-laugh event. Moreover, there are occurrences in the SMC where the gap between the two laughter events is short. This gives rise to the possibility that they can be considered as a single laughter event, given the definition used here. For example, some gaps between laughter events in the SMC were under 200 ms, which is shorter than the average gap between bouts of vocalisation during voiced laughter found in other studies [7]. Other studies have used arbitrary cut-offs to merge inter-word gaps to create one larger overarching speech event of 0.3 s [33]. In the present work, a probabilistic based merging methodology was developed. Given a set of gaps between laughter events $G = \{\Delta_1, \Delta_2, \ldots, \Delta_E\}$, where E is the total number of gaps and each gap Δ_E is the number of frames from where one laughter event ends and the next begins. Outliers where then identified as those gaps that had a length in the shortest 5% of G. This cut-off is a standard threshold in statistical testing for identifying outliers [61].

Every conversation includes, on average, 16.8 ± 12.9 laughter events showing that there is high variability between one dyad and the other. However, one cannot exclude that this is because there are different conversations' duration. The average frequency (number of laughter events per minute) is 1.5 ± 1.0, with minimum and maximum of 0.0 and 4.6, respectively. The high standard deviation suggests that the tendency to laugh changes significantly across conversations. Summary statistics about the conversations by gender and role are displayed in Table 3.1. Further descriptive statistics by gender pairing are given in Table 3.2; gender pairing refers to the two speaker's gender and, in the SMC, this can be either male-male (MM), male-female (MF) or female-female (FF).

Examining the descriptive statistics for gender, an independent t-test found that male speakers (M = 6.46, SD = 4.20) tended to utter sounds more often in a conversation than women (M = 4.96, SD = 2.55, t(118) = 2.37, p = 0.019). However, in terms of total amount of laughter, male (M = 13.18, SD = 12.05) and female (M = 16.71, SD = 14.91) speakers showed no significant difference (t(118) = 1.41, p = 0.16). This is explained by the significant differences between male (M = 1.04, SD = 0.75) and female (M = 1.64, SD = 1.26) laughs per minute of conversation (t(118) = 3.13, p = 0.0021). Showing that female speakers tend to laugh with greater frequency than male speakers but, as male speakers vocalise more, there is no significant difference in total amount of laughter. Of further interest is the significant difference in length of laughter events with female speakers (M = 0.72, SD = 0.20) having longer laughs than males (M = 0.62, SD = 0.29, t(118) = 2.21, p = 0.030).

In relation to the gender pairing of the conversations, a one-way Anova test found only one significant difference between pairings. It was shown that the average length of laughter was significantly different between pairings (F(2, 237) = 4.49, p = 0.012). A post-hoc Tukey HSD test found that the average laugh was longer for MF (M = 0.76, SD = 0.26) than MM pairings (M = 0.59, SD = 0.16, p = 0.004, 95% C.I. = [0.047, 0.29]). The Tukey HSD test found no significant difference between MM and FF (M = 0.69, SD = 0.15, p = 0.19, 95% C.I. = [0.23, 0.035]) nor MF and FF (p = 0.29, 95% C.I. = [0.040, 0.18]).

Table 3.1: Descriptive Statistics for the SMC by Gender and Role

	Male	Female	Caller	Receiver
Number	57	63	60	60
Age	29.95 ± 11.01	28.11 ± 13.06	30.72 ± 12.83	27.25 ± 11.19
Total Laughs	13.18 ± 12.05	16.71 ± 14.91	15.87 ± 14.38	14.20 ± 13.02
Laugh Frequency (minute)	1.04 ± 0.75	1.64 ± 1.26	1.41 ± 1.13	1.30 ± 1.04
Length of Laughs (seconds)	0.62 ± 0.29	0.72 ± 0.20	0.70 ± 0.22	0.65 ± 0.28

Table 3.2: Descriptive Statistics for the SMC by Gender Pairing

	MM	FF	MF
Number	14	17	29
Age	31.39 ± 11.16	29.62 ± 14.46	27.45 ± 10.87
Conversation Length (minute)	14.10 ± 7.85	9.73 ± 3.67	10.97 ± 6.03
Total Laughs	12.36 ± 11.75	15.56 ± 12.96	16.02 ± 14.86
Laugh Frequency (minute)	0.88 ± 0.69	1.59 ± 1.21	1.45 ± 1.10
Length of Laughs (seconds)	0.59 ± 0.16	0.69 ± 0.15	0.76 ± 0.26

This suggests that the gender composition of a conversation should have limited impact. The shorter laughter duration of laughter in MM pairings may pose a challenge, but in all other cases there is no difference. Gender is likely to present the most difficult issue. Male laughs are significantly shorter and less numerous than female laughs, this results in less data available for a system to learn about male laughs and could lead to performance issues in relation to males.

Finally, the SMC can be compared with the other datasets in the field for which type 3 tasks can be carried out. The 5 relevant datasets are as follows: the teleoperated ERICA [62], the BEA Hungarian spoken language database (conversation sub-set) [39], IEMOCAP [49], ICSI meeting corpus [63] and the switchboard corpus [57]. All of these corpora contain spontaneous speech much like the SMC, although the elicitation method differs between them. In terms of total audio, only two corpora (i.e., ICSI (72 h) and switchboard (260 h)) are longer than the SMC. Both of them also contain more laughter events than the SMC. Furthermore, the teleoperated ERICA corpus reports 1206 laughs. However, they count shared/overlapping laughter as two separate laughs; using the taxonomy of this project, the 508 shared laughs are each treated as one laugh creating a total laughter event count of 952. This means that only the ICSI and switchboard corpora have the larger laugh counts. When examining instead the percentage of

audio that is laughter, both the BAE (8.11%) and ICSI (9.37%) have higher laugh percentages than the SMC, with the SMC having a total laugh percentage of 3.19%. In terms of the number of participants, the switchboard corpus has 543 participants, around 4.5 times more than the SMC. The ICSI and all the other datasets have around half as much, or less, than the SMC. Finally, in terms of gender balance, the SMC and switchboard corpora have close to a 50/50 split whereas in all the other corpora, which report the gender split, the ratio of male to female is 3:1 or worse.

3.2 SSPNet Vocalisation Corpus

The SVC was created by extracting clips from the SMC [1]. Clips were taken from all 60 conversations in the SMC and, as a result, participant information remains the same as was detailed in the previous section. The SVC consists of 2763 audio clips. Each clip is 11 s in length, leading to a total audio time of 8 h 26 min. Each clip contains at least one laughter or filler event of between 1.5 and 9.5 s duration. After carrying out the same merging task as detailed in the previous section, there are 1158 laughter events (475 male and 683 female) in the corpus with a total duration of 18 min 11 s of laughter. The average laughter duration is 0.94 ± 0.70 s. Laughter composes 3.59% of the total audio data. A total of 936 clips contain laughter. Of the clips that contain laughter, it can account for up to 25% of the clip audio, with an average of 10%. This amount is particularly important when considering the evaluation of laughter detection methods for the SVC and how clips containing no laughter are treated (see Section 4.1 for full details).

Chapter 4

Baseline Approaches

4.1 Motivation

The first goal of this work was to replicate the state-of-the-art approaches for laughter detection based on the SSPNet vocalisation corpus (SVC) and then apply them to the SSPNet mobile corpus (SMC). The SVC has received much attention within the field of laughter detection after its publication and use in the 2013 INTERSPEECH paralinguistic challenge [1]. Section 3 offers a full overview of the SVC. It is a type 2 corpus comprising short clips that are guaranteed to contain at least one laughter or filler. Table 4.1 shows the performance achieved by studies on the SVC.

The goal of this chapter is to build a detection system that could attain similar results using the SVC and then apply such a system to the SMC. This was done to address RQ1. The SVC was used to test the methods developed on a publicly available type 2 task and to allow comparison with previous studies to ensure that those methods were operating at a SOTA level. Once those methods were validated, they were then applied to the SMC. Laughter detection on the SMC is a type 3 task as it does not have the constraints of laughter to non-laughter ratio and total

Table 4.1: Laughter Detection Results for the SVC. FFN: feed forward neural network. HMM: hidden Markov model. Bi-LSTM: bidirectional long short-term memory neural network. LSTM: long short-term memory neural network

Study	**System Architecture**	**Precision (%)**	**Recall (%)**	**F1 (%)**	**AUC (%)**
[38]	Stacked FFN	NR	NR	NR	97.3
[1]	Not Reported	NR	NR	NR	82.9
[41]	HMM	35-67	61-80	48-64	NR
[42]	FFN	NR	NR	NR	95.1
[26]	FFN	80.9-87.4	87.3-94.5	84.0-90.8	NR
[48]	FFN	NR	NR	NR	95.3
[51]	Bi-LSTM-CTC	0.65-0.79	0.49-0.66	0.54-0.66	NR
[43]	Bi-LSTM	NR	NR	NR	93.4
[50]	LSTM	NR	NR	54.31	92.24

audio time. As the SVC is extracted from the SMC (as outlined in Section 3), any different in performance from the type 2 to type 3 task can readily be concluded to be caused, at least in part, by the experimental constraints used in type 2 tasks and allow RQ1 to be answered.

From Table 4.1, it can be seen that studies have attained AUC results from as low as 80% up to 97.3%. This provides a clear target for this section. Precision, recall and F1 are much less reported in the field, with only three studies using the SVC having reported them. The best results using the SVC show precision, recall and F1 at 80%-95% [26]. However, in this study, the researchers carried out undersampling of the non-laughter class until both classes had similar distributions. They did this for both the training and testing sets. This means that these results were obtained by carrying out a type 1 task and, as such, provide an over-estimation of the detector's ability and an inflated baseline. Instead, previously obtained results in the range of 65-79% are considered [51]. In this paper, they did not use a standard frame-by-frame classification model and instead utilised a connectionist temporal classification (CTC) method. CTC is a procedure that produces a sequence of labels, which may or may not be aligned with the original input sequence. This means that the output of the CTC approach can be considered an event level metric. This is one of two levels for which metrics can be calculated for; the other one is a frame level metric. In the latter, each frame is assigned a classification and the employed metrics take into account every frame. In contrast, the event level metrics identify either time-stamps or sub-sequences of the frames that are classified as the target class. When calculating the event level metrics, each target event is considered. If a time-stamp label occurs within the ground truth event, then that event is considered detected (a true positive) otherwise it is considered missed (a false negative). When a time-stamp occurs beyond a target event, this is considered a mistake (a false positive). With these three measures, it is then possible to calculate event level precision, recall and F1. CTC acts as an event level detection system since it produces time-stamps rather than classifying each frame.

Previous results can provide an event level baseline. A frame level baseline can be drawn from ref. [41], which has variable results ranging from 48-64% for F1 and has greater variation in precision (35-60%) and recall (61-80%). In this paper, the authors utilised a multi-class classification approach. Four classifiers were trained, each one a binary classifier for each of the four classes (i.e., laughter, filler, speech and silence). Each model, therefore, produced the posterior probability of a frame belonging to that model's target class. The frame classification was then set by selecting the class with the highest estimated posterior probability.

Taken together, these studies provide a baseline for performance on the SVC for AUC and frame and event level precision, recall and F1. However, it is important to note one final issue. Clips in the SVC contain either laughter (20.59%), fillers (66.12%) or both (13.28%). In the case where a clip contains no laughter, it is not possible to calculate recall mathematically since

recall is calculated using the following equation:

$$\text{Recall} = \frac{\text{TruePostives}}{\text{TruePositives} + \text{FalseNegatives}}. \tag{4.1}$$

Namely, in the case where a clip contains no laughter, the value of both true positives and false negatives will be zero creating a mathematically undefinable division by zero. Furthermore, AUC is calculated as the area under the receiver operator characteristic curve. This curve is plotted by calculating the recall and the false positive rate for every classification cut-off value. AUC is then determined from the area beneath the resulting curve; however, as above, recall cannot be calculated in the given circumstance. This again means that, for all the clips without laughter, AUC cannot be calculated. How the above studies handled this issue is not clearly reported. In one study, the authors report that "We adopt precision, recall, F-measure and their average over all social signal events as evaluation metrics". [51]. It is unknown if clips without laughter were just ignored, treated as 100% or 0% recall or if some other method was used. If they were ignored, this offers an advantage to the resulting system because any false positives generated in the clips containing no laughter would simply be discarded. If instead all the clips without laughter were assigned a recall score of either 100% or 0%, these clips would either advantage or disadvantage the model unfairly. One option is to treat all the clips in the test set as one. With precision, recall and F1 being calculated for all the clips at once rather than for each individual clip and then averaged. Since it is unknown how this issue was solved by the papers presented above, two different methods are applied in the present work. The first method, henceforth termed *exclusion*, calculates the average performance for each clip of the SVC, while ignoring all the clips that do not contain laughter. The second procedure, henceforth termed *merging*, merges all the clips in the test set together, enabling the inclusion of the false positives in the clips with no laughter for the calculation of the results.

For the following novel research, the type 3 corpus called SMC is used in automatic laughter detection for the first time. This represents a more difficult challenge for the following reasons: (i) the proportion of laughter to non-laughter is lower than in type 2 tasks (see Section 2.3); (ii) the classification is carried out over conversations that can last 20 minutes in length, rather than the 11 seconds of the SVC; (iii) laughter can overlap with any other audio events, including laughter from others; (iv) laughter is not guaranteed to occur in a conversation, unlike the SVC which guarantees either laughter or fillers will occur [41]. Taken together, these differences mean that the SMC represents a closer-to-real-world approximation than the SVC and a more difficult task.

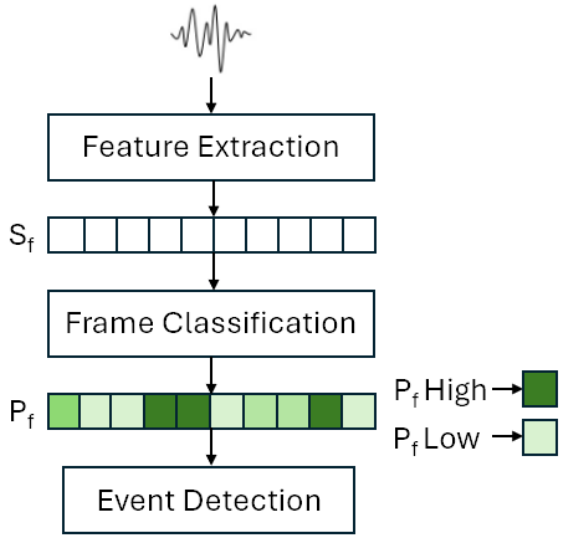

Figure 4.1: General Form of a Laughter Detection System

4.2 The Approach

Figure 4.1 displays the general form of the laughter detectors used in this section, all the detectors followed the same approach for feature extraction, frame classification and event detection. This general approach was then modified using different methodologies. These methods include feature vector extension, class weighing and smoothing.

4.2.1 Feature Extraction

The process of feature extraction was the same for both datasets, i.e., the SVC and SMC. The audio was initially split into windows of 20 ms duration, with a time step of 10 ms between window start times. Both of these values are standard in the literature (see Section 3) and no fine-tuning was attempted. Each window had the following features extracted from it:

- the first 13 mel frequency cepstral coefficients (MFCC),
- the signal intensity,
- the root mean square (RMS) energy,
- the fundamental frequency contour.

The features were extracted using the Python library named Surfboard v0.2.0 [64]. MFCCs, signal intensities and RMS energies have all been shown to be effective representations of speech for automatic speech recognition [65], pathology detection [66] and emotion recognition [67]. This feature extraction step resulted in a sequence of vectors $F = \{\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_T\}$, where T is the total number of vectors in the sequence. Each vector had the dimension $D = 16$.

4.2.2 Frame Classification

The goal of the frame classification step is to map each vector $\vec{f}$ to its probability of belonging to class laughter π_f. To carry out this mapping, two deep learning architectures were used: feed forward neural networks (FFN) and long short-term memory networks (LSTM). Table 4.1 shows that both of these architectures have been applied multiple times to the SVC and so were replicated here.

Feed forward neural networks consist of an input layer, a number of hidden layers and an output layer. Each layer consists of a pre-set number of neurons. Each neuron has an associated weight and bias. In a binary classification task, such as laughter/non-laughter, the final output layer consists of a single neuron that produces a single output. This output can be thought of as the estimated posterior probability of the input feature vector belonging to the target class.

To train an FFN to predict a target class, a set of examples were used in which the class of each frame f is known. During training for the weights and bias values of individual neurons, all such values are adjusted through back-propagation [68]. In back-propagation, the error of a network is calculated by comparing the target output for a frame with the output from the network. This comparison is undertaken using a loss-function. In the current work, binary-cross entropy loss was used, which is calculated via the following expression:

$$L_f = -y_f log(p_f) - (1-y_f) log(1-p_f), \tag{4.2}$$

where L_f is the loss for frame f, y_f is the label of f and p_f is the predicted probability of f being the target class (in this case laughter) by the network. This loss represents the error in the network, with higher values of L_f corresponding to worse predictions. The goal of training is, therefore, to lower the value of L_f. This is done by calculating the derivative of the loss function with respect to the weights and biases. As a result, each weight and bias is altered, according to the learning rate, to lower the value of L_f. In practice, this is not undertaken frame-by-frame but rather in batches where the loss for multiple frames is averaged before updates to the weights and biases are carried out. In the current work, a batch size of 100 was used.

The input layer size is determined by the dimensionality of the feature vectors; in this case, there are 16 input neurons. The tested feed forward networks contain two hidden layers. These layers are composed of 100 densely connected neurons that use the ReLu activation function [69]. Finally, there is a single output node, which utilises Sigmoid activation [69]. The number of layers and nodes were set through a hyper-parameter optimisation process, which is detailed in Section 4.2.5.

In addition to the FFN described above, the LSTM networks were also tested. LSTMs are specialised recurrent neural networks that are able to process sequential data by creating both long and short term memory representations of previous frames. LSTMs cells are composed of five key components: forget, input and output gates, a cell state and a hidden state. Figure 4.2

shows the LSTM cell used in this work.

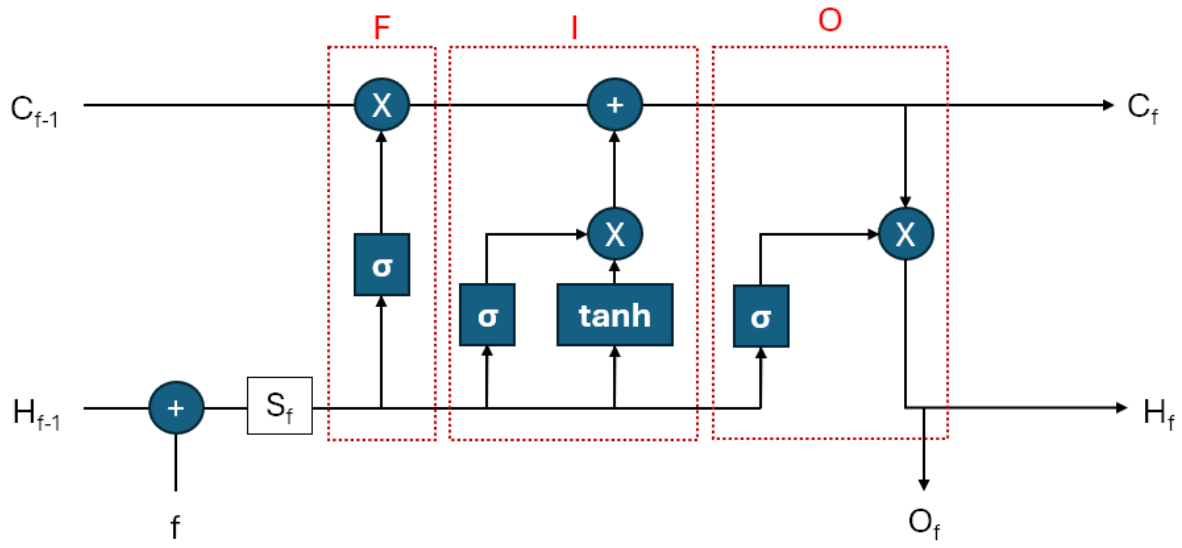

Figure 4.2: Information Flow Through a Single LSTM Cell. Top Line Follows the Cell State C. Bottom Line Follows the Hidden State H for the Current Frame f. The Three Gates: Forget F, Input I and Output O are Outlined in Red

.

The process of computing the output for a given frame via an LSTM is as follows. First, the current hidden state from the previous step H_{f-1} and the input frame f are concatenated to create the current state of the network S_f for a given frame f. If this is the first frame in the sequence, the values for both the cell state and the hidden state are zeroes. The variable S_f is then used to update the cell state through the following equation:

$$C_f = F(S_f)C_{f-1} + I(S_f), \tag{4.3}$$

where C_f is the cell state and C_{f-1} is the cell state of the previous frame; F is the forget gate function and I is the input gate function. The forget gate uses a sigmoid activation function to produce a value between zero and one for every value of S_f. By then multiplying the output of the forget gate and $C_t - 1$, it is possible to 'forget' the information by setting it to zero. The sigmoid function is used to ensure that no values increase in size, meaning that the information at this step can only be forgotten. The input gate I does the opposite, where initially a tan function is used to constrain all the values between -1 and 1 multiplied again by the output of a sigmoid function. This is done to calculate what information from the current state of the network should be incorporated in the cell state through addition. After these steps have been completed, the cell state C_f is then used to update the current hidden state of the network via:

$$H = O(C_f)S_f, \tag{4.4}$$

where H is the hidden state and O is the output gate function. The latter is a tan function,

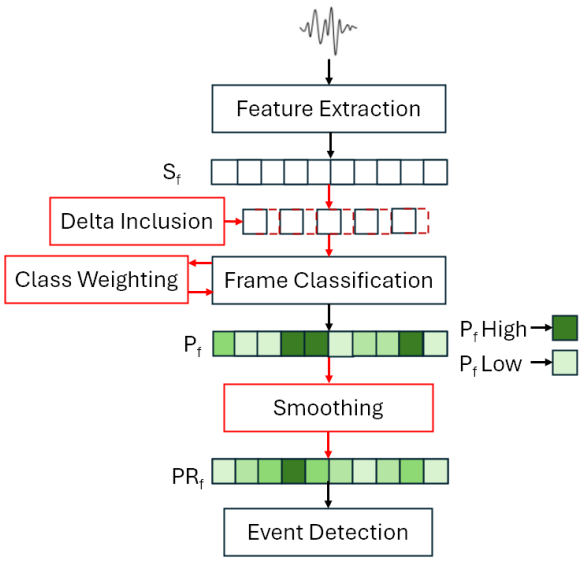

Figure 4.3: General Form of Laughter Detector With Pre/Post-Processing Methods Shown in Red

which means that the long-term memory alters the output of the current step of the LSTM. This step produces the hidden state for frame f, which can then be used for class prediction O or passed as an input into the next step of the LSTM H_f if there are more frames in the sequence. The cell state and the hidden state of the LSTM can be considered its memory since both hold the information sent between inputs. They differ in that the hidden state captures short term dependencies and has an immediate effect on the current prediction (detailed below), while the cell state maintains longer term dependencies and enables the long-term portion of the LSTM's memory.

The LSTM networks used in the present work were composed of 1 input layer, 4 hidden layers and 1 output layer. The input layer took the target feature vector and the nine immediately prior feature vectors, meaning it had 160 inputs. The first two hidden layers were composed of 100 densely connected standard neural network nodes while the third layer was composed of 100 LSTM nodes, all of which use ReLu activation. The output layer contained a single output node that uses Sigmoid activation. Again, these parameters were set using the hyper-parameter optimisation process detailed in Section 4.2.5.

4.2.3 Pre/Post-Processing

The previous text outlines the general approach for laughter detection. However, there are multiple ongoing issues in the field. Examples of these include: class imbalance, spurious detections and context inclusion, with multiple methodologies having been developed to address these issues. Figure 4.3 shows the extra pre/post-processing steps used in addition to applying the above classifiers to the SMC dataset; these steps are explained below.

In type 1 tasks, datasets are often created so that laughter and non-laughter are balanced. This is unrealistic in spontaneous conversations, as explained in Section 2.4; for type 3 tasks, laughter comprises 10% or less of the total audio time. These large class imbalances present issues for deep learning techniques because there are not enough frames for the target class to learn how to properly distinguish them from the non-target class [70]. One solution to this problem is to add a weight to the loss function. This technique, called class weighting, alters the calculated loss for each prediction made during training according to a pre-determined weight. The standard manner in which to calculate this predetermined weight is given by:

$$w_c = \frac{n_f}{n_c n_{fc}}, \tag{4.5}$$

where w_c is the weight the loss function is multiplied by for a particular class c, n_f is the total number of frames, n_c is the total number of classes and n_{fc} is the total number of frames belonging to class c. This means that each frame's loss is adjusted by multiplying the loss by the inverse frequency of the frame's class. Class weighting has been applied in the laughter detection field [71] and other analogue fields [70], leading to improvements in detector error rates; it has been tested in this work.

Due to speech audio events being produced by the same mechanisms, they share commonalities that can lead to spurious detections in frame-by-frame classification [42]. Spurious detections are characterised by sharp spikes in the outputs of the frame classifier step. To remove these spikes, the sequence of posterior probabilities estimated by the classifier, π_f, can be convolved with a Hamming window according to the following expression:

$$p_k = \sum_{n=-\infty}^{\infty} \frac{1}{L} \pi_n H_{k-n+L/2}, \tag{4.6}$$

where H_n is a Hamming window of length L. This means that p_k is the convolution of the samples in an interval of length L, centred on π_k, with the Hamming window sample values. Every sample is weighted by the Hamming window to emphasise the samples closest to π_k. The effect of this smoothing step is shown in Figure 4.4. The motivation behind the choice of the L values is that it is comparable with the average duration of the laughter events in the corpus (see Chapter 3). This step is able to effectively remove spikes in π_k. Furthermore, it can fill gaps in the detections where silence occurs between laughter bouts. However, it is unable to improve detection of completely missed laughter events.

A further issue for FFNs is a lack of context. Context is recognised as the key for identifying paralinguistic events in speech [51,72–75]. FFNs consider each frame in isolation, meaning that they have no ability to receive context from the surrounding frames. This can result in detection errors for both type 1 and 2. Pre-processing can incorporate context through the extension of the input feature vector. This extension can be carried out by concatenating neighbouring frames [26,58] or the inclusion in the feature vector of the deltas between frames [26,31,32,36,

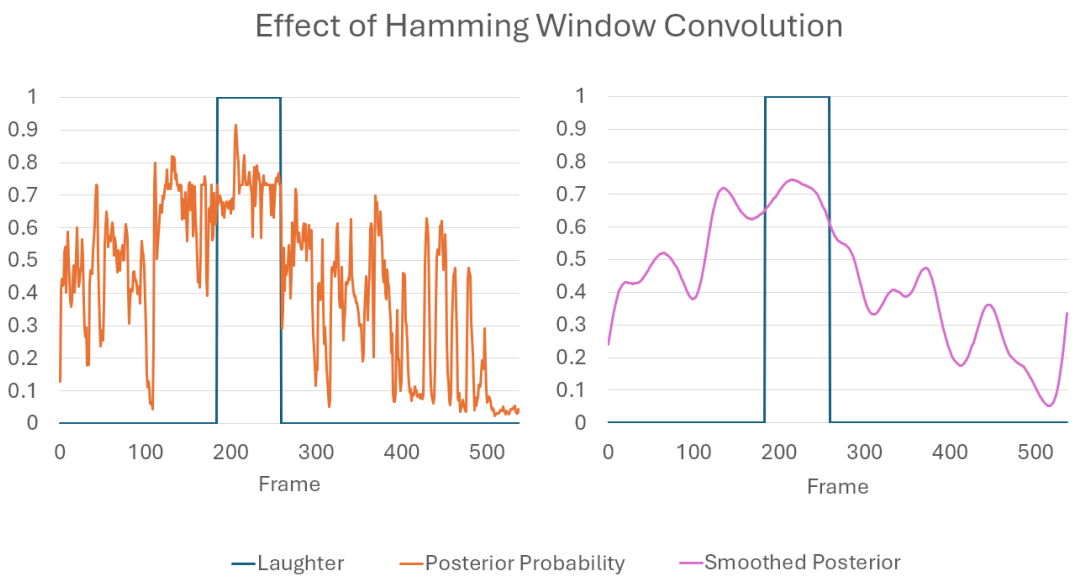

Figure 4.4: Frame Probabilities Before (left) and After (right) Hamming Window Convolution

58, 76]. In this part of the work, feature vectors were expanded to include the Δ of each of the 16 features, calculated as the difference between the target frame's values and the immediately prior frame. This means that detection methods using this time-dependant information had input feature vectors with the dimension of $D = 32$.

4.2.4 Event Detection

The frame classification step produces a sequence $P_s = \{p_{s1}, p_{s2}, \ldots, p_{sk}\}$ or $P = \{p_1, p_2, \ldots, p_k\}$ (depending on the presence or absence of smoothing), which are the estimated posterior probability that a frame belongs to class 'laughter'. Of more interest than frame level classifications is the ability to detect laughter events. This step is to enable downstream uses as outlined in Section 2.5. Event detection was carried out by identifying time stamps in the sequence of probabilities that should correspond to laughter events.

Laughter events were assumed to occur where there were peaks in the sequence P_s. Peaks were defined as sub-sequences within P_s where $p_k > p_{k-1}$ and $p_k > p_{k+1}$. As this definition allowed the value of p_k to be small, a further condition was added where a cut-off was applied such that peaks were only considered as event detections if $p_k > M$. Here, M was a value between 0 and 1 and was set through hyper-parameter optimisations (full details are given in Section 4.2.5). It was a further possibility that sequences of P_s, which were above M, would still oscillate locally, leading to many peaks occurring close together. As outlined in Section 3.1, where the ground truth labels were merged, the same probabilistic-based merging was used here. The values derived from the ground truth labels were applied to the merging of the event peaks. In the case where two peaks were merged, a point equidistant from each peak in P_s was selected

as the new time-stamp. Merging was carried out repeatedly until all the remaining peaks were beyond the merge range from one peak to another. In accordance with the methodology in ref. [51], only those peaks that co-occurred with laughter were considered true positives.

4.2.5 Training and Testing

A k-fold approach was used for the creation of the training and testing sets. For the SVC dataset, each of the 11-second clips were initially grouped by which conversation they had been extracted from. They were then split so that all the clips from a conversation would always be either in the training or test fold; this ensured speaker independence. The fold splits were carried out following a 40-10-10 rule, i.e., with 40 conversations being used for training, 10 for validation and 10 for testing. For the SMC, a similar approach was used in which the training, development and testing splits were again carried out at a conversational level with the same 40-10-10 split.

In the case of both datasets, this resulted in a total of 6 folds. To improve upon the reliability of the results, each detection method experiment was repeated three times, each time initialising random underlying models. The results are presented as the average AUC, precision, recall and F1 across all six folds and the three repetitions of each experiment. Before training and testing, each feature was normalised to a scale from 0 to 1. The normalization function was fitted for each fold: this ensured no crossover of information between the training and testing data sets. For evaluation purposes, the standard metrics of AUC, precision, recall and F1 were calculated at a frame level. In addition, the event detection scheme was also evaluated using precision, recall and F1.

The networks for frame classification were trained using binary cross entropy as a loss function coupled with the Adam optimizer, with a default learning rate of 0.001. The number of training epochs was 5 and the batch size was 1000. All these values were set a-priori and no attempts were made to use alternative values. The models were implemented using the Python library called Keras (version = 2.4.3) [77].

4.2.6 Hyper-Parameter Optimisation

There were four hyper-parameters to be set. These were the number of hidden layers, the number of nodes per layer, the size of the Hamming window used for smoothing and the percentage cut-off for classifying peaks as events. In addition to the parameters that are optimised, the metric that assists the optimisation of them could be AUC or F1 at either a frame or event level. These three metrics were investigated to determine if there was an effect on the results of the optimisation.

The number of hidden layers and nodes per layer were initially investigated with the Hamming window (size = 51) and cut-off (percent = 11) set. For FFNs, the number of hidden layers tested was 2 to 10 with a step of 2. The number of nodes tested was 100 to 1000 with a step

of 100. For LSTMs, the same number of nodes was tested. The LSTM networks can have two different types of hidden layer: an FFN layer or a LSTM layer. The number of LSTM layers tested was 2 to 10 with a step of 2. The number of FFN layers tested was 0 to 4 with a step of 2. A general rule was adopted where, if two parameters achieved the same performance, the smaller network was selected. This would aid future investigation by reducing the training and testing time. A complete grid search method was used, in which every possible combination of parameters were paired.

A two-way ANOVA test was utilised to examine the differences in performance. For the FFN networks, there was no significant difference in F1 due to the number of layers (F(4, 250) = 0.58, $p = 0.82$) or nodes (F(9, 250) = 0.62, $p = 0.65$), nor a significant interaction (F(36, 250) = 1.05, $p = 0.41$). As such, the number of layers was set at 2 and the nodes per layer was set to 100.

For the LSTM, a three-way ANOVA test was used. It was found that there was no significant effect due to the number of LSTM layers (F(4, 750) = 1.92, $p = 0.11$) or nodes per layer (F(9, 750) = 1.39, $p = 0.19$). Nor was there any significant interaction between the number of LSTM layers and the nodes per layer (F(36, 750) = 0.43, $p = 1.00$), the number of LSTM and FFN layers (F(8, 750) = 1.77, $p = 0.080$), nor the number of FFN layers and nodes per layer (F(18, 750) = 1.02, $p = 0.43$). There was a significant interaction between all three variables (F(72, 750) = 1.54, $p = 0.004$) and a significant effect due to the number of FFN layers (F(2, 750) = 27481.2, $p < 0.001$). A post-hoc Tukey HSD test found that two FFN layers ($M = 24.09$, $SD = 1.58$) were significantly better than zero FFN layers ($M = 1.55$, $SD = 0.26$, $p < 0.0001$, 95% C.I. = [23.21, 23.76]). Furthermore, four FFN layers ($M = 25.08$, $SD = 1.26$) were significantly better than zero layers ($p < 0.0001$, 95% C.I. = [23.40, 24.30]). However, there was no significant difference between two and four FFN layers ($p = 0.24$, 95% C.I. = [-0.086, 0.46]). As such, the LSTM networks were given 2 FFN hidden layers and 2 LSTM hidden layers, with each of the layers having 100 nodes.

Having set the architecture of the deep learning networks, the values for the Hamming window size and the percent cut-off were then optimised. The values for the Hamming window ranged from 11 to 101 (inclusive) with a step of 10. For the percent cut-off, the values ranged from 1 to 91% (inclusive) with a step of 10. A parameter optimisation was run for each of the underlying deep learning architectures (i.e., FFN and LSTM) and for each underlying dataset (i.e., SVC and SMC). The results of the optimisation for each of the models and the dataset options are shown in Table 4.2. In cases where there was disagreement between the metrics with regard to what parameters were optimal, the parameters found using event level F1 were initially selected. If the event level F1 showed no effect, frame level F1 was then used. If that showed no effect, then AUC was selected.

Table 4.2: Hyper-Parameter Optimisation Results for Each Detector, Metric and Dataset. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. D: delta. S: smoothing. NE: no effect. All FFN have 2 FFN hidden layers. All LSTM detectors have 2 LSTM and 2 FFN hidden layers. All hidden layers have 100 nodes

Model Architecture	Dataset	Window Size			Cut-Off M
		AUC	F1 (frame)	F1 (event)	
FFN	SVC	No Effect	11	11	10
FFN+CW		No Effect	11	41	10
FFN+D		11	11	11	100
FFN+CW+D		No Effect	No Effect	11	11
LSTM		No Effect	11	11	70
LSTM+CW		11	No Effect	51	10
LSTM+D		11	No Effect	31	100
LSTM+CW+D		No Effect	No Effect	11	11
FFN	SMC	51	No Effect	11	60
FFN+CW		No Effect	No Effect	31	1
FFN+D		No Effect	No Effect	51	40
FFN+D+CW		11	No Effect	51	10
LSTM		41	31	11	80
LSTM+CW		61	No Effect	31	11
LSTM+D		41	No Effect	71	11
LSTM+CW+D		No Effect	No Effect	51	11

4.2.7 Experiments and Results

This section displays the results achieved for the SVC and SMC datasets. A total of sixteen detection methods were trialled. These methods differed in terms of the underlying architecture and post/pre-processing. Tables 4.3 and 4.4 show the results obtained using the merging evaluation and the exclusion evaluation methodologies, respectively, for each detection method at a frame level. Together they show the effect of the evaluation method, as discussed in Section 4.1.

Initially, the effect of either merging the test set or excluding non-laughter clips was tested. This was undertaken using an independent t-test, which found no significant effect in relation to AUC using merging ($M = 81.29$, $SD = 4.50$) compared to exclusion ($M = 80.88$, $SD = 4.64$, t(574) = 1.16, $p = 0.28$). Since there is no significant effect on AUC performance, the exclusion dataset was selected for further testing.

A one-way ANOVA test discovered significant effect in relation to the detection system (F(15, 272) = 17.23, $p < 0.0001$). Post-hoc Tukey HSD tests were carried out to examine which of the detectors performed best. No significant difference in AUC performance was found between LSTM+S and LSTM+CW+S, LSTM+D+S and LSTM+CW+D+S. All the other detectors performed significantly worse than LSTM+D+S. Figure 4.5 displays the significant differences between LSTM+D+S and all the other detectors. Table 4.4 gives exact values of AUC for all the detectors, with LSTM+D+S achieving 90.02 ± 3.62. For exact p-values, lower and upper confidence values see Table A.1. These results can be considered close to the state-of-the-art

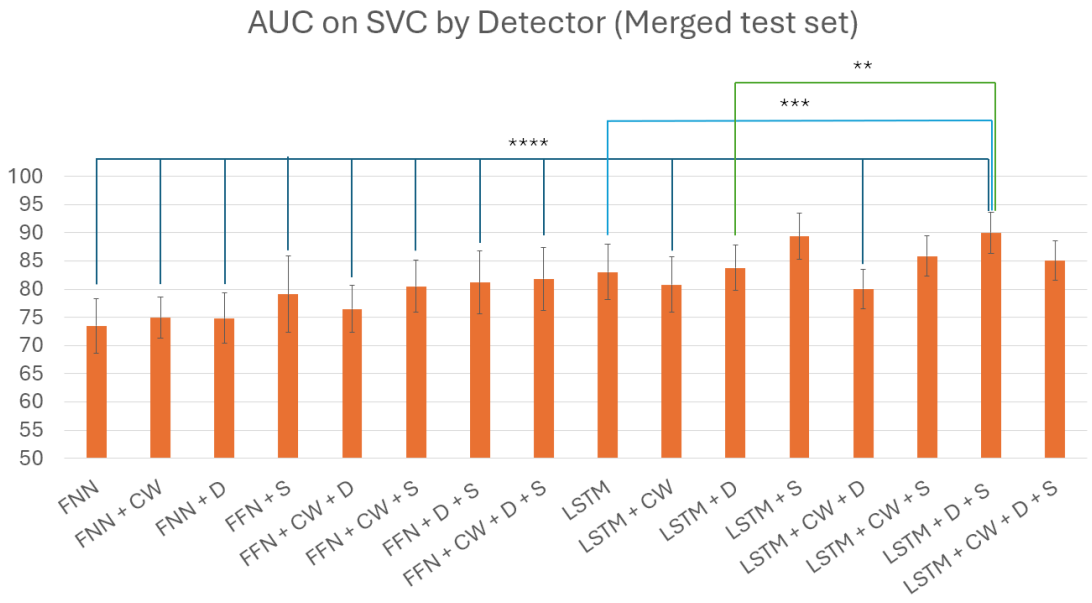

Figure 4.5: Average (STD) AUC Achieved by Detection System. Significant Differences Shown in Relation to LSTM+D+S (**$p < 0.005$, ***$p < 0.0005$, ****$p < 0.00005$)

analyses. The lack of reporting on the standard deviation and the number of repetitions makes it impossible to carry out significance testing to fairly compare these outcomes against other studies in the field. However, these results suggest that the best detectors are achieving state-of-the-art performance and that both LSTMs and smoothing are key to these optimal results.

The effect of the evaluation criteria was then tested for frame level precision, recall and F1. The independent t-tests were once again used and found that the evaluation method does have significant effects. Precision was significantly higher when using exclusion ($M = 49.85$, $SD = 17.82$) as opposed to merging ($M = 36.59$, $SD = 16.33$, t(574) = 86.68, $p < 0.0001$). F1 was also significantly higher when using exclusion ($M = 23.50$, $SD = 6.89$) as opposed to merging ($M = 14.42$, $SD = 5.63$, t(574) = 295.27, $p < 0.0001$). However, there was no significant effect on recall by the method (exclusion: $M = 33.06$, $SD = 8.84$. merging: $M = 33.06$, $SD = 8.85$, t(574) = 0, $p = 1.00$). These results contrast with the above AUC results, which suggested no effect due to the evaluation methodology. Improvements in F1 and precision were caused by exclusion of the non-laughter clips. This makes logical sense since only false positives could be removed through exclusion, thus improving precision. This also explains why some detectors' performance increased significantly. Namely, if a detector's precision was poor then, when non-laughter clips were excluded, the detector's precision significantly increased and led to a significant increase in F1.

Examining the results of the exclusion methodology further, a one-way ANOVA test found there were significant differences in F1 between detectors (F(15, 272) = 74.84, $p < 0.0001$). Figure 4.6 shows the significance differences found using a post-hoc Tukey HSD test. For exact p-values, lower and upper confidence intervals see Table A.2. LSTM+CW+S had the

Table 4.3: Frame Level Performance of Each Detection Method for the SVC Using Merging of All Test Clips - Allowing the Inclusion of Clips Without Laughter. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. D: delta. S: smoothing. Bold highlights the best performing detector for each underlying architecture.

Model Type	Precision	Recall	F1	AUC
FFN	42.23 ± 20.38	3.92 ± 2.01	6.38 ± 2.85	73.50 ± 4.87
FFN+CW	9.86 ± 2.69	53.76 ± 8.94	16.38 ± 3.60	75.02 ± 3.66
FFN+D	50.82 ± 22.44	4.31 ± 3.13	6.45 ± 3.27	74.87 ± 4.51
FFN+S	49.33 ± 44.32	0.71 ± 0.86	1.15 ± 1.32	79.11 ± 6.81
FFN+CW+D	12.42 ± 4.57	49.16 ± 11.05	19.16 ± 5.00	76.52 ± 4.23
FFN+CW+S	16.27 ± 5.14	51.12 ± 11.58	23.78 ± 5.28	80.54 ± 4.64
FFN+D+S	58.44 ± 45.48	1.04 ± 1.41	1.47 ± 1.55	81.19 ± 5.55
FFN+CW+D+S	**22.65 ± 9.08**	**46.09 ± 13.22**	**28.43 ± 7.68**	**81.87 ± 5.57**
LSTM	60.94 ± 20.15	11.27 ± 4.78	17.73 ± 5.78	83.06 ± 4.92
LSTM+CW	10.78 ± 4.24	72.88 ± 10.74	18.35 ± 6.34	80.84 ± 4.90
LSTM+D	69.45 ± 15.34	9.96 ± 7.82	15.71 ± 10.38	83.80 ± 3.97
LSTM+S	72.57 ± 24.12	5.47 ± 3.55	9.61 ± 5.63	89.36 ± 4.06
LSTM+CW+D	9.16 ± 3.70	66.77 ± 21.70	15.77 ± 5.99	80.03 ± 3.56
LSTM+CW+S	**13.43 ± 5.72**	**76.64 ± 11.76**	**22.10 ± 8.34**	**85.88 ± 3.59**
LSTM+D+S	75.68 ± 28.77	5.43 ± 6.11	9.26 ± 9.24	90.02 ± 3.62
LSTM+CW+D+S	11.36 ± 5.10	70.45 ± 22.86	18.93 ± 7.86	85.07 ± 3.49

Table 4.4: Frame Level Performance of Each Detection Method for the SVC Using Exclusion of All Clips Not Containing Laughter. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. D: delta. S: smoothing. Bold highlights the best performing detector for each underlying architecture.

Model Type	Precision	Recall	F1	AUC
FFN	60.60 ± 21.90	3.92 ± 2.01	6.84 ± 3.10	73.33 ± 5.22
FFN+CW	24.95 ± 4.93	53.76 ± 8.94	33.50 ± 4.52	75.05 ± 3.77
FFN+D	65.19 ± 23.06	4.31 ± 3.13	7.02 ± 3.86	74.86 ± 5.02
FFN+S	54.57 ± 46.72	0.71 ± 0.86	1.23 ± 1.43	78.49 ± 6.68
FFN+CW+D	29.75 ± 7.87	49.16 ± 11.05	35.82 ± 6.02	76.29 ± 4.48
FFN+CW+S	34.18 ± 8.66	51.11 ± 11.57	39.68 ± 6.70	79.78 ± 4.60
FFN+D+S	61.08 ± 46.19	1.04 ± 1.41	1.63 ± 1.87	80.74 ± 5.59
FFN+CW+D+S	**41.76 ± 13.12**	**46.06 ± 13.20**	**41.43 ± 8.58**	**81.01 ± 5.54**
LSTM	73.18 ± 15.96	11.27 ± 4.78	18.79 ± 6.68	82.52 ± 5.61
LSTM+CW	27.49 ± 7.02	72.88 ± 10.74	38.87 ± 7.82	80.92 ± 4.75
LSTM+D	81.29 ± 9.98	9.96 ± 7.82	16.48 ± 11.42	83.21 ± 4.43
LSTM+S	79.87 ± 21.74	5.45 ± 3.54	9.81 ± 5.90	88.20 ± 4.66
LSTM+CW+D	**24.51 ± 8.93**	**66.77 ± 21.70**	**34.57 ± 11.31**	**80.17 ± 3.54**
LSTM+CW+S	31.18 ± 8.82	76.65 ± 11.75	42.83 ± 9.47	85.71 ± 3.23
LSTM+D+S	80.14 ± 29.23	5.42 ± 6.10	9.46 ± 9.74	88.93 ± 3.69
LSTM+CW+D+S	27.81 ± 10.96	70.46 ± 22.86	38.09 ± 13.26	84.89 ± 3.48

highest overall score. However, the Tukey HSD test showed that it was not significantly different to other detectors using class weighting other than FFN+CW. Furthermore, it was shown that LSTM+CW+S was significantly better than all other detectors without class weightings. These results suggest that class weighting is key to improving frame level F1. However, in contrast to the AUC results, there are few significant differences between the FFN- and LSTM-based architectures. Class weighted detectors had the highest average frame level F1 of 40%, with some folds achieving state-of-the-art performance.

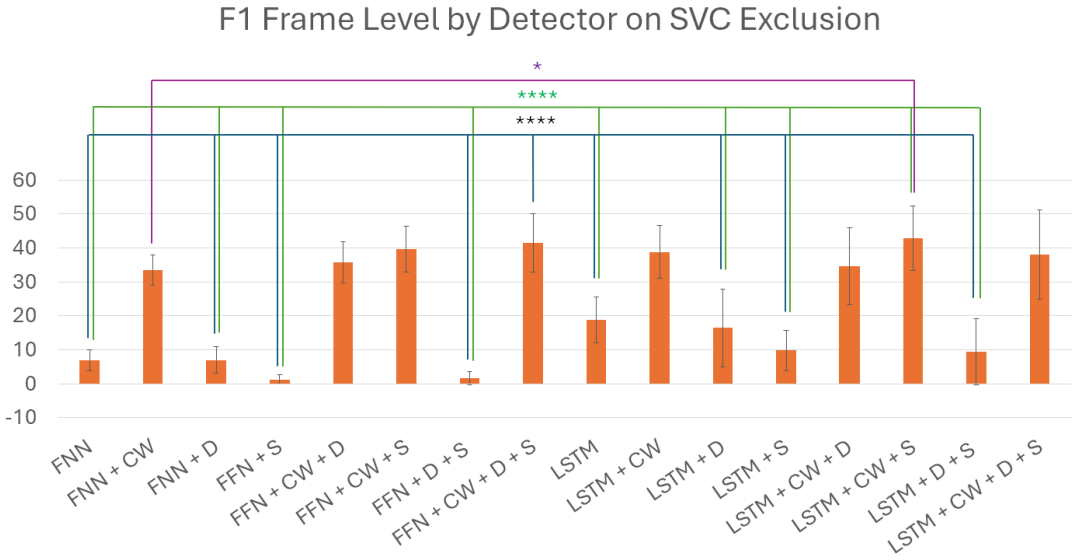

Figure 4.6: Average (STD) Frame Level F1 Achieved by Detection System on the SVC Using Exclusion Evaluation Method. Green and Pink Significance Lines Relate to LSTM+CW+S. Blue Significance Line Relates to FFN+CW+D+S (*$p < 0.05$, ****$p < 0.00005$)

For exact p-values, lower and upper confidence intervals see Table A.2

Finally, the event level performance is displayed in Table 4.5 for merging and Table 4.6 for exclusion. In agreement with the frame level F1 results, an ANOVA test once again found significant differences in detector performance depending on the evaluation method and detector type. An independent sample t-test found a significant decrease in precision when using merging as opposed to exclusion (t(574) = 579.38, $p < 0.0001$). This effect was also found in precision, with merging showing significantly poorer results than exclusion (t(574) = 289.07, $p < 0.0001$), and F1 with merging showing significantly poorer results than exclusion (t(574) = 118.68, $p < 0.0001$). These results follow the frame level precision, recall and F1 results of the evaluation methods.

As stated earlier, the exclusion-based results were carried forward for further analysis. A one-way ANOVA test was used to compare detector performance, it found that detector type had a significant effect (F(15, 272) = 317.55, $p < 0.0001$). A post-hoc Tukey HSD test was used to compare model performance with the results shown in Figure 4.7. For exact p-values, lower

and upper confidence intervals see Table A.3. The LSTM+CW detector had the highest overall F1 score, although this was not significantly better than LSTM+CW+D. However, LSTM+CW was significantly better than all the other detectors. Again, this is probably caused by the class weighting increasing recall at the cost of producing more false positives, some of which are removed by the exclusion of non-laughter clips. The best performing detector (LSTM+CW) achieves an F1 of 57.65 ± 0.78 at an event level that is on par with optimal event level performance in the field of 54-66% [51].

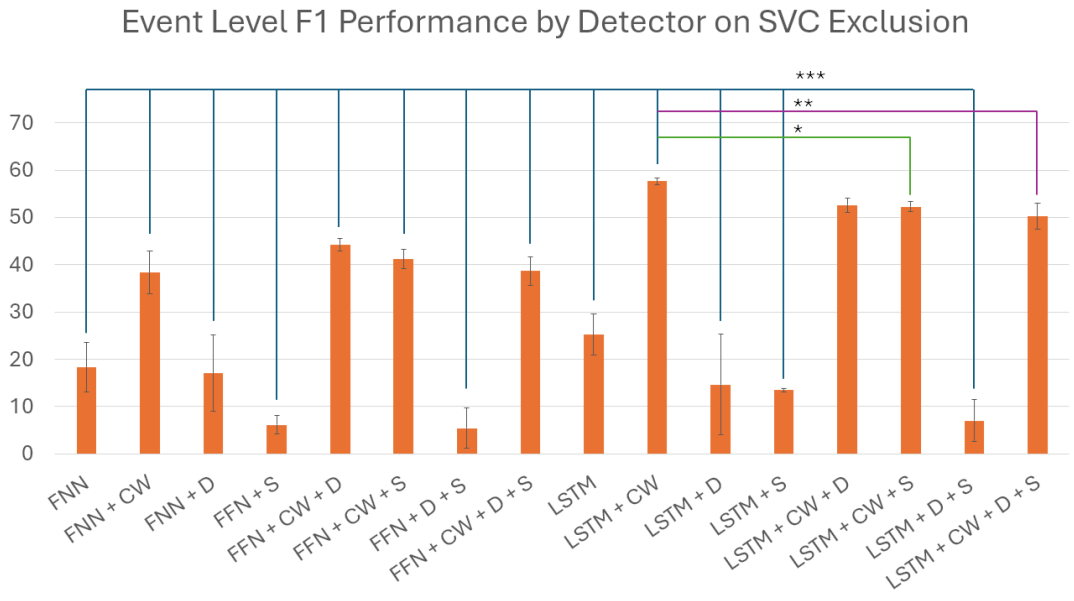

Figure 4.7: Average (STD) Event Level F1 Achieved by Detection System on the SVC Using Exclusion Evaluation. Significance Shown in Relation to LSTM+CW (*$p < 0.05$, **$p < 0.005$, ***$p < 0.0005$)

4.2.8 SSPNet Mobile Corpus

The above results show that the detection system performance can approach the state-of-the-art in frame level metrics and can be in line with the state-of-the-art in event level metrics. In this section, the detection systems are now applied to the SMC. Table 4.7 shows the attained frame level results for each detector on the SMC. Conversations in the test set were merged into one and results were calculated based on this. This decision has less of an impact compared to the SVC since all conversations have laughter, meaning none would be excluded if a conversation-by-conversation approach was used instead.

As with the SVC corpus, AUC was examined using a one-way ANOVA test, which found significant differences between detector performance ($F(15, 272) = 16.25$, $p < 0.0001$). Figure 4.8 displays the differences shown by a post-hoc Tukey HSD test between the LSTM+S detector, which achieved the overall best score, and all the others. For exact p-values, lower

Table 4.5: Event Level Performance of Each Detection Method on the SVC Using Merging of All Test Clips - Allowing the Inclusion of Clips Without Laughter. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. D: delta. S: smoothing. Bold highlights the best performing detector for each underlying architecture.

Detector	Precision	Recall	F1
FFN	40.71 ± 19.22	18.06 ± 7.26	23.55 ± 8.87
FFN+CW	11.19 ± 3.30	77.98 ± 8.12	19.38 ± 5.01
FFN+D	48.36 ± 21.79	14.47 ± 7.89	20.89 ± 10.86
FFN+S	70.14 ± 30.79	4.03 ± 1.92	7.51 ± 3.58
FFN+CW+D	14.54 ± 6.09	74.06 ± 11.71	23.78 ± 7.81
FFN+CW+S	24.72 ± 7.83	45.93 ± 16.31	31.31 ± 9.12
FFN+D+S	73.97 ± 32.98	3.40 ± 2.18	6.34 ± 4.07
FFN+CW+D+S	**32.71 ± 13.06**	**45.98 ± 15.91**	**37.01 ± 12.03**
LSTM	63.37 ± 22.33	20.84 ± 7.74	29.56 ± 8.31
LSTM+CW	18.01 ± 8.48	81.74 ± 5.01	28.58 ± 11.32
LSTM+D	73.19 ± 15.67	20.30 ± 13.31	28.57 ± 15.04
LSTM+S	75.84 ± 24.52	8.75 ± 2.59	15.32 ± 4.15
LSTM+CW+D	14.68 ± 5.53	82.12 ± 8.43	24.42 ± 8.03
LSTM+CW+S	**29.51 ± 12.91**	**68.45 ± 7.37**	**39.48 ± 13.69**
LSTM+D+S	87.12 ± 12.25	8.07 ± 5.60	14.20 ± 9.03
LSTM+CW+D+S	24.85 ± 9.22	68.10 ± 11.60	34.89 ± 10.81

Table 4.6: Event Level Performance of Each Detection Method for the SVC Using Exclusion of All Clips Not Containing Laughter. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. D: delta. S: smoothing. Bold highlights the best performing detector for each underlying architecture.

Detector	Precision	Recall	F1
FFN	55.59 ± 13.05	11.02 ± 3.43	18.28 ± 5.26
FFN+CW	26.46 ± 4.04	70.62 ± 1.87	38.38 ± 4.46
FFN+D	67.49 ± 2.27	10.11 ± 5.49	17.09 ± 8.15
FFN+S	95.24 ± 6.73	3.20 ± 1.04	6.17 ± 1.95
FFN+CW+D	**34.47 ± 3.21**	**63.46 ± 6.25**	**44.31 ± 1.35**
FFN+CW+S	45.76 ± 5.79	37.94 ± 2.17	41.21 ± 2.04
FFN+D+S	100.00 ± 0.00	2.85 ± 2.26	5.45 ± 4.21
FFN+CW+D+S	57.40 ± 7.44	30.38 ± 6.48	38.67 ± 3.01
LSTM	84.39 ± 4.48	14.88 ± 2.92	25.21 ± 4.31
LSTM+CW	48.64 ± 0.90	70.89 ± 3.00	57.65 ± 0.78
LSTM+D	91.67 ± 6.80	8.39 ± 6.38	14.66 ± 10.66
LSTM+S	100.00 ± 0.00	7.21 ± 0.23	13.46 ± 0.40
LSTM+CW+D	**41.65 ± 1.54**	**71.43 ± 1.91**	**52.61 ± 1.53**
LSTM+CW+S	58.60 ± 2.42	47.28 ± 0.79	52.31 ± 1.02
LSTM+D+S	100.00 ± 0.00	3.70 ± 2.34	7.03 ± 4.41
LSTM+CW+D+S	54.85 ± 2.26	46.47 ± 3.64	50.25 ± 2.76

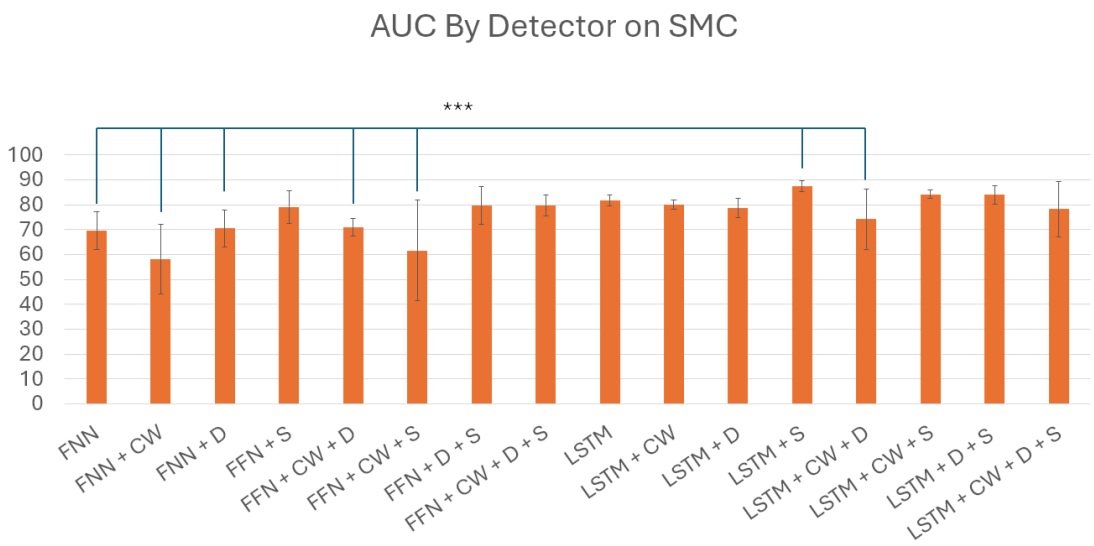

Figure 4.8: Average (STD) AUC Achieved by Detection System on the SVC. Significance Shown in Relation to LSTM+S (***$p < 0.0005$)

and upper confidence intervals see Table A.4. The tests show many of the detectors achieved AUCs that were not significantly different from the LSTM+S. All the other detectors were shown to perform significantly worse than the LSTM+S system. Only one LSTM-based detector (LSTM+CW+D) showed a significantly poorer AUC performance. Meanwhile, the FFN-based detectors showed weaker performance in all but three cases, suggesting that the LSTM networks are better suited to a type 3 task. Comparing the best performing detector on the SVC (LSTM+D+S exclusion: 88.93 ± 3.69) and the SMC (LSTM+S: 87.48 ± 2.19), an independent sample t-test found no significant difference between performance (t(34) = 2.06, $p = 0.16$). This suggests that the optimal methods found in the literature to date can effectively operate on the larger and more difficult type 3 task presented by the SMC.

Further to the AUC results, the frame level precision, recall and F1 were examined. An ANOVA test found significant differences in performance between detectors at an F1 level (F(15, 272) = 13.70, $p < 0.0001$). A post hoc-Tukey HSD test was used to examine differences between the detectors. Figure 4.9 shows the significant differences found between the FFN+CW+D+S detector and all the others. For exact p-values, lower and upper confidence intervals see Table A.5. The FFN+CW+D+S system was found to have the highest overall performance. However, Multiple detectors were shown to have no significant difference from it. In terms of LSTM-based detectors, LSTMs with class weighting performed significantly better than those without. This suggests class weighting is a requirement for LSTM performance. Meanwhile, using the FFN architecture, the results are less clear. FFN detectors with smoothing perform significantly worse, except when coupled with both class weighting and delta, in which case they achieve results in line with the best detectors on the corpus. Furthermore, just

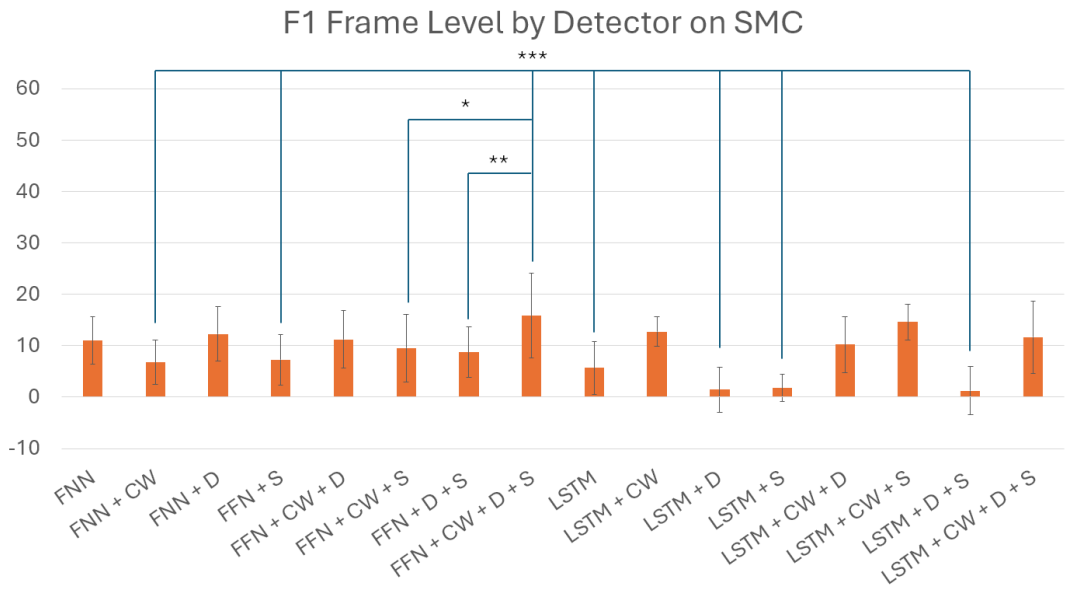

Figure 4.9: Average (STD) Frame Level F1 Achieved by Detection System on the SMC. Significance Shown in Relation to FFN+CW+D+S (*$p < 0.05$, **$p < 0.005$, ***$p < 0.0005$)

using class weighing with the FFN on the SMC results in significant drops in performance. Now comparing the best F1 results on the SMC (FFN+CW+D+S: 15.86 ± 8.27) to those on the SVC (FFN+CW+D+S exclusion: 41.43 ± 8.58), an independent sample t-test found significant differences, with the SMC detectors performing significantly worse than the SVC detectors (t(34) = 82.87, $p < 0.0001$). These results contrast to those found using AUC and suggest that the detection methods previously developed in the field are less effective when used on the SMC when F1 is considered.

Finally, the detector results based on the SMC at an event level are shown in Table 4.8. A one-way ANOVA test found significant differences in terms of F1 between the models (F(15, 272) = 9.65, $p < 0.0001$). A post-hoc Tukey HSD test was used to examine differences between detectors, the results of which can be seen in Figure 4.10. For exact p-values, lower and upper confidence intervals see Table A.6. The overall best performing detector was the LSTM+CW+S. The event level F1 is generally in agreement with the frame level for LSTMs, with LSTM-based detectors with class weighting significantly outperforming all the cases without it (except in the case of LSTM and LSTM+CW+D). However, the advantage of the LSTM-based detectors over the FFN is reduced at an event level. Only three of the FFN detectors performed significantly worse than the LSTM+CW+S detector; in all cases, these involved coupling the FFN with class weighting. A second point of interest is the precision, recall and F1 for the LSTM, LSTM+D, LSTM+S and LSTM+D+S. They have relatively good precision and recall averages, but their average F1 is lower. This fact seems unlikely since F1 is the harmonic mean of precision and recall. However, all four models have high standard deviation in relation to both precision and recall. This, coupled with the low average F1, suggests that, although both precision and recall

Table 4.7: Frame Level Performance of Each Detection Method for the SMC. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. D: delta. S: smoothing. Bold highlights the best performing detector for each underlying architecture.

Model Type	Precision	Recall	F1	AUC
FFN	21.49 ± 9.05	9.62 ± 5.07	10.99 ± 4.62	69.63 ± 7.50
FFN+CW	8.25 ± 11.10	46.99 ± 29.86	6.75 ± 4.27	58.21 ± 14.02
FFN+D	18.80 ± 9.20	10.45 ± 4.91	12.29 ± 5.34	70.52 ± 7.49
FFN+S	43.65 ± 31.75	4.42 ± 3.40	7.28 ± 4.92	79.05 ± 6.70
FFN+CW+D	6.59 ± 3.43	40.95 ± 19.05	11.23 ± 5.65	70.89 ± 3.52
FFN+CW+S	7.63 ± 7.45	44.52 ± 29.20	9.54 ± 6.55	61.59 ± 20.16
FFN+D+S	43.57 ± 23.89	5.16 ± 3.16	8.75 ± 4.90	79.78 ± 7.58
FFN+CW+D+S	**10.03 ± 5.59**	**43.70 ± 20.93**	**15.86 ± 8.27**	**79.79 ± 4.24**
LSTM	47.41 ± 26.19	3.27 ± 3.21	5.69 ± 5.15	81.73 ± 2.05
LSTM+CW	**7.09 ± 1.81**	**69.77 ± 10.13**	**12.73 ± 2.83**	**79.96 ± 1.78**
LSTM+D	17.88 ± 32.01	1.12 ± 3.77	1.44 ± 4.43	78.69 ± 3.96
LSTM+S	43.39 ± 38.98	0.95 ± 1.49	1.79 ± 2.71	87.48 ± 2.19
LSTM+CW+D	5.86 ± 3.55	63.90 ± 26.09	10.21 ± 5.44	74.17 ± 12.14
LSTM+CW+S	8.25 ± 2.30	73.30 ± 11.72	14.62 ± 3.50	84.16 ± 1.70
LSTM+D+S	8.31 ± 24.96	0.76 ± 2.96	1.22 ± 4.69	83.95 ± 3.66
LSTM+CW+D+S	7.07 ± 5.66	65.97 ± 28.20	11.65 ± 6.99	78.23 ± 11.12

can be high, neither of them are both high at the same time, suggesting an unreliability in some LSTM detectors. Finally, the best performing SVC event level detector (LSTM+CW: 57.65 ± 0.78) was compared with the best performing SMC event level detector (LSTM+CW+S: 26.30 ± 5.14) by using an independent sample t-test, which found a significant difference in performance ($t(34) = 654.54$, $p < 0.0001$). This again supports the above conclusion that the current methods in the field are ineffective when applied to a type 3 dataset.

An immediate issue with the above results is the disagreement between AUC and F1. With regard to the SVC results, there is a ∼20% difference between AUC and F1. For the SMC, this difference is inflated to a ∼50-60% decrease. Table 4.1 shows that AUC is a widely used metric in the laughter detection field, especially when evaluating performance on the SVC. Despite the widespread use of this metric in the field, its validity has previously been called into question [78]. This is due to how it is calculated, specifically that specificity includes true negatives in its calculation. To examine this point further, consider three cut-off conditions: low, medium and high. At a high cut-off, there will be very few false positives and a high number of true negatives - leading to specificity being close to zero. For both the medium and low cut-offs, in comparison, there will be more false positives. However, due to the number of non-laughter frames, the ratio of true negatives to false positives is likely to remain imbalanced, with many more true negatives than false positives; thus, still resulting in a specificity value close to zero. Therefore, when plotting the ROC, the specificity value remains close to zero for most cut-off values. As specificity is plotted along the x-axis, this in effect pulls the curve to the left and, as a result, inflates the AUC value above what is reasonable.

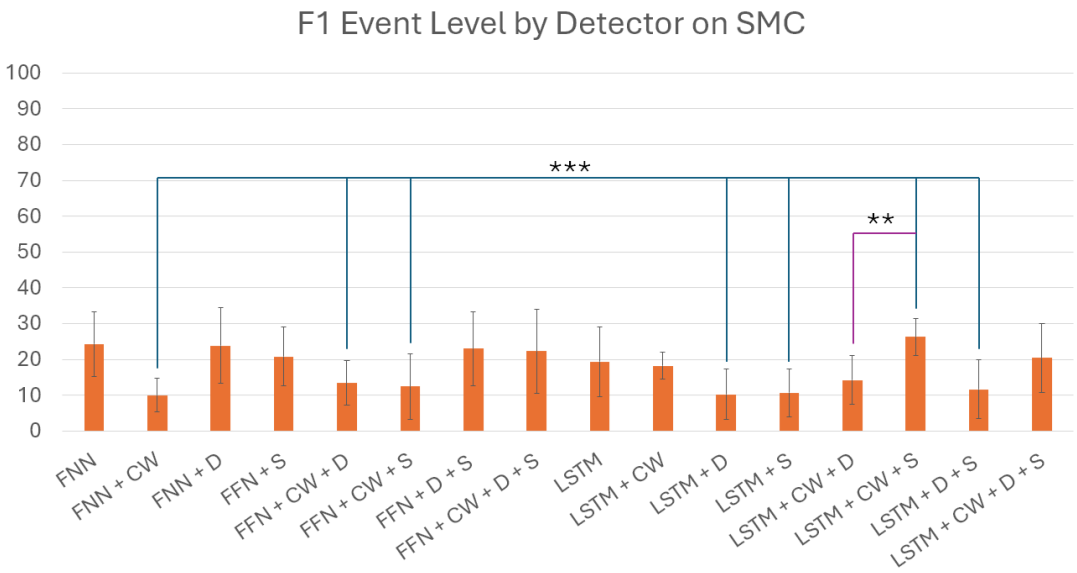

Figure 4.10: Average Event Level F1 Achieved by Detection System on the SMC. Significance Differences Shown in Relation to LSTM+CW+S (**$p < 0.005$, ***$p < 0.0005$)

Table 4.8: Event Level Performance of Each Detection Method for the SMC. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. D: delta. S: smoothing. Bold highlights the best performing detector for each underlying architecture.

Detector	Precision	Recall	F1
FFN	**21.55 ± 10.50**	**42.89 ± 16.93**	**24.30 ± 9.11**
FFN+CW	8.59 ± 9.36	70.02 ± 29.89	10.06 ± 4.60
FFN+D	17.69 ± 8.89	46.77 ± 20.55	23.87 ± 10.49
FFN+S	50.45 ± 23.80	15.85 ± 6.84	20.86 ± 8.28
FFN+CW+D	7.92 ± 4.52	79.05 ± 25.12	13.45 ± 6.13
FFN+CW+S	13.00 ± 15.31	39.71 ± 24.09	12.52 ± 9.15
FFN+D+S	41.52 ± 20.82	17.55 ± 7.92	23.01 ± 10.32
FFN+CW+D+S	17.00 ± 9.88	49.74 ± 18.39	22.32 ± 11.67
LSTM	49.93 ± 26.12	28.27 ± 31.55	19.36 ± 9.76
LSTM+CW	10.23 ± 2.39	86.94 ± 7.70	18.18 ± 3.76
LSTM+D	22.56 ± 30.80	67.56 ± 40.81	10.28 ± 7.07
LSTM+S	62.89 ± 31.76	19.07 ± 30.79	10.72 ± 6.69
LSTM+CW+D	8.06 ± 4.28	76.73 ± 26.98	14.28 ± 6.86
LSTM+CW+S	**16.41 ± 3.95**	**70.62 ± 9.73**	**26.30 ± 5.14**
LSTM+D+S	32.03 ± 38.32	61.02 ± 39.66	11.68 ± 8.18
LSTM+CW+D+S	12.77 ± 7.38	63.66 ± 24.21	20.41 ± 9.68

Table 4.9: Frame Level Performance of LSTM+CW+S for Each Metric by Group Split

Group	Precision	Recall	F1	AUC
Caller	10.58 ± 9.55	77.80 ± 20.66	17.00 ± 12.78	80.54 ± 9.59
Receiver	20.46 ± 14.30	66.99 ± 23.12	28.20 ± 16.28	88.67 ± 5.82
Male	14.45 ± 14.75	67.85 ± 26.75	20.32 ± 16.76	84.33 ± 8.35
Female	16.49 ± 11.37	76.50 ± 16.97	24.67 ± 14.30	84.84 ± 9.35
MM	8.51 ± 10.13	62.73 ± 25.91	13.19 ± 13.73	84.26 ± 6.91
FF	11.88 ± 8.10	78.37 ± 13.25	19.07 ± 11.21	87.21 ± 5.75
MF	9.69 ± 7.27	72.56 ± 18.61	16.01 ± 9.81	84.02 ± 6.25

Table 4.10: Event Level Performance of LSTM+CW+S for Each Metric by Group Split

Group	Precision	Recall	F1
Caller	11.51 ± 8.60	81.07 ± 12.52	18.96 ± 12.07
Receiver	25.19 ± 17.08	58.33 ± 20.36	31.34 ± 16.49
Male	15.04 ± 12.90	63.96 ± 21.39	21.21 ± 14.32
Female	21.14 ± 16.30	74.52 ± 18.11	28.47 ± 16.08
MM	21.85 ± 9.62	61.18 ± 13.03	31.62 ± 11.57
FF	21.33 ± 2.84	62.78 ± 4.89	31.81 ± 3.64
MF	21.82 ± 5.49	63.64 ± 7.41	32.10 ± 6.21

Furthermore, the above issue results in the AUC score being mostly dependent on the sensitivity value. It was shown above that detectors that produce more false positives are not penalised heavily. However, the presence of more positive cases raises the chance of identifying true positives. The creation of more false positives has little impact on the final score, while true positives have a greater impact. This means that, if a detector were to take more risky chances, then both the true positive and false positive scores would increase. While the false negative and true negative scores would decrease. However, because true negative scores begin so much higher than all other scores, this has little negative impact on the specificity score and so leads to an overall net positive. This means that AUC encourages the increasing of sensitivity almost universally compared with specificity and does not give a clear view on how effective a given detector is.

4.2.9 Performance Analysis

A performance analysis on the best performing detector was carried out in an attempt to understand when a specific approach works and when it does not. The LSTM+CW+S detector was selected for this analysis. Initial testing compared the detector's performance by gender, role and conversation pairing. Frame level results from this analysis are displayed in Table 4.9 with event level results shown in Table 4.10.

By first addressing the effect of role, an independent sample t-test was run that compares caller and receiver scores for each metric. It was found that there was a significant difference

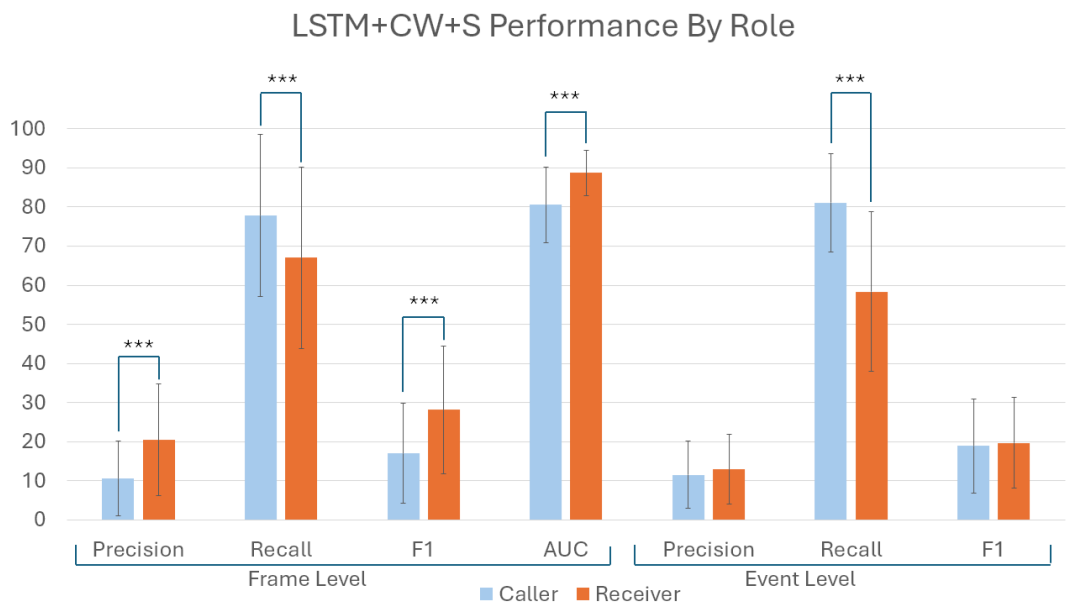

Figure 4.11: LSTM+CW+S Performance by Role on the SMC (***$p < 0.0005$)

between roles across five out of the seven metrics (see Figure 4.11 and for exact t statistics and p values Table A.7). Receivers saw a significantly better performance than callers in frame level AUC, precision and F1. However, in frame and event level recall callers showed significantly better performance than receivers. These results are surprising given that both callers and receivers were treated the same across all the conversations, with the same type of microphone and mobile phone being used.

Now examining the effect of the speaker's gender. Independent sample T-tests were again run to compare the performance of each metric (the results are shown in Figure 4.12 for exact t statistics and p values Table A.8). For frame level recall, F1, event level precision, recall and event level F1 female speakers showed significantly better performance than male speakers. There were no significant differences in frame level precision or AUC between female and male speakers.

These gender differences are probably due to the underlying distribution of laughs in the dataset. As described in Section 3, women laugh significantly more than men during the conversations. This issue of imbalances in the training data has caused similar issues for performance in other applications. In face recognition, it was shown that systems with high effectiveness experienced performance drops when applied to races, age groups and genders that were under-represented in the training set [79]. Mitigation of this issue is possible by training classifiers on datasets that represent a more under-performing group [80, 81]. As such, the gender performance imbalance in the current system would probably be overcome through the creation of more data with male laughter.

In addition to performance differences at the level of a speaker's gender, the effect of gender pairing on a conversation was investigated. One-way ANOVA tests were used to compare the

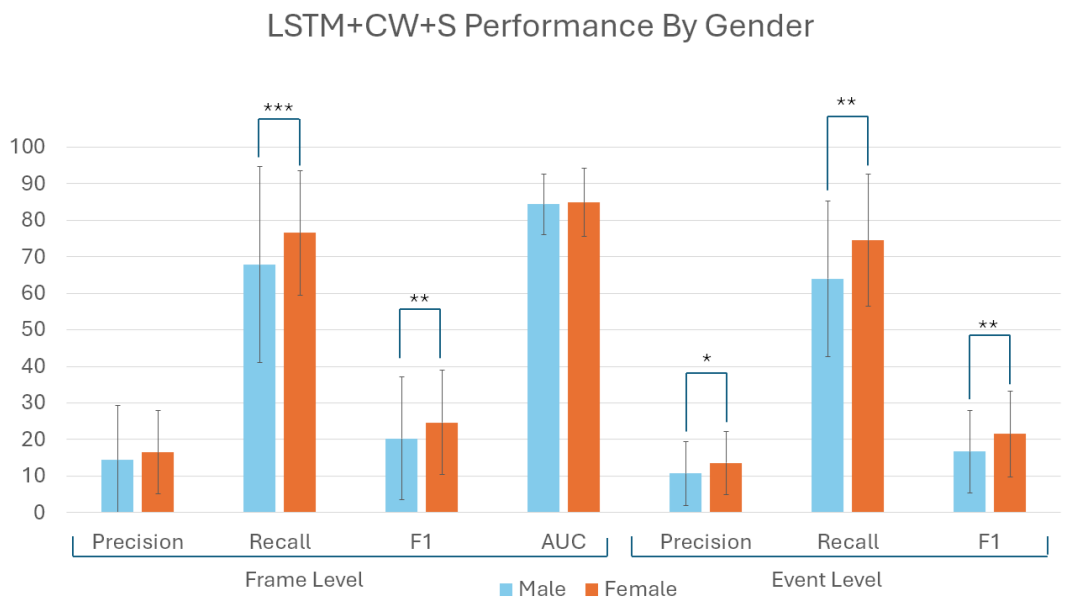

Figure 4.12: LSTM+CW+S Performance by Gender on the SMC (*$p < 0.05$, **$p < 0.005$, ***$p < 0.0005$)

three gender pairings on each metric (the results are shown in Figure 4.13, for exact confidence intervals and p values see Table **??**). No significant differences were found due to pairing for frame level precision (F(2, 177) = 2.07, $p = 0.13$), nor were significant differences found for event level precision (F(2, 177) = 0.12, $p = 0.89$), recall (F(2, 177) = 1.18, $p = 0.31$) or F1 (F(2, 177) = 0.067, $p = 0.94$). However, significant differences were found for frame level recall (F(2, 177) = 7.64, $p = 0.0007$) with a post-hoc Tukey HSD test showing MM pairings performed significantly worse than FF pairings and that MM pairings performed significantly worse than MF pairings. However, there were no significant differences from MF to FF.

Regarding frame level F1, a one-way ANOVA test found significant differences due to pairing (F(2, 177) = 3.19, $p < 0.044$), with a post-hoc Tukey HSD test showing MM pairings had significantly worse performance than FF pairings. However, MM pairings were not significantly different from MF pairings. Nor were MF pairings significantly different from FF pairings.

Regarding AUC, a one-way ANOVA test found significant differences due to pairing (F(2, 177) = 4.51, $p < 0.012$), with a post-hoc Tukey HSD test showing that MF pairings were significantly worse than FF pairings. However, MM pairings and FF pairings were not significantly different. Nor were MM pairings and MF pairings. The gender effects described above go some way towards explaining these differences. Conversations that only contained female speakers generally saw better results because the detectors perform best for this gender.

The selected detector has relatively good recall (frame = 73.30 ± 11.72, event = 70.62 ± 9.73). However, its precision is poor (frame = 8.25 ± 2.30, event = 16.41 ± 3.95). This suggests that this detector creates too many false positives. Although smoothing is shown to improve this problem by improving precision by ∼6% at an event level compared with the pre-smoothing

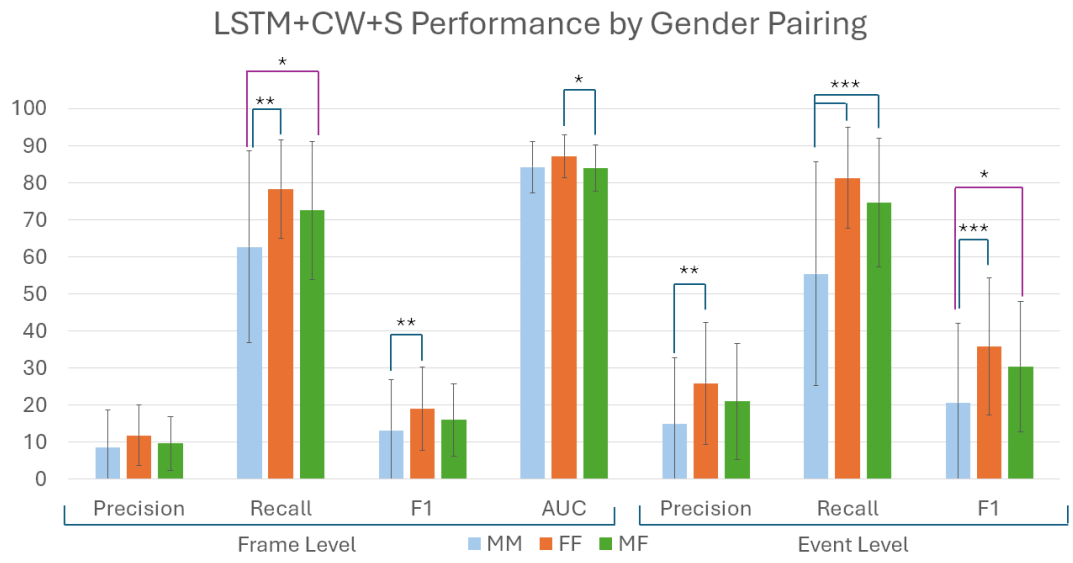

Figure 4.13: LSTM+CW+S Performance by Gender Pairing on the SMC (*$p < 0.05$, **$p < 0.005$, ***$p < 0.0005$)

detector, these gains come at the cost of only a ∼16% recall. Figure 4.14 shows that, although the Hamming window is effective, the gains made reach a ceiling, in terms of F1, due to the effect of recall dropping at around twice the rate of the rise in precision.

Given the above results, false positives have the largest impact on detector performance. Table 4.11 displays the average count of audio events that caused false positives across the 6 folds and 3 repetitions. A chi-square goodness of fit test found that the distribution of event level false positives differed significantly from the underlying distribution of events (X^2 (4, N = 120) = 10044.22, $p < 0.001$). Pearson residuals showed this effect was driven by three large differences. Pauses accounted for less false positives than expected with a residual value of -14.27 ± 1.68. This is surprising given previous results in the field, which found breathing (classified in this dataset as pauses) to be commonly mistaken for unvoiced laughter [33]. It is possible that there is less unvoiced laughter in the SMC, so the detectors were maybe not as badly influenced as in other works. However, no data exists to confirm or refute this. Receiver fillers also have a large negative residual (-4.95 ± 3.85), with a similar although less strong effect seen for caller fillers, suggesting that fillers are mistaken for laughter less often than expected. This suggests that laughter and fillers are sufficiently different to not be confused with each other, even with the poorly performing detectors. This finding is also replicated in the field more generally [27]. Finally, the largest residual was found for caller speech (17.95 ± 1.94). Coupled with the negative residual associated with receiver speech (-6.14 ± 3.73), this suggests that caller speech is more often confused with laughter than was expected. This result offers support for the findings above, in terms of performance differences by role, with caller speech seeming to differ from receiver speech; the former also causes more false positives. Finally,

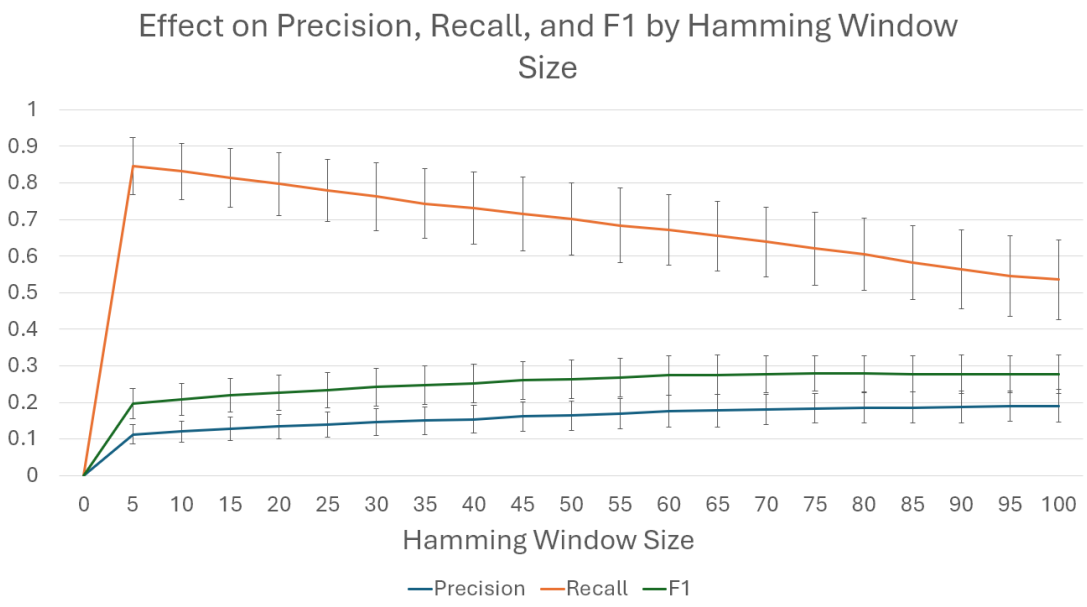

Figure 4.14: Effect of Different Hamming Window Sizes on Precision, Recall and F1 at an Event Level

although the Pearson residual for the caller is larger compared to the receivers, both callers and receivers contribute the most to the false positives, showing that speech is the main source of the false positives. This must be addressed to improve laughter detectors.

Table 4.11: False Positives by Class. Class_r: receiver. Class_c: caller

	Observed	Expected	Residual	Pearson Residual
Pause	38.28 ± 33.65	276.07 ± 88.65	-237.79 ± 62.52	-14.27 ± 1.68
Receiver	301.33 ± 96.03	202.22 ± 138.24	-99.11 ± 57.40	-6.14 ± 3.73
Caller	659.39 ± 148.23	337.23 ± 116.98	322.16 ± 45.90	17.95 ± 1.94
Filler_r	24.61 ± 33.84	63.85 ± 22.04	-39.24 ± 32.41	-4.95 ± 3.85
Filler_c	52.00 ± 30.50	75.61 ± 32.28	-23.61 ± 18.12	-2.63 ± 1.97
Overlapping Speech	314.94 ± 92.32	261.10 ± 103.16	53.85 ± 50.06	3.74 ± 3.49
Overlapping Breakdown:				
Receiver - Caller	197.17 ± 78.72	131.60 ± 61.06	65.57 ± 35.65	5.83 ± 3.59
Filler_r - Caller	11.44 ± 8.88	8.87 ± 6.95	2.57 ± 5.02	0.90 ± 1.55
Filler_c - Receiver	15.78 ± 10.83	18.37 ± 9.60	-2.59 ± 5.05	-0.62 ± 1.14
Receiver - Bc_c	24.17 ± 13.23	22.81 ± 10.74	1.36 ± 9.82	0.38 ± 2.07
Caller - Bc_r	7.47 ± 4.55	8.89 ± 8.01	1.42 ± 5.23	0.42 ± 1.52

A final possibility for the event level false positives is that the detectors are ‘near missing’ the events. Given that the peaks are merged across time, it is possible that a correct detection at the edge of an event and a incorrect detection next to an event, may result in a merged peak that is slightly off from the event. The first step to test this possibility is to define a ‘near miss’. The average gap between laughter events is 41.66 $\pm$ 64.00s with a median of 18.27 s. This suggests that the furthest a false positive could generally be from a laughter event is 9.10 s.

Figure 4.15: Number of False Positives (Average (STD) Per Fold) by Their Distance, in Seconds, from a Laughter Event (95% of False Positives Shown)

A near miss was defined as anything within 5% of this distance, or 0.46 s. If false positives were distributed randomly, it would therefore be expected that 5% of false positives would exist within this distance. Figure 4.15 shows the average number of false positives by distance from a laughter event. It clearly shows that the majority of the missed events are within a few seconds of the laughter events. Table 4.12 shows the percentage of false positives that fell within a given time and the associated event level precision, recall and F1 if those near missed were reclassified as true positives. It shows that, by reclassifying near misses as true positives, the event level F1 rises by $\sim$5% to 32.17 $\pm$ 5.78. This rise is caused by increases in both precision and recall, which suggests that near misses existed close to both already detected events and missed events. However, despite the increases in all event level metrics with the reclassification of near missed false positives, the detector's F1 remains below 40%. This shows that it is still ineffective; false positives remain the greatest issue.

Table 4.12: Percentage of False Positives Within a Given Time of Laughter Events and the Associated Precision, Recall and F1 if They Were Reclassified as True Positives

Time (s)	Percentage of False Positives	Precision	Recall	F1
Original	-	16.41 ± 3.95	70.62 ± 9.73	26.30 ± 5.14
0.1	9.90 ± 2.44	17.22 ± 4.01	72.74 ± 9.86	27.51 ± 5.18
0.46	23.99 ± 5.55	20.49 ± 4.64	78.89 ± 9.27	32.17 ± 5.78
0.5	25.06 ± 5.72	20.95 ± 4.65	79.52 ± 9.06	32.79 ± 5.74
1	36.99 ± 8.21	24.34 ± 5.12	83.90 ± 8.28	37.36 ± 6.12
2	53.21 ± 10.08	28.94 ± 5.80	88.22 ± 7.20	43.17 ± 6.60

4.2.10 Conclusion

This chapter replicated the work undertaken in laughter detection for the SVC. It was successful in applying the methods developed in the field to laughter detection in terms of AUC and precision, recall and F1 at both a frame and event level. The chapter then extended this work by applying the same methods to the SMC. Using the SMC for laughter detection ensured that two common experimental constraints were removed, those being the ratio of laughter to non-laughter and the total audio time of the corpus. It was clearly shown that, in terms of precision, recall and F1 at both the frame and event level, that the state-of-the-art methods reach a performance ceiling of $\sim$25%. These experiments were carried out with the goal of answering RQ1: Are state of the art laughter detectors effective when common experimental constraints are removed? The results provide strong evidence that state-of-the-art methods are not effective.

This was shown to be driven by a decrease in precision due to false positives caused by speech. This issue is directly caused by the removal of the constraints on total audio time and the ratio of laughter to non-laughter. Longer audio and a lower percentage of laughter inevitably leads to more speech frames. To improve the performance of laughter detectors in a type 3 context, therefore, it is imperative to develop methods that can reduce these false positives. Chapter 5 addresses this major issue with the development of novel approaches to laughter detection.

Chapter 5

Improving Laughter Detection

5.1 Motivation

Chapter 4 demonstrated that the current methods used in the laughter detection field suffer significant drops in performance when applied to a type 3 task. In this chapter, novel approaches are created and tested in an attempt to mitigate these drops in performance. The chapter is considered in two-parts. The first leverages automatic speech recognition (ASR) and voice activity detection (VAD) in an attempt to reduce false positives and carry out automatic undersampling to mitigate the class imbalance issue. This work was carried out to provide answers to RQ2: Can the incorporation of linguistic data lead to improvements in laughter detection? The second part extends the detectors to address fillers and back-channel events to also address the class imbalance issue. Other works have examined laughter and filler detection concurrently [26,27,29,41]; however, one study also included back-channels [51]. Furthermore, this work is the first to apply multi-cue detection to a type 3 task. This portion of the work directly addresses RQ3: What is the effect of broadening the scope of laughter detectors to include multiple cues?. Section 5.2 explores this multi-cue work, while the remainder of this section focuses on the ASR/VAD related approaches.

The performance analysis at the end of Chapter 4 showed that the best performing models had high recall but poor precision. This precision issue was caused by false positives, which were shown to be generally caused by speech events. This led to the creation of RQ2. It was hypothesised that by providing linguistic information to the laughter detectors these false positives could be removed. Exactly what information to provide and how to provide it was the subject of the research and experiments of the first half of this chapter.

ASR information has been used in the field for laughter detection purposes. In one study, the authors used the entropy in ASR predictions to mask the posteriors produced by a laughter detection system [41]. Where the entropy in the ASR predictions was higher, it was hypothesised that laughter would occur. This method led to improvements in laughter detection AUC of between 5-10%. In a separate study, an ASR system was used to produce phoneme predictions,

a second-stage then attempted to learn patterns in these phonemes, which aligned with laughter [37]. This approach achieved a precision and recall of around 90% in a type 1 task. In the present work, ASR is leveraged in multiple novel ways and applied to a type 3 task for the first time.

Further to the above issue of false positives, there remains the issue of large class imbalances in laughter detection. Large class imbalances present obstacles to leveraging machine learning algorithms [82]. In laughter detection, it has been shown that undersampling the data, which leads to more balanced classes, improves laughter detection in the SVC [26]. However, no method of automatic undersampling was proposed. Developing an automatic method, which does not need to know a-priori the labels of the data, could improve laughter detection results. Four ASR-based systems were developed and termed as follows:

- anti-detector,
- undersampling,
- feature vector extension,
- confidence-based alteration.

For each of the above methods, the approach as described in Section 4.2 is modified. Figure 5.1 shows the general approach along with each of the additional methods tested here. Building an ASR system was outside the scope of this project and, as such, an off-the-shelf system was used. ASR was performed using VOSK [83]. VOSK accepts audio files as input and returns the start and end time of each word detected. For each word detected, a list of possible words that might have been spoken is generated. Each word in this list has an estimated risk value attributed to it, calculated using minimum Bayes risk (MBR). VAD was detected using ref. [84]. For each audio file, the system returns time stamps for the estimated start and end times for all the voiced segments of that audio file. All frames that started within these start and end times received a VAD value of 1, otherwise a value of zero was assigned. The following subsections describe how each method operated. Results are displayed and discussed at the end of the chapter.

5.1.1 Anti-Detector

The anti-detection system attempts to solve both the class imbalance issue and the false positive issue. The system works by identifying sequences in the audio in which the VAD system detects vocal activity, but the ASR system does not output any words. In these sequences, the VAD system would be used to identify all the vocalisations and ASR would be used to remove vocalisation detections that were words, leaving behind only paralinguistic utterances. This anti-detection system would then be trained to discriminate between all the remaining utterances to

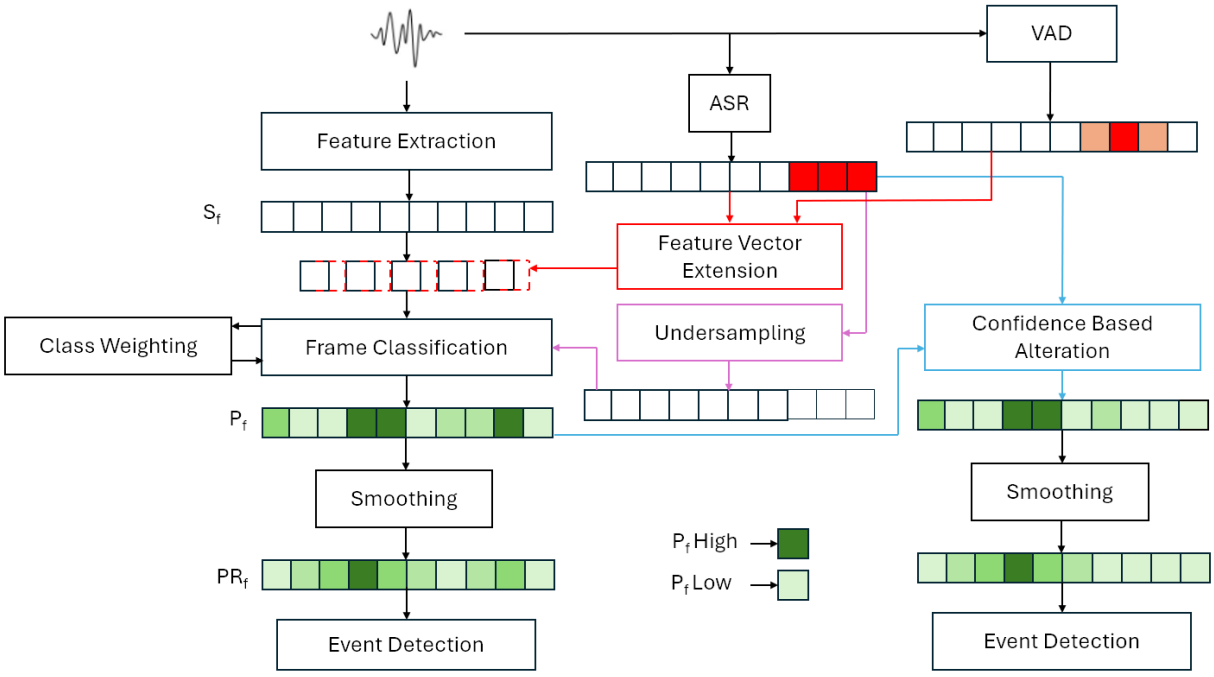

Figure 5.1: Modified Version of the Laughter Detection System. Red Route Shows Feature Vector Extension. Blue Route Shows Confidence-Based Alteration. Purple Route Shows Undersampling

group them into the various paralinguistic classes. By removing all ASR detections, a majority of the false positives created by speakers speech should also be removed, leading to an increase in precision. Furthermore, by only training on frames that the VAD system identified as speaker vocalisations, it was theorised that the class imbalance issue would also reduce.

The success of the anti-detector system hinges on the presupposition that paralanguage exists in frames where VAD occurs, independent of ASR detection. Before attempting to train/test automatic detectors, this assumption was tested using the ground truth labels.

In total, there were 4533002 frames in the dataset, of these 52.52% were labelled as containing words by the ASR system, while 54.11% were detected as frames that contained voicing by the VAD system. In terms of overlap, 55.03% of frames with a word associated with it also had a voice activity detected. This suggests there was weak agreement between the two systems. Some disagreement is expected given that the VAD system is attempting to detect all the vocalisations instead of just linguistic ones. However, all ASR frames should be detected by VAD as well.

In terms of frames where VAD occurred independently of ASR (which is where paralanguage was initially theorised to occur), the actual labels of these frames can be broken down as follows: 75.07% were speech frames, 16.50% were pauses, 4.34% were laughter, 2.27% were fillers and 0.80% were backchannels. The 4% of laughter that occurred in the independent VAD frames represents 38.70% of all the laughter frames. This means that, on a frame level, the anti-detector was not a viable solution. This is because, before even beginning to detect laughter

in the independent VAD frames, 61.30% of laughter would already be mislabelled as a false negative.

However, it was possible that the anti-detector system could work at an event level. For example, if the 38% of laughter frames were spread throughout all the laughter events, it would be possible to detect them in the final system. To test this theory, each laughter event was examined and tested for the presence of VAD, independent of word detection. Only 39.86% of the laughter events contained at least one frame of independent VAD detection. Unfortunately, these results replicated the issue seen above that around 60% of the events would be false negatives before any detection took place, essentially rendering the anti-detector useless.

5.1.2 Undersampling

Undersampling has been shown in the literature to improve results in paralinguistic detection, with ref. [26] showing some of the best results in both the BEA and the SVC, with F1 scores for both in the range of 75-95%. However, these results were demonstrated on an undersampled test set. The authors selected all the laughter segments and then extracted an equal number of non-laughter segments, essentially ensuring that the class-distribution in both the training and test sets was 50/50. Although undersampling in this manner is accepted practice when creating a training set, the method is rarely applied to the test dataset. Undersampling the test set in this way makes the task significantly easier for a detector to perform, meaning results are often unjustifiably inflated. Furthermore, by undersampling the test set, it becomes impossible to generalise the models' efficacy on real-world audio since undersampling of the real-world data would be impossible. The proposed system leverages the idea that the words detected by the ASR system may be wrong but will probably line up with where actual words occurred.

In this system, during training, all the frames that overlapped with words output by the ASR were removed from the dataset and the detectors were trained on what remained. The remaining frames should be comprised of silence, paralanguage and any linguistic vocalisations the ASR missed. After being trained on the undersampled dataset, the system was tested by providing predictions on the entirety of the test dataset. However, all the frames in the test set that were labelled as words by the ASR had their predictions set to zero.

Around 82% of the audio data is speech. The ASR system labelled 61.20% of that speech data correctly as containing speech. Only 5% of the frames that the ASR detected as having words were incorrect. Some of these incorrect labels did occur on laughter frames. A total of 9.42% of the laughter frames were labelled as speech by the ASR. At an event level, 126 events had more than 50% of their frames labelled as speech, with only 22 of those being fully labelled as speech. This means that 93.01% of laughter events had at least 50% of their frames untouched by the ASR, with a total of 73.06% of events having none of their frames labelled as speech. These results supported the idea of using the ASR system to undersample the data prior to training. This method enabled the removal of around 50% of the training data while only removing

a small amount of laughter and was a method of undersampling that was independent of human coded labels, meaning it could be used in real world applications. These results showed that the undersampling technique maintains the majority of the laughter data while removing a large chunk of the overall data. All the systems that utilise undersampling are henceforth labelled with U.

5.1.3 Feature Vector Extension

In this detection method, the ASR information was included in the feature vectors created in the feature extraction step. This method does not address the class imbalance issue, as none of the underlying data is removed. However, it was theorised that the additional information would enable the neural networks to reduce the false positives associated with speech. In both of the above methods, mistakes in the ASR and VAD systems can lead to the removal of laughter frames/events. In this method, there is a chance that, in those cases where mistakes are made by ASR and VAD, the remainder of the feature vector's information may enable the system to still identify it.

To extend the feature vectors in this way, the ASR and VAD information was summarised in terms of two features: whether there was speech or not (1 or 0) and the MBR confidence in the estimated most likely word (a value between 0 and 1). Aside from the alteration of the size of the input feature vector, the remainder of the approach is unchanged. All the systems that use the extended feature vectors are henceforth labelled with E.

5.1.4 Confidence-Based Alteration

In this system, the ASR information is incorporated through alteration of the posterior probabilities output by each detector. The alteration is based on the MBR estimated for each frame. Alterations are undertaken using the following equation:

$$p_a = p_e(1 - p_w), \tag{5.1}$$

where p_a is the probability for each frame after adjustment, p_e is the original estimated posterior probability that a frame is laughter and p_w is the estimated MBR probability that the detected word is the actual word. The MBR adjustment is carried out before the convolution with the Hamming window. This method ensures that, in cases where the ASR is certain or nearly certain of a detected word, the laughter detector will zero the probability for the necessary frames. However, in the case where the ASR is less certain, as is predicted to be the case when the ASR mistakenly detects paralanguage as language, these probabilities will be lowered but not zeroed. This approach has a similar advantage as the feature vector extension system, in comparison to both undersampling and anti-detection, in that it incorporates an amount of uncertainty into the

Table 5.1: Hyper-Parameter Optimisation Results for Each Detector and Metric Using ASR Approaches. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. U: undersampling. E: feature vector extension. C: confidence-based alteration. All FFN detectors had 2 FFN hidden layers. All LSTM detectors had 2 FFN hidden layers and 2 LSTM hidden layers. All hidden layers had 100 nodes

Model Architecture	Window Size			Cut-off M
	AUC	F1 (frame)	F1 (event)	
FFN+CW	No Effect	No Effect	31	10
FFN+CW+U	31	No Effect	31	1
FFN+CW+C	11	No effect	11	1
FFN+CW+E	41	No effect	71	10
LSTM+CW	61	No Effect	31	10
LSTM+CW+U	No Effect	No Effect	41	1
LSTM+CW+C	51	No Effect	11	1
LSTM+CW+E	No Effect	No Effect	81	10

ASR prediction rather than removing everything labelled as words by the ASR. All the systems that utilise confidence-based alteration are henceforth labelled with *C*.

5.1.5 Training and Testing

Since the above methods generally attempt to improve F1 by improving precision, it was decided that the models with the best recall would be used as the underlying detector systems. From the results in Chapter 4, the model with the highest recall from each underlying architecture was the LSTM+CW and the FFN+CW. Each of the above four methods were applied to each of these models and the results examined. Multiple of the tested methods adjusted the posteriors produced by the detectors to zero. If there are mistaken detections centred around these areas, it is possible therefore that these methods may create extra peaks in the sequence, which would negatively impact both frame and event level metrics. As such, all the tested methods and architectures results are reported with and without the Hamming window convolution applied.

A k-fold approach was again used by employing the same folds as the previous chapter. The same hyper-parameter optimisation process was used, with each detector being optimised. The results are displayed in Table 5.1.

5.1.6 Results: Frame Level

Table 5.2 displays the frame level metrics by model and method. First examining AUC, a one-way ANOVA test found significant differences between detectors ($F(15, 272) = 25.70$, $p < 0.0001$). A post-hoc Tukey HSD test was used to identify which models differed. The baseline detector LSTM+CW+S was not significantly outperformed by any of the new methods on either architecture. For exact p-values, lower and upper confidence intervals see Table B.1. These results would suggest that none of the new methodologies have a positive impact.

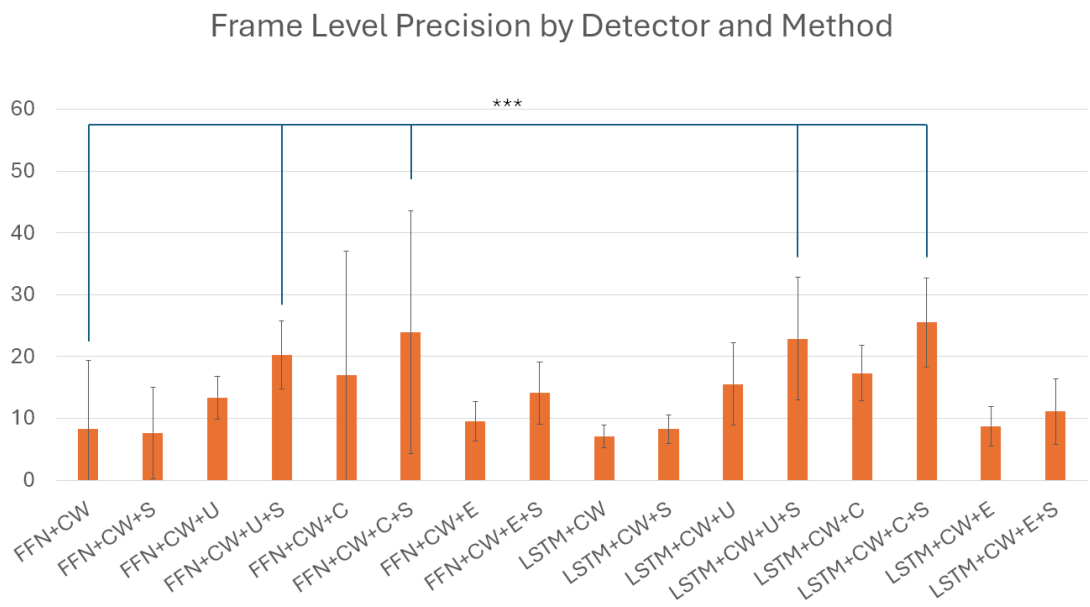

Figure 5.2: Frame Level Precision Performance by Detector and Method Using ASR Approaches. Significance Differences Shown in Relation to FFN+CW (***$p < 0.0005$)

Examining instead the effect that the tested methods had on the frame level precision of the detectors, a one-way ANOVA test showed significant differences between models ($F(15, 272) = 8.55$, $p < 0.0001$). A post-hoc Tukey test found that the baseline model FFN+CW was significantly outperformed by FFN+CW+U+S, FFN+CW+C+S, LSTM+CW+U+S and LSTM+CW+C+S detectors. Figure 5.2 displays the changes in precision and their significance. For exact p-values, lower and upper confidence intervals see Table B.2. LSTM+CW+C+S saw the highest frame level precision; it significantly outperformed FFN+CW+C+S. However, there was no significant difference between LSTM+CW+C+S and the two other detectors that outperformed the baseline, FFN+CW+U+S and LSTM+CW+U+S, suggesting equal improvements between them. The results indicate that both architectures benefit from undersampling and confidence-based alteration, but only after convolution with the Hamming window.

Turning now to the frame level recall results, a one-way ANOVA test once again found significant differences between detectors ($F(15, 272) = 15.70$, $p < 0.0001$). A post-hoc Tukey HSD test revealed multiple significant differences, which can be seen in Figures 5.3 and 5.4. For exact p-values, lower and upper confidence intervals see Table B.3. Due to the LSTM and FFN baseline architectures having significant performance differences between them, the results of the additional methods trialled here are given with respect to both baselines. FFN+CW+C+S had significantly worse recall than the FFN+CW baseline. While four of the new methods, i.e., FFN+CW+E, FFN+CW+E+S, LSTM+CW+E and LSTM+CW+E+S, and the two LSTM baseline detectors, all performed significantly better than the FFN+CW baseline. However, when comparing these four detectors with the LSTM+CW-based baseline, there were no significant differences. Furthermore, three of the new methods, i.e., FFN+CW+C+S, LSTM+CW+U and

LSTM+CW+U+S performed significantly worse than the LSTM+CW baseline. These results are promising since the methods tested were intended to increase precision without affecting recall. Unfortunately, although the negative impact on recall was localised to a few detectors, two of these detectors (FFN+CW+C+S and LSTM+CW+U+S), which showed significant decreases in recall, were also two of the detectors that saw significant increases in precision. This may result in any gains being negated by the losses.

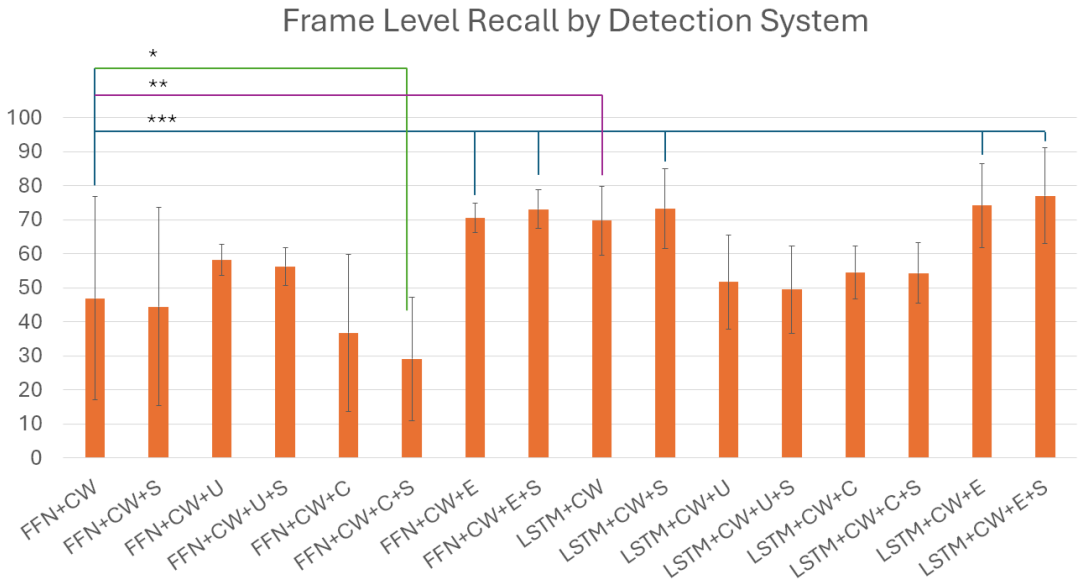

Figure 5.3: Frame Level Recall by Detector and Method Using ASR Approaches. Significance Differences Shown in Relation to FFN+CW (*$p < 0.05$, **$p < 0.005$, ***$p < 0.0005$)

Figure 5.4: Frame Level Recall by Detector and Method Using ASR Approaches. Significance Differences Shown in Relation to LSTM+CW+S (*$p < 0.05$, ***$p < 0.0005$)

Examining frame level F1, a one-way ANOVA test was run that found significant differences in F1 performance between detectors ($F(15, 272) = 21.93$, $p < 0.0001$). A post-hoc Tukey HSD test was used to identify differences between methods, the results of which (given in relation to the LSTM+CW+S baseline) are provided in Figure 5.5. For exact p-values, lower and upper confidence intervals see Table B.4. The best performing baseline (LSTM+CW+S) was significantly improved upon by six detectors FFN+CW+U+S, FFN+CW+E+S, LSTM+CW+U, LSTM+CW+U+S, LSTM+CW+C and LSTM+CW+C+S. Of these LSTM+CW+C+S achieved the highest overall score. The post-hoc Tukey test found that two of these six were significantly worse than this (FFN+CW+E+S and LSTM+CW+U). The remaining three saw no significant difference (FFN+CW+U+S, LSTM+CW+U+S and LSTM+CW+C) with exact p-values, lower and upper confidence intervals in relation to LSTM+CW+C+S given in Table B.5. This leaves a total of four top performing detectors that significantly outperformed the best baseline detector. Taken together, the main driver for three of these improved detectors appears to be the significant improvements in precision (FFN+CW+U+S, LSTM+CW+U+S, LSTM+CW+C+S). The final significantly better detector, LSTM+CW+C, did not show significant improvements in precision over the best performing baseline. However, it did have significant precision improvements over the LSTM architecture baselines, which enabled it to significantly outperform that baseline in terms of F1.

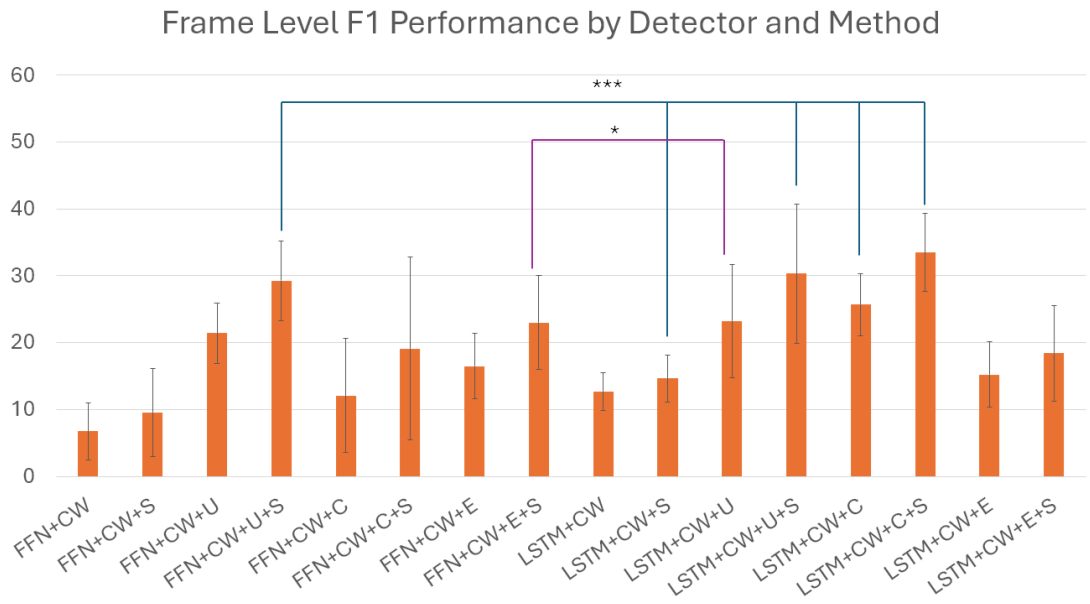

Figure 5.5: Frame Level F1 Performance on the SMC by Detector and Method Using ASR Approaches. Significant Differences Shown in Relation to LSTM+CW+S ($*p < 0.05$, $***p < 0.0005$)

In the case of the FFN architecture, undersampling is the most effective method. With confidence-based alteration and undersampling effectively performing when coupled with the LSTM architecture. The FFN-based architecture shows an increase in recall when using un-

Table 5.2: Frame Level Performance of Each Detection Method and Architecture for the SMC Using ASR Approaches. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. S: smoothing. C: confidence-based alteration. E: feature vector extension. U: undersampling. Bold highlights the best performing detector for each underlying architecture

Model	Precision	Recall	F1	AUC
FFN+CW	8.25 ± 11.10	46.99 ± 29.86	6.75 ± 4.27	58.21 ± 14.02
FFN+CW+S	7.63 ± 7.45	44.52 ± 29.20	9.54 ± 6.55	61.59 ± 20.16
FFN+CW+U	13.32 ± 3.44	58.22 ± 4.58	21.42 ± 4.50	78.21 ± 1.44
FFN+CW+U+S	**20.28 ± 5.53**	**56.36 ± 5.55**	**29.24 ± 5.97**	**83.87 ± 1.58**
FFN+CW+C	16.97 ± 20.01	36.77 ± 23.17	12.09 ± 8.55	70.38 ± 8.65
FFN+CW+C+S	23.92 ± 19.65	29.19 ± 18.09	19.13 ± 13.65	75.27 ± 12.25
FFN+CW+E	9.50 ± 3.24	70.54 ± 4.25	16.48 ± 4.90	83.70 ± 0.99
FFN+CW+E+S	14.07 ± 5.07	73.13 ± 5.64	23.02 ± 6.97	89.26 ± 1.06
LSTM+CW	7.09 ± 1.81	69.77 ± 10.13	12.73 ± 2.83	79.96 ± 1.78
LSTM+CW+S	8.25 ± 2.30	73.30 ± 11.72	14.62 ± 3.50	84.16 ± 1.70
LSTM+CW+U	15.51 ± 6.66	51.80 ± 13.81	23.20 ± 8.44	78.66 ± 1.75
LSTM+CW+U+S	22.90 ± 9.91	49.57 ± 12.81	30.34 ± 10.41	83.54 ± 2.09
LSTM+CW+C	17.31 ± 4.45	54.57 ± 7.71	25.67 ± 4.68	82.06 ± 1.61
LSTM+CW+C+S	**25.53 ± 7.19**	**54.40 ± 8.77**	**33.47 ± 5.85**	**86.43 ± 1.97**
LSTM+CW+E	8.72 ± 3.17	74.19 ± 12.36	15.24 ± 4.84	83.20 ± 3.99
LSTM+CW+E+S	11.10 ± 5.34	77.09 ± 14.16	18.42 ± 7.14	87.44 ± 2.92

dersampling. This is despite some laughter frames predictions being set to zero since the ASR misclassified them as speech. The results suggest that the effect of better balanced classes outweighs these mistakes.

With respect to the LSTM-based architecture. A decrease in recall in the undersampling methods was expected. However, this drop was compensated for by the intended increase in precision. This suggests that some of the methods employed were effective in reducing the number of false positives produced by the detection methods, achieving the stated goal at a frame level. These methods led to significant improvements in terms of F1. However, despite these improvements, the overall performance of all the detectors remains below 40%.

5.1.7 Results: Event Level

Table 5.3 displays the event level performance by model and method applied. Once again examining the effect on precision, a one-way ANOVA test was carried out that revealed significant differences between detectors ($F(15, 272) = 11.23$, $p < 0.0001$). Figure 5.6 displays the results of a post-hoc Tukey test, in relation to the LSTM+CW+S baseline. For exact p-values, lower and upper confidence intervals see Table B.6. It found that three detectors (FFN+CW+U+S, LSTM+CW+U+S and LSTM+CW+C+S) showed significantly better precision. The detector with the highest overall score, LSTM+CW+C+S, showed no significant difference from the other two best performing detectors (FFN+CW+U+S: $p = 0.86$, 95% C.I. = [-5.76, 20.28],

Table 5.3: Event Level Performance of Each Detection Method and Architecture for the SMC Using ASR Approaches. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. S: smoothing. C: confidence-based alteration. E: feature vector extension. U: undersampling. Bold highlights the best performing detector for each underlying architecture

Model	Precision	Recall	F1
FFN+CW	8.59 ± 9.36	70.02 ± 29.89	10.06 ± 4.60
FFN+CW+S	13.00 ± 15.31	39.71 ± 24.09	12.52 ± 9.15
FFN+CW+U	13.93 ± 4.25	88.89 ± 3.08	23.84 ± 6.21
FFN+CW+U+S	**31.14 ± 7.58**	**67.49 ± 4.12**	**42.12 ± 7.47**
FFN+CW+C	18.53 ± 20.17	69.61 ± 28.37	17.20 ± 11.09
FFN+CW+C+S	23.62 ± 25.04	38.80 ± 24.80	22.00 ± 18.09
FFN+CW+E	12.11 ± 3.84	91.92 ± 2.87	21.17 ± 5.92
FFN+CW+E+S	24.23 ± 7.95	75.72 ± 4.45	35.91 ± 9.05
LSTM+CW	10.23 ± 2.39	86.94 ± 7.70	18.18 ± 3.76
LSTM+CW+S	16.41 ± 3.95	70.62 ± 9.73	26.30 ± 5.14
LSTM+CW+U	23.42 ± 10.28	75.31 ± 7.71	34.25 ± 11.58
LSTM+CW+U+S	34.54 ± 14.60	60.79 ± 8.58	41.59 ± 11.63
LSTM+CW+C	27.26 ± 6.19	78.69 ± 7.48	39.90 ± 6.51
LSTM+CW+C+S	**38.40 ± 8.41**	**60.25 ± 7.96**	**45.89 ± 6.00**
LSTM+CW+E	13.81 ± 5.65	86.74 ± 5.57	23.32 ± 8.28
LSTM+CW+E+S	19.02 ± 6.99	75.13 ± 7.61	29.50 ± 8.82

LSTM+CW+U+S: $p = 1.00$, 95% C.I. = [-16.88, 9.16]). These results are, in general, in agreement with the frame level results. This is because the same detectors, at both levels, achieved better results with the exception of FFN+CW+C+S, which failed to achieve a significant difference at an event level.

Examining the recall scores, a one-way ANOVA test found significant differences ($F(15, 272) = 20.23$, $p < 0.0001$). The results of a post-hoc Tukey test, shown in relation to the LSTM+CW baseline, are displayed in Figure 5.7. For exact p-values, lower and upper confidence intervals see Table B.7. No detector achieved a significant increase in recall with respect to the best performing baseline LSTM+CW . Five of the new methods (FFN+CW+U+S, FFN+CW+C, FFN+CW+C+S, LSTM+CW+U+S and LSTM+CW+C+S) performed significantly worse in recall compared with this baseline. Unfortunately, all three of the detectors that showed improvements for precision are in this group, suggesting that, at an event level, the methods may be too detrimental to recall for them to be useful.

Examining event level F1, a one-way ANOVA test once again found significant differences between detectors ($F(15, 272) = 27.27$, $p < 0.0001$). Figure 5.8 displays the results of a post-hoc Tukey HSD test in relation to the LSTM+CW+S baseline. For exact p-values, lower and upper confidence intervals see Table B.8. It found that four detectors (FFN+CW+U+S, LSTM+CW+U+S, LSTM+CW+C and LSTM+CW+C+S) significantly out-performed the baseline. This suggests that, despite the negative impact on recall, the improvements in precision

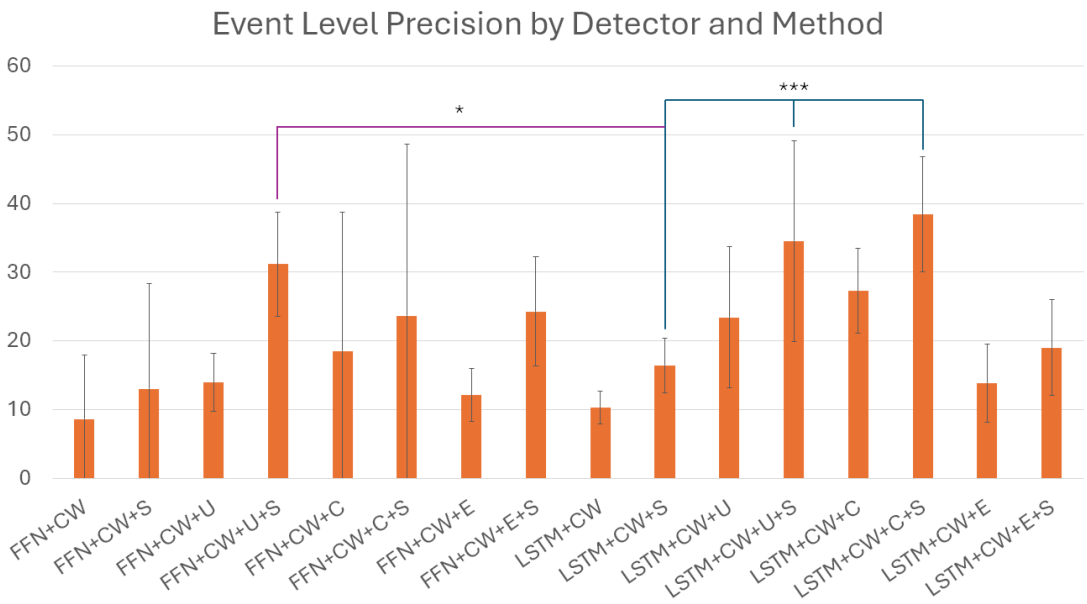

Figure 5.6: Event Level Precision on SMC by Detector and Method Using ASR Approaches. Significant Differences Shown in Relation to LSTM+CW+S Baseline Detector (*$p < 0.05$, ***$p < 0.0005$)

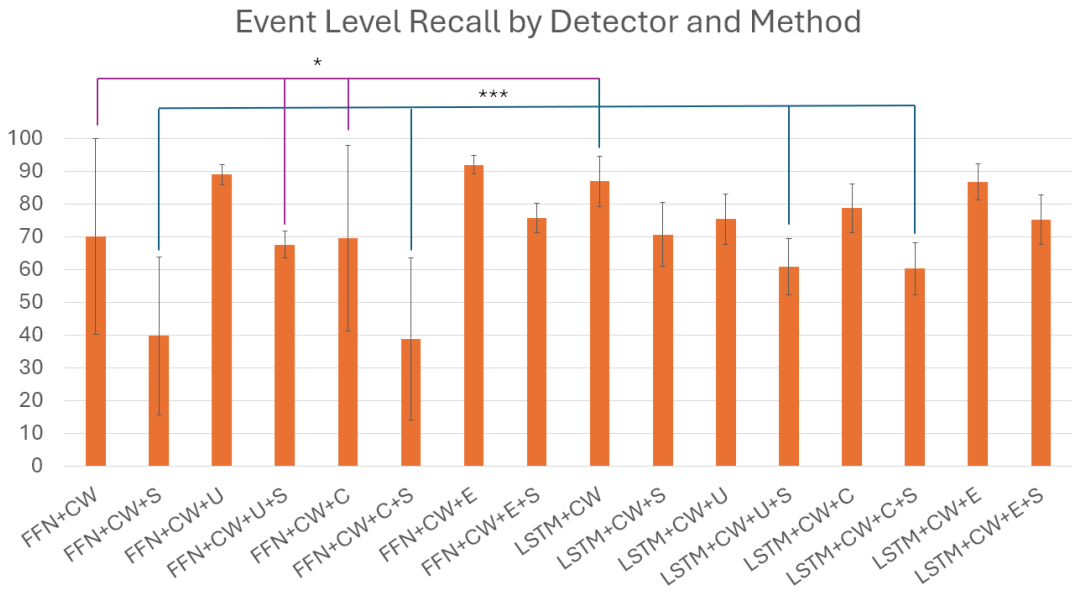

Figure 5.7: Event Level Recall by Detector and Method on SMC Using ASR Approaches. Significant Differences Shown in Relation to LSTM+CW Baseline Detector (*$p < 0.05$, ***$p < 0.0005$)

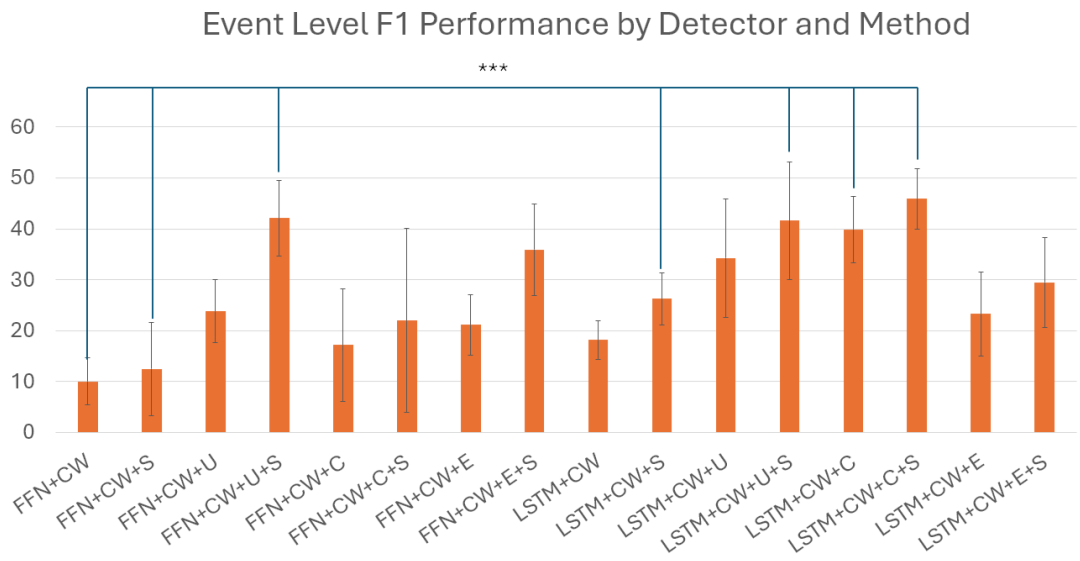

Figure 5.8: Event Level F1 Performance on the SMC by Detector and Method Using ASR Approaches. Significant Differences Shown in Relation to LSTM+CW+S Baseline Detector ($***p < 0.0005$)

are enough to still create significant improvements overall. Taken together, the frame and event level results agree that the confidence-based alteration and undersampling method are the most effective, with the underlying neural network architecture mediating each method's effectiveness.

5.1.8 Performance Analysis

The following section describes a performance analysis carried out on the best performing detector from this chapter: LSTM+CW+C+S. The latter is henceforth termed the confidence-based alteration system (CBA). This detector saw a significant increase in precision compared with the baseline, but this was coupled with a significant decrease in recall. However, overall, the increase in precision outweighed the recall issue, resulting in a significant improvement in F1 performance at both a frame and an event level.

The first step in this performance analysis is to examine between-group differences that may be affecting detector performance. Descriptive statistics for the CBA's system performance by gender, role and gender pairing are displayed in Table 5.4 for frame level metrics and Table 5.5 for event level metrics.

In terms of role, independent t-tests were used to compare the performance of the CBA system for each metric. Much like in the previous chapter, there were significant differences. However, where those differences were found, the direction of them had changed. These differences, for both the CBA system and the baseline, are shown in Figure 5.9 for frame level results and Figure 5.10 for event level outcomes.

Table 5.4: Frame Level Performance of LSTM+CW+C+S for the SMC by Group Split

	Precision	Recall	F1	AUC
Caller	39.93 ± 23.89	57.97 ± 20.32	41.76 ± 18.79	88.48 ± 6.55
Receiver	41.71 ± 24.56	46.99 ± 20.28	38.53 ± 18.37	84.83 ± 9.59
Male	35.48 ± 24.87	48.37 ± 21.97	35.45 ± 19.49	84.79 ± 9.28
Female	45.65 ± 22.61	56.20 ± 19.39	44.39 ± 16.76	88.22 ± 7.24
MM	25.04 ± 21.02	47.38 ± 21.34	26.76 ± 18.33	86.74 ± 6.92
FF	34.21 ± 19.64	60.31 ± 18.14	37.00 ± 15.01	89.85 ± 5.86
MF	27.41 ± 15.02	54.56 ± 17.18	33.54 ± 13.61	86.50 ± 6.94

Table 5.5: Event Level Performance of LSTM+CW+C+S for the SMC by Group Split

	Precision	Recall	F1
Caller	24.70 ± 16.05	72.97 ± 15.55	33.84 ± 16.98
Receiver	22.06 ± 13.66	45.47 ± 18.07	26.88 ± 13.48
Male	20.43 ± 15.16	53.27 ± 21.52	26.41 ± 15.06
Female	25.87 ± 14.39	64.77 ± 20.48	33.78 ± 15.55
MM	31.24 ± 26.30	51.59 ± 24.38	33.24 ± 22.76
FF	52.77 ± 26.90	69.09 ± 20.54	52.14 ± 19.41
MF	43.61 ± 22.80	62.71 ± 17.75	48.39 ± 19.22

CBA frame level precision has no significant performance differences in relation to role (t(358) = 1.16, $p = 0.69$), suggesting that the CBA system negated the issues of the baseline system. This change also impacted the frame level F1, which is also no longer significantly different (t(358) = 1.65, $p = 0.10$). However, in both the CBA and baseline systems, there were significant differences in frame level recall with callers, which sees significantly better performance than receivers (t(358) = 5.12, $p < 0.0001$). In AUC, the CBA system was significantly better with callers than with receivers (t(358) = 4.13, $p < 0.0001$). However, in contrast to this, the baseline system saw receivers with significantly better performance.

At an event level, the CBA saw the introduction of a significant difference in performance in both recall (t(358) = 7.55, $p < 0.001$) and F1 (t(358) = 2.67, $p = 0.0086$). Recall was significantly better for callers compared with receivers, mirroring the frame level results, and shows the same trend from baseline to CBA. However, the CBA system also saw the introduction of a significant difference in the event level F1 metric, with callers seeing significantly better performance. This differs from the baseline system, in which no significant differences between roles were found in event level F1. Since the CBA system did not attempt to address the differences found between callers and receivers, as in the previous chapter's performance analysis, it is unsurprising that significant differences in performance remain. The differences in the trends between the baseline and CBA systems is probably due to the low caller precision seen in the baseline detector. As the CBA system specifically addressed precision, it removed the differences in roles. Further support for this can be seen in the recall results, which have the same size and direction of the difference as seen in the baseline.

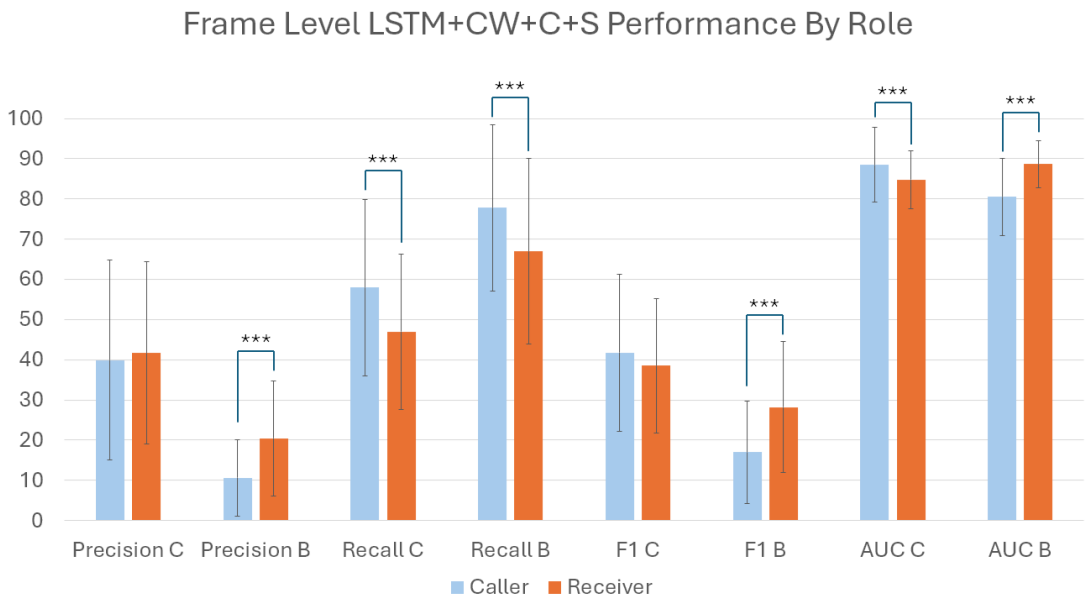

Figure 5.9: Frame Level LSTM+CW+C+S Performance by Role for the SMC. B: baseline LSTM+CW+S. C: confidence-based alteration LSTM+CW+C+S (***$p < 0.0005$)

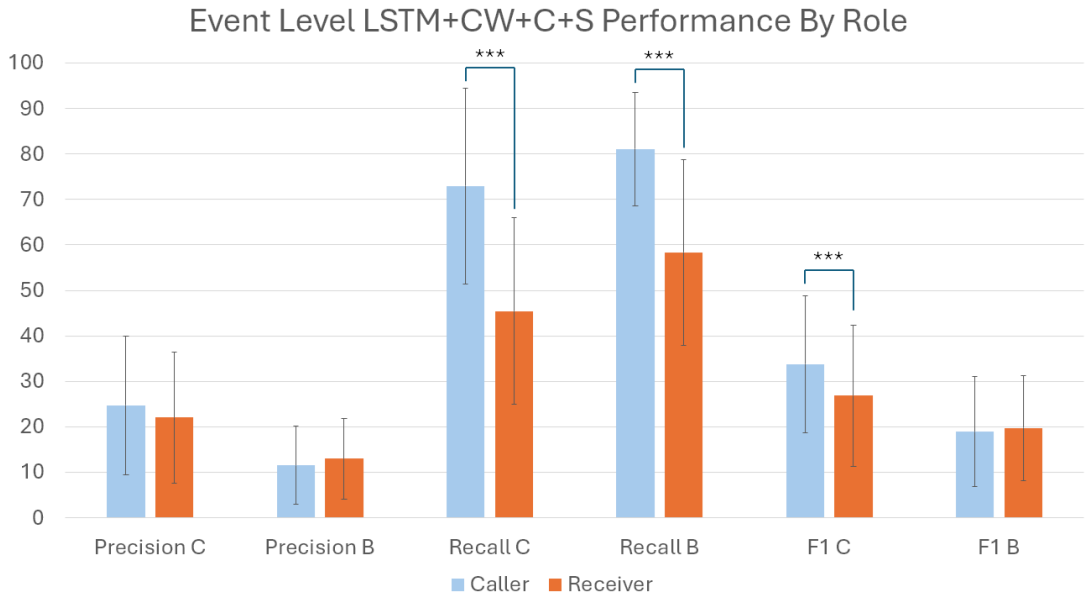

Figure 5.10: Event Level LSTM+CW+C+S Performance by Role for the SMC. B: baseline LSTM+CW+S. C: confidence-based alteration LSTM+CW+C+S (***$p < 0.0005$)

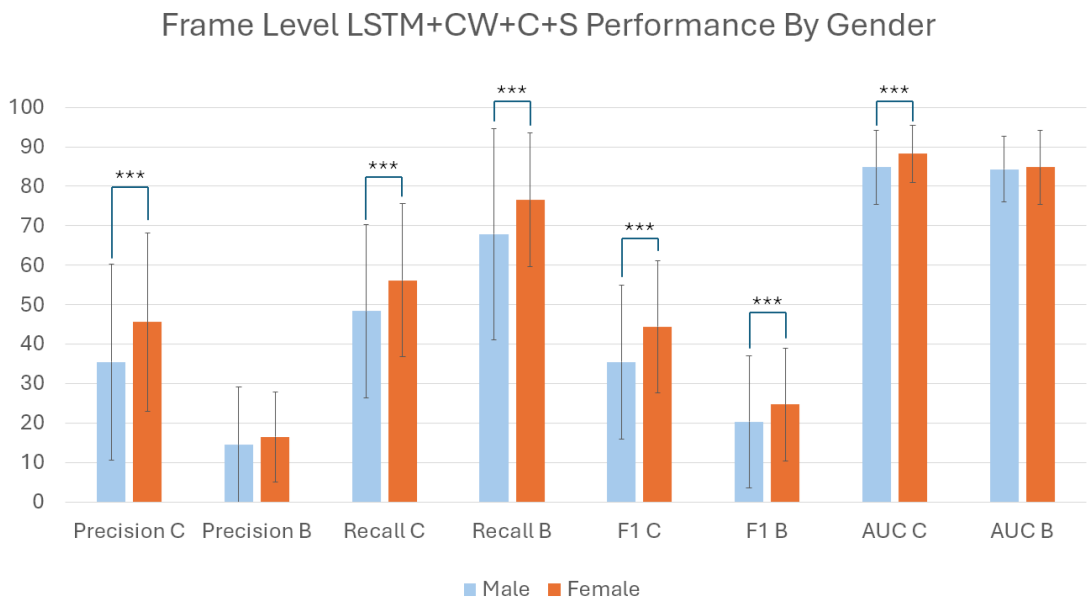

Figure 5.11: Frame Level Performance by Gender for the SMC. B: baseline LSTM+CW+S. C: confidence-based alteration LSTM+CW+C+S (***$p < 0.0005$)

Now examining the differences in performance by gender, the t-tests were once again used to test for significant differences in performance by metric. Figure 5.11 displays frame level and Figure 5.12 shows event level differences for both the CBA and baseline systems. Significant differences in CBA performance by gender were found in all metrics (exact significance values given in Table B.9) In all cases, female performance was significantly better than male performance. This suggests a worsening of the performance differences found in the baseline system. This is because there was no significant difference by gender in the baseline system in frame level precision or AUC. The underlying distribution of laughter, with female speakers tending to laugh more than males, was suggested as an explanation in the previous chapter for the gender differences seen. However, this interpretation cannot explain why the CBA system would introduce more/worse differences. It is possible that, when the CBA system reduces false positives, the higher number of possible true positive events for females leads to unequal increases in precision. The equation for calculating precision is given in Section 2.5. If there are an equal proportion of male and female laughs spotted, the value of female true positives would be greater than male false positives. With an equal reduction of false positives for both genders, the resulting precision statistic would, therefore, rise further for female speakers. As explained in the performance analysis at the end of the previous chapter, similar data distribution effects have been solved in other fields through the collection of more data representing the under-represented class.

In relation to performance by the gender pairing of a conversation, Figures 5.13 (frame level) and 5.14 (event level) show the significant differences found using a one-way ANOVA test for each metric. Exact values for each ANOVA and associated post-hoc Tukey tests can be found

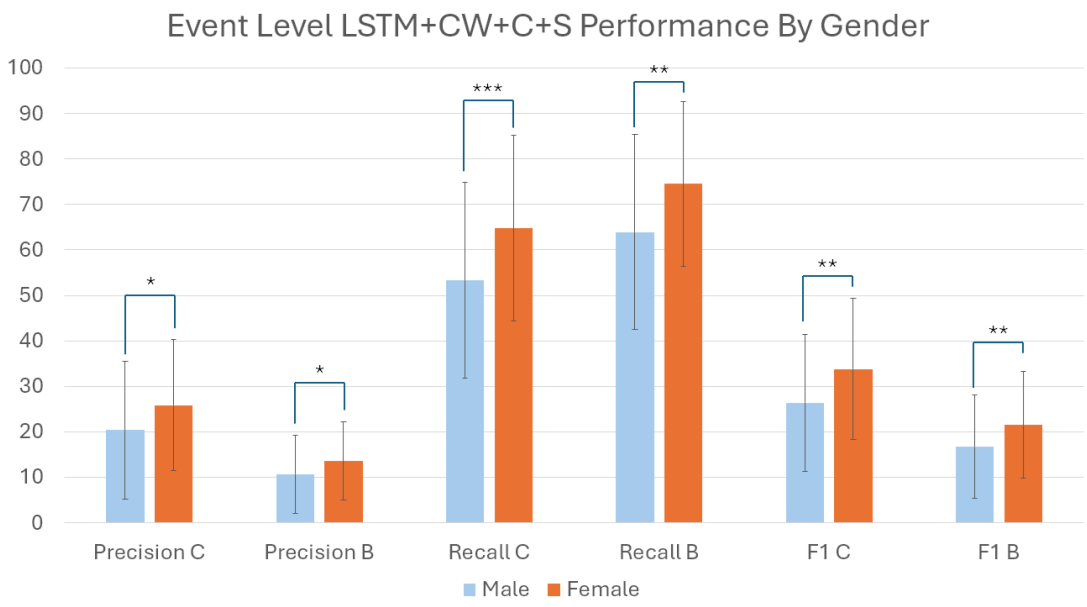

Figure 5.12: Event Level Performance by Gender for the SMC. B: baseline LSTM+CW+S. C: confidence-based alteration LSTM+CW+C+S (*$p < 0.05$, **$p < 0.005$, ***$p < 0.0005$)

in Table B.10. As seen with the gender effects above, the increase in precision at both a frame and event level has not been distributed equally and, thus, significant differences in pairing performance are seen in the CBA system.

In frame level precision, recall, and F1 MM pairings had significantly worse performance than FF pairings. However, in all three metrics there was no significant difference between FF pairings and MF pairings, nor between MF and MM pairings. In AUC MF pairings performed significantly worse than FF pairings. There were no significant differences found between MF and MM pairings, nor between MM and FF. Similarly in event level precision, recall, and F1 MM pairings performed significantly worse than FF pairings. In addition, MM pairings performed significantly worse than MF pairings in all three event level metrics. For FF and MF pairings there was no significant difference in any event level metric. The differences in pairing performance are probably due to the gender differences seen above, especially given the increase in the differences seen in gender introduced by the CBA system.

The CBA system saw a significant decrease in recall compared with the baseline detectors. To investigate the cause of this fact, a comparison between the LSTM+CW+C+S and the baseline was carried out. This investigated which events were spotted by the baseline but missed by the CBA system. It was found, on average, that the confidence-based alteration detector failed to detect 36.40 ± 16.08 laughter events per fold that the baseline detector did detect. The average length of these events was 0.77 ± 0.08 s, which is above the average for laughter events in the corpus (see Chapter 3, Table 3.1, for full details), suggesting that the issue is not the length of the laughter events.

Of the missed events, $91.38 \pm 12.29\%$ of them had speech detected by the ASR system.

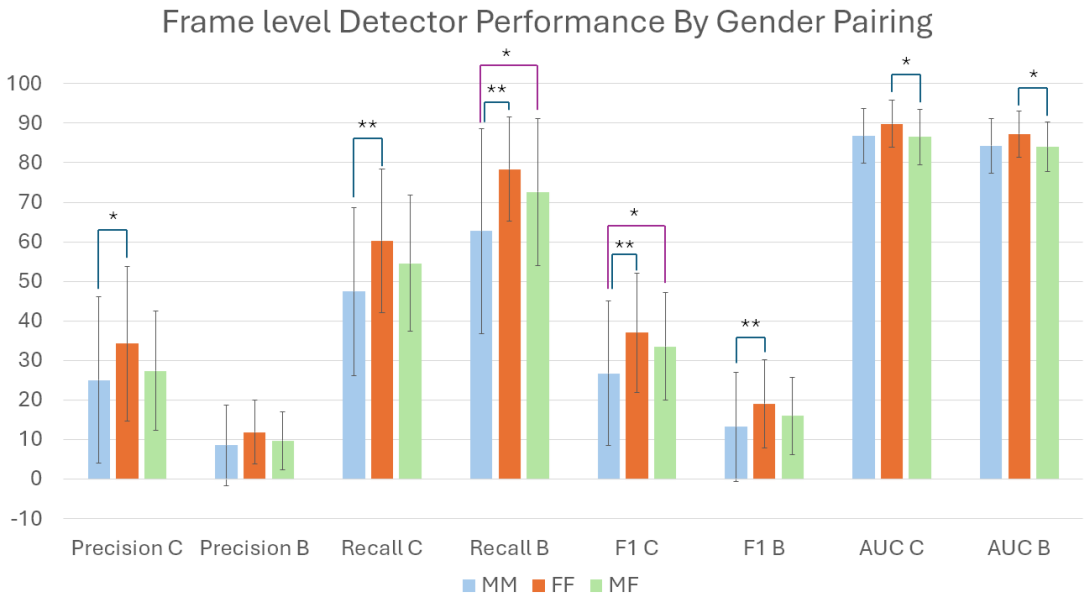

Figure 5.13: Frame Level Performance by Gender Pairing for the SMC. B: baseline LSTM+CW+S. C: confidence-based alteration LSTM+CW+C+S (*$p < 0.05$, **$p < 0.005$)

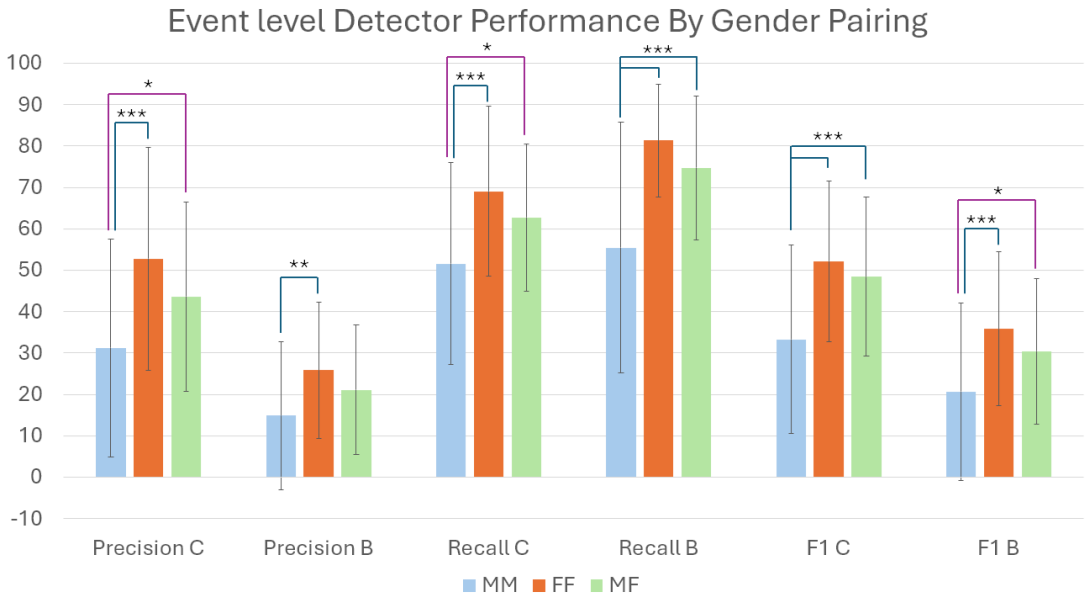

Figure 5.14: Event Level Performance by Gender Pairing for the SMC. B: baseline LSTM+CW+S. C: confidence-based alteration LSTM+CW+C+S (*$p < 0.05$, ***$p < 0.0005$)

Table 5.6: Average Percentage of Events That Overlapped with Laughter Events Missed by the LSTM+CW+C+S Detector but Detected by the LSTM+CW+S Baseline

Event Type	Missed Events
Speech	76.57 ± 12.03
Filler	9.96 ± 6.28
Back-Channel	1.70 ± 3.33
Laughter	11.77 ± 8.01

The average coverage of events by the ASR detections was $49.03 \pm 10.56\%$, meaning that, for $\sim$90% of the events, half of that event's frames were classed as having ASR. Given that $\sim$10% of missed events had no ASR detection and of the 90% of events that did have ASR detections around half of their frames were not tagged as ASR, these events could still be detected by the CBA system. However, it is likely that the confidence-based alteration manages to suppress any peaks in the posterior sequence, which could give rise to an event detection in these cases. This suppression is done by creating zero or near zero values in the posterior sequence that, when coupled with the smoothing step, suppress the amplitude of any peaks below the cut-off percent threshold that would have counted them as event detections. For the 10% of events that have no ASR detections, this suppression is probably caused by zeroing or near zeroing of the frames either side of an event, having the similar effect of lowering the maximum amplitude after the smoothing step.

Despite the majority of events having ASR detections, only 62.19 ± 9.31 of these events actually had overlapping events according to the ground truth labels; the distribution of what type of event was overlapping with the missed laughter event is displayed in Table 5.6. This clearly shows that the majority of the overlaps were caused by speech, so ASR detections are expected. However, as only $\sim$60% of events actually had overlapping events and only $\sim$75% of these overlaps were caused by speech, this leaves around 50% of the mistaken ASR detections unexplained. This suggests that, in around half the cases of these missed laughter events, the ASR system made a mistake. Support for this claim can be found by examining the MBR confidence. The MBR confidence of the ASR system detections of these missed events was on average $51.05 \pm 7.50\%$ (min = 11.04 ± 5.76, max = 85.65 ± 3.69), whereas the average MBR confidence across each fold was $83.27 \pm 0.62\%$. An independent T-test found these averages to be significantly different ($t(1002) = 8893.51$, $p < 0.0001$), showing that the ASR was indeed less confident at detecting overlapping with laughter. However, the average confidence remains over 50%, which means that the equation used to implement the adjustment (see Section 5.1) would halve the posterior probability of the laughter frames - leaving them below the frame level classification for laughter and suppressing the smoothed posteriors. This leads to the lowering of any peaks that may have occurred during the event and, thus, the event not being detected even in the case of the ASR system being uncertain.

It may be possible that a more conservative incorporation of the ASR MBR confidence may

Table 5.7: Event Level Precision, Recall and F1 for LSTM+CW+C+S Using a Threshold-Based Alteration System

Confidence Level	Precision	Recall	F1
Original System	38.40 ± 8.41	60.25 ± 7.96	45.89 ± 6.00
0	38.99 ± 8.76	59.92 ± 7.90	46.16 ± 6.11
0.1	38.99 ± 8.76	59.92 ± 7.90	46.16 ± 6.11
0.2	38.96 ± 8.78	59.96 ± 7.94	46.15 ± 6.13
0.3	38.27 ± 8.45	60.28 ± 7.85	45.80 ± 6.03
0.4	36.50 ± 7.73	61.31 ± 7.95	44.84 ± 5.86
0.5	33.67 ± 7.17	62.62 ± 7.82	42.98 ± 5.82
0.6	30.85 ± 6.87	64.29 ± 8.06	40.92 ± 5.89
0.7	28.79 ± 6.38	65.62 ± 7.99	39.35 ± 5.96
0.8	26.80 ± 6.15	66.47 ± 8.27	37.55 ± 6.04
0.9	25.06 ± 6.00	67.67 ± 8.64	35.98 ± 6.30
1	23.67 ± 5.60	68.28 ± 8.40	34.64 ± 6.14

result in better recall. To test this possibility, a new system was tested where if the MBR for a frame was above a threshold value that frame's posterior would be set to zero. This means that the effect of high confidence ASR detections would remain the same as the original system, with the posteriors being zeroed. However, for lower confidence ASR detections, the posteriors would be preserved. Table 5.7 shows the precision, recall and F1 for difference threshold levels. At a threshold level of zero, there is no significant difference between the threshold approach and the original approach. This offers further evidence that, in the event when the ASR MBR confidence is low, it is still high enough to practically zero the posteriors. Moving from a threshold value of 0.3 to 0.4, there is some recovery of recall; however, this is offset by a lowering of precision. This drop in precision is greater than the increase in recall, meaning that the threshold system has no advantage over the original one. As was the case in Chapter 4, there is a ceiling on how effective methods can be and, in the case of confidence-based alterations, that ceiling is an F1 of $\sim$45%.

Finally, the issue of near misses was investigated. It was shown in the previous chapter that 25% of the false positives generated by the best performing detector were near-misses. This was 20% above what would be expected if false positives were randomly distributed. The same analysis was carried out for the CBA system, with results shown in Table 5.8. The results show that 29.49 ± 8.95% of the false positives were near misses. This is around 5% higher than the baseline model. This increase suggests that the CBA system removed false positives that existed further from laughter events, meaning the system was effective in removing the type of false positive that it was intended to address. Furthermore, when reclassifying the near misses as true positives, the performance of the system in terms of event level F1 increases to 50.61 ± 5.70. This is the first time that any system has had an average performance above 50%.

Table 5.8: Percentage of False Positives Within a Given Time of Laughter Events and the Associated Precision, Recall and F1 if They Were Reclassified as True Positives for the CBA System

Time (s)	Percentage of False Positives	Precision	Recall	F1
Original	-	38.40 ± 8.41	60.25 ± 7.96	45.89 ± 6.00
0.1	14.59 ± 5.16	39.45 ± 8.55	61.79 ± 8.05	47.11 ± 5.90
0.46	29.49 ± 8.95	42.73 ± 8.66	65.35 ± 8.99	50.61 ± 5.70
0.5	30.26 ± 9.20	43.11 ± 8.73	65.74 ± 9.01	50.99 ± 5.73
1	40.92 ± 10.17	46.16 ± 8.99	69.69 ± 9.77	54.46 ± 6.03
2	57.72 ± 11.65	49.68 ± 9.27	74.26 ± 9.65	58.51 ± 6.24

5.2 Multi-Cue Detection

The previous section explored the feasibility and effectiveness of the inclusion of ASR data through various new pre/post-processing methods. It showed that these methods all led to significant improvements over the state-of-the-art procedures displayed in Chapter 4, with the performance of the best detector achieving an event level F1 of $\sim$50%. This section explores another method for improving laughter detection by addressing the class imbalance issue.

The SMC contains annotations for two other paralanguage cues: filler and back-channel. Fillers comprise 4.51% of the SMC and back-channel events consist of 1.00%. Fillers, also called filled pauses, are short utterances (“Um", “ah", “er") made by a speaker [85]. They have been found to signal a speaker’s uncertainty [86] and to help manage the turn-taking and structure of a conversation [87, 88]. Back-channel are utterances that can consist of short words or sounds similar to fillers. Back-channel are used by a conversation participant who is not currently holding the floor of the conversation, to signal that they are still engaged in the conversation [89]. This means that they always overlap with another speaker’s speech. All three of these cues (i.e., laughter, filler and back-channel) can be considered forms of non-verbal communication (NVC). When considered as a single super class, they account for around 9% of the total data in the SMC. It is possible that by extending the detection systems by including these labels, the class imbalance issue may be lessened and the performance of the detectors improved. Further, this may aid in the reduction of the false positives cause by speech. By drawing a clear distinction between verbal and non-verbal communication it was theorised that possible causes of confusion during training would be lessened. These ideas led to the creation of RQ3: What is the effect of broadening the scope of laughter detectors to include multiple cues? As with the above inclusion of linguistic information there were multiple ways of extending the detectors to other cues. Three approaches were developed to test this research question:

- multi-label system,
- joint NVC with individual detectors,
- joint NVC with NVC distinguisher.

In the multi-label system the detectors were modified from binary classifiers to multi-class classifiers. Rather than defining the problem as a target class versus not, the problem was instead extended so that all the classes in the SMC (i.e., laughter, filler, back-channel, speech and pause) were assigned a label. The detectors were then trained to output five posteriors, one associated with each class. Multiple cues can occur at the same time since speakers can speak over each other. As such, during classification, a single frame could be classed as multiple classes.

In both forms of joint NVC fusion approaches, the detectors comprised of a two-stage classification system. The first stage was shared and involved grouping all three paralanguage classes (i.e., laugh, filler and back-channel) into one super-class of NVC. A first stage binary classification detector was then trained to distinguish between NVC and all the other classes. After jointly detecting the cues as a super-class, two systems were then developed that take as input the frames marked as NVC by the joint-cue detector. These systems then attempt to differentiate these cues and class them as either laughter, filler or back-channel. These approaches were differentiated based on the amount of the initial training data the secondary detection systems underwent and how the decisions of the NVC detectors and the secondary detectors were fused.

In the case of the joint NVC fusion with independent detectors, the second stage was developed as separate detectors. With an individual detector for laughter, filler and back-channel each being created, each detector was trained on the entirety of each training fold in the same manner as laughter detectors were in the previous chapter. This resulted in a unique detector for each of the target classes. Each frame identified as containing NVC by the first stage detector was then passed to the cue independent detectors.

Similarly to the multi-label system, each frame could again receive multiple classes. Here, if any of the individual detectors estimated a posterior above 0.5 and the NVC had detected paralanguage, the frame received this label. If none of the individual detectors had a posterior greater than 0.5, the frame was instead labelled as not NVC.

In the second approach, i.e., joint NVC fusion with a NVC distinguisher, a single secondary detector is developed that carried out cue classification. This detector has three possible outputs: laughter, filler and back-channel. It is trained on the laughter, filler and back-channel frames present in the training fold only. During the testing of all the frames that are detected by the multi-cue detection system, they are passed to this secondary differentiation system. The second-stage NVC distinguisher then produces a posterior for each class. Again, a frame can receive multiple classes if more than one of the class posteriors is above 0.5; it will instead be labelled as not NVC if none of the class posteriors are above 0.5.

For each of the above three approaches, it is possible to calculate precision, recall, F1 and AUC for each cue using a one-versus-rest process. Results are then given that align with each cue and then averaged across them. This prevents superior performance on a single cue from making a detection system appear better (or worse) than it is in reality. Furthermore, if a system is effective for one cue and none of the others, it could then be used as a detector for that cue

alone.

5.2.1 Multi-Label System

This approach mirrors that of all the previous detectors, save for the extension of the target class to include filler, back-channel, speech and pauses. As described in Section 4.2.5, the audio data is split into a sequence of frames, each representing a 20 ms window of audio through 16 features. Unlike in the laughter detection systems where each frame is classed as either laughter or not, in the multi-label system the frames are labelled using a vector of dimension $D = 5$. Each position in this vector is either a one or a zero, with the position representing the class. As multiple audio events can co-occur, this labelling vector will always have at least one positive position and may have multiple.

With the resulting labelled frames, the remainder of this approach remains the same. Both LSTM and FFN architectures were tested. The experiments were again performed according to the k-fold protocol used in previous sections and chapters. The results are calculated for each class before an average is taken, to give an overall score for each detector. This average performance is computed using a "macro-F1" approach, due to the existence of heavily under-represented classes [90, p. 260]. Macro-F1 is calculated by taking the unweighted F1 of each class and averaging them. Similarly, the macro-precision and recall are also calculated and displayed. No previous study has attempted a multi-label classification system over a type 3 corpus for paralanguage cue detection. Therefore, there is no baseline in the field to set against. As such, in this section, results are compared against the CBA detector from the previous section. This is done to provide context on how effective or not the detector is at detecting paralinguistic cues compared with the best laughter detector that is currently available. Hyper-parameter optimisation was carried out using macro-F1. The results from this optimisation for each of the tested models are displayed in Table 5.9.

Results

Tables 5.10 (frame level) and 5.11 (event level) show the performance by metric, architecture and cue by the multi-label system. Independent T-tests were used to compare the average performance by metric of the underlying architectures (for exact results see Table B.11. It was found that across all metrics the FFN architecture performed significantly better than the LSTM architecture. Given this, FFN was selected for further comparisons and analyses.

One-way ANOVA tests were then used to compare the performance across cues. Figure 5.15 shows the performance by metric and cue of the FFN multi-label system against the best performing laughter detector from earlier in this chapter, i.e., CBA, which acts as a baseline. For frame level AUC, a one-way ANOVA test found significant differences by cue ($F(6, 119) = 261.86$, $p < 0.0001$). A post-hoc Tukey HSD test determined that the CBA laughter perfor-

Table 5.9: Hyper-Parameters by Metric and Architecture of the Multi-Cue Detection Systems. All hidden layers had 100 nodes. No effect on frame level F1 was found, so it is omitted from the table. All systems used smoothing. FFN: feed forward neural network. LSTM: long short-term memory. All FFN detectors had two FFN hidden layers. All LSTM detectors had two FFN hidden layers and two LSTM hidden layers. All hidden layers had 100 nodes

System	Architecture	Window Size			Percent Cut-Off
		AUC	F1 (frame)	F1 (event)	
Multi-Label System	FFN	11	11	41	10
	LSTM	41	11	11	1
Joint NVC with Individual Detectors	FFN	11	21	21	10
	LSTM	31	71	61	10
Joint NVC with NVC Distinguisher	FFN	11	11	31	1
	LSTM	51	11	31	1

mance was significantly better than the multi-label system performed for every cue (for exact statistics see Table B.12).

One-way ANOVAs found that for frame level precision ($F(6, 119) = 187.07$, $p < 0.0001$), recall ($F(6, 119) = 381.47$, $p < 0.0001$), and F1 ($F(6, 119) = 432.17$, $p < 0.0001$) there were significant differences between the CBA baseline and the multi-label performance by cue. Post-hoc Tukey tests found that in all three metrics the CBA system was significantly better than the multi-label system for back-channel, laughter and filler. However, the multi-cue system was significantly better at pause and speech. There was no significant difference when comparing CBA and the average performance across cues of the multi-label system (for exact post-hoc Tukey values see Table B.13 for precision, Table B.14 for recall and Table B.15 for F1)

Similarly for the event level metrics of precision ($F(6, 119) = 236.63$, $p < 0.0001$), recall ($F(6, 119) = 106.36$, $p < 0.0001$), and F1 ($F(6, 119) = 368.98$, $p < 0.0001$) one-way ANOVAs found significant differences between the CBA and multi-label systems. In all three metrics post-hoc Tukey HSD tests determined that the CBA system was significantly better than the multi-label system at back-channel, laughter, filler, pause detection and on average. However, the multi-label system was significantly better at speech detection (for exact post-hoc Tukey values see Table B.16 for precision, Table B.17 for recall and Table B.18 for F1) .

A consistent relationship between cue and performance was found across frame level precision, recall and F1, with speech and pause seeing significantly better performance than that of the CBA baseline; while back-channel, laughter and fillers all saw significantly worse performance. The average performance across all the cues for the multi-label system was not significantly different compared with the CBA. The fact that both speech and pause saw significant performance increases was expected, given that both speech and pause classes comprise a larger proportion of the dataset relative to the other three cues. Their comparatively good performance is also

Table 5.10: Frame Level Performance by Metric, Cue and Architecture of the Multi-Label System. BC: back channel. FFN: feed forward neural network. LSTM: long short-term memory. CBA: confidence-based alteration system. Both architectures included smoothing

Cue	Architecture	Precision	Recall	F1	AUC
BC		3.16 ± 0.99	6.57 ± 4.64	3.67 ± 1.96	76.27 ± 2.48
Laugh		14.66 ± 4.04	35.40 ± 5.44	20.24 ± 3.91	78.85 ± 2.02
Filler	FFN	11.05 ± 3.20	28.23 ± 13.71	14.62 ± 3.19	72.12 ± 1.66
Pause		37.88 ± 4.95	90.51 ± 4.53	53.14 ± 4.45	91.24 ± 1.03
Speech		44.13 ± 5.15	100.00 ± 0.00	61.05 ± 5.08	56.14 ± 5.99
BC		1.70 ± 0.58	18.14 ± 15.70	2.51 ± 0.79	60.64 ± 5.27
Laugh		13.96 ± 14.58	3.76 ± 5.32	2.94 ± 2.51	55.82 ± 4.52
Filler	LSTM	5.83 ± 1.48	60.85 ± 27.28	9.96 ± 2.45	61.44 ± 4.62
Pause		25.67 ± 10.70	36.23 ± 31.48	26.07 ± 16.57	71.39 ± 15.66
Speech		44.13 ± 5.16	99.96 ± 0.12	61.05 ± 5.10	48.49 ± 1.85
CBA	LSTM	25.53 ± 7.19	54.40 ± 8.77	33.47 ± 5.85	86.43 ± 1.97

Table 5.11: Event Level Performance by Metric, Cue and Architecture of the Multi-Label System. BC: back channel. FFN: feed forward neural network. LSTM: long short-term memory. CBA: confidence-based alteration system. Both architectures included smoothing

Cue	Architecture	Precision	Recall	F1
BC		3.33 ± 1.27	27.21 ± 11.34	5.75 ± 2.09
Laugh		7.82 ± 2.45	66.12 ± 7.89	13.92 ± 4.06
Filler	FFN	6.29 ± 1.83	34.30 ± 10.24	10.54 ± 2.90
Pause		25.78 ± 4.42	68.77 ± 7.80	37.22 ± 4.69
Speech		44.95 ± 5.66	88.05 ± 4.74	59.39 ± 5.81
BC		1.27 ± 0.32	26.84 ± 6.62	2.42 ± 0.59
Laugh		3.58 ± 1.63	45.22 ± 13.09	6.51 ± 2.68
Filler	LSTM	4.54 ± 0.90	35.90 ± 7.28	8.02 ± 1.50
Pause		15.36 ± 4.61	52.10 ± 15.42	23.60 ± 6.96
Speech		46.36 ± 6.05	71.97 ± 15.33	55.37 ± 7.08
CBA	LSTM	38.40 ± 8.41	60.25 ± 7.96	45.89 ± 6.00

why there is no significant difference when comparing the average cue performance against the CBA, as they balance the comparatively poorer performance of the other cues. The performance in terms of laughter shows a significant decrease in performance, suggesting that adding more classes, and with it more explicit label information to the models, does not result in any gains in performance.

Also of interest is the event level metrics. Figure 5.16 displays the result of the post-hoc Tukey HSD tests used to compare performance on each metric across all cues. In terms of event level precision and F1, the multi-label system saw significantly worse performance in all the cases, except speech where the multi-label system was significantly better. The event level results reinforce the above conclusion that extending the number of labels/classes available to the system does not lead to better performance for individual cues. As before, the better performance

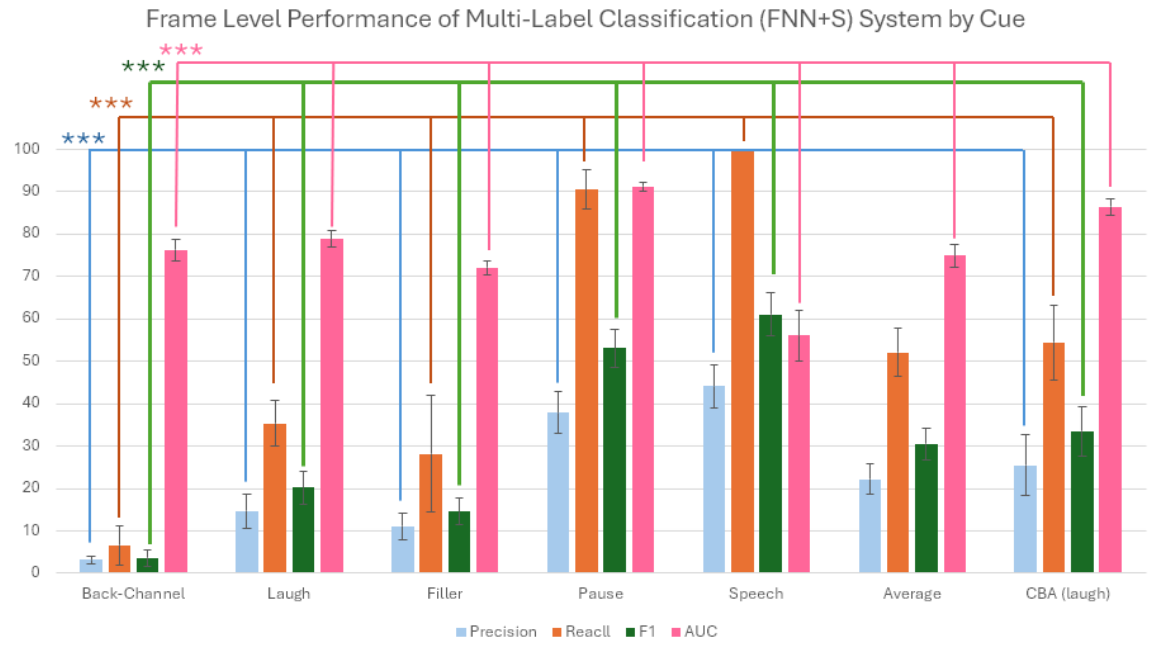

Figure 5.15: Frame Level Performance by Cue and Metric of Multi-Label Classification System. Significance Differences Shown in Relation to CBA Baseline Detector (***$p < 0.0005$)

in some metrics for pause and speech classes is probably due to the larger proportion of training data available for these classes.

5.2.2 Two-Stage Detection

Here, the inclusion of additional cues is incorporated through a two-stage detection process. In the first stage, all the non-verbal cues (i.e., laugh, filler and back-channel) are grouped into one super-class. This detector classes all the frames as either NVC or verbal communication (VC). The second stage then differentiates the NVC frames into individual classes. Two approaches were developed as second stage detectors: individual detectors and NVC distinguisher.

For the second stage consisting of individual detectors, three separate detectors were trained. Each detector followed the same approach and training procedure as the laughter detectors in Chapter 4, but each had a different target cue, i.e., either laughter, filler or back-channel. The effectiveness of each of these individual detectors without the first stage NVC detector is presented in this section. Then the impact of combining the NVC detector and the individual detectors is given. To combine the NVC detector and the individual cues, the following rule-based fusion was used. The NVC detector and each of the three individual detectors all produced classifications. Where the NVC detector classified frames as VC, the individual detector's classifications were zeroed. In the case where a frame was classed by the NVC system as containing NVC, the individual detector classifications were unchanged and used as classifications for each of the three paralanguage classes. The resulting filtered classifications are used to compute perfor-

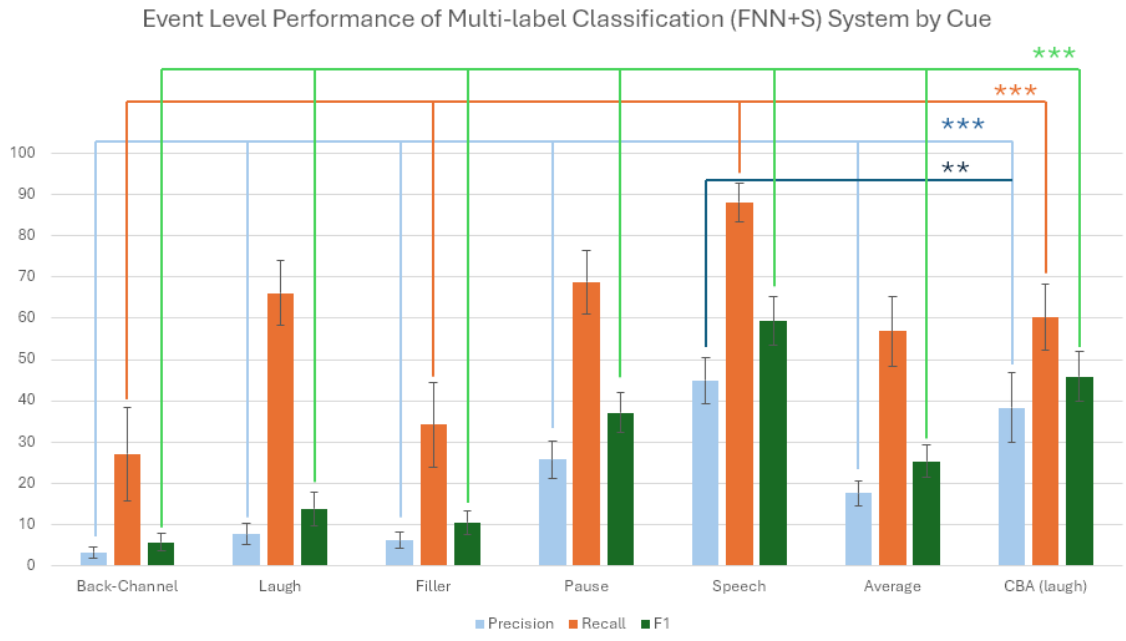

Figure 5.16: Event Level Performance by Cue and Metric of Multi-Label Classification System. Significant Differences Shown in Relation to CBA Baseline Detector (**$p < 0.005$, ***$p < 0.0005$)

mance metrics for each paralanguage class.

The other two-stage approach uses a NVC distinguisher. Here, the frames produced by the feature-extraction step are down-sampled to only include frames that are labelled as having laughter, filler or back-channel. This reduced dataset is then used to train a single detector, which outputs estimated posteriors for each of the three classes. This two-stage system mirrors the one given above but, rather than individual detectors producing a class probability for each frame, the paralanguage differentiator does these tasks in one place. Again, the NVC detector is used as a first stage and only frames classed as NVC are considered by the paralanguage differentiator.

Results: NVC super-class detector

The following three subsections address the two-stage detectors. In this first section, the ability to classifying NVC rather than VC is examined. Then, in the following two sections, the NVC detectors are coupled with the secondary stages to create individual cue detectors. Table 5.12 displays the performance of the NVC detector that treats laughter, filler and back-channel as one super-class. Since the intention is to use this as a first stage that filters out false positives, performance is given according to different threshold cut-offs for the frame level classification. Performance is given for frame level metrics because the detector is intended to be used as a frame level filter. Note that, for all the threshold levels, the AUC score is unaffected since it is calculated by varying the threshold between zero and one.

By first examining precision, a two-way ANOVA test was used to compare the effect of

Table 5.12: Performance by Metric and Architecture of the NVC Super-Class Detection System. FFN: feed forward neural network. LSTM: long short-term memory

Threshold	Architecture	Precision	Recall	F1	Percent of Data Filtered
0.5	FFN	17.55 ± 4.95	53.85 ± 9.69	25.52 ± 4.44	74.69 ± 7.90
0.4		12.48 ± 2.56	79.43 ± 6.82	21.38 ± 3.62	49.67 ± 8.64
0.3		10.07 ± 1.73	93.49 ± 3.05	18.12 ± 2.79	27.60 ± 6.51
0.2		8.85 ± 1.50	98.65 ± 0.81	16.21 ± 2.52	13.36 ± 4.37
0.1		8.02 ± 1.32	99.90 ± 0.13	14.82 ± 2.26	3.35 ± 2.37
0.5	LSTM	31.70 ± 33.11	1.69 ± 5.43	1.39 ± 3.11	92.92 ± 23.26
0.4		31.39 ± 30.20	4.08 ± 11.47	2.76 ± 4.58	96.43 ± 11.66
0.3		21.90 ± 19.07	8.23 ± 18.00	4.87 ± 5.76	92.52 ± 18.37
0.2		16.11 ± 14.24	16.73 ± 23.95	8.64 ± 5.82	84.61 ± 23.94
0.1		11.41 ± 8.45	39.28 ± 27.66	12.91 ± 3.90	63.44 ± 28.23

architectures and thresholds. It found that both architecture ($F(1, 9) = 19.55$, $p < 0.001$) and threshold values ($F(4, 9) = 4.97$, $p < 0.001$) had a significant effect on precision, but there was no significant interaction between them ($F(4, 9) = 1.15$, $p = 0.34$). Since the architecture had only two variable levels (i.e., FFN and LSTM), the two-way ANOVA test results directly show that the LSTM architecture performs significantly better than the FFN. Post-hoc Tukey HSD tests were used to examine the effect of the thresholds, with the results being shown in Figure 5.17 (for exact p-values and confidence values see Table B.19). It was found that threshold 0.1 performed significantly worse than thresholds 0.4 and 0.5. Furthermore, threshold 0.2 performed significantly worse than threshold 0.5. This suggests that, as expected, higher thresholds lead to better precision scores.

Regarding recall, a two-way ANOVA test once again found significant differences between architectures ($F(1, 9) = 1087.34$, $p < 0.001$) and thresholds ($F(4, 9) = 43.26$, $p < 0.001$), with a significant interaction between them ($F(4, 9) = 8.66$, $p < 0.001$). In terms of the architecture, the opposite effect compared with the above is determined, with FFN significantly outperforming LSTM architectures. Figure 5.18 shows the interaction between the threshold and architecture for recall. It shows that, as the threshold is lowered, the recall increases for both architectures. A post-hoc Tukey HSD test was employed to compare the threshold recall performance; the results are shown in Figure 5.18 (for exact p-values and confidence values see Table B.20). Threshold 0.5 is significantly worse than 0.4; threshold 0.4 is significantly worse than 0.3. However, threshold 0.3 is not significantly different from 0.2 but is significantly worse than 0.1. Threshold 0.2 is not significantly different from 0.1. These results suggest that there is no effect when lowering the threshold for filtrating below 0.2. Although a lower threshold leads to higher recall it also limits the effectiveness of the super-class filtration system with only $\sim$15% of the overall corpus data being removed when using the FFN architecture and a 0.2 threshold.

Performance in terms of F1 was compared for each threshold and architecture using a two-

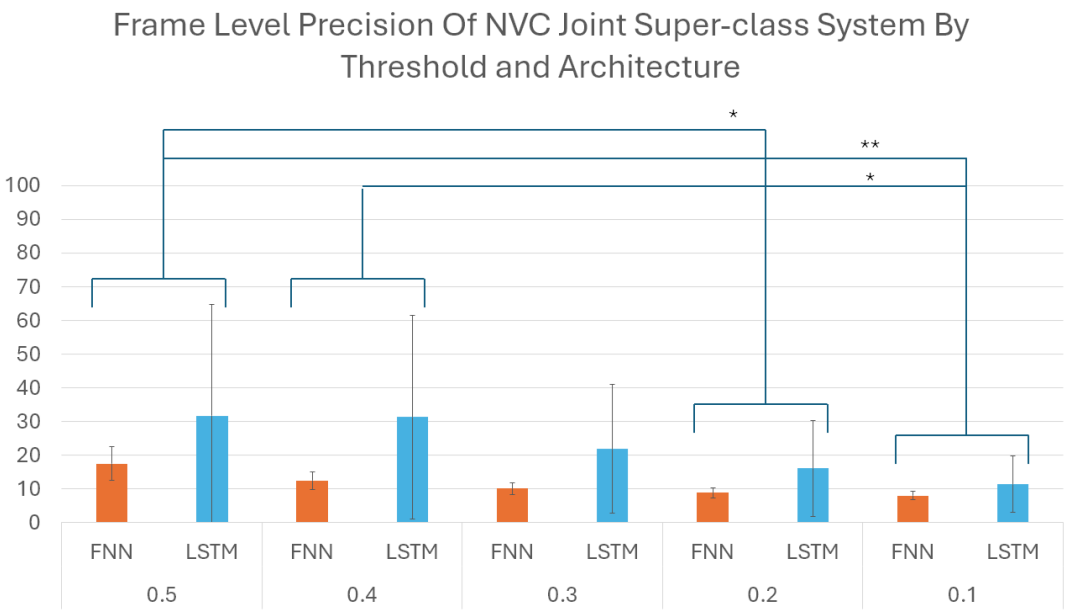

Figure 5.17: Precision of NVC Detection System by Threshold and Architecture ($*p < 0.05$, $**p < 0.005$)

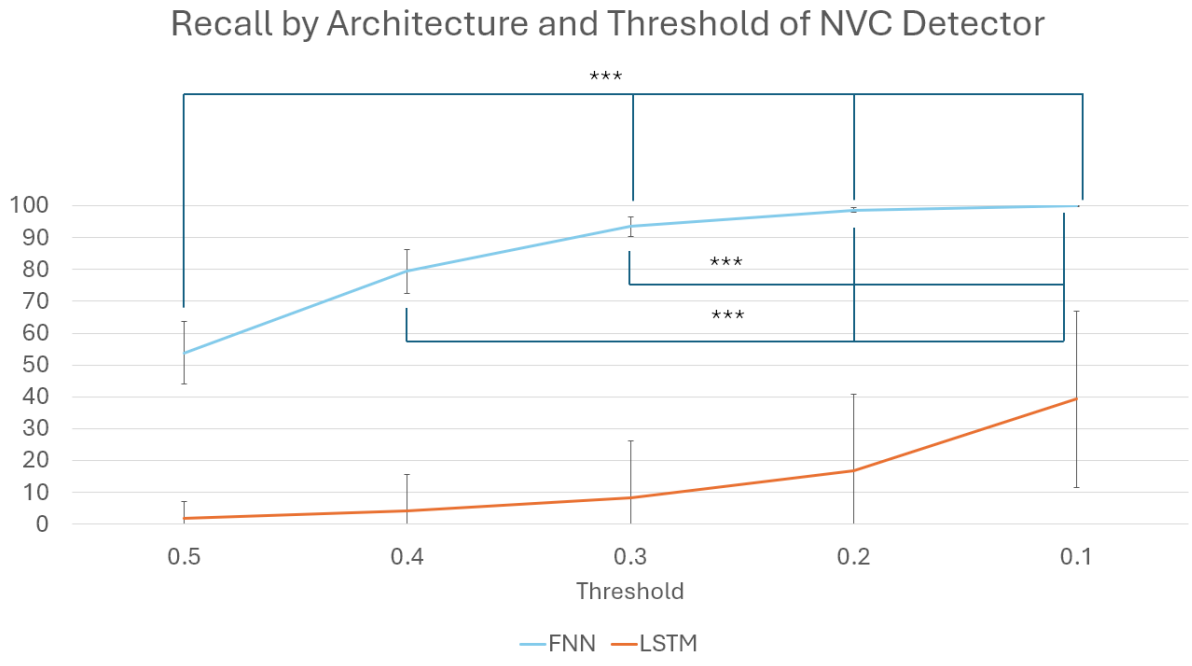

Figure 5.18: Interaction Between Threshold and Architecture on Recall for the NVC Detector ($***p < 0.0005$)

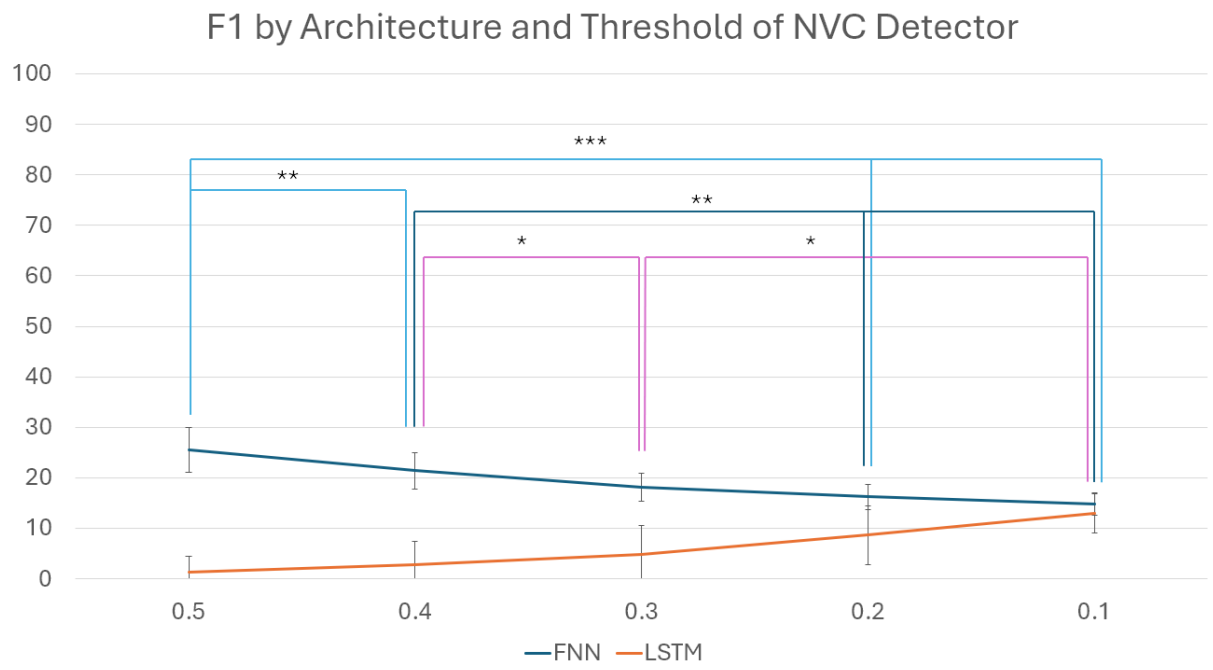

Figure 5.19: Interaction Between Threshold and Architecture on F1 for the NVC Detector (*$p <$ 0.05, **$p < 0.005$, ***$p < 0.0005$)

way ANOVA test. It found a significant effect due to architecture (F(1, 9) = 441.52. $p < 0.001$) but not for threshold (F(4, 9) = 1.98. $p = 0.10$). However, a significant interaction between architecture and threshold was found (F(4, 9) = 39.63. $p < 0.001$). The significant differences found in architecture show that the FFN architecture performed significantly better than the LSTM architecture. The interaction between architecture and threshold is shown in Figure 5.19. It shows that the difference between architectures is diminished as the threshold increases, until performance is almost equal at 0.1. Given that threshold has a significant effect when considered with architecture, and given that the FFN architecture was found to significantly outperform the LSTM architecture, a one-way ANOVA test was carried out to compare threshold level with regard to the FFN architecture alone. It found that threshold had a significant effect on F1 (F(1, 3) = 30.23. $p < 0.0001$). A post-hoc Tukey test was used to examine the effect of thresholds on FFN F1; the results of which are displayed in Figure 5.19 (for exact p-values and confidence values see Table B.21). It was determined that threshold 0.5 had the highest average F1 score that was significantly better than all the lower thresholds.

Finally, the percentage of data removed was compared across both the thresholds and architectures. A two-way ANOVA test found that there were significant differences due to the threshold (F(4, 9) = 51.43, $p < 0.0001$) and architecture (F(1, 9) = 448.30. $p < 0.0001$); there was also a significant interaction between the threshold and architecture (F(4, 9) = 14.54, $p < 0.0001$). In terms of the architecture, the LSTM-based detector removed significantly more than the FFN across all the thresholds. One-way ANOVA and post-hoc Tukey tests were run for each architecture, with the results displayed in Figures 5.20 (for the LSTM-based system) and 5.21 (for

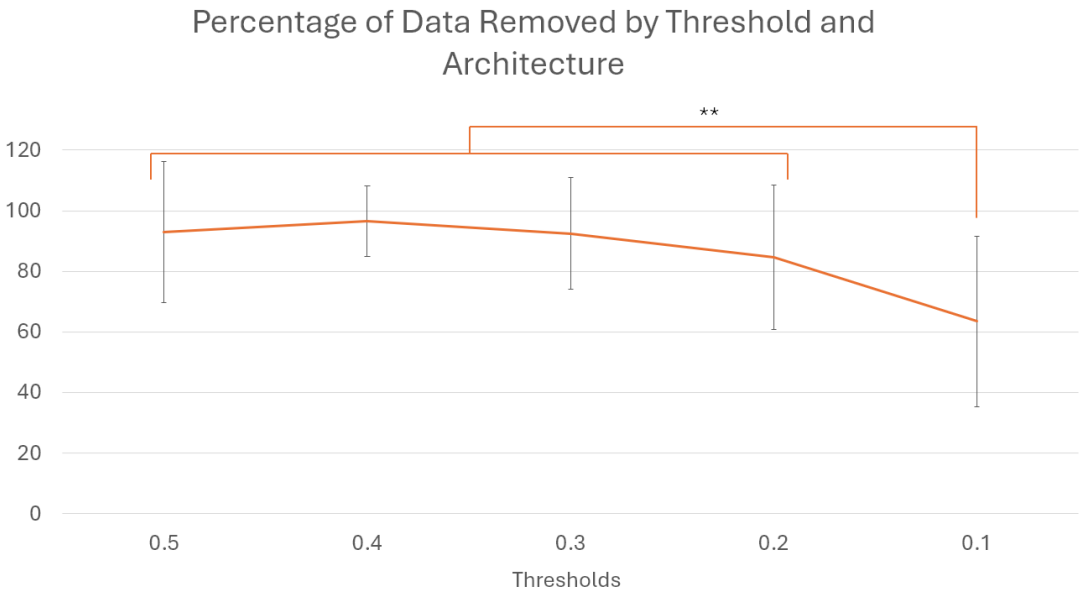

Figure 5.20: Interaction Between Threshold and Architecture on the Percentage of Data Removed by LSTM-Based NVC Detector (**$p < 0.005$)

the FFN-based system). The LSTM-based architecture removed more data than the FFN. Due to the two-way ANOVA test showing a significant interaction between the threshold and architecture, a one-way ANOVA test was carried out on the LSTM performance only to investigate the optimal threshold for the LSTM system. It found significant differences between the thresholds ($F(4, 1) = 6.33$, $p < 0.0001$). Post-hoc Tukey HSD tests showed that the highest overall percent of data removed was achieved by threshold 0.4. However, this was not significantly different to thresholds 0.5, 0.3 or 0.2. It was significantly better than threshold 0.1 (for exact p-values and confidence values see Table B.22). This would suggest that if the LSTM architecture was utilized the threshold value of 0.2 would be best as it one of the highest F1 scores with no significant change in the percentage of data removed. A one-way ANOVA test also showed significant differences in percent removed by the FFN architecture ($F(4, 1) = 30.23$, $p < 0.0001$). The results of a post-hoc Tukey HSD test found that a threshold of 0.5, 0.4 and 0.3 each removed significantly more data than all values below them (for exact p-values and confidence values see Table B.23).

The NVC detector is intended to be used as a filter. The perfect filter would have a recall of 100% and remove all the frames that do not contain NVC (around 90% of the data). Therefore, the goal of choosing an architecture and threshold is to maximise both a high recall and a high percentage of data removed. Most important is recall since, with a low recall, the second stages will be unable to detect any NVC and this would, therefore, make the system ineffective. As FFN architectures have significantly better recall, they were selected as the base architecture. Threshold 0.3 and those with a value below it all had a recall of over 90%. However, they removed less than a third of the data, meaning their effect as a filter may be considered insub-

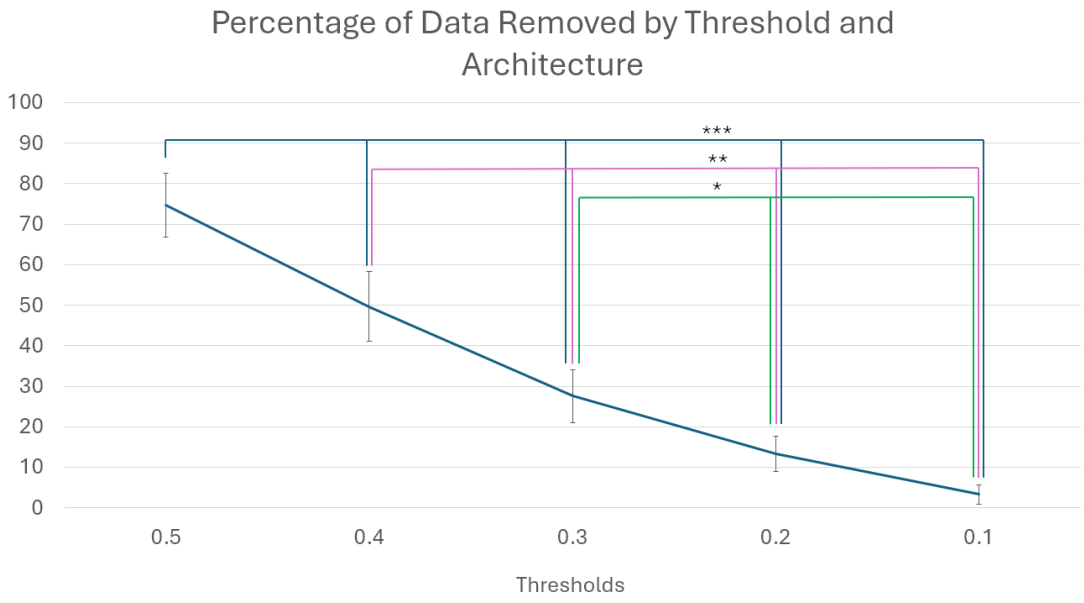

Figure 5.21: Interaction Between Threshold and Architecture on the Percentage of Data Removed by FFN-Based NVC Detector (*$p < 0.05$, **$p < 0.005$, ***$p < 0.0005$)

stantial. Increasing the threshold to 0.4 reduces recall to ∼80%; however, around half of the data would then be filtered out. Therefore, selection of threshold 0.4 is the best balance between capturing NVC and filtering out non-NVC frames.

Results: Individual Detectors

The performance of each individual cue detector is shown in Table 5.13 for frame level results and Table 5.14 for event level outcomes.

Table 5.13: Frame Level Performance of Each Individual Class Detector for the SMC. FFN: feed forward neural network. LSTM: long short-term memory. CW: class weight. S: smoothing

Cue	Architecture	Precision	Recall	F1	AUC
Back-Channel	FFN+CW+S	1.98 ± 0.20	76.16 ± 6.92	3.85 ± 0.38	75.58 ± 2.29
	LSTM+CW+S	1.26 ± 0.64	53.01 ± 31.51	2.27 ± 1.10	49.78 ± 10.27
Filler	FFN+CW+S	9.65 ± 2.45	40.37 ± 10.59	15.16 ± 2.95	71.04 ± 1.64
	LSTM+CW+S	8.73 ± 8.95	27.48 ± 28.94	6.90 ± 4.62	55.51 ± 7.06
Laughter	FFN+CW+S	7.63 ± 7.45	44.52 ± 29.20	9.54 ± 6.55	61.59 ± 20.16
	LSTM+CW+S	8.25 ± 2.30	73.30 ± 11.72	14.62 ± 3.50	84.16 ± 1.70

First examining the effect of architecture, independent t-tests were used to compare the performance of each architecture for each cue on each metric. Figure 5.22 displays the results of this significance testing. For back-channel detection, the FFN architecture saw significantly better performance using the FFN architecture than the LSTM (for exact t statistics and p-values see Table B.24). This offers strong support for using FFN as the architecture for the back-channel detector.

Table 5.14: Event Level Performance of Each Individual Class Detector for the SMC. FFN: feed forward neural network. LSTM: long short-term memory. CW: class weight. S: smoothing

Cue	Architecture	Precision	Recall	F1
Back-Channel	FFN+CW+S	3.12 ± 0.79	34.35 ± 6.92	5.71 ± 1.37
	LSTM+CW+S	1.51 ± 0.87	21.40 ± 6.07	2.71 ± 1.38
Filler	FFN+CW+S	8.03 ± 2.17	28.30 ± 5.78	12.44 ± 3.06
	LSTM+CW+S	8.33 ± 9.43	19.16 ± 12.43	8.39 ± 3.92
Laughter	FFN+CW+S	13.00 ± 15.31	39.71 ± 24.09	12.52 ± 9.15
	LSTM+CW+S	16.41 ± 3.95	70.62 ± 9.73	26.30 ± 5.14

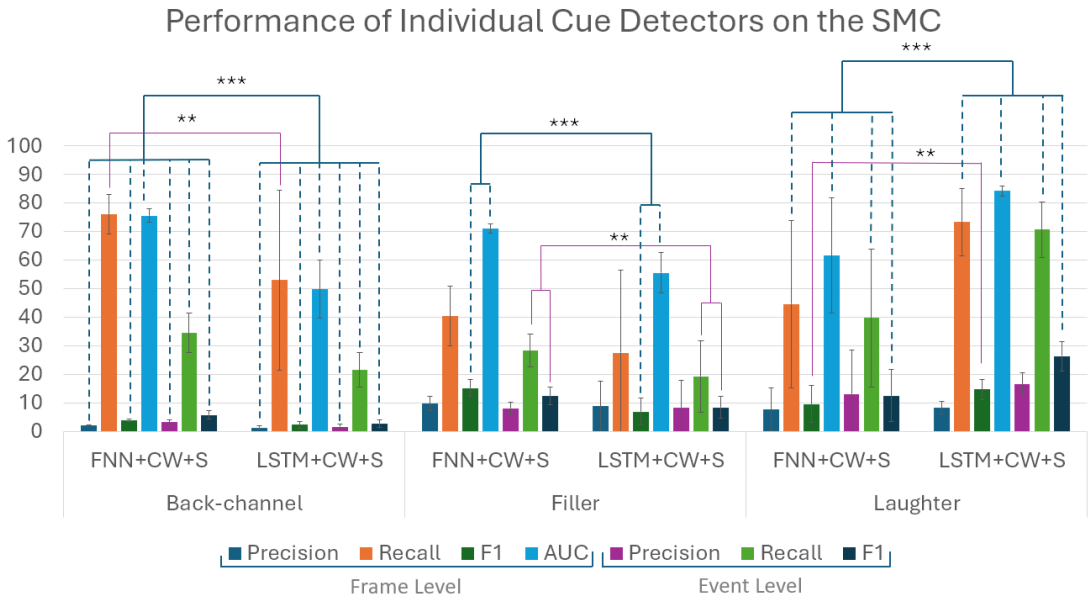

Figure 5.22: Comparison of Performance by Metric and Architecture for Each Cue (**$p < 0.005$, ***$p < 0.0005$)

For filler detection, the FFN architecture was significantly better in frame level F1, AUC, event level recall and F1. There was no significant difference in frame level precision, recall or event level precision (for exact t statistics and p-values see Table B.25). Although some metrics saw no significant difference, these results once again suggest that the FFN-based detector is the better choice for filler detection.

Finally, the opposite to the above is found in regard to laughter. Here, t-tests found that the LSTM architecture significantly outperformed the FFN architecture in frame level recall, F1, AUC, event level recall and F1. There was no significant difference found in frame or event level precision (for exact t statistics and p-values see Table B.26). These results suggest that, for the two-stage detection process using the individual cues, the FFN-based networks should be used for back-channel and fillers and the LSTM network for laughter.

Results: Joint NVC with Individual Detectors

Tables 5.15 (frame level) and 5.16 (event level) display the results achieved for each cue and the overall average of the two-stage detector using individual cue detectors. Furthermore, the table displays the results achieved by each individual detector without the second stage to act as a baseline.

Table 5.15: Frame Level Performance on Each Class for the Two-Stage Detector Using Individual Detectors for Each Cue Alongside the Baseline. FFN: feed forward neural network. LSTM: long short-term memory

	Architecture	Precision	Recall	F1	AUC
Back-Channel	FFN	2.34 ± 0.27	51.71 ± 10.50	4.46 ± 0.49	74.81 ± 2.29
Filler	FFN	10.63 ± 2.99	33.04 ± 7.39	15.58 ± 3.02	71.07 ± 1.20
Laugh	LSTM	21.90 ± 10.69	1.46 ± 2.26	2.27 ± 2.84	56.90 ± 3.77
Average	-	11.62 ± 4.65	28.74 ± 6.72	7.44 ± 2.12	67.59 ± 2.42
			Baseline		
Back-Channel	FFN	1.98 ± 0.20	76.16 ± 6.92	3.85 ± 0.38	75.58 ± 2.29
Filler	FFN	9.65 ± 2.45	40.37 ± 10.59	15.16 ± 2.95	71.04 ± 1.64
Laugh	LSTM	8.25 ± 2.30	73.30 ± 11.72	14.62 ± 3.50	84.16 ± 1.70

Table 5.16: Event Level Performance on Each Class for the Two-Stage Detector Using Individual Detectors for Each Cue Alongside the Baseline. FFN: feed forward neural network. LSTM: long short-term memory

	Architecture	Precision	Recall	F1
Back-Channel	FFN	3.21 ± 0.78	33.87 ± 7.03	5.85 ± 1.38
Filler	FFN	8.45 ± 2.31	29.68 ± 5.87	13.09 ± 3.24
Laugh	LSTM	12.90 ± 7.91	30.07 ± 11.99	16.33 ± 5.12
Average	-	8.19 ± 3.67	31.21 ± 8.30	11.76 ± 3.25
		Baseline		
Back-Channel Baseline	FFN	3.12 ± 0.79	34.35 ± 6.92	5.71 ± 1.37
Filled Baseline	FFN	8.03 ± 2.17	28.30 ± 5.78	12.44 ± 3.06
Laugh Baseline	LSTM	16.41 ± 3.95	70.62 ± 9.73	26.30 ± 5.14

The effect of the two-stage detector was tested using t-tests to compare its performance against the baseline for each cue and metric. The results of this testing are displayed in Figure 5.23 (for exact t statistics and p-values for back channel see Table B.27). Regarding back-channel, the two-stage detector saw significantly better performance than the baseline in frame level precision and F1. While the baseline was significantly better in frame level recall. Furthermore, there were no significant differences in AUC, event level precision, recall or F1. Despite the success in increasing precision and F1 at a frame level, the effect was small. Moreover, at an event level, these effects disappear, suggesting that the two-stage detector is ineffective for back-channels.

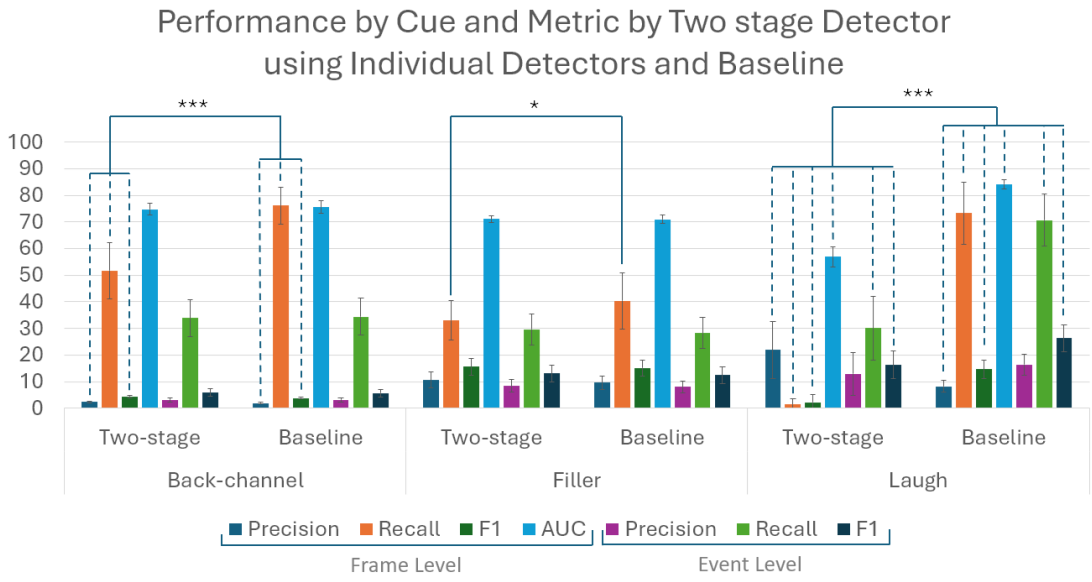

Figure 5.23: Results Achieved by Individual Class Detectors Alone (Baseline) and With the Two-Stage System ($*p < 0.05$, $***p < 0.0005$)

Regarding fillers, the baseline system was significantly better in frame level recall than the two-stage system. In all the other metrics no significant difference was found (for exact t statistics and p-values for filler see Table B.28). These results again suggest that the two-stage detector was ineffective in aiding in the detection of fillers.

Finally, with regard to laughter, the two-stage detector was significantly better in frame level precision. However, the two-stage detector was significantly worse in frame level recall, F1, AUC, event level recall and F1. There was no significant difference found in event level precision (for exact t statistics and p-values for laughter see Table B.29). This again reinforces the conclusion that the two-stage detector, which utilises the individual detectors as a second stage, is ineffective for improving detection of paralinguistic cues.

Results: Paralanguage Distinguisher Effectiveness

This section reports the results on the detection of each cue in the undersampled SMC dataset. In this case, the undersampled dataset only contained frames that have a paralinguistic cue label. Tables 5.17 (frame level) and 5.18 (event level) show the results for each cue individually and the overall macro average for each metric.

Initially, the underlying architecture was compared by cue using independent t-tests; the results are shown in Figure 5.24. For back-channel, the FFN architecture was significantly better than the LSTM architecture in frame level precision, F1, AUC, event level precision and F1. There was no significant difference found for frame level recall or event level recall (for exact t statistics and p-values for back-channel see Table B.30). This suggests that the FFN distinguisher should be used for back-channel detection.

Table 5.17: Frame Level Performance on Each Class for the Paralanguage Distinguisher on the Down-Sampled SMC. FFN: feed forward neural network. LSTM: long short-term memory

	Architecture	Precision	Recall	F1	AUC
BC	FFN	21.82 ± 7.10	35.94 ± 17.56	23.58 ± 5.71	66.47 ± 5.45
Laugh		53.96 ± 9.28	83.01 ± 5.44	64.64 ± 5.81	82.60 ± 2.92
Filler		64.99 ± 5.29	83.80 ± 7.06	72.91 ± 4.14	74.21 ± 3.32
Average		40.92 ± 7.22	67.58 ± 10.02	53.71 ± 5.22	74.43 ± 3.90
BC	LSTM	12.67 ± 3.15	33.33 ± 34.29	13.29 ± 9.24	52.23 ± 4.68
Laugh		51.23 ± 18.13	56.40 ± 27.33	46.23 ± 12.58	66.43 ± 7.53
Filler		58.43 ± 11.07	74.35 ± 28.84	61.45 ± 17.58	59.35 ± 9.94
Average		40.78 ± 10.78	54.69 ± 30.15	40.32 ± 13.13	59.34 ± 7.38

Table 5.18: Event Level Performance on Each Class for the Paralanguage Distinguisher on the Down-Sampled SMC. FFN: feed forward neural network. LSTM: long short-term memory

	Architecture	Precision	Recall	F1
BC	FFN	23.15 ± 8.07	99.22 ± 1.47	36.85 ± 9.80
Laugh		56.02 ± 9.61	100.00 ± 0.00	71.33 ± 7.89
Filler		67.15 ± 5.33	99.74 ± 0.20	80.13 ± 3.92
Average		48.77 ± 7.67	99.65 ± 0.56	62.77 ± 7.20
BC	LSTM	12.91 ± 2.55	97.21 ± 4.38	22.70 ± 3.94
Laugh		46.15 ± 18.29	99.96 ± 0.16	61.23 ± 15.29
Filler		51.32 ± 13.87	98.20 ± 3.68	66.44 ± 12.59
Average		36.79 ± 11.57	98.46 ± 2.74	50.12 ± 10.61

Regarding the filler cue, the FFN architecture was significantly better for frame level precision, F1, AUC, event level precision and F1. There was no significant difference in frame level recall or event level recall (for exact t statistics and p-values for filler see Table B.31).

Examining the performance of the architectures with respect to laughter detection, independent t-tests found that the FFN architecture significantly outperformed the LSTM architecture in frame level recall, F1, AUC, event level recall and event level F1. No significant differences between architectures were found in frame or event level precision (for exact t statistics and p-values for laughter see Table B.32).

Finally, regarding the macro-average across cues, the FFN architecture was again significantly better than the LSTM in frame level F1, AUC, event level precision and F1. There were no significant differences found for frame level precision, recall or event level recall (for exact t statistics and p-values for the macro average see Table B.33).

These results further support that the FFN architecture is a significantly better choice than LSTM for the cue distiguisher. As such, it was selected as the base architecture moving forward. It is important to note that the increase in absolute performance for both architectures is almost certainly due to the down-sampling of the SMC. The task addressed by the distinguisher is closer to a type 1 task rather than a type 3 task.

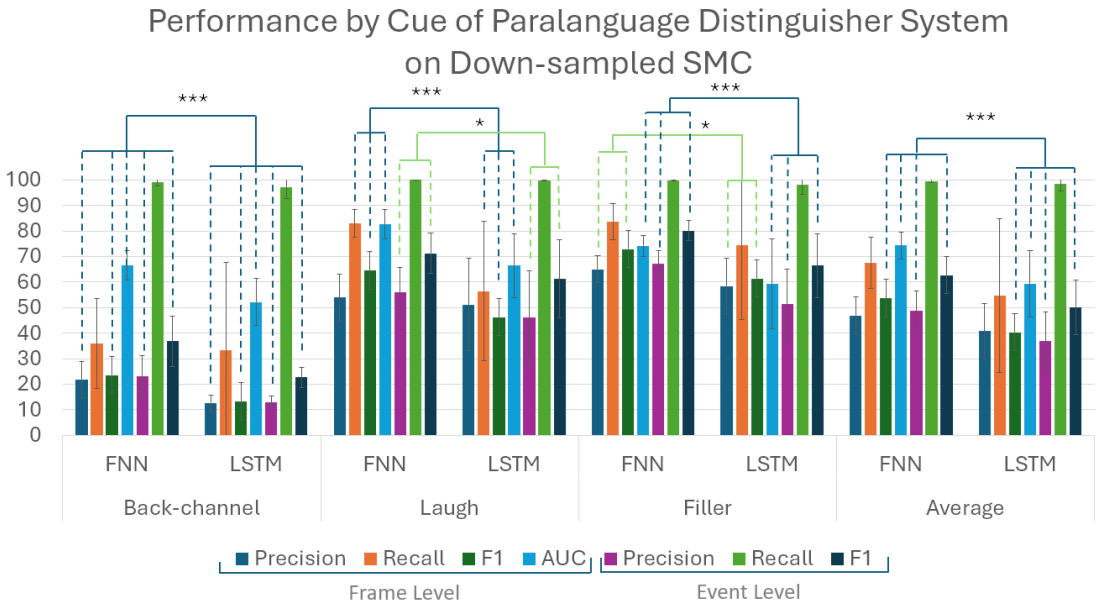

Figure 5.24: Performance by Architecture and Metric for Each Cue on the Down-Sampled SMC ($*p < 0.05$, $***p < 0.0005$)

Results: Joint NVC Distinguisher effectiveness

In this section, the paralanguage distinguisher presented in the previous section is coupled with the NVC detector. This two-stage system was applied to the entirety of the SMC dataset. Frame level results by cue are shown in Table 5.19, with event level results shown in Table 5.20. The individual cue detectors are once again shown to provide a baseline.

Table 5.19: Frame Level Performance on Each Class for the Two-Stage Detector with NVC Distinguisher on the SMC

	Precision	Recall	F1	AUC
BC	3.23 ± 0.94	15.12 ± 8.40	4.81 ± 1.22	75.02 ± 2.25
Laugh	10.10 ± 3.36	56.84 ± 6.00	16.77 ± 4.59	78.62 ± 1.66
Filler	9.04 ± 2.62	46.38 ± 7.78	14.81 ± 3.29	70.66 ± 1.52
		Baseline		
BC	1.98 ± 0.20	76.16 ± 6.92	3.85 ± 0.38	75.58 ± 2.29
Laugh	9.65 ± 2.45	40.37 ± 10.59	15.16 ± 2.95	71.04 ± 1.64
Filler	8.25 ± 2.30	73.30 ± 11.72	14.62 ± 3.50	84.16 ± 1.70

The performance of the two-stage detector with paralanguage distinguisher was compared with the baseline for each metric using independent t-tests. The results of this analysis are displayed in Figure 5.25. With regard to back-channel performance, there were mixed results. The two-stage system was significantly better in terms of frame level precision and F1. However, the baseline system was significantly better in frame level recall and event level recall. Furthermore, there was no significant difference in AUC, event level precision or F1 (for exact t statistics and p-values for back channel see Table B.34).

Table 5.20: Performance on Each Class for the Two-Stage Detector with NVC Distinguisher on the SMC

	Precision	Recall	F1
BC	3.54 ± 0.96	24.09 ± 7.55	5.99 ± 1.32
Laugh	8.82 ± 2.85	66.14 ± 5.95	15.46 ± 4.54
Filler	7.10 ± 2.13	31.64 ± 7.85	11.55 ± 3.22
		Baseline	
BC	3.12 ± 0.79	34.35 ± 6.92	5.71 ± 1.37
Laugh	8.03 ± 2.17	28.30 ± 5.78	12.44 ± 3.06
Filler	16.41 ± 3.95	70.62 ± 9.73	26.30 ± 5.14

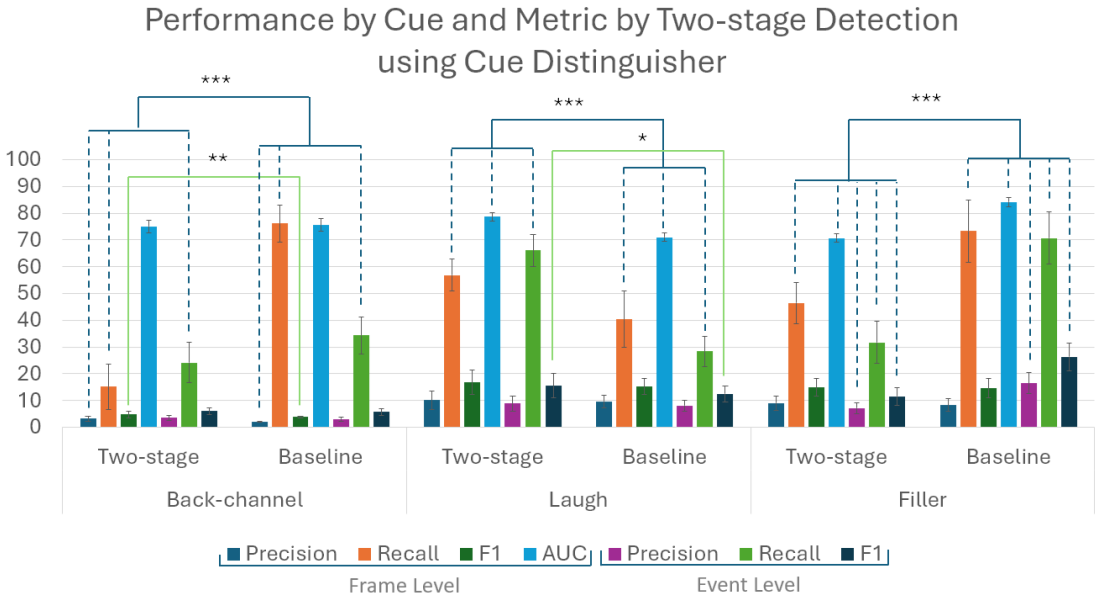

Figure 5.25: Performance by the Two-Stage Detection System with Cue Distinguisher on the SMC Alongside Individual Cue Detector Baseline

Finally, when applied to the filler class, it was found that the baseline approach was significantly better for frame level recall, AUC, event level precision, recall and F1. No significant difference was found in frame level precision or F1 (for exact t statistics and p-values for filler see Table B.35).

With regard to laughter, different results were found, with the two-stage detector having significantly better frame level recall, AUC, event level recall and F1. With no significant difference found in frame level precision, F1 or event level precision (for exact t statistics and p-values for laughter see Table B.36).

These results suggest that the two-stage approach with a cue distinguisher was somewhat successful. However, this success was very limited in, for example, the back-channel class in which the absolute size of the significant increase in performance for frame level precision and F1 is comparatively small. Moreover, with regard to fillers, the two-stage system is consistently worse than the baseline.

5.3 Conclusion

The methods explored in the first half of this chapter addressed RQ2: Can the incorporation of linguistic data lead to improvements in laughter detection? The results were variable. The inclusion of the ASR information was effective at increasing F1, with the effect being more pronounced at the event level compared with the frame level. For the ASR-based methods, this increase was driven by these new methods improving precision, with a smaller drop in recall when compared with the methods explored in Chapter 4. Overall, this led to an almost two-fold increase in frame and event level F1. Unfortunately, due to how the methods operate, it is not possible to couple multiple of them together. Furthermore, improvements to the methods require improvements to the underlying ASR technology. This means that, as ASR improves, it is likely that these methods will provide better results. However, improving ASR is outside the scope of this project and so, for the moment, the current results are the limit of what these methodologies can achieve. Therefore, in relation to RQ2, it has been demonstrated that linguistic information can improve laughter detection results. However, limitations in ASR create a performance ceiling that is difficult to improve upon using these methods.

With respect to the multi-cue multi-layer approaches presented in the second half of the chapter, the majority of these approaches saw significant decreases in performance by cue. Moreover, any gains made were generally at the cost of significant drops in other performance metrics. These methods addressed RQ3: What is the effect of broadening the scope of laughter detectors to include multiple cues? The results suggest that, although some gains could be made by further researching these methods, any gains are likely to be limited.

Having investigated both RQ2 and RQ3 it is apparent that, although ASR data was effective at improving laughter detection, both methodological approaches have performance ceilings. Further, although approaches centred on RQ2 were more effective that the state-of-the-art, their performance ceiling peaked at around an F1 of 50%. As such new approaches had to be investigated to achieve effective laughter detection.

Chapter 6

Transformer-Based Laughter Detection

6.1 Motivation

In Section 4.2.9, it was shown that pre/post processing methods from the field were unable to increase recall without severe drops in precision. The previous Chapter explored novel pre/post-processing methods that aimed to improve laughter detection through the improvement of recall without effecting precision. This was done by specifically targeting the cause of the poor precision while attempting to not affect recall at all, namely, speech being mistaken for laughter. These methods were shown to be more effective, with a significant increase in F1 of between 10-20%. However, these methodologies also reach a performance ceiling of below 50% F1 on average. Passing this performance ceiling was not possible without improving the underlying ASR or VAD effectiveness.

In this chapter a different approach was trialled. Rather than attempting to improve alter the output from the detectors to address mistakes made by them a model driven approach was used. New underlying deep learning architectures were explored alongside new methods of extracting features from the underlying data to improve detector performance. Both of these goals were addressed through the use of transformers. Transformers have been widely used for ASR [91–93], image processing [94–96] and healthcare [94, 97, 98]. In all of the above fields, transformers have yielded significantly better results than other machine learning architectures based on benchmarks in multiple fields [95, 99]. Due to these promising results RQ4 was created: Are transformers effective when applied to the task of laughter detection? Transformers were applied in two distinct ways. Firstly, encoder only transformers were built and trained on the SMC. Secondly, pre-trained transformers were used to extract features and these features were then employed to train detectors. The results show this second method lead to significant improvements in all metrics compared with all other models. The limits of these more effective detectors are then tested before possible use-cases are explored, including understanding the effects of gender on laughter in a conversation.

Transformers leverage an attention mechanism to enable a longer look back through sequen-

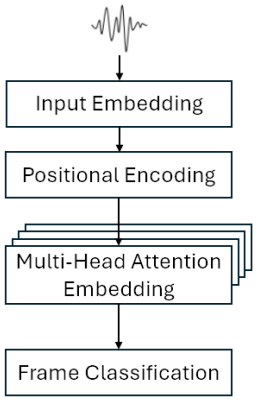

Figure 6.1: General Form of an encoder only Transformer

tial data than other deep learning architectures such as LSTMs. They can do this because the attention mechanism only selects relevant parts of the input data [100]. This targeted look-back allows input sequences to be of any length, while also allowing information contained at any point in the input sequence to have an effect on the output. The LSTM models used in previous chapters had an input sequence length of 10 frames, meaning it had access to data spanning a context of 0.12s to make a prediction. Although it is possible to extend this context window further, the forget gates within LSTMs (explained in Section 4.2.2) ensure that data from previous frames is forgotten meaning that there are limits to how far back an LSTMs context can be extended before new frames become meaningless [101]. As the average length of laughter in the SMC is 0.5s it seems likely that a longer context, than afforded by LSTMs, could result in better detection results. Transformers are capable of handling longer context and it was for this reason that transformers were applied to the current work of laughter detection.

The general form of an encoder only transformer is shown in Figure 6.1 from ref. [100]. The encoder layer takes as input a sequence of real world data (i.e., an audio clip, a sequence of words or frames from a video) and creates a numerical embedding of this data. This embedding represents each individual part of the input sequence (words, sound frames and picture frames) in relation to all the other relevant parts of the sequence. What is considered most relevant is a learned parameter of the attention mechanism described below. This embedding can then be used to create labels for classification tasks or tokens representing words, objects or sounds. Since this project labels data as laughter or not there is no practical use for a decoder and, as such, encoder only transformers were utilised.

All transformers begin by taking input data and creating input embeddings of that data. In the case of audio classification, the process for creating these input embeddings is similar to what has been done in previous chapters: the audio signal is split into distinct overlapping windows. In previous chapters, these windows then have summary statistics extracted to form a

descriptive feature vector of the underlying audio. These feature vectors could be used as input into a transformer. However, it is more common to use the underlying waveform of the audio window as an input. This waveform is a 1-dimensional sequence of integers that represent the amplitude of the audio signal at a given time. If a 20 ms window size is used along with a sampling rate of 16000 Hz (used by refs [102, 103]) then each window would be composed of 320 amplitude values. These values are then passed through a small convolution neural network (CNN) to create a feature vector embedding of these values. It is important to note that, as the initial representation of the audio data, some architectures use spectrograms rather than waveforms [104]. Spectrograms are again passed through a CNN to create a feature vector embedding. At the conclusion of the input embedding step, the initial audio stream is converted into a sequence of vectors $I = \{\vec{F}_1, \vec{F}_2, \ldots, \vec{F}_T\}$, where T is the total number of vectors in the sequence.

Transformers do not use recursion, as such the position of each vector in I must have its relative position in the sequence encoded within itself. To undertake this task, a positional encoding layer is applied to I. Here, the input embeddings I are either convolved with a function or another CNN. When a function is used to inject positional information, each feature f within $\vec{F}_{pos}$ at position pos is added to the value of the sine or cosine curve. The value added to each feature depends on the feature's position in the input vector and the pos of the vector in sequence I, with the value being calculated as follows:

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}), \tag{6.1}$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}), \tag{6.2}$$

where pos is the position of $\vec{F}_{pos}$ in I, d_{model} is the total dimension of $\vec{F}$ and i is the position of the feature within $\vec{F}_{pos}$ being addressed. If instead a CNN is used, then the vector $\vec{F}_{pos}$ is utilised as the input and the positional encoding is a learned parameter of the network, which outputs a vector of the same dimensionality as the input. It has been demonstrated that there is no performance difference between a functional as opposed to a learned approach. However, the functional approach results in fewer parameters in the final network and, thus, less computation at training time and, as such, is often used [100]. Furthermore, the functional approach can be extended to any length of I. The positional encoding step takes as input the input embedding sequence I and outputs the sequence $M = \{\vec{m}_1, \vec{m}_2, \ldots, \vec{m}_T\}$, where each vector $\vec{m}$ represents both the frame audio content and its relative position in the input audio.

Key to the effectiveness of the transformers is the ability to select relevant sections of the input sequence upon which to create the internal embedding for each window. This selection is carried out using an attention mechanism. Attention is calculated as follows. Given the sequence $M = \{\vec{m}_1, \vec{m}_2, \ldots, \vec{m}_T\}$, where T is the total number of vectors in the sequence, the goal of the

attention layer is to create an embedding that represents $\vec{p}e_t$ in the context of the surrounding vectors. To undertake this task for every vector in PE, three values are calculated: the Query, Key and Value. These three values are calculated by multiplying each vector in sequence S by weights Q_w, K_w and V_w. This creates, for each vector $\vec{p}e_t$, an associated Q, K and V vector. These vectors are then used to calculate the similarity of $\vec{p}e_t$ compared with all the other vectors in the input sequence and itself. Similarity is calculated by determining the dot-product of

To begin this stage of the work, multiple transformer models were created and trained on the SMC. Two approaches were used for the initial input embedding: waveform and spectrogram. While, for the remaining of the encoder, only transformer architecture was shared across both models.

For input embedding, the two methods attempted were spectrogram and waveform. In both cases, features were extracted from windows of length 20 ms and time-step of 10 ms. In the case of waveform features, this resulted in feature vectors of dimension $D = 320$ while, for the spectrogram, the dimensionality was $D = 128$. In both cases, these input dimensions were convolved using a CNN to create a dimensionality of $D = 512$.

After creating the initial input embeddings of the audio data, the remainder of the approach was shared across the transformers. The positional encoding step was carried out using the functional approach, as described above. For the attention step, multiple numbers of attention heads were trialled (5 to 20 with a step of 5). Finally, a feed forward neural network (FFN) was applied to the attention embeddings to produce a classification for each frame. This FFN had 3 layers: an input layer with 512 nodes, a hidden layer with 100 nodes and an output layer composed of a single node using Sigmoid activation.

None of the tested transformers showed performance beyond chance. In most folds, the transformer model learned to predict the non-laughter class. In all the other cases, the transformer predictions were random. A possible explanation for this failure can be found in the number of parameters in the model in comparison to the amount of available data. Each model contained $\sim$700 K parameters, which is around 10 times the size of the models used elsewhere in this work. With around 10 hours of training material for each fold, a total of 3600 K examples, only around 108 K of these were laughter frames. It appears that despite the SMC being one of the largest datasets in the field in terms of laughter and total audio time, it is not large enough to train an encoder-only transformer.

6.2 Pre-Trained Transformers for Laughter Detection

In this section, pre-trained audio processing transformers were used to extract attention-based embeddings to be used as input for laughter detectors. Due to the limitations found in training transformers from initialisation in the previous section, it was instead decided to test pre-trained

Table 6.1: Descriptive Statistics for Each Pre-Trained Transformer Model

Transformer	Embedding Size	Training Size (hours)	Parameters
HuBERT S	768	960	95 M
HuBERT L	1280	60 K	1 B
WHISPER S	512	680 K	74 M
WHISPER L	1280	5 M	1.55 B
Wav2Vec2 S	768	960	95 M
Wav2Vec2 L	1920	436 K	2 B

transformers ability to transfer learning from another task. In this case, the embeddings created by a transformer are extracted after the attention embedding step but before the CNN step. These attention embeddings are then used as input to a newly initialised neural network (either FFN or LSTM based), which is trained to carry out a different task to the original transformer; in this case, laughter detection. It was theorised that the attention embeddings created by transformers, which had been trained on audio tasks, would effectively represent the underlying audio for other tasks. A similar approach of using detectors for laughter detection, which were pre-trained on a different audio task, was previously attempted in the field [37]. However, this was only attempted on a type 1 task and used patterns in the phonetic outputs of an ASR to detect laughter.

Three pre-trained transformers were selected for testing: HuBERT [102], Whisper [104] and Wav2Vec2 [103]. Each transformer has multiple different sizes, differentiated by the number of parameters, embedding size and training set size. To test the effectiveness of these different sizes for each transformer, the 'small' and 'large' versions were tested (henceforth denoted as L for large and S for small). The differences between them are displayed in Table 6.1.

Each transformer has a unique methodology, which may offer advantages to using its attention embeddings for laughter detection. Wav2Vec2 creates learned embeddings for the input audio through self-supervised unlabelled training data. This creates a unique learned representation of the input audio and was shown to improve results compared with standard feature extraction methodologies [103]. HuBERT similarly creates learned representations for audio data. However, in the HuBERT approach, the goal is to incorporate both auditory and linguistic information into the representation of the audio [102]. Whisper does not use custom speech representations, instead using Mel spectrograms to represent audio inputs. Whisper is differentiated by its training data. The data is around 1000 times larger than what was used to train the other two transformers (when comparing similar sized models). Furthermore, $\sim$30% of the data is not English speech audio. By using non-English audio, the system is exposed to, and can learn, a wider array of sounds and accents [104].

For attention embedding extraction, the window and hop length for splitting the audio was constrained by each underlying transformer, as the same length and hop had to be used as when they were trained. Furthermore, the size of the resulting feature vector was constrained by the internal embedding size each transformer used. These values are included in Table 6.1. Attention

embeddings were extracted using pre-trained transformers available through Hugging Face [1]. This resulted in a sequence of feature vectors. These feature vectors could then follow the same approach as described in Section 4.2.1. Both LSTMs and FFNs based architectures were tested. Initially, no additional methodologies tested in the previous chapters were applied to ensure the effect of the transformer embeddings in isolation could be understood. After displaying the results of the various transformer embeddings, the methods described in the previous chapters are then applied to the best performing transformer-based detector. Next, the best performing detector is applied to the SVC to enable comparison with the field at large.

6.2.1 Training and Testing Methodology

As in the previous chapters, the training and testing methodology followed a k-fold approach. Fold splitting was carried out at a conversational level, using the same splits as previous chapters to ensure comparability between results. Descriptive statistics for each fold can be found in Section 4.2.5. Networks again used binary cross entropy as a loss function coupled with the Adam optimizer with a default learning rate of 0.001. The number of training epochs was 5 and the batch size was 1000.

As with previous chapters, hyper-parameter optimisation was carried out for each transformer and size. Results of the hyper-parameter optimisation step found no advantage compared with deeper or wider networks. For both frame level F1 and AUC, there was no significant effect due to Hamming window size in most of the architectures. However, in Whisper L, Wav2Vec2 S and Whisper S, the smaller Hamming windows performed significantly better. This contrasts, in parts, with the previous chapters in which multiple models saw that larger windows led to better results. This suggests that the transformer embedding results required less alteration to achieve the best frame level results.

Regarding event level F1, a Hamming window size of 11 was consistent across all the detectors and provided significantly better results. Again, this offers support for the lack of a need to adjust posteriors. Moreover, all the detectors had a percent cut-off of 50% or more. This again contrasts with the previous chapters, in which the most effective detectors had lower cut-offs between 1 and 10%. This suggests that the peaks that occurred in the posteriors, produced by the transformer embedding networks, were better aligned with actual events; thus, a great proportion of them could be included in the final classification decision. Overall, the hyper-parameter optimisation process suggests that the transformer embedding detectors achieve more reliable detections.

Figures 6.2, 6.3 and 6.4 show why this is the case. Figure 6.2 displays the posteriors as produced by Whisper L and CBA for a laughter event. The Whisper L detector output is characterised by sharp changes from $\sim$0 to $\sim$100%, whereas the hand-crafted feature detector shows greater variation and uncertainty in its posteriors. The two histograms shown in Figure 6.3

[1] `https://huggingface.co/`

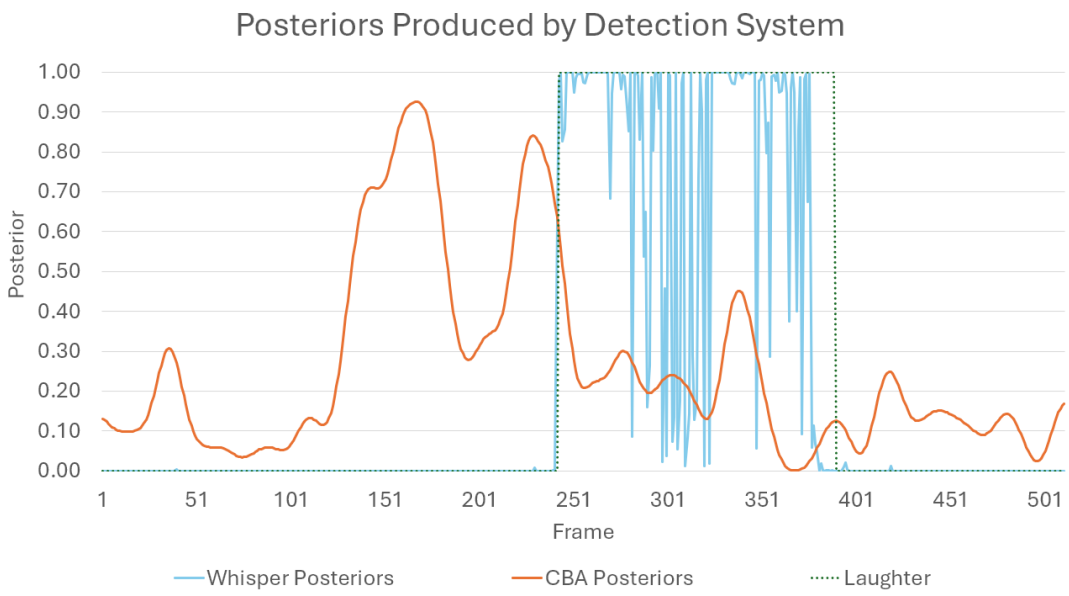

Figure 6.2: Posteriors Produced by the CBA and Whisper L Detection Systems

Table 6.2: Hyper-Parameter Optimisation Results for Each Transformer Based Detector and Metric. S: small. L: large. All systems had two FFN hidden layers with 100 nodes per layer

Model Architecture	Window Size			Percent Cut-Off
	AUC	F1 (frame)	F1 (event)	
Wav2Vec2 S	No Effect	11	11	70
Wav2Vec2 L	No Effect	No Effect	11	50
HuBERT S	No Effect	No Effect	11	60
HuBERT L	No Effect	No Effect	11	50
Whisper S	No Effect	11	11	80
Whisper L	21	No Effect	11	70

(Whisper L) and 6.4 (CBA) confirm that this trend is not an isolated one. Posteriors are almost exclusively constrained to bins 0-0.1 and 0.9-1 for Whisper L. Whereas, in the case of the CBA detection system, there is a greater spread across the bins. The lack of variability in the output of the transformer-based detector leads to fewer spurious peaks in the output sequence and, therefore, a greater percentage of the output can be considered without increasing the number of peaks considered as events compared with previous detectors.

6.2.2 Results: Transformer Embeddings as Input

This section explores the performance of each transformer and how that performance is affected by each different network type. In all cases, LSTM-based architectures saw performance of below 5% in both frame and event level precision, recall and F1. As such, the LSTM architecture was discounted as a possibility and only FFN networks were carried forward. The metrics used mirror those in previous chapters to enable comparison with previous models. Table 6.3 displays

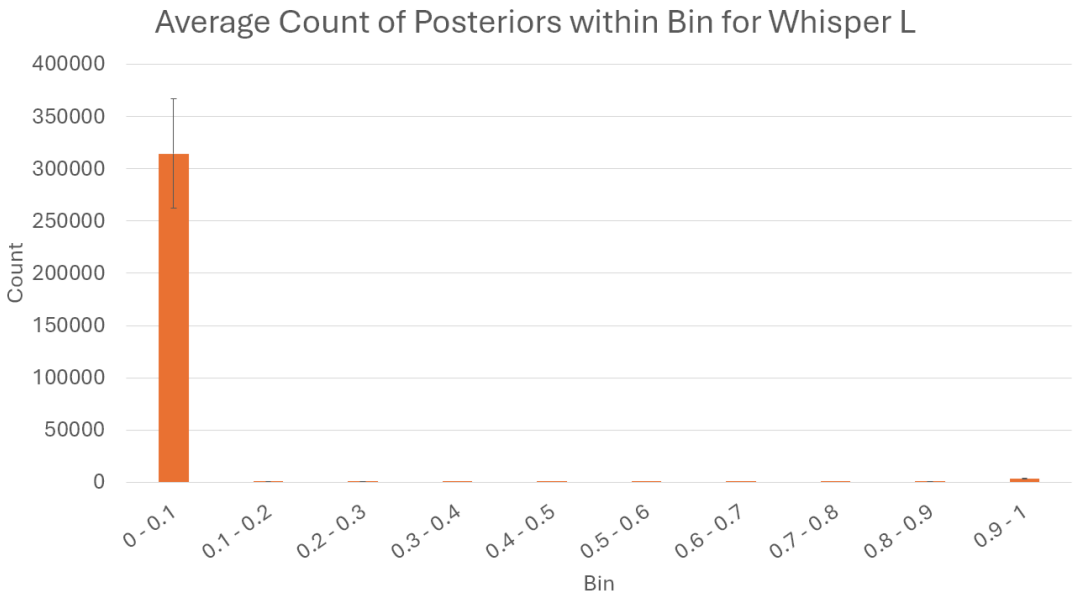

Figure 6.3: Histogram Showing the Average Count of Posteriors Probabilities as Estimated by the Whisper-L-Based Detection System

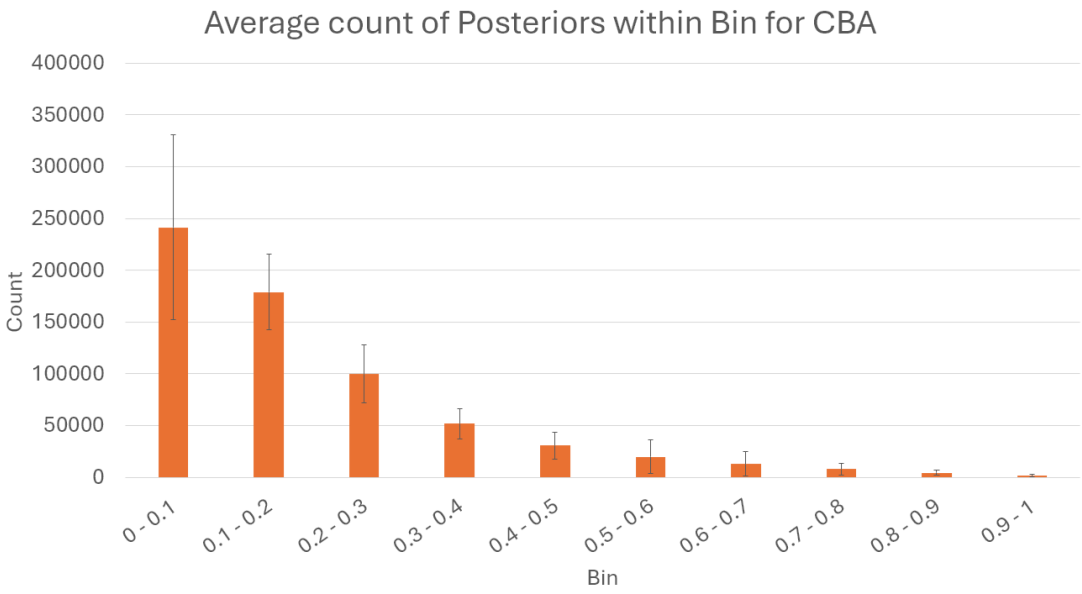

Figure 6.4: Histogram Showing the Average Count of Posteriors Probabilities as Estimated by the CBA Detection System

Table 6.3: Frame Level Precision, Recall, F1 and AUC Results for Each Feature Extraction Methodology Using a Feed Forward Neural Network on the SMC. S: small. L: large. Bold highlights the best performing detector.

	Precision	Recall	F1	AUC
Baseline	10.03 ± 5.59	43.70 ± 20.93	15.86 ± 8.27	79.79 ± 4.24
Wav2Vec S	74.64 ± 11.24	37.74 ± 3.85	49.48 ± 2.70	95.40 ± 0.86
Wav2Vec L	**77.96 ± 8.24**	**52.52 ± 4.52**	**62.25 ± 2.29**	**96.75 ± 0.66**
HuBERT S	73.78 ± 8.75	35.10 ± 7.14	46.82 ± 7.32	95.01 ± 0.97
HuBERT L	70.39 ± 12.84	51.54 ± 12.29	57.27 ± 9.17	96.35 ± 0.69
Whisper S	81.89 ± 5.36	48.38 ± 1.97	60.74 ± 2.22	94.41 ± 1.09
Whisper L	81.94 ± 4.34	46.27 ± 6.85	58.68 ± 4.38	93.09 ± 1.33

Table 6.4: Event Level Precision, Recall and F1 for Each Feature Extraction Methodology Using a Feed Forward Neural Network. S: small. L: large. Bold highlights the best performing detector.

Model	Precision	Recall	F1
Baseline	15.77 ± 8.58	48.09 ± 21.86	23.28 ± 11.83
Wav2Vec S	54.92 ± 13.18	85.46 ± 3.54	65.75 ± 10.30
Wav2Vec L	58.83 ± 13.82	89.23 ± 2.96	69.77 ± 10.08
HuBERT S	51.08 ± 10.79	86.43 ± 3.79	63.36 ± 8.28
HuBERT L	55.23 ± 16.12	84.66 ± 7.67	64.66 ± 10.78
Whisper S	61.21 ± 8.86	88.12 ± 2.76	71.78 ± 6.08
Whisper L	**81.57 ± 6.72**	**84.27 ± 4.10**	**82.60 ± 3.04**

the results for frame level precision, recall, F1 and AUC for FFN. Event level metrics are shown in Table 6.4.

First addressing the AUC scores seen in Table 6.3, a one-way ANOVA test was carried out that showed significant differences between the models (F(6, 119) = 188.86. $p < 0.0001$). A post-hoc Tukey HSD was carried out and found that the baseline was significantly outperformed by all the transformer embedding models. Wav2Vec L achieved the highest overall score. Wav2Vec L was significantly better than Whisper S and Whisper L. However, there were no significant differences between Wav2Vec L and the other transformer based detectors (for exact confidence intervals and p-values see Table C.1). Comparing the large and small variants of each foundational model, it was found there was no significant difference between Wav2Vec L and S ($p = 0.30$, 95% C.I. = [-3.18, 0.48]), HuBERT L and S ($p = 0.31$, 95% C.I. = [-3.17, 0.49]) or Whisper L and S ($p = 0.33$, 95% C.I. = [-0.51, 3.15]). This suggests no advantage to using large over small variants of the transformer models. These results show a consistent and significant increase of $\sim$15% from hand-crafted features to transformer embeddings.

In terms of frame level recall, a one-way ANOVA test found significant differences between detectors (F(6, 119) = 7.66, $p < 0.0001$). A post-hoc Tukey HSD test determined that the baseline detector ($M = 43.70$, $SD = 20.93$) was not significantly different from any of the transformer embedding detectors. Wav2Vec L again achieved the highest overall score and the post hoc tests

found this to be significantly better than Wav2Vec S and HuBERT S. However, no significant difference was found from it to HuBERT L, Whisper S or Whisper L (for exact confidence intervals and p-values see Table C.2). Comparing foundational model size, there was no significant difference between Whisper L and S ($p = 1.00$, 95% C.I. = [-8.07, 12.29]). However, there were significant differences between Wav2Vec L and S ($p = 0.00056$, 95% C.I. = [24.96, 4.60]) and HuBERT L and S ($p < 0.0001$, 95% C.I. = [26.62, 6.26]), suggesting some advantage to using the larger foundational models.

With regard to frame level precision, a one-way ANOVA test found significant differences between detectors (F(6, 119) = 160.26. $p < 0.0001$). A post-hoc Tukey HSD test determined that the baseline detector was significantly outperformed by all the transformer-based detectors. Whisper L achieved the highest overall precision. The post-hoc tests found this to be significantly better than HuBERT L (for exact confidence intervals and p-values see Table C.3). However, there was no significant difference between it and any other transformer detector. In terms of foundational model size, there was no significant difference found between Whisper L and S ($p < 1.00$, 95% C.I. = [-8.62, 8.52]), Wav2Vec L and S ($p < 0.91$, 95% C.I. = [-11.89, 5.25]) or HuBERT L and S ($p < 0.90$, 95% C.I. = [-5.18, 11.96]).

For frame level F1, a one-way ANOVA test found significant differences between detectors (F(6, 119) = 135.78. $p < 0.0001$). A post-hoc Tukey HSD test determined that the baseline detector was significantly outperformed by all the transformer-based detectors. The highest overall score was achieved by Wav2Vec L. This was significantly better than Wav2Vec S and HuBERT S. However, it was not significantly better than any other detector (for exact confidence intervals and p-values see Table C.4). Examining the difference in performance based on the foundational model size, there was a significant difference found between Wav2Vec L and S ($p < 0.0001$, 95% C.I. = [18.66, 6.88]) and HuBERT L and S ($p < 0.0001$, 95% C.I. = [16.34, 4.56]). However, there was no significant difference between Whisper L and S ($p = 0.94$, 95% C.I. = [-3.83, 7.95]).

The above results show a clear advantage to using transformer embeddings as input over hand-crafted features. The effect of using large as opposed to small foundational models is less clear and varies by the metric tested.

Now addressing event level precision, a one-way ANOVA test found significant differences between detectors (F(6, 119) = 51.65. $p < 0.0001$). A post-hoc Tukey HSD test determined that the baseline detector was significantly outperformed by all the transformer-based detectors. Whisper L achieved the highest overall precision. Post-hoc tests found this to be significantly better than all the other detectors (for exact confidence intervals and p-values see Table C.5). Comparing foundational model size, there were significant differences found between Whisper L and S ($p < 0.0001$, 95% C.I. = [31.94, 8.78]) but no significant differences between Wav2Vec L and S ($p = 0.95$, 95% C.I. = [-15.48, 7.67]) or HuBERT L and S ($p = 0.93$, 95% C.I. = [-15.73, 7.43]).

For event level recall, a one-way ANOVA test found significant differences between detectors (F(6, 119) = 44.86. $p < 0.0001$). A post-hoc Tukey HSD test determined that the baseline detector was significantly outperformed by all the transformer-based detectors. Wav2Vec L achieved the highest overall score; however, this was not significantly better than any other transformer-based detector (for exact confidence intervals and p-values see Table C.6). Examining the effect of foundational model size, post-hoc tests found no significant difference between Wav2vec L and S ($p = 0.88$, 95% C.I. = [-13.00, 5.46]), HuBERT L and S ($p = 1.00$, 95% C.I. = [-7.46, 11.00]) and Whisper L and S ($p = 0.87$, 95% C.I. = [-5.38, 13.08]).

A one-way ANOVA test, similarly, found significant differences in detector event level F1 performance (F(6, 119) = 76.04. $p < 0.0001$). A post-hoc Tukey HSD test determined that the baseline detector was significantly outperformed by all the transformer-based detectors. The highest overall F1 was achieved by Whisper L, post-hoc tests found this to be significantly better than all the other transformer-based detectors (for exact confidence intervals and p-values see Table C.7). Comparing the performance based on the size of the foundational model, there was a significant difference found between Whisper L and S ($p = 0.0091$, 95% C.I. = [1.73, 19.91]). However, there was no significant difference found between Wav2Vec L and S ($p = 0.84$, 95% C.I. = [-13.11, 5.07]) or HuBERT L and S ($p = 1.00$, 95% C.I. = [-10.39, 7.79]).

Taken together, these results show that the transformer embedding based detectors are significantly and consistently better than the baseline. The best performing model at an event level was Whisper L, with the main driver of the differences in F1 being caused by differences in precision performance. Of interest is the improvement of the recall metrics from frame to event level for Whisper L, rising by $\sim$40%. This would suggest that the relatively poor recall seen in the frame level metrics is caused by the detectors missing frames within laughter events that were actually detected. Furthermore, the difference from small to large transformer models is much less pronounced at an event level, suggesting the increase in F1 in the frame level statistics is caused by the transformers missing fewer frames within events but both sizes of model being able to detect the same events.

6.2.3 Results: Effect of Pre/Post-Processing Methods on Whisper Large

From the previous section, multiple detectors that utilised the attention embeddings performed equally effectively while outperforming all the previous detectors. These results were achieved without using any of the pre/post-processing methods examined in previous chapters. In an attempt to further increase the performance of the detectors, this section applies these methods on the Whisper Large attention embedding based detectors. Table 6.5 displays the frame level results for each additional method trialled.

For frame level AUC, a one-way ANOVA test found significant differences between detectors (F(13, 238) = 147.09. $p < 0.0001$). The results of a post hoc Tukey test are displayed in Figure 6.5 (for exact confidence intervals and p-values see Table C.8). The post-hoc test deter-

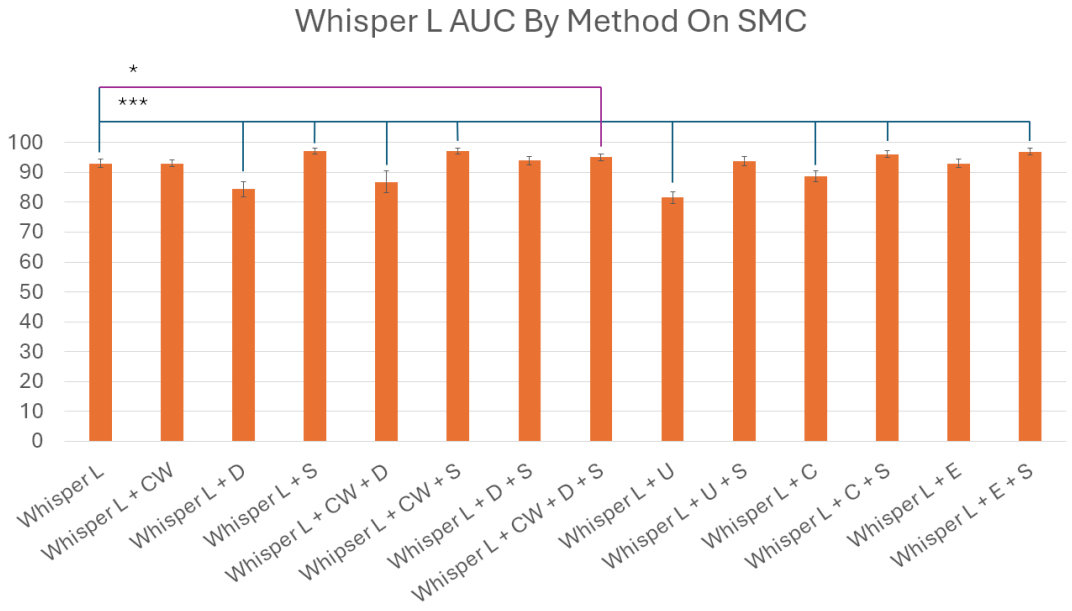

Figure 6.5: AUC Score for Each Different Detection Method on the SMC. Significant Differences Shown in Relation to Whisper L ($*p < 0.05$, $***p < 0.0005$)

mined that Whisper L was significantly better than Whisper L+D, Whisper L+CW+D, Whisper L+U and Whisper L+C. Some methods did lead to significant improvements in AUC (Whisper L+S, Whisper L+CW+S, Whisper L+CW+D+S, Whisper L+C+S, Whisper L+E+S). All others had no significant effect (Whisper L+CW, Whisper L+D+S, Whisper L+U+S, Whisper L+E). Whisper L+CW+S achieved the highest overall AUC. Compared with the other detectors that improved on the baseline, it was not significantly different to Whisper L+E+S or Whisper L+S. It was, however, significantly better than both Whisper L+CW+D+S and Whisper L+C+S (for exact confidence intervals and p-values see Table C.9). Overall, this suggests that some of the additional methods trialled in the previous chapters were successful in increasing AUC but at a maximum of $\sim$5%.

In terms of frame level precision, a one-way ANOVA test found significant differences in performance between detectors ($F(13, 238) = 32.62$. $p < 0.0001$). A post-hoc Tukey HSD test, shown in Figure 6.6, determined that, in three cases, Whisper L performed significantly better (Whisper L+D, Whisper L+CW+D, Whisper L+D+S). In all the other cases, there was no significant effect (for exact confidence intervals and p-values see Table C.10).

In terms of recall, a one-way ANOVA test found significant differences in performance between detectors ($F(13, 238) = 108.75$. $p < 0.0001$). A post-hoc Tukey HSD test, shown in Figure 6.7, determined that Whisper L was significantly outperformed by Whisper L+CW. Four detectors were significantly worse (Whisper L+D, Whisper L+CW+D, Whisper L+D+S and Whisper L+CW+D+S). All the other detectors showed no significant effect (for exact confidence intervals and p-values see Table C.11).

For frame level F1, a one-way ANOVA test found significant differences in performance

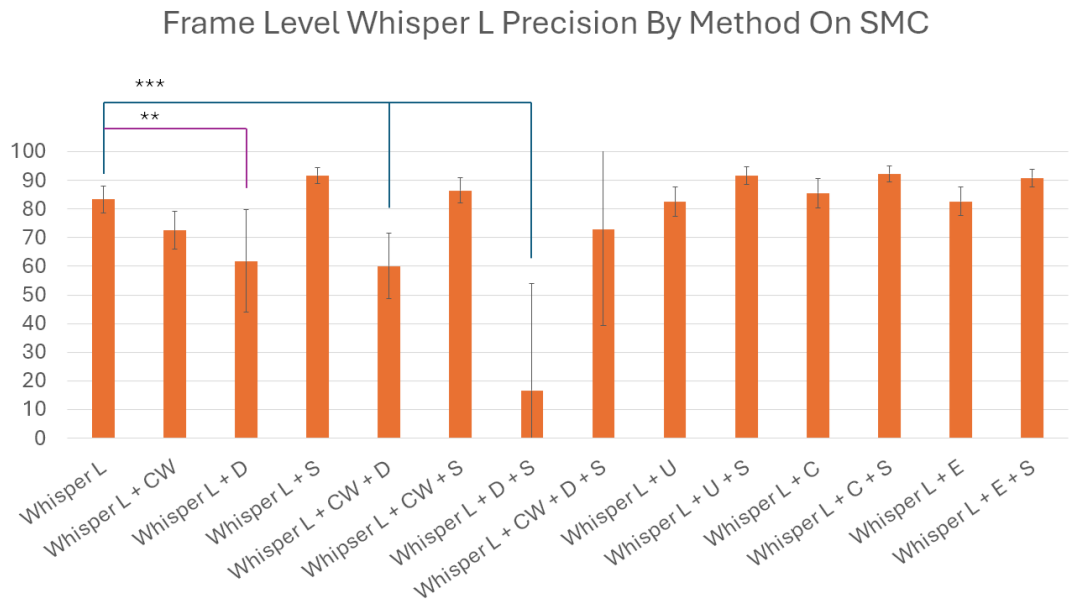

Figure 6.6: Frame Level Precision for Each Different Detection Method on the SMC. Significant Differences Shown in Relation to Whisper L (**$p < 0.005$, ***$p < 0.0005$)

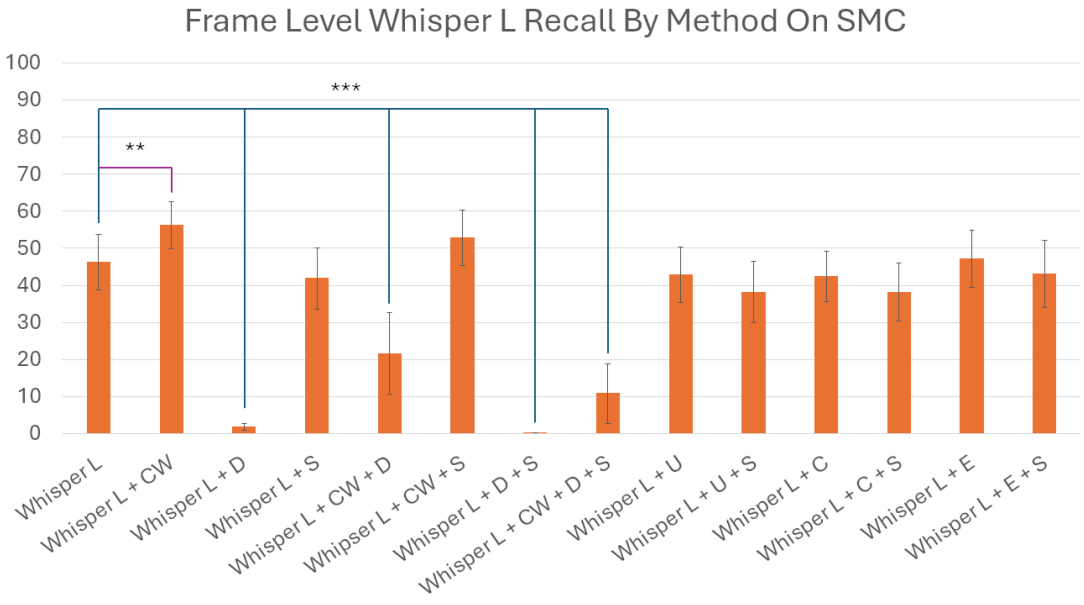

Figure 6.7: Frame Level Recall for Each Different Detection Method on the SMC. Significant Differences Shown in Relation to Whisper L (**$p < 0.005$, ***$p < 0.0005$)

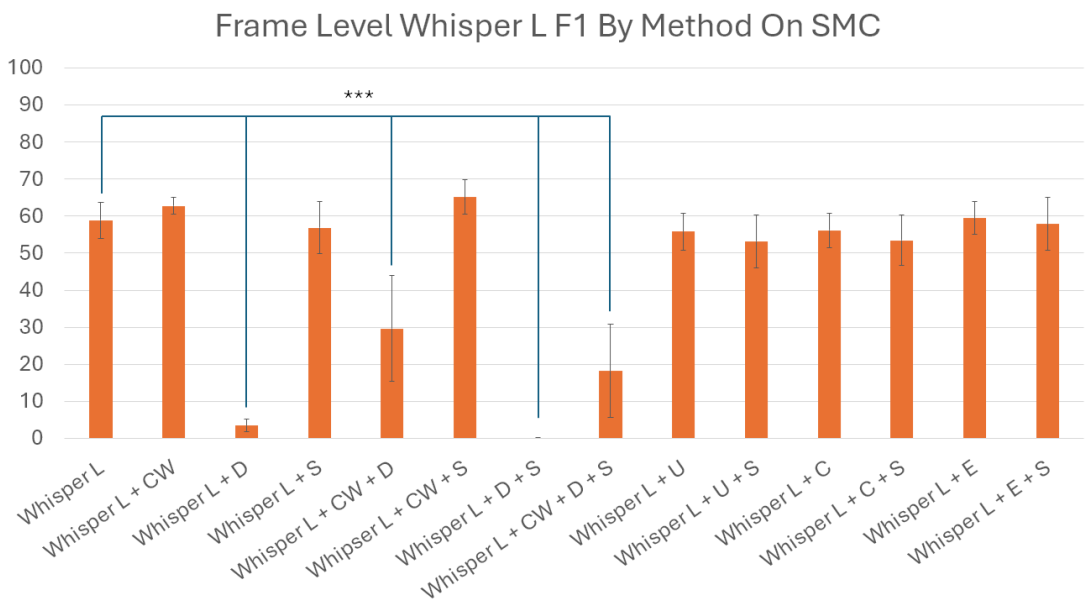

Figure 6.8: Frame Level F1 for Each Different Detection Method on the SMC. Significant Differences Shown in Relation to Whisper L (***$p < 0.0005$)

(F(13, 238) = 185.58. $p < 0.0001$). Figure 6.8 shows the results of a post-hoc Tukey HSD test. The latter determined that Whisper L was significantly better than four of the new detectors (Whisper L+D, Whisper L+D+S, Whisper L+CW+D and Whisper L+CW+D+S). No other significant differences were found (for exact confidence intervals and p-values see Table C.12). Though some methods saw significant increase in precision and recall, there was no improvement in F1. This is due to each method only seeing increases in one of the two metrics and not both. This means that, when F1 is calculated, the differences negate each other. One further point to note is that the inclusion of deltas has a universal detrimental impact on Whisper L performance. This is probably due to lack of temporal constraints in the transformer embeddings, resulting in deltas that are meaningless.

Table 6.6 displays the event level results. First examining the effect on event level precision, a one-way ANOVA test found significant differences in performance (F(13, 238) = 19.68. $p < 0.0001$). Figure 6.8 shows the results of a post-hoc Tukey HSD test. The latter determined that Whisper L was significantly better than two of the new detectors (Whisper L+D and Whisper L+CW+D). No other significant differences were found (for exact confidence intervals and p-values see Table C.13). These results align with the frame level results above in that including deltas in the input feature vector has a significant negative effect on performance. However, at the event level, smoothing appears to mitigate this effect since the method Whisper L+D+S was not significantly different at an event level but was at a frame level.

With regard to event level recall, a one-way ANOVA test again found significant differences between methods (F(13, 238) = 109.78. $p < 0.0001$). Figure 6.10 displays the results of a post-hoc Tukey HSD test. This determined that Whisper L was not significantly different to

Table 6.5: Affect of Methodology Presented So Far on the Performance of the Whisper Large Transformer Embedding Extractions on the SMC at a Frame Level. All models used the FFN base architecture. CW: class weight. D: delta. S: smoothing. U: undersampling. C: confidence-based alteration. E: feature vector extension. Bold highlights the best performing detector.

Detector	Precision	Recall	F1	AUC
Whisper L	83.35 ± 4.67	46.19 ± 7.47	58.90 ± 4.82	92.99 ± 1.38
Whisper L+CW	72.56 ± 6.63	56.24 ± 6.37	62.79 ± 2.23	92.98 ± 1.08
Whisper L+D	61.83 ± 17.86	1.80 ± 0.89	3.48 ± 1.70	84.34 ± 2.48
Whisper L+S	91.66 ± 2.68	41.84 ± 8.38	56.90 ± 7.03	97.16 ± 0.90
Whisper L+CW+D	60.13 ± 11.42	21.58 ± 11.01	29.70 ± 14.26	86.86 ± 3.74
Whipser L+CW+S	**86.44 ± 4.35**	**52.86 ± 7.50**	**65.13 ± 4.65**	**97.19 ± 0.95**
Whisper L+D+S	16.67 ± 37.27	0.03 ± 0.10	0.06 ± 0.20	93.90 ± 1.33
Whisper L+CW+D+S	72.76 ± 33.44	10.78 ± 7.97	18.24 ± 12.61	95.03 ± 1.11
Whisper L+U	82.45 ± 5.13	42.83 ± 7.41	55.80 ± 5.04	81.53 ± 2.03
Whisper L+U+S	91.60 ± 3.10	38.15 ± 8.27	53.25 ± 7.16	93.89 ± 1.54
Whisper L+C	85.50 ± 5.04	42.40 ± 6.83	56.14 ± 4.57	88.80 ± 1.80
Whisper L+C+S	92.26 ± 2.75	38.18 ± 7.75	53.47 ± 6.85	96.08 ± 1.06
Whisper L+E	82.68 ± 5.05	47.15 ± 7.70	59.45 ± 4.43	92.99 ± 1.46
Whisper L+E+S	90.85 ± 3.08	43.15 ± 9.12	57.86 ± 7.13	96.97 ± 1.04

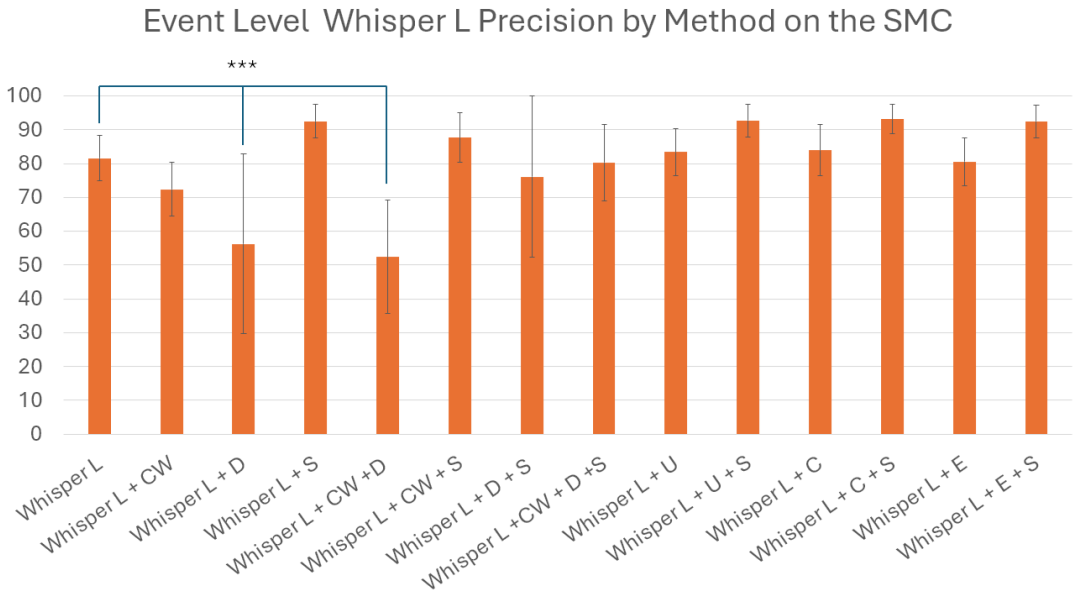

Figure 6.9: Event Level Precision for Each Different Detection Method on the SMC. Significant Differences Shown in Relation to Whisper L (***$p < 0.0005$)

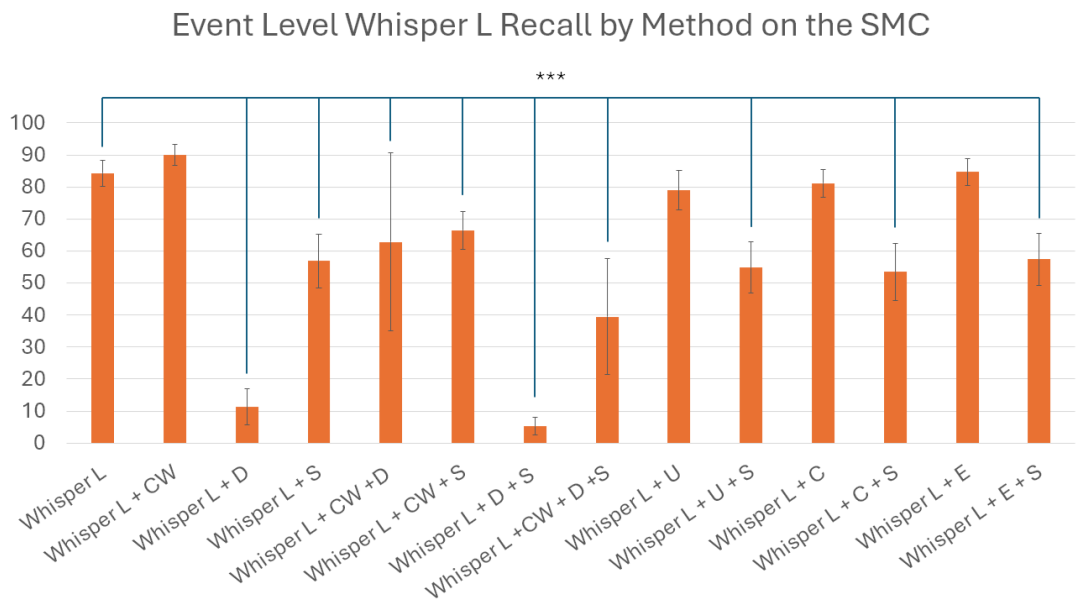

Figure 6.10: Event Level Recall for Each Different Detection Method on the SMC. Significant Differences Shown in Relation to Whisper L (***$p < 0.0005$)

Whisper L+CW, Whisper L+U, Whisper L+C or Whisper L+E. In all other cases Whisper L was significantly better (for exact confidence intervals and p-values see Table C.14). Of note is the fact that the methods developed in Chapter 5 have no significant effect on recall until smoothing was also incorporated, in which case they saw significant drops. This is to be expected because smoothing is a method that improves precision at the cost of recall. However, as precision was already high, smoothing failed to contribute positively at an event level and instead only the negative effects relating to recall descent were seen. As for the methods explored in Chapter 4, only CW had no significant impact on recall.

Regarding event level F1, a one-way ANOVA found significant differences between detectors ($F(13, 238) = 118.00$. $p < 0.0001$). Figure 6.11 displays the results of a post-hoc Tukey HSD test. There was no significant difference between Whisper L and Whisper L+CW+S, Whisper L+CW, Whisper L+U, Whisper L+C and Whisper L+E. In all other cases Whisper L was significantly better (for exact confidence intervals and p-values see Table C.15). These results are expected given the negative impact on recall seen above.

Taken together, this section shows that using transformer embeddings as input into FFN is effective. However, improving on these results using previously developed methods in the field and those presented in this work have shown conflicting results depending on the metric considered. AUC suggests that smoothing can have significant positive effects when used alone or when coupled with class weighting, confidence based alteration or feature vector extension. However, at both a frame and an event level, all the methods had either no effect or a significantly negative effect on performance. Given the issues with AUC, perhaps a misleading metric as described at the end of Section 4.2.8, the results in terms of F1 should be given more weight.

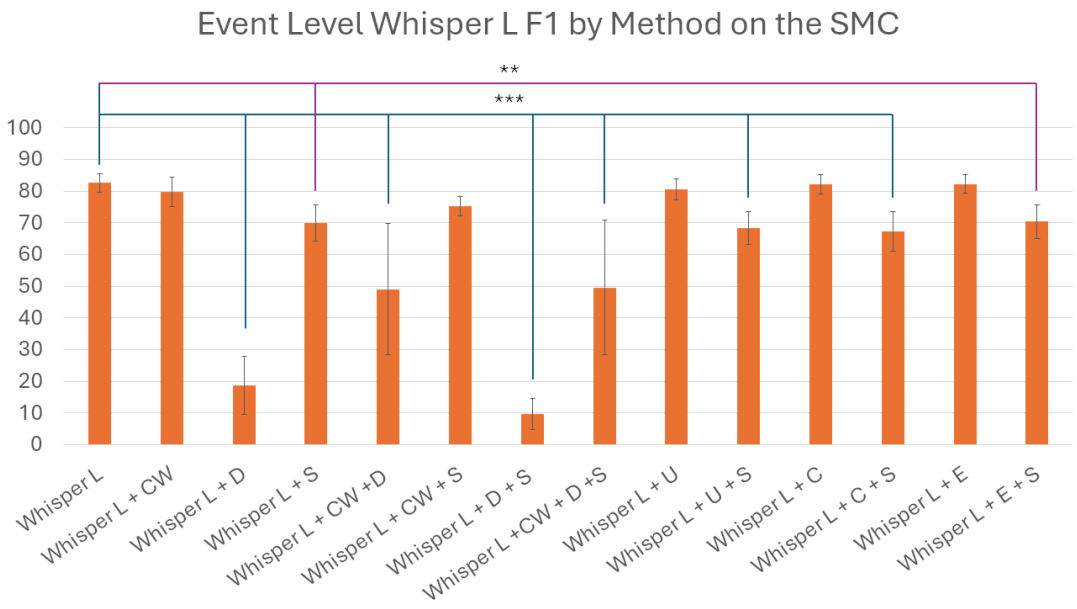

Figure 6.11: Event Level F1 for Each Different Detection Method on the SMC. Significant Differences Shown in Relation to Whisper L (**$p < 0.005$, ***$p < 0.0005$)

Table 6.6: Affect of Methodology Presented So Far on the Performance of the Whisper Large Transformer Embedding Extractions on the SMC at an Event Level. All models used the FFN base architecture. CW: class weight. D: delta. S: smoothing. U: undersampling. C: confidence-based alteration. E: feature vector extension. Bold highlights the best performing detector.

Detector	Precision	Recall	F1
Whisper L	**81.57 ± 6.72**	**84.27 ± 4.10**	**82.60 ± 3.04**
Whisper L+CW	72.40 ± 7.99	89.94 ± 3.28	79.83 ± 4.58
Whisper L+D	56.27 ± 26.59	11.31 ± 5.63	18.75 ± 9.16
Whisper L+S	92.56 ± 4.85	56.86 ± 8.45	69.98 ± 5.83
Whisper L+CW+D	52.43 ± 16.85	62.87 ± 27.81	49.13 ± 20.67
Whisper L+CW+S	87.71 ± 7.39	66.45 ± 5.84	75.18 ± 3.06
Whisper L+D+S	76.10 ± 23.84	5.22 ± 2.74	9.69 ± 4.87
Whisper L+CW+D+S	80.27 ± 11.37	39.49 ± 18.17	49.66 ± 21.21
Whisper L+U	83.42 ± 6.97	78.90 ± 6.15	80.69 ± 3.32
Whisper L+U+S	92.71 ± 4.92	54.82 ± 7.94	68.40 ± 5.19
Whisper L+C	83.97 ± 7.56	81.16 ± 4.36	82.17 ± 3.16
Whisper L+C+S	93.11 ± 4.34	53.48 ± 8.85	67.39 ± 6.19
Whisper L+E	80.56 ± 7.07	84.68 ± 4.11	82.23 ± 2.95
Whisper L+E+S	92.46 ± 4.92	57.44 ± 8.10	70.38 ± 5.23

Table 6.7: Frame Level Performance of Whisper L for Each Metric by Group Split

Group	Precision	Recall	F1	AUC
Caller	83.78 ± 18.90	45.96 ± 18.28	57.47 ± 18.03	93.37 ± 4.82
Receiver	78.85 ± 21.62	39.31 ± 17.91	50.51 ± 17.97	90.86 ± 5.72
Male	76.91 ± 26.51	40.98 ± 19.93	51.52 ± 21.49	92.18 ± 5.64
Female	85.30 ± 11.30	44.13 ± 16.76	56.22 ± 14.55	92.06 ± 5.26
MM	68.26 ± 26.06	44.12 ± 20.87	50.60 ± 19.66	93.97 ± 3.37
FF	80.83 ± 7.78	47.07 ± 15.48	57.57 ± 10.94	93.30 ± 4.51
MF	82.82 ± 10.65	45.39 ± 14.04	57.28 ± 13.09	92.82 ± 4.12

Table 6.8: Event Level Performance of Whisper L on Each Metric by Group Split

Group	Precision	Recall	F1
Caller	86.42 ± 19.56	87.21 ± 20.83	86.00 ± 19.40
Receiver	81.06 ± 22.67	83.86 ± 22.94	81.65 ± 22.07
Male	79.13 ± 28.07	80.24 ± 29.12	78.64 ± 27.88
Female	87.92 ± 10.85	90.32 ± 10.20	88.51 ± 9.04
MM	71.48 ± 26.33	80.93 ± 26.42	74.90 ± 24.99
FF	83.54 ± 9.37	90.52 ± 8.14	86.32 ± 6.08
MF	85.55 ± 10.95	88.93 ± 11.69	86.42 ± 9.85

Hence, it can be concluded that the additional methodology is ineffective when applied to Whisper L and, as such, Whisper L remains the best performing detection approach.

6.2.4 Performance Analysis

A performance analysis of the best detector, Whisper L, was carried out to better understand the effectiveness of the transformer embeddings. The initial testing compared the detector's performance by gender, role and conversation pairing. Frame level results from this analysis are displayed in Table 6.7, with event level results shown in Table 6.8.

First examining the effect of role, independent t-tests were used to compare Whisper L performance by role on each metric. Figure 6.12 displays the frame level results and Figure 6.13 shows the event level outcomes. Significant differences were found in frame level precision, recall, F1, AUC, event level precision and F1. No significant difference was found in event level recall (for exact t statistics and p-values see Table C.16). These findings indicate Whisper L introduced new significant differences by role compared with the previous best performing system: the CBA system. Although the size of the difference found in the Whisper L system is smaller than those seen in the CBA.

With regard to gender, independent t-tests were once again used to test for significant differences, with frame level results being shown in Figure 6.14 and event level outcomes displayed in Figure 6.15. The tests found significant differences in frame level precision, F1, event level precision, recall and F1. No significant difference was found for frame level recall or AUC (for

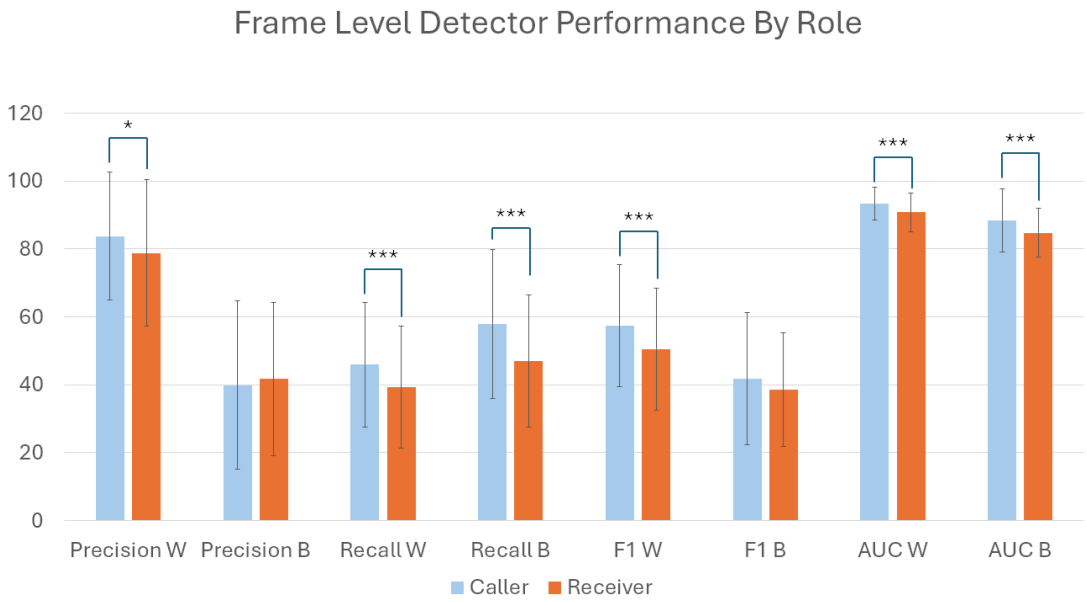

Figure 6.12: Frame Level Whisper L Performance by Role on the SMC. B: baseline LSTM+CW+C+S. W: Whisper L (*$p < 0.05$, ***$p < 0.0005$)

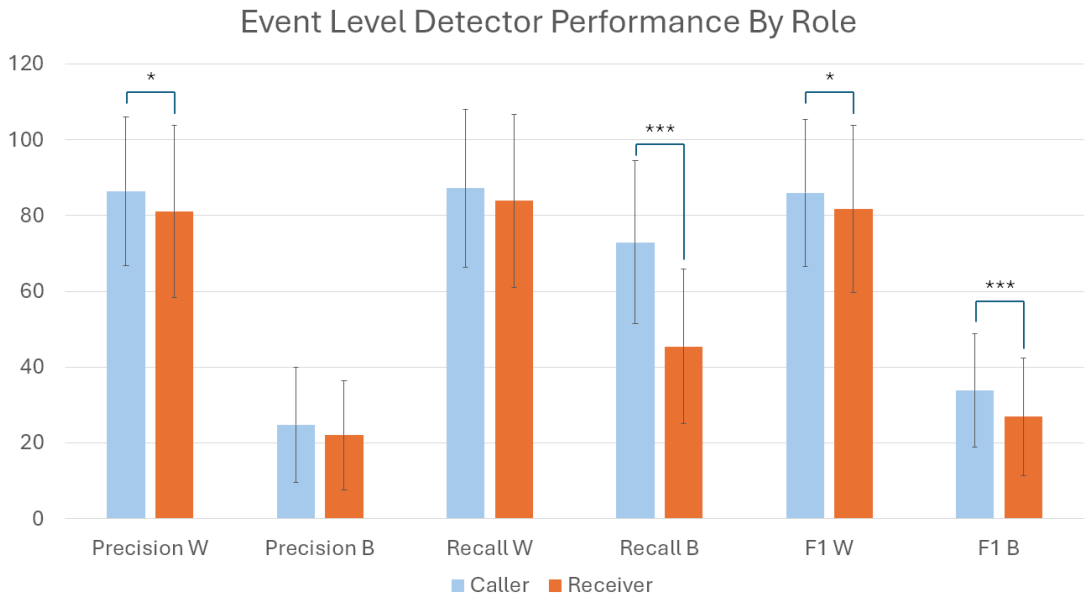

Figure 6.13: Event Level Whisper L Performance by Role on the SMC. B: Baseline LSTM+CW+C+S. W: Whisper L (*$p < 0.05$, ***$p < 0.0005$)

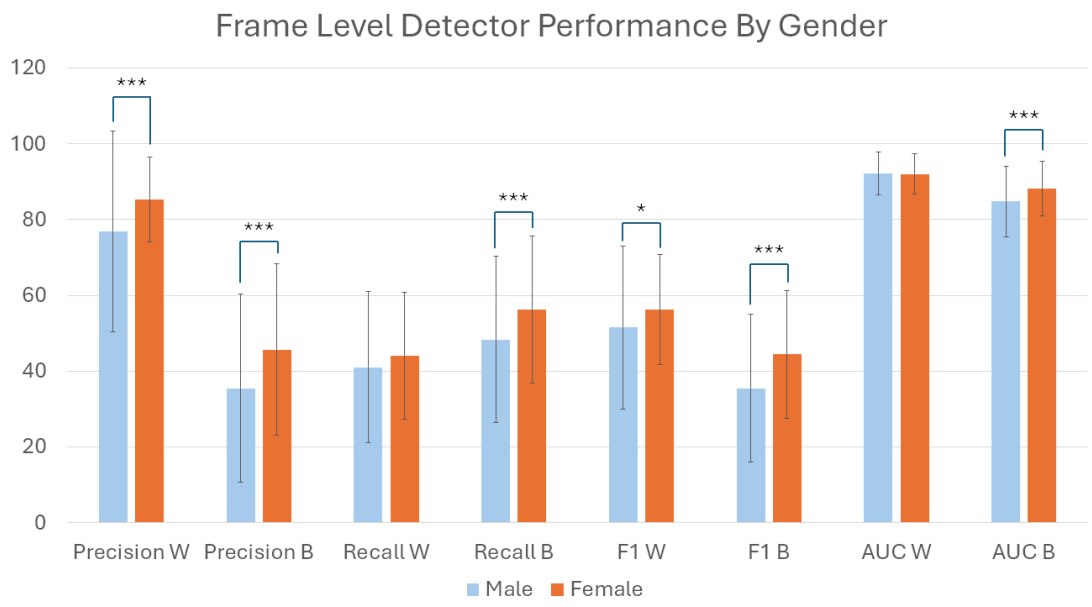

Figure 6.14: Frame Level Whisper L Performance by Gender on the SMC. B: baseline LSTM+CW+C+S. W: Whisper L (*$p < 0.05$, ***$p < 0.0005$)

exact t statistics and p-values see Table C.17). This means that, in 5 out of the 7 metrics, the Whisper L system saw significantly better performance for female than male speakers. This is a continuation of the trend seen in the baseline CBA system and in all the previous systems. However, in frame level recall and AUC, there was no significant difference found in terms of gender performance. In Chapter 4, the LSTM+CW+S system also saw no significant difference in terms of AUC, but the Whisper L system is the first time that a detector has seen no significant difference in frame level recall. Although these results seem promising, the significant differences in performance at an event level are seen in all the metrics, which suggests that the Whisper L detector still suffers from gender bias.

As has been the case in previous chapters, the gender differences described above led to significant differences in performance by conversational pairing. One-way ANOVA tests were used to compare the performance of each of the three gender pairings. Post-hoc Tukey HSD tests were utilised to identify which pairings experienced significant differences, with results shown in Figure 6.16 for frame level and Figure 6.17 for event level. In frame level precision, a one-way ANOVA test found significant differences in pairing performance (F(2, 177) = 13.68, $p < 0.0001$). Post-hoc Tukey HSD tests determined that MM pairings had significantly worse performance than FF and MF pairings. No significant difference was found between FF and MF pairings. For frame level recall, a one-way ANOVA test revealed no significant differences by pairing (F(2, 177) = 0.39, $p = 0.68$). For frame level F1, a one-way ANOVA test found significant differences by pairing (F(2, 177) = 3.60, $p = 0.029$). A post-hoc Tukey HSD test determined that MM pairings performed significantly worse than MF pairings. No significant differences were found between MM and FF pairings nor between FF and MF pairings. In AUC,

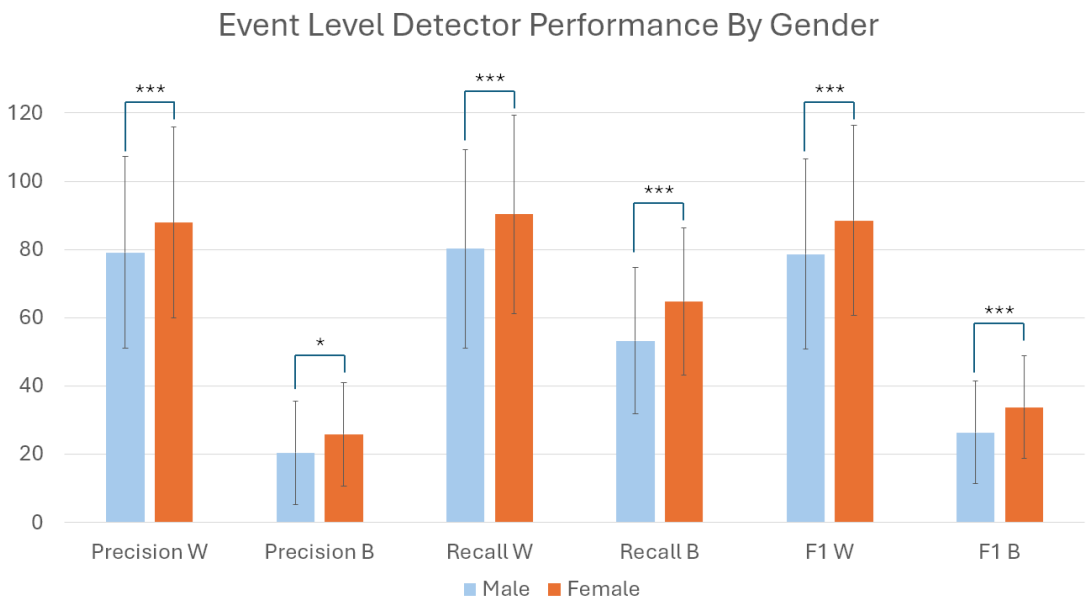

Figure 6.15: Event Level Whisper L Performance by Gender on the SMC. B: baseline LSTM+CW+C+S. W: Whisper L (*$p < 0.05$, ***$p < 0.0005$)

a one-way ANOVA test found no significant differences by pairing (F(2, 177) = 1.14, $p = 0.32$). For event level precision, a one-way ANOVA test determined significant differences by pairing (F(2, 177) = 12.00, $p < 0.0001$). Post-hoc Tukey tests found that MM pairings performed significantly worse than FF and MF pairings. No significant difference was found between FF and MF pairings. For event level recall, a one-way ANOVA test found significant differences by pairing (F(2, 177) = 4.98, $p = 0.0079$). MM pairings performed significantly worse than FF and MF pairings. For event level F1, a one-way ANOVA test determined significant differences by pairing (F(2, 177) = 10.50, $p < 0.0001$). Post-hoc Tukey HSD tests found that MM pairings performed significantly worse than FF and MF pairings. No significant difference was found between MF and FF pairings (for exact confidence intervals and p-values see Table C.18).

In addition to the above analysis of performance by group, an examination of the false positives created by the system were also investigated. The frame and event level precision of the Whisper L detection system are both high, at around ∼80%. However, there remains multiple false positive detections. In previous chapters, it was shown that an above random chance number of the event level false positives occurred within 0.46 s of a laughter event, with the LSTM+CW+S detector seeing 23.99 ± 5.55% and the CBA system seeing 29.49 ± 8.95%. With regard to Whisper L, Table 6.9 shows the percentage of false positives within various times of a laughter event. At the 0.46 s cut-off, there was around 72.40 ± 10.06% of event level false positives. This represents an increase of ∼40% in comparison to the CBA system and results in an event level F1 of almost 90%. This suggests that, even when the Whisper L system makes mistakes, in the majority of cases these mistakes are almost correct. It is possible, therefore, that the merging of peaks may be causing the system to move peaks away from correctly identified

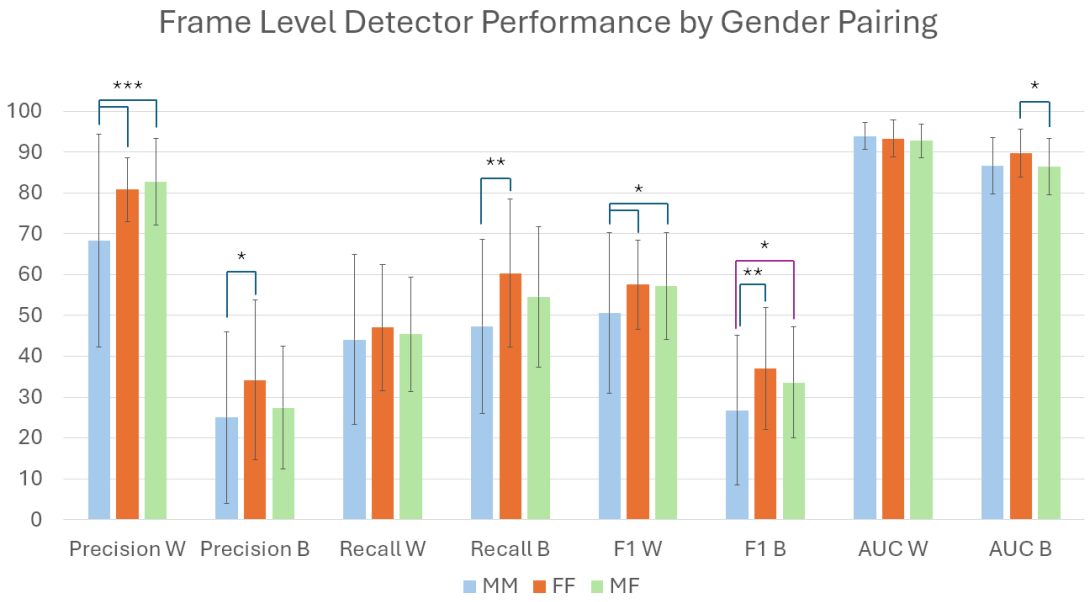

Figure 6.16: Frame Level Whisper L Performance by Gender Pairing on the SMC. B: baseline LSTM+CW+C+S. W: Whisper L (*$p < 0.05$, **$p < 0.005$, ***$p < 0.0005$)

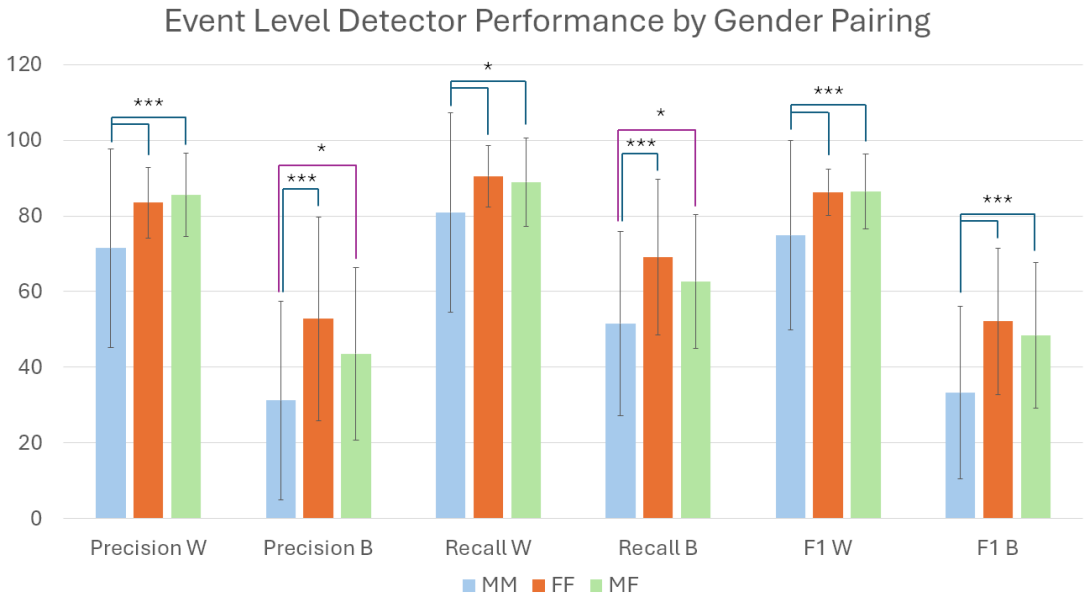

Figure 6.17: Event Level Whisper L Performance by Gender Pairing on the SMC. B: baseline LSTM+CW+C+S. W: Whisper L (*$p < 0.05$, ***$p < 0.0005$)

events.

Table 6.9: Percentage of False Positives Within a Given Length of Time from Laughter and the Associated Precision, Recall and F1 if They Were Reclassified as True Positives

Time (s)	Percentage of False Positives	Precision	Recall	F1
Original	-	81.57 ± 6.72	84.27 ± 4.10	82.60 ± 3.04
0.1	31.39 ± 12.11	85.99 ± 5.70	85.54 ± 4.24	85.52 ± 2.30
0.46	72.40 ± 10.06	91.09 ± 5.37	87.68 ± 4.03	89.13 ± 1.91
0.5	73.44 ± 10.76	91.13 ± 5.38	87.81 ± 4.02	89.22 ± 1.91
1	81.98 ± 9.86	92.56 ± 5.06	89.38 ± 3.84	90.74 ± 1.80
2	91.00 ± 6.04	93.45 ± 4.43	91.29 ± 3.37	92.22 ± 1.73

It was theorised that, if multiple peaks occurred close together and that the current system of merging is causing the final merged peak to be shifted away from the actual event, then a different approach to merging may reduce the issue. It was hypothesised that, in the case where there are multiple peaks close to each other with some occurring in conjunction with laughter and others occurring close to an event, then those peaks that occur over the event would have a higher posterior, as estimated by the Whisper Large detector. As such, a new merging process was developed and tested. Where multiple peaks were close enough together to trigger the merging process, rather than picking a point equidistant between the peaks (as had been done before), the position of the peak with the highest associated posterior probability of belonging to laughter was instead selected as the merged position. Table 6.10 shows the event level performance of the two different merging methods for the Whisper L system. A t-test found no significant difference between the two merging methods, suggesting that the near misses are not caused by the merging method moving peaks away from actual events, but that the detector is missing these events.

Table 6.10: Whisper L Event Level Precision, Recall and F1 by Method of Peak Merging

	Precision	Recall	F1
Equidistant	81.57 ± 6.72	84.27 ± 4.10	82.60 ± 3.04
Maximum	81.93 ± 6.67	84.20 ± 4.10	82.75 ± 2.98

A further issue with the Whisper L system is that, at a frame level, its recall is relatively poor with a value at under 50%. The high event level recall and precision of $\sim$80% suggest that the Whisper L detector is effectively detecting most events and making few mistakes, suggesting that, at a frame level, the system is missing frames within events that are detected. In previous chapters, this issue was addressed by using a Hamming window to convolve the posteriors initially output by the system. However, when using the Hamming window with Whisper L, no significant difference in recall was found, although there were significant differences in AUC and event level recall and F1 (as presented above). As explained in Section 6.2.1, the distribution of the posteriors produced by Whisper L is different from that of the previous detectors.

Whisper L posteriors are characterised by sharp changes from the ∼0-10 to 90-100% range. Here, the Hamming convolution is less effective since multiple values are zero or close to zero, leading to the posteriors being suppressed beyond the goal. A different filter option is to use a median filter.

Median filters follow a similar approach to the Hamming window procedure that was used previously. A sliding window is applied to the sequence of posteriors, all posteriors in the window are ordered by magnitude and the value in the middle position is adopted as the new posterior value [105]. Median filters have been used in denoising [106] and image processing [107]. A key characteristic of the median filter is that it can preserve edges in a sequence, periods where the sequence switches from one average to another, while also removing noise [108]. In the case of Whisper L, it is theorised that a median filter maintains the switch between no laughter and detected laughter, while removing the noise in the detected laughs by maintaining the detection from the onset to offset. Further to the median filter, a variant was tested termed the MinMax filter. In this case, a sliding window was once again applied to the output posteriors. For each window, a detection decision was made in which, if a majority of posteriors were over 0.5, the maximum posterior would be output for the window. In the case where the majority of posteriors were below 0.5, the output would be the minimum of the window. The size of the window for both novel features was treated as a hyper-parameter and was tuned in the same manner as the Hamming window size, as described in Section 4.2. Table 6.11 shows the frame level results achieved from each filter when it was applied to the Whisper L system; Table 6.12 displays the event level results. A one-way ANOVA test was used to compare the unaltered Whisper L performance on each metric with the performance given by each filter method.

Table 6.11: Whisper L Frame Level Precision, Recall and F1 by Method of Filtering/Smoothing. Bold highlights the best performing detector.

	Window Size	Precision	Recall	F1
Original	0	83.35 ± 4.67	46.19 ± 7.47	58.90 ± 4.82
Hamming	10	91.66 ± 2.68	41.84 ± 8.38	56.90 ± 7.03
Median Filter	10	90.08 ± 3.61	45.36 ± 8.41	59.75 ± 6.23
Maximum Filter	**10**	**65.08 ± 7.30**	**73.36 ± 5.46**	**68.44 ± 3.32**

Table 6.12: Whisper L Event Level Precision, Recall and F1 by Method of Filtering/Smoothing. Bold highlights the best performing detector.

	Window Size	Precision	Recall	F1
Original	**0**	**81.57 ± 6.72**	**84.27 ± 4.10**	**82.60 ± 3.04**
Hamming	10	92.56 ± 4.85	56.86 ± 8.45	69.98 ± 5.83
Median Filter	10	91.50 ± 4.17	71.35 ± 6.29	79.92 ± 3.38
Maximum Filter	10	78.08 ± 7.66	78.43 ± 4.74	77.86 ± 3.57

In terms of improving frame level precision, a one-way ANOVA test found significant differences between filter methods ($F(3, 68) = 112.19$, $p < 0.0001$). A post-hoc Tukey HSD test

determined that, compared with the Hamming window approach, there was significantly better performance by the MinMax approach. However, there was no significant difference from Hamming to median. For frame level recall, a one-way ANOVA test found significant differences by approach (F(3, 68) = 67.47, $p < 0.0001$). A post-hoc Tukey test determined that the Hamming approach was significantly improved upon by the MinMax approach with no significant difference found between the Hamming and median approaches. In frame level F1, a one-way ANOVA test found significant differences between approaches (F(3, 68) = 15.31, $p < 0.0001$). Post-hoc Tukey HSD tests determined that the Hamming approach was significantly outperformed by the MinMax approach. No significant difference was found between Hamming and median approaches. For event level precision, a one-way ANOVA test found significant differences between approaches (F(3, 68) = 25.80, $p < 0.0001$). A post-hoc Tukey HSD test determined no significant difference between the Hamming approach and the median approach. However, the Hamming approach was significantly better than the MinMax approach. In event level recall, a one-way ANOVA test found significant differences between approaches (F(3, 68) = 67.00, $p < 0.0001$). Post-hoc Tukey HSD tests determined that no filtering was significantly better than the median approach and the MinMax approach. For event level F1, a one-way ANOVA test found significant differences by approach (F(3, 68) = 31.52, $p < 0.0001$). Post-hoc Tukey HSD tests determined that no filtering was significantly better than MinMax. However, no significant difference was found between no filtering and the median approach (for exact p-values and confidence intervals see Table C.19). These results suggest that the MinMax filter was partially successful at a frame level, with the balancing of precision and recall, for the Whisper L system. However, at an event level, it only negatively affected performance. These results mean that the usefulness of the filter rests upon the end goal or tasks that the system intends to fulfil. If frame level F1 is important, the MinMax filter can lead to significant gains of performance, i.e., $\sim$10%. However, if event level metrics are the focus then the original unaltered system is best.

6.2.5 Results: Whisper Large Performance on SVC

The previous sections have demonstrated that the transformer embeddings are effective input feature vectors for training feed forward neural networks for laughter detection. Furthermore, it was shown that, at both an event and a frame level, the methods used to improve results in the previous chapters are ineffective when coupled with the transformer embeddings. In this section, the transformer embeddings are tested on the SVC. It is expected that the results will remain stable or increase, as the work presented in this section represents moving from a type 3 to a type 2 task. However, the results enable comparison with the wider field and, as a consequent, they are included here.

The methodology applied to the SVC is the same as explained in Chapter 4, with the feature extraction step replaced with the embedding extraction of Whisper Large. For complete-

Table 6.13: Frame Level Performance on the SVC for Whisper L Using Both Merging and Exclusion Criteria. CW: class weight. D: delta. S: smoothing

	Evaluation Criteria	Precision	Recall	F1	AUC
Whisper L	Merged	83.42 ± 2.99	71.93 ± 4.33	77.18 ± 3.01	96.62 ± 1.78
Whisper L	Excluded	82.45 ± 25.11	70.33 ± 28.38	72.47 ± 25.44	94.75 ± 10.87
FFN+CW+D+S	Merged	22.65 ± 9.08	46.09 ± 13.22	28.43 ± 7.68	81.87 ± 5.57
FFN+CW+D+S	Excluded	41.76 ± 13.12	46.06 ± 13.20	41.43 ± 8.58	81.01 ± 5.54
[26]	x	80.9-87.4	87.3-94.5	84.0-90.8	x
[38]	x	x	x	x	97.3

Table 6.14: Event Level Performance on the SVC for Whisper L Using Both Merging and Exclusion Criteria. CW: class weight. D: delta. S: smoothing

	Evaluation Criteria	Precision	Recall	F1
Whisper L	Merged	82.20 ± 0.34	95.17 ± 1.02	88.21 ± 0.55
Whisper L	Excluded	78.81 ± 30.64	92.21 ± 25.74	83.19 ± 27.56
LSTM+CW+S	Merged	29.51 ± 12.91	68.45 ± 7.37	39.48 ± 13.69
LSTM+CW+S	Excluded	58.60 ± 2.42	47.28 ± 0.79	52.31 ± 1.02

ness, both the exclusion and merging results are included here (see Section 4.1 for full details). Table 6.13 displays the frame level performance by the Whisper L model alongside the best performing detectors from Chapter 4 and the field at large. Table 6.14 displays the event level results.

Initially, the effect of merging or exclusion was tested for each metric using t-tests, which found no significant difference on any metric. As the exclusion criteria was assumed to be used in the field, these results were selected for further analysis. Independent t-tests were used to compare the performance of the best performing detectors from Chapter 4, i.e., FFN+CW+D+S for frame level and LSTM+CW+S for event level, on the SVC with Whisper L. A comparison was carried out over the results using the exclusion evaluation criteria, as explained in Chapter 4. In all cases Whisper L was significantly better than baseline (for exact t statistics and p-values see Table C.20). These results show that, for all the metrics, Whisper L significantly outperformed all the other detectors. This is unsurprising given that Whisper L performed better than all these detectors on the SMC, which is a more difficult task. However, when examining the Whisper L system's performance against the best reported performances in the field, it appears that its performance is worse. Statistical testing was not possible given that standard deviations and fold numbers have not been published for the best performing models detailed in Tables 6.13, as such it is impossible to state whether the difference in performance is significant or not. Especially given that, in multiple folds, the Whisper L performance exceeded that of the best in the field.

Table 6.15: Frame Level Performance on Filler and Back-Channel Detection by Whisper L and Baseline Detectors. CW: class weight. S: smoothing

Detector	Precision	Recall	F1	AUC
BC Whisper L	54.75 ± 8.83	15.36 ± 4.04	23.77 ± 5.26	84.68 ± 2.66
BC FFN+CW+S	1.98 ± 0.20	76.16 ± 6.92	3.85 ± 0.38	75.58 ± 2.29
Filler Whisper L	83.24 ± 2.34	63.53 ± 4.54	71.90 ± 2.66	93.97 ± 0.72
Filler FFN+CW+S	9.65 ± 2.45	40.37 ± 10.59	15.16 ± 2.95	71.04 ± 1.64

Table 6.16: Event Level Performance on Filler and Back-Channel Detection by Whisper L and Baseline Detectors. CW: class weight. S: smoothing

Detector	Precision	Recall	F1
BC Whisper L	46.87 ± 8.75	47.47 ± 6.56	46.80 ± 6.79
BC FFN+CW+S	3.12 ± 0.79	34.35 ± 6.92	5.71 ± 1.37
Filler Whisper L	80.66 ± 2.40	83.38 ± 2.75	81.95 ± 1.77
Filler FFN+CW+S	8.03 ± 2.17	28.30 ± 5.78	12.44 ± 3.06

6.3 Transformer Embedding for Filler and Back-Channel Detection

The previous sections demonstrate the effectiveness of the transformer embeddings for automatic laughter detection in both type 2 and 3 tasks. In this section, the same process is now applied to filler and back-channel events. The only adjustment to the approach compared with previous sections was the target class. Tables 6.15 (frame level) and 6.16 (event level) display the results achieved by Whisper L alongside the filler and back-channel individual detectors created in Section 5.2.2, which act here as a baseline.

First examining the performance on filler detection, independent t-tests were used to compare the performance of the Whisper Large detector against the baseline. The results of these tests are shown in Figure 6.18. These results show that, for every metric, the Whisper L detector performed significantly better than the baseline (for exact t statistics and p-values see Table C.21). These results again show advantages when using the attention embeddings as input features, as the performance of filler detection is on par with laughter detection.

The performance regarding back-channel was also examined using independent t-tests. Figure 6.19 displays the results of this significance testing. Independent t-tests found Whisper L was significantly better in all metrics except frame level recall were it was significantly worse than baseline (for exact t statistics and p-values see Table C.22). These results suggest there were some advantages when using the Whisper L embeddings. The significant increase in precision at both a frame and an event level is promising. Furthermore, the event level recall results suggest that around half of all the back-channels are being detected, although the detectors are struggling to identify boundaries (as shown by the low frame level recall). The difference in performance from laughter and filler events to back-channel is probably due to the nature of

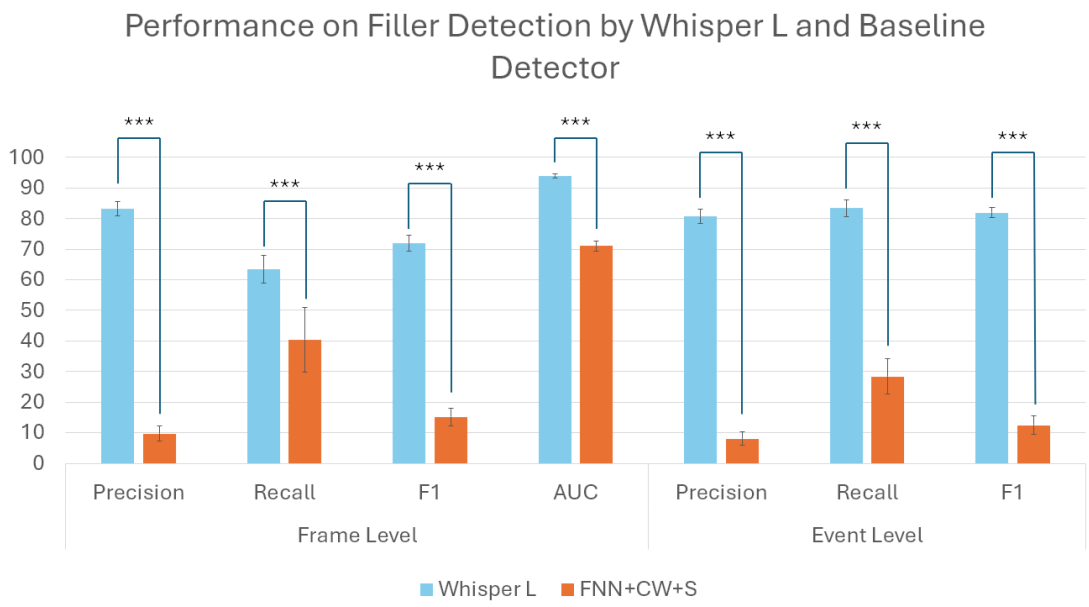

Figure 6.18: Performance of Whisper L and Baseline Detectors at Filler Detection in the SMC (***$p < 0.0005$)

Table 6.17: Frame Level Performance on Back-Channel Detection by Whisper L with Pre/Post-Processing. CW: class weight. S: smoothing. D: delta. Bold highlights the best performing detector.

Model Type	Precision	Recall	F1	AUC
Whisper L	54.75 ± 8.83	15.36 ± 4.04	23.77 ± 5.26	84.68 ± 2.66
Whisper L+CW	27.89 ± 4.08	35.80 ± 6.11	31.01 ± 3.87	85.45 ± 2.29
Whisper L+D	55.44 ± 7.84	20.57 ± 4.93	29.76 ± 5.89	87.29 ± 2.66
Whisper L+S	1.98 ± 0.20	76.16 ± 6.92	3.85 ± 0.38	75.58 ± 2.29
Whisper L+CW+D	22.60 ± 2.73	56.48 ± 5.06	32.20 ± 3.29	90.45 ± 1.65
Whisper L+CW+S	45.71 ± 5.40	31.62 ± 6.40	36.99 ± 5.51	89.80 ± 2.38
Whisper L+D+S	68.50 ± 11.31	16.47 ± 4.78	26.20 ± 6.44	92.17 ± 1.91
Whisper L+CW+D+S	**35.62 ± 3.91**	**57.74 ± 5.64**	**43.88 ± 3.74**	**93.99 ± 1.41**

these events. Back-channel, as outlined in Chapter 5.2, are short utterances that a listener makes to signal they are still engaged and listening to what the other speaker is currently saying. They always overlap with another speaker's speech. Given the overlapping nature of back-channel and speech, the methods developed in Chapter 5 are not applicable to back-channel detection. However, it is possible that the pre/post-processing methods explored in Chapter 4 would be effective, so they were tested.

Tables 6.17 (frame level) and 6.18 (event level) show the results achieved by the various pre/post-processes. For each metric, one-way ANOVA tests were used to compare the detection methods results against the baseline Whisper L performance. In cases where significant differences were found, post-hoc Tukey tests were carried out to identify which methods were significantly different from others.

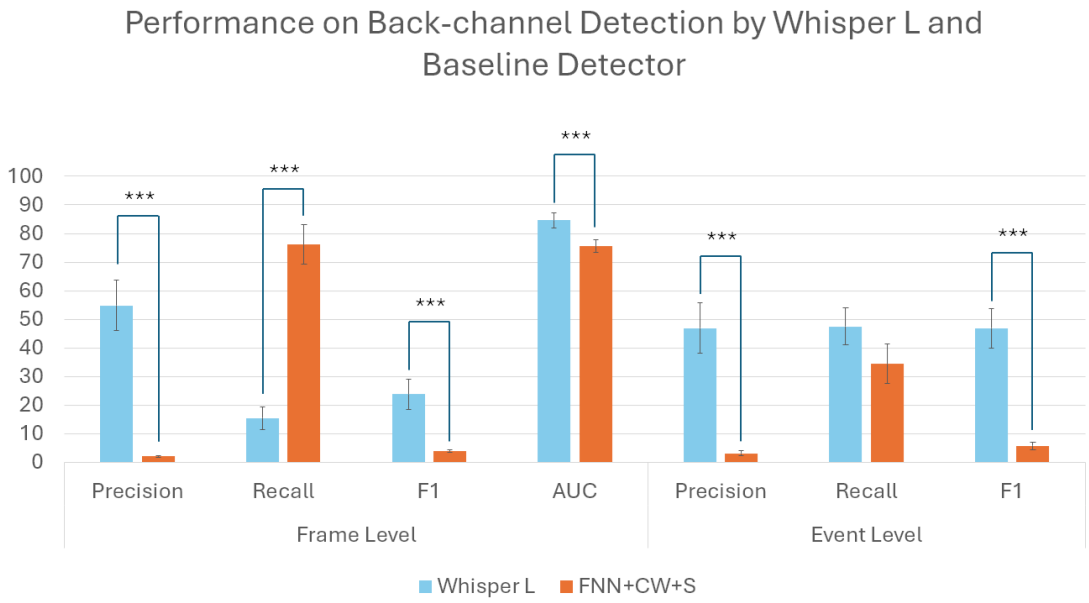

Figure 6.19: Performance of Whisper L and Baseline Detectors at Back-Channel Detection in the SMC (***$p < 0.0005$)

Table 6.18: Event Level Performance on Back-Channel Detection by Whisper L with Pre/Post-Processing. CW: class weight. S: smoothing. D: delta. Bold highlights the best performing detector.

Model Type	Precision	Recall	F1
Whisper L	46.87 ± 8.75	47.47 ± 6.56	46.80 ± 6.79
Whisper L+CW	17.86 ± 3.50	71.42 ± 9.64	28.39 ± 4.77
Whisper L+D	**46.59 ± 7.11**	**56.33 ± 6.61**	**50.77 ± 6.32**
Whisper L+S	3.12 ± 0.79	34.35 ± 6.92	5.71 ± 1.37
Whisper L+CW+D	12.25 ± 2.90	74.25 ± 8.27	20.94 ± 4.40
Whisper L+CW+S	31.82 ± 5.55	60.34 ± 8.31	41.31 ± 5.70
Whisper L+D+S	63.69 ± 8.89	37.43 ± 4.59	46.92 ± 5.21
Whisper L+CW+D+S	24.82 ± 4.50	73.09 ± 6.24	36.93 ± 5.61

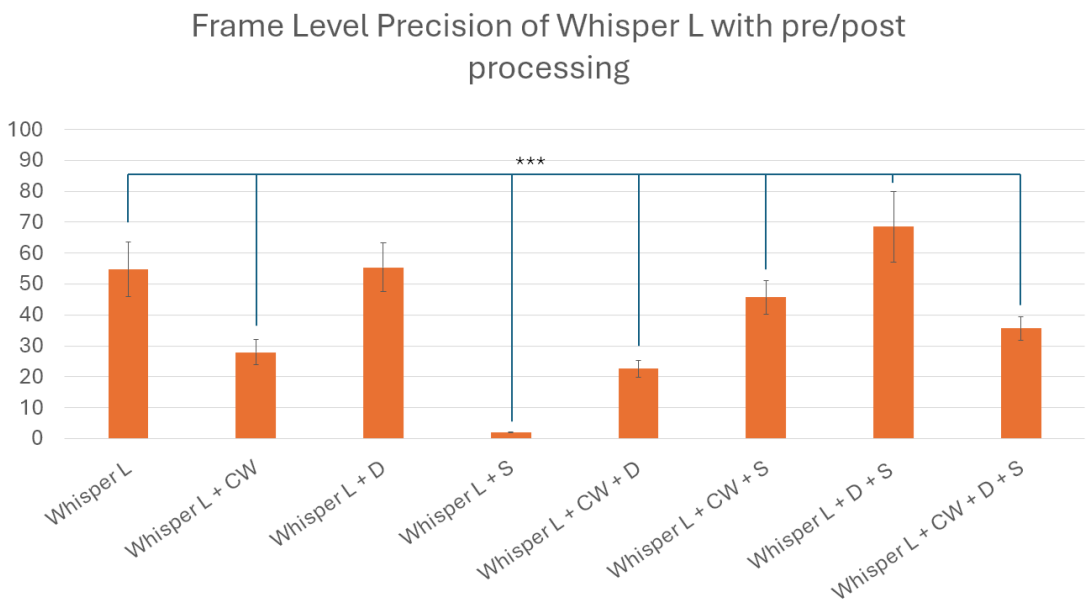

Figure 6.20: Frame Level Precision on Back-Channel Detection by Method and Metric. Significant Differences Shown in Relation to Whisper L (***$p < 0.0005$)

In terms of frame level precision, a one-way ANOVA test found significant differences between detectors (F(7, 136) = 196.44, $p < 0.0001$). A post-hoc Tukey HSD test, shown in Figure 6.20, determined that Whisper L was significantly worse than Whisper L+D+S. One detector was not significantly different (Whiser L+D). All others were significantly worse (for exact confidence intervals and p values see Table C.23).

In frame level recall, a one-way ANOVA test found significant differences between detectors (F(7, 136) = 292.09, $p < 0.0001$). A post-hoc Tukey HSD test, shown in Figure 6.21, found no significant difference was found between Whisper L and Whisper L+D or Whisper L+D+S. In all other cases the detectors with post processing out-performed Whisper L (for exact confidence intervals and p values see Table C.24).

In frame level F1, a one-way ANOVA test found significant differences between detectors (F(7, 136) = 114.06, $p < 0.0001$). A post-hoc Tukey HSD test, shown in Figure 6.22, determined that Whisper L was significantly better than Whisper L+S. No significant difference was found between Whisper L and Whisper L+D+S. In all other cases Whisper L was significantly worse (for exact confidence intervals and p values see Table C.25)

In AUC, a one-way ANOVA test found significant differences between detectors (F(7, 136) = 123.53, $p < 0.0001$). A post-hoc Tukey HSD test, shown in Figure 6.23, determined that Whisper L was significantly better than Whisper L+S. No significant difference was found between Whisper L and Whisper L+CW. In all other cases Whisper L was significantly worse (for exact confidence intervals and p values see Table C.26). These results show that multiple of the pre/post-processing methods presented in Chapter 4 are effective at improving back-channel detection using transformer embeddings as input. However, overall F1 remains below the 50%

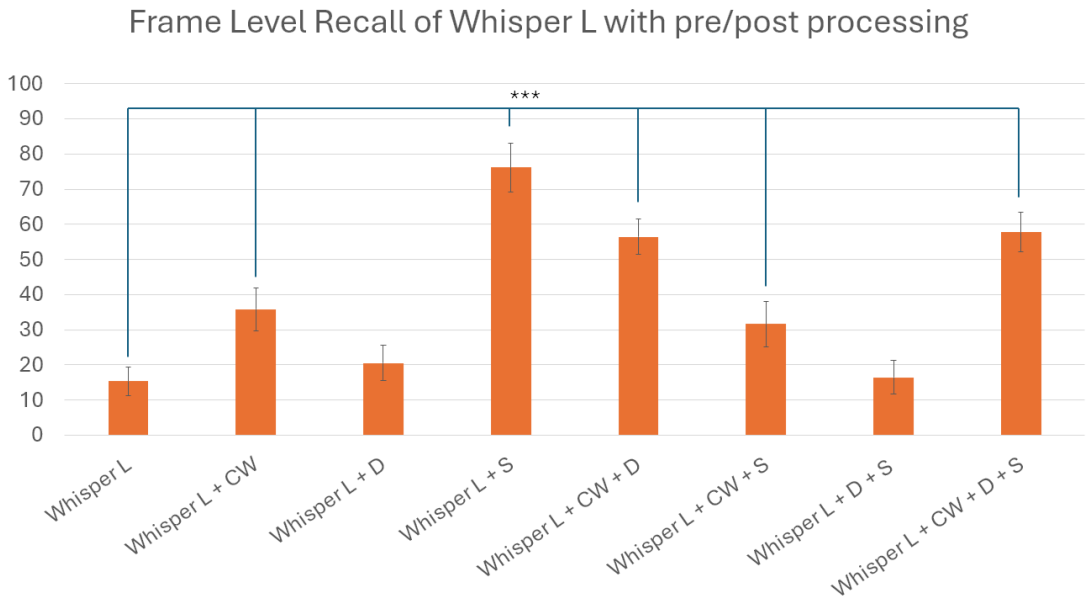

Figure 6.21: Frame Level Recall on Back-Channel Detection by Method and Metric. Significant Differences Shown in Relation to Whisper L (***$p < 0.0005$)

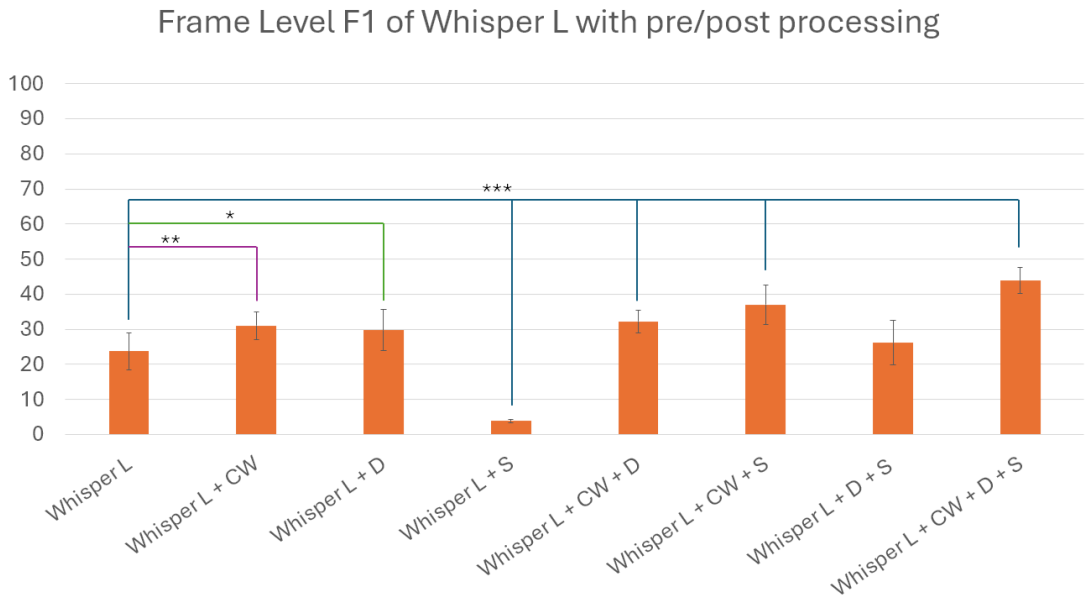

Figure 6.22: Frame Level F1 on Back-Channel Detection by Method and Metric. Significant Differences Shown in Relation to Whisper L (*$p < 0.05$, **$p < 0.005$, ***$p < 0.0005$)

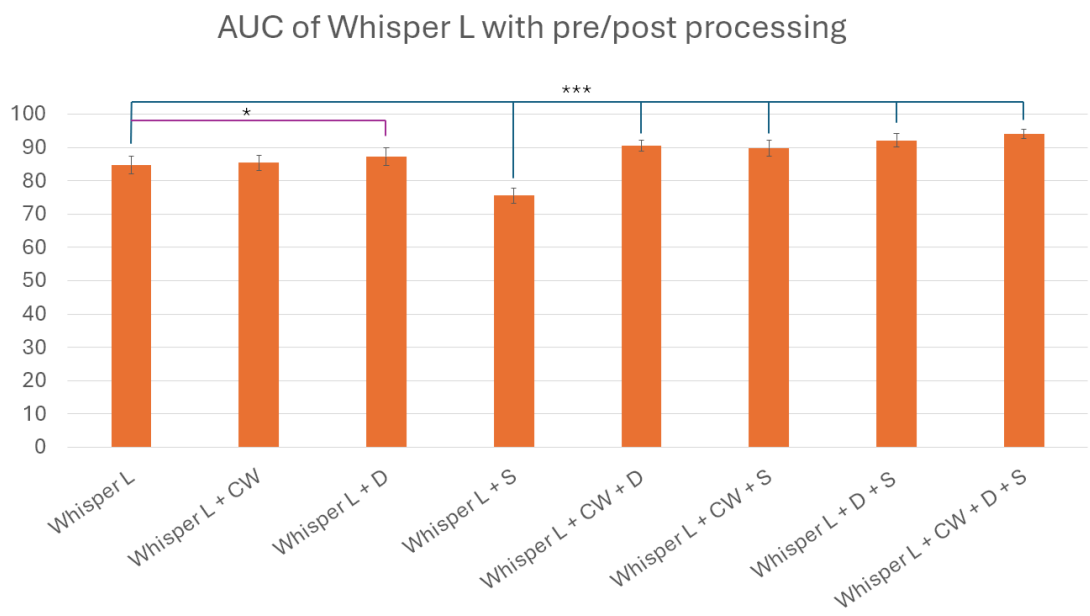

Figure 6.23: AUC on Back-Channel Detection by Method and Metric. Significant Differences Shown in Relation to Whisper L (*$p < 0.05$, ***$p < 0.0005$)

mark.

For event level precision, a one-way ANOVA test found significant differences (F(7, 136) = 214.50, $p < 0.0001$). A post-hoc Tukey HSD test, shown in Figure 6.24, determined that Whisper L was significantly worse than Whisper L+D+S. No significant difference was found between Whisper L and Whisper L+D. In all other cases Whisper L was significantly better (for exact confidence intervals and p values see Table C.27).

In event level recall, a one-way ANOVA test found significant differences (F(7, 136) = 85.32, $p < 0.0001$). A post-hoc Tukey HSD test, shown in Figure 6.25, determined that Whisper L was significantly better than Whisper L+S and Whisper L+D+S. Whisper L was significantly worse than all others (for exact confidence intervals and p values see Table C.28).

For event level F1, a one-way ANOVA test found significant differences (F(7, 136) = 155.56, $p < 0.0001$). A post-hoc Tukey HSD test, shown in Figure 6.26, determined that Whisper L was not significantly different from Whisper L+D and Whisper L+D+S. In all other cases Whisper L was better (for exact confidence intervals and p values see Table C.29). These event level results suggest that the pre/post-processing methods presented in Chapter 4 are ineffective at improving back-channel detection.

Transformer-based embeddings were shown to enable significantly better detection of laughter and fillers but failed to reliably detect back-channel. The above results show that some of the methods developed in the field of laughter detection can be transferred to the back-channel detection problem and led to significant increases in performance at a frame level but not at an event level. Moreover, the best detectors here only achieve an F1 of $\sim$50%, leaving much room for improvement. This leaves back-channel detection an open question, especially com-

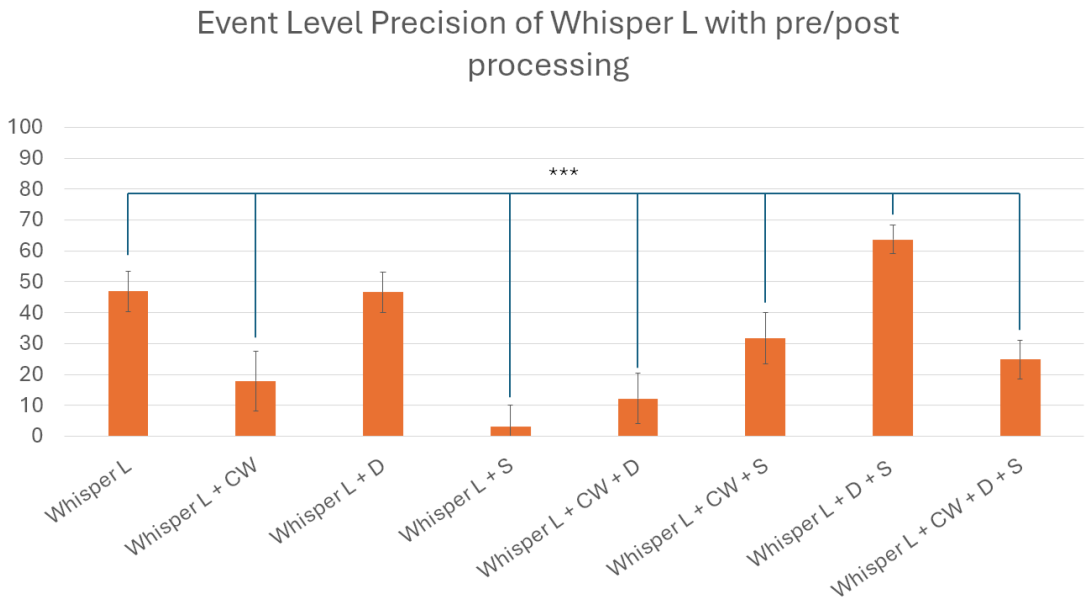

Figure 6.24: Event Level Precision on Back-Channel Detection by Method and Metric. Significant Differences Shown in Relation to Whisper L (***$p < 0.0005$)

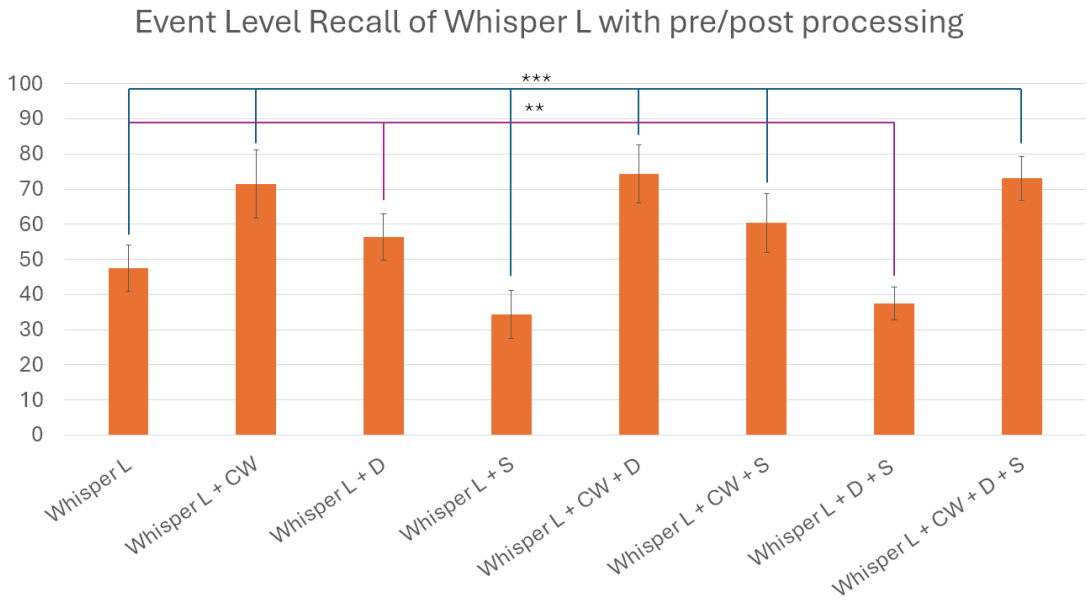

Figure 6.25: Event Level Recall on Back-Channel Detection by Method and Metric. Significant Differences Shown in Relation to Whisper L (**$p < 0.005$, ***$p < 0.0005$)

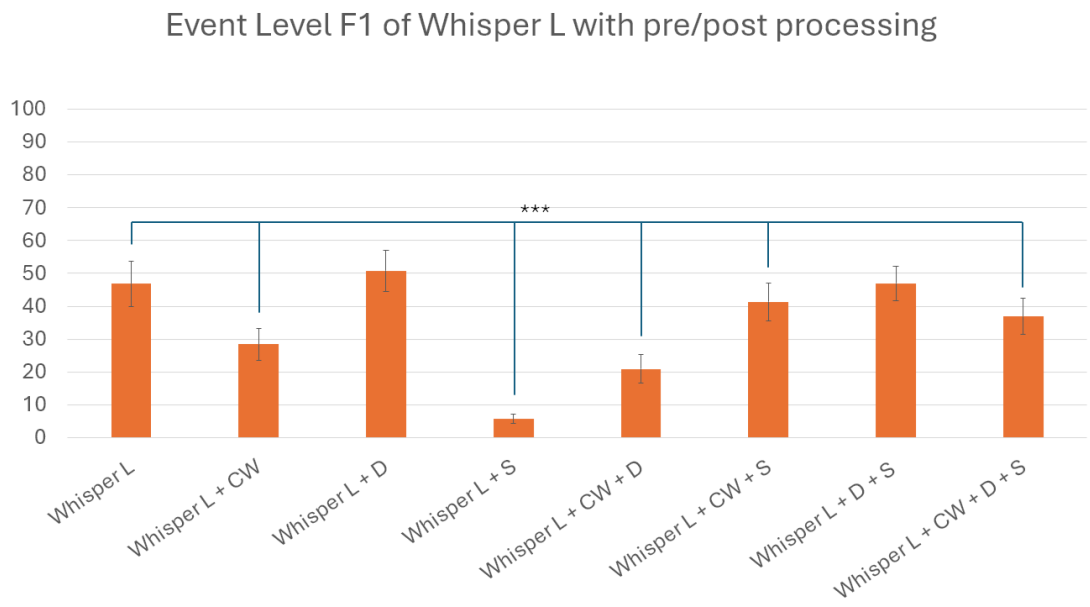

Figure 6.26: Event Level F1 on Back-Channel Detection by Method and Metric. Significant Differences Shown in Relation to Whisper L (***$p < 0.0005$)

pared with laughter and fillers, and shows that, although effective for some tasks, transformer embeddings are not a silver bullet that can solve all detection challenges.

6.4 Automatic Detection of Common Laughter Traits

In this final section, the now effective laughter detectors are applied to the SMC in an attempt to automatically extract speaker information. Although the most effective detectors discovered so far use the Whisper Large attention embeddings as the input, this analysis was extended and carried out on the HuBERT S detector and the CBA detectors from Chapter 5 as well. This was done to test how effective a detector is required to be before it can reliably automatically detect speaker information.

There are well-documented gender effects on laughter behaviour. Namely, that female speakers tend to laugh more than male speakers [109]. This effect is found in the SMC, for the exact numbers see Chapter 3. In this section of the work, three detection methods are tested for their ability to reliably detect this tendency to laugh differently by gender pairing at a conversation level in both laughter frequency and total amount.

The three systems used are as follows: Whisper L, HuBERT Base and CBA. For each conversation in the SMC, the detector that had not seen that conversation during training was used to automatically estimate the number of laughter events present using the laughter event schema described in Section 4.2.4. Furthermore, the same event detection steps were applied to the ground truth labels to find the actual number of laughter events per conversation. This gave the total laughter events per conversation. As the conversations were of variable length, the fre-

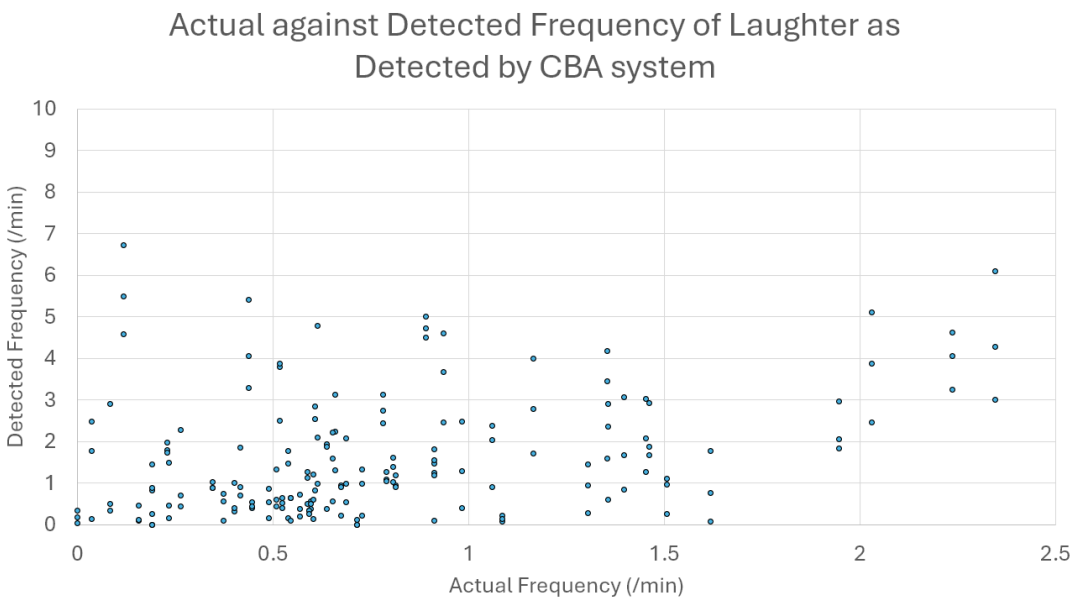

Figure 6.27: Correlation of Actual and Detected Frequency of Laughter Per Minute of Conversation Achieved by CBA System

quency of events (both actual and detected) was also calculated in terms of laughs per minute of conversation.

Initially, the correlation between actual and detected total laughs and laughter frequency was tested. In terms of total laughter, both HuBERT S (r(2) = 0.76 $\pm$ 0.020, $p < 0.0001$) and Whisper L (r(2) = 0.86 $\pm$ 0.014, $p < 0.0001$) detectors achieved a statistically significant correlation. However, the CBA system's correlation was non-significant after a bonferroni adjustment (r(2) = 0.36 $\pm$ 0.13, $p = 0.043$). Both HuBERT and Whisper achieved strong correlations of around 0.8. With regard to frequency, all three models achieved significant correlations (CBA: r(2) = 0.44 $\pm$ 0.095, $p = 0.0023$, HuBERT S: r(2) = 0.81 $\pm$ 0.017, $p < 0.0001$, Whisper L: r(2) = 0.83 $\pm$ 0.0062, $p < 0.0001$). The correlation achieved by the CBA system was low, whereas both the transformer-based systems achieved strong correlations. Figures 6.27, 6.28 and 6.29 show the laughter frequency correlation achieved by each detection system.

It was shown above that there were strong and significant correlations between detected and actual laughter. Therefore, detected laughter events should be sufficiently reliable to reflect differences associated with social and psychological phenomena. For each detection system, two-way ANOVA tests were used to test the effect of gender pairing (MM, FF and MF) and method (actual or detected) and their interaction. Both total laughter and laughter frequency were tested.

With regard to the CBA system and total laughter per conversation, a two-way ANOVA test found no significant effect on gender pairing (F(2, 114) = 0.27, $p = 0.77$) and no significant interaction between gender pairing and method (F(2, 114) = 0.10, $p = 0.90$). However, there was a significant difference by method (F(1, 114) = 14.26, $p < 0.0001$), suggesting that the CBA

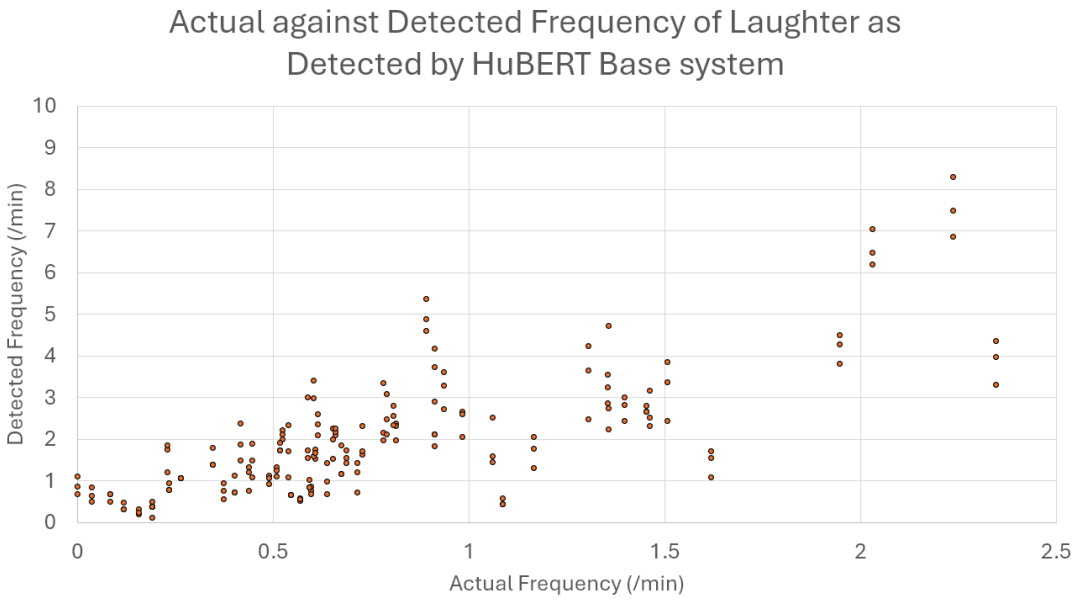

Figure 6.28: Correlation of Actual and Detected Frequency of Laughter Per Minute of Conversation Achieved by HuBERT S System

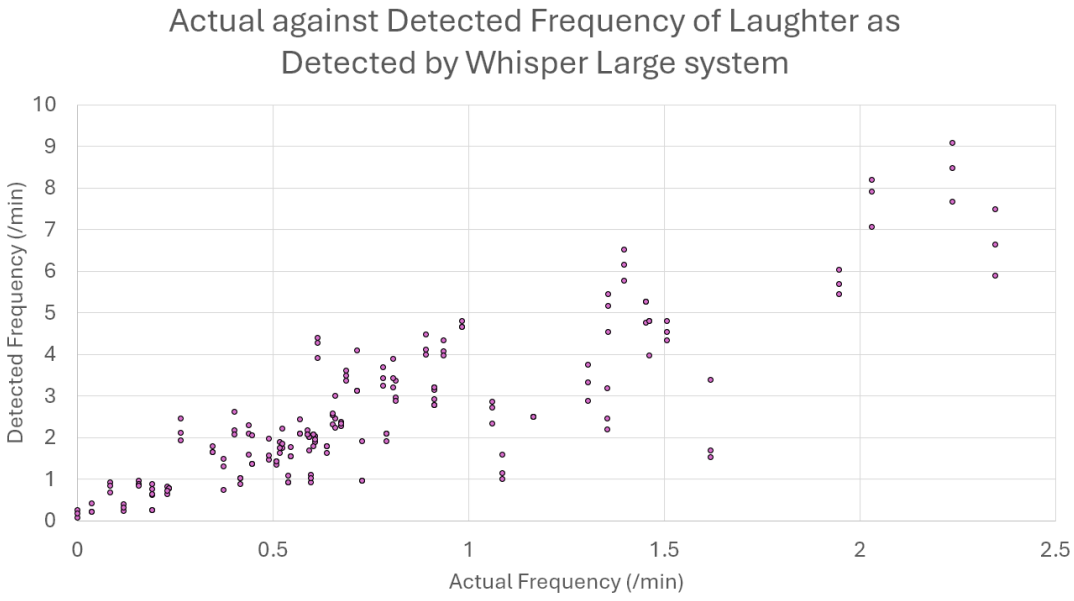

Figure 6.29: Correlation of Actual and Detected Frequency of Laughter Per Minute of Conversation Achieved by Whisper L System

detector performance was significantly different to the ground truth labels. This agrees with the above findings, in which the CBA system did not correlate with the ground truth labels when the total laughter is considered. When examining the frequency of laughter, the same behaviour is seen with the two-way ANOVA test; finding no significant difference by gender pairing (F(2, 114) = 2.24, $p = 0.11$) and no significant interaction between gender pairing and method (F(2, 114) = 0.36, $p = 0.70$). However, once again a significant effect by method was found (F(1, 114) = 19.92, $p < 0.0001$). Again, this confirms the above correlation finding that the CBA system had no significant correlation with the actual labels. Taken together, these results show that the CBA system does not perform well enough to be useful for this task.

Examining the HuBERT S system, in terms of total laughter, a two-way ANOVA test found no significant effect on gender pairing (F(2, 114) = 1.66, $p = 0.20$), method (F(1, 114) = 2.92, $p = 0.090$) or interaction between gender pairing and method (F(2, 114) = 0.21, $p = 0.81$). This suggests that, in terms of total laughter, there are no significant differences between HuBERT S and the ground-truth labels. In terms of laughter frequency, a two-way ANOVA test did find a significant effect on gender pairing (F(2, 114) = 7.04, $p = 0.001$). There was no significant difference by method (F(1, 114) = 3.83, $p = 0.053$) or interaction between method and gender pairing (F(2, 114) = 0.36, $p = 0.70$). This suggests that, at a frequency level, there were differences in laughter tendency by gender pairing and that HuBERT S was as capable as the ground truth labels at representing this. A post-hoc Tukey HSD test determined that MM pairings ($M = 1.08$, $SD = 0.63$) laughed significantly less than FF pairings ($M = 2.20$, $SD = 1.55$, $p = 0.001$, 95% C.I. = [0.39, 1.85]). MM pairings also laughed significantly less than MF pairings ($M = 1.90$, $SD = 1.20$, $p = 0.011$, 95% C.I. = [0.16, 1.47]). No significant difference was found between MF and FF pairings ($p = 0.48$, 95% C.I. = [-0.31, 0.92]).

Whisper L saw significant effects regarding total laughter by method (F(1, 114) = 11.97, $p < 0.0001$). No significant effects were seen by gender pairing (F(2, 114) = 0.40, $p = 0.67$) and no significant interaction was seen between method and pairing (F(2, 114) = 0.039, $p = 0.96$). This suggests that, in terms of total laughter, Whisper L produces different results compared with group truth labels for estimating laughter in a conversation. When addressing frequency of laughter, a two-way ANOVA test found significant differences by gender pairing (F(2, 114) = 5.20, $p = 0.007$) and method (F(1, 114) = 15.01, $p < 0.0001$). No significant effect was found in the interaction between method and gender pairing (F(2, 114) = 0.30, $p = 0.74$). A post-hoc Tukey HSD test determined that MM pairings ($M = 1.08$, $SD = 0.75$) laughed significantly less than FF pairings ($M = 1.99$, $SD = 1.49$, $p = 0.007$, 95% C.I. = [0.21, 1.61]). Furthermore, MM pairings laughter significantly less than MF pairings ($M = 1.77$, $SD = 1.24$, $p = 0.027$, 95% C.I. = [0.063, 1.33]). No significant difference was found between MF and FF pairings ($p = 0.66$, 95% C.I. = [-0.38, 0.81]). These differences match those found with the HuBERT S system. However, a significant effect by method was also seen in Whisper L. To examine this further, a one-way ANOVA test was used to examine only Whisper L's detected laughter frequency.

The test found no significant difference between the detected laughter frequency of each gender pairing (F(1, 57) = 2.47, $p = 0.094$). However, a significant effect was found when a one-way ANOVA test was used to test the differences by gender pairing with the actual labels (F(1, 57) = 3.60, $p = 0.034$). A post-hoc Tukey HSD test determined that MM ($M = 0.73$, $SD = 0.42$) laughed significantly less than MF pairings ($M = 1.38$, $SD = 0.86$, $p = 0.048$, 95% C.I. = [0.0058, 1.29]). No significant difference was found between MM and FF pairings ($M = 1.44$, $SD = 0.98$, $p = 0.053$, 95% C.I. = [-0.0068, 1.42]), nor between FF and MF pairings ($p = 0.66$, 95% C.I. = [-1.42, 0.67]). These results suggest that Whisper L was unable to correctly identify gender pairing laughter differences.

The above results show that the CBA system is ineffective at estimating the laughter in a conversation (both frequency and total). However, both HuBERT S and Whisper L approaches were able to capture both the frequency and total amount of laughter in conversations relative to the other conversations. However, only HuBERT S was able to reliably capture the differences in tendency to laugh based on the gender pairing of a conversation.

6.5 Conclusion

This chapter addressed RQ4: Are transformers effective when applied to the task of laughter detection? Two methods using transformers were tested, using them as an underlying deep learning architecture and as a method of feature extraction. It was found that, with the current amount of data available in the SMC for laughter detection, training a transformer from initialisation is ineffective. However, transformers pre-trained on other related audio detection tasks were found to provide an effective means of feature extraction. This suggests that with enough data it would be possible to train a transformer laughter detector from initialisation but that this is unnecessary given the effectiveness of the transfer learning approach. These results offer strong evidence in answer to RQ4: that transformers are effective when applied to laughter detection.

Furthermore, these features were also effective when applied to fillers. However, they were less effective when applied to back-channel events, requiring the pre/post-processing methods to achieve an F1 of 50%, at both a frame and an event level. Finally, three different laughter detection systems were tested for their ability to effectively detect the frequency and total amount of laughter in a conversation. They were tested for their ability to detect common differences in tendencies to laugh in a conversation due to the gender pairing of the speakers. It was shown that Whisper L was the most effective at estimating both laughter frequency and total amount, in terms of the correlation with the ground truth. HuBERT S was shown to effectively capture differences in gender pairings' tendency to laugh. This is despite having an event level F1 score of 60%, around $\sim$20% less than Whisper Large.

Chapter 7

Conclusions and Future Work

This thesis addressed the task of automatic laughter detection in spontaneous conversations. It was shown, through an initial review of the field, that the task definition and evaluation methods varied. The choices made affected the estimation of the effectiveness of the field's methods. In particular, the use of area under the (receiver operator) curve (AUC) had led to inflated estimations of detector effectiveness due to its inclusion of true negatives in its calculation and the inherent class imbalances present in laughter detection datasets. Moreover, the more constraints placed upon the task of laughter detection, the easier the task and the higher the score. Important constraints include the length of the clips analysed, the laughter to non-laughter class balance (both within each clip and the datasets as a whole), the number of laughs per clip and the spontaneity of the laughter generation method. A total of 4 research questions addressed these issues over the course of the thesis. These were:

- RQ1: **Are State-Of-The-Art laughter detectors effective when common experimental constraints are removed?**
- RQ2: **Can the incorporation of linguistic data lead to improvements in laughter detection?**
- RQ3: **What is the effect of broadening the scope of laughter detectors to include multiple cues?**
- RQ4: **Are transformers effective when applied to the task of laughter detection?**

Chapter 4 addressed RQ1. It was initially demonstrated that detection systems were able to achieve state-of-the-art (SOTA) detection performance on the publicly available SSPNet vocalisation corpus (SVC [41]). These systems were then applied to the SSPNet mobile corpus (SMC [55]) and it was shown that the best-performing detection system achieved a frame level F1 of 15% and an event level F1 of 25%. These results provided evidence that SOTA methods were ineffective when the constraints on total audio time and the ratio of laughter to non-laughter were removed. These results offered answers to RQ1. A performance analysis found that the

central issue limiting performance was the creation of false positives due to speech being mistaken for laughter. Furthermore, it showed that the most effective method for improving laughter detection, i.e., smoothing, could only improve precision at the cost of decreased recall. The results of this performance analysis led to the creation of two more research questions RQ2 and RQ3.

Chapter 5 addressed both of these questions. RQ2 was investigated through the creation of novel pre/post-processing methods which incorporated linguistic information in a variety of ways. This was done as speech was the leading cause of mistakes. Linguistic data was automatically extracted by both an automatic speech recognition (ASR) and a voice activity detection (VAD) system. It was demonstrated that the incorporation of this data into the detection system could double the F1 achieved (i.e., a frame level of 33% and event level of 45%) by the SOTA methods in the previous chapter. This offered evidence that laughter detection could be improved through the incorporation of linguistic information, answering RQ2. RQ3 was addressed by the second half of Chapter 5. The detectors were broadened to detect multiple cues in an effort to address the class imbalance issue and provide the networks with more context during training. However, the results of this section of the work were not promising with large reductions in F1. The result of RQ3 was therefore negative and this direction of research was abandoned.

Chapter 6 investigated RQ4 but using new underlying architecture - transformers. It was found that the amount of data currently available was not sufficient to train a randomly initialized laughter detection transformer. However, multiple pre-trained transformer models exist that address related tasks. These pre-trained models were investigated for their ability to effectively represent audio data with their attention embeddings. These embeddings were used as features to train a new classifier in laughter detection. This was found to be the most effective means of carrying out laughter detection, with F1 scores almost doubling from the previous chapter (i.e., a frame level of 62% and event level of 82%). These results addressed RQ4 and suggested that Transformers are effective when applied to the task of laughter detection. Having achieved effective laughter detection it was then shown that this method of detection was equally effective when applied to filler detection. However, back-channel detection proved a more difficult task. Finally, the ability of different laughter detection systems to detect differences in laughter behaviour by the speakers' gender pairing was investigated. It was shown that one of the transformer-based detection systems was able to reliably replicate differences in laughter tendency by gender pairing and that it did not differ significantly from the results found when using the ground truth labels.

In summary, laughter detection was achieved in a type 3 task setting. Over the entirety of the work, the effectiveness of SOTA detectors was almost quadrupled.

7.1 Directions for Future Work

This section discusses possible future directions for this work: both to improve laughter detection, but also to improve paralanguage detection, and understanding of it more broadly.

- **Further reduce constraints.** This work addresses a type 3 laughter detection task. This represents the least constrained type of task in the field of laughter detection. However, constraints were still present, such as the phone conversations occurring in a controlled environment, each conversation being comprised of only two speakers and all conversations addressing the same topics. It is possible that, by removing these constraints, a similar performance difference from type 1 to 2 and 2 to 3 task would be seen.

- **Investigate and address the effect of gender/gender pairing.** Each of the three best-performing laughter detection systems saw significant differences in their performance by gender and gender pairing, with female speakers generally seeing better performance. It was suggested that, by gathering more data from male speakers, this performance gap could be narrowed or eliminated, but this has not been tested yet.

- **Investigate the detection of other non-verbal cues.** Initial work addressing filler and back-channel cues is presented. Filler events were shown to be as reliably detected as laughter. However, detection systems were shown to be ineffective at detecting back-channel at both a frame and an event level. This shows the need for further work that properly addresses back-channel, which calls for an investigation into other non-verbal cues such as sighs, cries and gasps. Each non-verbal cue carries important communication between speakers; this work shows that methods that are effective at detecting one non-verbal cue may not generalise to all of them.

- **Expand the ability to understand speaker traits.** In Section 6.4, it was shown that common differences in the tendency to laugh by pairing can be detected as well as when the ground truth labels are used. This work could be extended to include other traits. For example, personality detection, conflict detection and gender detection.

- **Explore contrastive learning.** Laughter detection was improved through the application of longer contexts and better representation of feature representation through the use of transformers. It is possible that through the use of contrastive learning the underlying representation of laughter may be improved to further increase the effectiveness of detectors and remove the need for large foundational transformer models to be used for feature extraction.

- **Investigate Loss Functions** Class weighting from Chapter 4 was the only method which directly changed the loss function calculation. It had a marked effect on the performance and behaviour of the laughter detectors. It is possible that further changes to the loss

function, or selecting different loss functions entirely, may have a positive impact on the detectors. There are loss functions designed for environments where there are large class imbalances such as Focal loss or Dice/Tversky loss. As with the direction above, improving the underlying detectors could remove the need for large foundational transformer models.

7.2 Concluding Remarks

This thesis achieved a four-fold increase in the event level F1 score over state-of-the-art approaches for the task of laughter detection. This was achieved through the application of the attention embeddings created by publicly available transformer models. It was shown, initially, that the least constrained laughter detection task was under-researched in the literature. Furthermore, the efficacy of laughter detection systems on this type of task was shown to be over-inflated by the use of AUC as a metric. An extensive performance analysis revealed multiple issues with the state-of-the-art methods. It was demonstrated that the state-of-the-art performance could be doubled, in terms of event level F1, through post-processing methods that specifically addressed the central issues identified. It was then demonstrated that transformer attention embeddings could be used to further double the performance of these laughter detection systems. This final method achieved effective laughter detection in spontaneous conversations. Finally, the methodology was then extended and was shown to achieve the same level of effectiveness for automatic filler detection in spontaneous conversations. This approach used publicly available pre-trained transformer models. This means it can be applied by anyone. As a result, this approach is practical and useful to the field at large.

Appendix A

Significance Statistics Tables for Chapter 4

Table A.1: Significance values for each model in relation to the best performing detector (LSTM+D+S) for frame Level AUC Performance on the SVC Using Merging of All Test Clips. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. D: delta. S: smoothing.

Model Type	Confidence Interval		p-value
	Lower	Upper	
FNN	11.2339	21.8061	0
FNN + CW	9.7139	20.2861	0
FNN + D	9.8639	20.4361	0
FFN + S	5.6239	16.1961	0
FFN + CW + D	8.2139	18.7861	0
FFN + CW + S	4.1939	14.7661	0
FFN + D + S	3.5439	14.1161	0
FFN + CW + D + S	2.8639	13.4361	0.00002
LSTM	1.6739	12.2461	0.00086
LSTM + CW	3.8939	14.4661	0
LSTM + D	0.9339	11.5061	0.00609
LSTM + S	-5.9461	4.6261	1
LSTM + CW + D	4.7039	15.2761	0
LSTM + CW + S	-1.1461	9.4261	0.33059
LSTM + D + S	x	x	x
LSTM + CW + D + S	-0.3361	10.2361	0.09497

Table A.2: Significance values for each model in relation to the best performing detectors (Right: LSTM+CW+S and Left: FFN+CW+D+S) for frame Level F1 Performance on the SVC Using Merging of All Test Clips. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. D: delta. S: smoothing.

Model Type	Confidence Interval		p-value	Confidence Interval		p-value
	Lower	Upper		Lower	Upper	
FNN	27.0389	44.9411	0	-43.5411	-25.6389	0
FNN + CW	0.3789	18.2811	0.03147	-16.8811	1.0211	0.15043
FNN + D	26.8589	44.7611	0	-43.3611	-25.4589	0
FFN + S	32.6489	50.5511	0	-49.1511	-31.2489	0
FFN + CW + D	-1.9411	15.9611	0.33067	-3.3411	14.5611	0.71884
FFN + CW + S	-5.8011	12.1011	0.99787	-10.7011	7.2011	1
FFN + D + S	32.2489	50.1511	0	-48.7511	-30.8489	0
FFN + CW + D + S	-7.5511	10.3511	1	x	x	x
LSTM	15.0889	32.9911	0	-31.5911	-13.6889	0
LSTM + CW	-4.9911	12.9111	0.97808	-11.5111	6.3911	0.99981
LSTM + D	-35.3011	-17.3989	0	-33.9011	-15.9989	0
LSTM + S	-41.9711	-24.0689	0	-40.5711	-22.6689	0
LSTM + CW + D	-0.6911	17.2111	0.108	-15.8111	2.0911	0.36847
LSTM + CW + S	x	x	x	-7.5511	10.3511	1
LSTM + D + S	-42.3211	-24.4189	0	-40.9211	-23.0189	0
LSTM + CW + D + S	-4.2111	13.6911	0.90283	-12.2911	5.6111	0.99598

Table A.3: Significance values for each model in relation to the best performing detector (LSTM+CW) for event Level F1 Performance on the SVC Using Exclusion Methodology. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. D: delta. S: smoothing.

Model Type	Confidence Interval		p-value
	Lower	Upper	
FNN	34.263	44.477	0
FNN + CW	14.163	24.377	0
FNN + D	35.453	45.667	0
FFN + S	46.373	56.587	0
FFN + CW + D	8.233	18.447	0
FFN + CW + S	11.333	21.547	0
FFN + D + S	47.093	57.307	0
FFN + CW + D + S	13.873	24.087	0
LSTM	27.333	37.547	0
LSTM + CW	x	x	x
LSTM + D	-48.097	-37.883	0
LSTM + S	-49.297	-39.083	0
LSTM + CW + D	-10.147	0.067	0.05739
LSTM + CW + S	-10.447	-0.233	0.03032
LSTM + D + S	-55.727	-45.513	0
LSTM + CW + D + S	-12.507	-2.293	0.00011

Table A.4: Significance values for each model in relation to the best performing detector (LSTM+S) for AUC Performance on the SMC. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. D: delta. S: smoothing.

Model Type	Confidence Interval		p-value
	Lower	Upper	
FNN	8.0201	27.6799	0
FNN + CW	19.4401	39.0999	0
FNN + D	7.1301	26.7899	0
FFN + S	-1.3999	18.2599	0.1908
FFN + CW + D	6.7601	26.4199	0
FFN + CW + S	16.0601	35.7199	0
FFN + D + S	-2.1299	17.5299	0.33028
FFN + CW + D + S	-2.1399	17.5199	0.33251
LSTM	-4.0799	15.5799	0.81029
LSTM + CW	-2.3099	17.3499	0.37166
LSTM + D	-1.0399	18.6199	0.13993
LSTM + S	x	x	x
LSTM + CW + D	3.4801	23.1399	0.00049
LSTM + CW + S	-6.5099	13.1499	0.99866
LSTM + D + S	-6.2999	13.3599	0.99734
LSTM + CW + D + S	-0.5799	19.0799	0.09088

Table A.5: Significance values for each model in relation to the best performing detector (FFN+CW+D+S) for Frame Level F1 Performance on the SMC. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. D: delta. S: smoothing.

Model Type	Confidence Interval		p-value
	Lower	Upper	
FNN	-10.8724	1.1324	0.27203
FNN + CW	-15.1124	-3.1076	0.00004
FNN + D	-9.5724	2.4324	0.79029
FFN + S	-14.5824	-2.5776	0.00015
FFN + CW + D	-1.3724	10.6324	0.35706
FFN + CW + S	-12.3224	-0.3176	0.02791
FFN + D + S	-13.1124	-1.1076	0.00549
FFN + CW + D + S	x	x	x
LSTM	-16.1724	-4.1676	0
LSTM + CW	-9.1324	2.8724	0.91328
LSTM + D	-20.4224	-8.4176	0
LSTM + S	-20.0724	-8.0676	0
LSTM + CW + D	-11.6524	0.3524	0.09063
LSTM + CW + S	-7.2424	4.7624	1
LSTM + D + S	-20.6424	-8.6376	0
LSTM + CW + D + S	-10.2124	1.7924	0.52994

Table A.6: Significance values for each model in relation to the best performing detector (LSTM+CW+S) for Event Level F1 Performance on the SMC. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. D: delta. S: smoothing.

Model Type	Confidence Interval		p-value
	Lower	Upper	
FNN	-7.5471	11.5471	1
FNN + CW	6.6929	25.7871	0
FNN + D	-7.1171	11.9771	0.99996
FFN + S	-4.1071	14.9871	0.83913
FFN + CW + D	3.3029	22.3971	0.00056
FFN + CW + S	4.2329	23.3271	0.00012
FFN + D + S	-6.2571	12.8371	0.99832
FFN + CW + D + S	-5.5671	13.5271	0.98752
LSTM	-2.6071	16.4871	0.46444
LSTM + CW	-9.5471	9.5471	1
LSTM + D	-25.5671	-6.4729	0
LSTM + S	-25.1271	-6.0329	0
LSTM + CW + D	2.4729	21.5671	0.00198
LSTM + CW + S	x	x	x
LSTM + D + S	-24.1671	-5.0729	0.00003
LSTM + CW + D + S	-3.6571	15.4371	0.74166

Table A.7: Significance values for t-tests comparing LSTM+CW+S laughter detection by role.

Level	Metric	t statistic	p-value
Frame	AUC	9.53	<0.0001
	Precision	7.69	<0.0001
	Recall	4.66	<0.0001
	F1	2.77	<0.0001
Event	Precision	0.26	0.38
	Recall	5.71	<0.0001
	F1	0.38	0.74

Table A.8: Significance values for t-tests comparing LSTM+CW+S laughter detection by Gender.

Level	Metric	t statistic	p-value
Frame	AUC	0.53	0.59
	Precision	1.48	0.14
	Recall	3.69	0.0003
	F1	2.65	0.0085
Event	Precision	2.29	0.024
	Recall	3.58	0.0005
	F1	2.75	0.0068

Table A.9: Significance values for one-way ANOVAs and associated post-hoc Tukey tests comparing performance of LSTM+CW+S detector on each metric by gender pairing in a conversation.

Level	Metric	MM-MF Confidence Interval		p-value	MM-FF Confidence Interval		p-value	FF-MF Confidence Interval		p-value
		Lower	Upper		Lower	Upper		Lower	Upper	
Frame	AUC	-5.81	-0.57	0.012	-0.14	6.04	0.065	-3.03	2.55	0.98
	Precision					No difference				
	Recall	1.25	18.41	0.02	6.12	25.16	0.0004	2.24	13.86	0.201
	F1	-2.16	7.8	0.38	0.35	11.41	0.034	-7.74	1.62	0.27
Event	Precision					No difference				
	Recall					No difference				
	F1					No difference				

Appendix B

Significance Statistics Tables for Chapter 5

Table B.1: Significance values for each model in relation to the baseline detector (LSTM+CW+S) for AUC Performance on the SMC Using ASR Approaches. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. S: smoothing. C: confidence-based alteration. E: feature vector extension. U: undersampling.

Model Type	Confidence Interval		p-value
	Lower	Upper	
FFN+CW	17.4052	34.4948	0
FFN+CW+S	14.0252	31.1148	0
FFN+CW+U	-2.5948	14.4948	0.54305
FFN+CW+U+S	-8.2548	8.8348	1
FFN+CW+C	5.2352	22.3248	0.00001
FFN+CW+C+S	0.3452	17.4348	0.03216
FFN+CW+E	-8.0848	9.0048	1
FFN+CW+E+S	-13.6448	3.4448	0.7859
LSTM+CW	-4.3448	12.7448	0.94535
LSTM+CW+S	x	x	x
LSTM+CW+U	-14.0448	3.0448	0.67787
LSTM+CW+U+S	-9.1648	7.9248	1
LSTM+CW+C	-6.4448	10.6448	0.99997
LSTM+CW+C+S	-10.8148	6.2748	0.99993
LSTM+CW+E	-7.5848	9.5048	1
LSTM+CW+E+S	-11.8248	5.2648	0.99458

Table B.2: Significance values for each model in relation to the baseline detector (FFN+CW) for Frame Level Precision Performance on the SMC Using ASR Approaches. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. S: smoothing. C: confidence-based alteration. E: feature vector extension. U: undersampling.

Model Type	Confidence Interval		p-value
	Lower	Upper	
FFN+CW	x	x	x
FFN+CW+S	-11.0319	9.7919	1
FFN+CW+U	-5.3419	15.4819	0.94939
FFN+CW+U+S	1.6181	22.4419	0.00803
FFN+CW+C	-1.6919	19.1319	0.22373
FFN+CW+C+S	5.2581	26.0819	0.00004
FFN+CW+E	-9.1619	11.6619	1
FFN+CW+E+S	-4.5919	16.2319	0.85816
LSTM+CW	-11.5719	9.2519	1
LSTM+CW+S	-10.4119	10.4119	1
LSTM+CW+U	-3.1519	17.6719	0.54059
LSTM+CW+U+S	4.2381	25.0619	0.00022
LSTM+CW+C	-1.3519	19.4719	0.17203
LSTM+CW+C+S	6.8681	27.6919	0
LSTM+CW+E	-9.9419	10.8819	1
LSTM+CW+E+S	-7.5619	13.2619	0.99989

Table B.3: Significance values for each model in relation to the baseline detectors (Right: FFN+CW, Left: LSTM+CW) for Frame Level Recall Performance on the SMC Using ASR Approaches. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. S: smoothing. C: confidence-based alteration. E: feature vector extension. U: undersampling.

Model Type	Confidence Interval		p-value	Confidence Interval		p-value
	Lower	Upper		Lower	Upper	
FFN+CW	x	x	x	5.0262	40.5338	0.00141
FFN+CW+S	-20.2238	15.2838	1	7.4962	43.0038	0.00017
FFN+CW+U	-6.5238	28.9838	0.70501	-6.2038	29.3038	0.6607
FFN+CW+U+S	-8.3838	27.1238	0.90518	-4.3438	31.1638	0.39448
FFN+CW+C	-27.9738	7.5338	0.8283	15.2462	50.7538	0
FFN+CW+C+S	-35.5538	-0.0462	0.04864	22.8262	58.3338	0
FFN+CW+E	5.7962	41.3038	0.00074	-18.5238	16.9838	1
FFN+CW+E+S	8.3862	43.8938	0.00008	-21.1138	14.3938	1
LSTM+CW	5.0262	40.5338	0.00141	x	x	x
LSTM+CW+S	8.5562	44.0638	0.00006	-14.2238	21.2838	1
LSTM+CW+U	-12.9438	22.5638	0.9999	-35.7238	-0.2162	0.04389
LSTM+CW+U+S	-15.1738	20.3338	1	-37.9538	-2.4462	0.01005
LSTM+CW+C	-10.1738	25.3338	0.98429	-32.9538	2.5538	0.19303
LSTM+CW+C+S	-10.3438	25.1638	0.98737	-33.1238	2.3838	0.17848
LSTM+CW+E	9.4462	44.9538	0.00003	-13.3338	22.1738	0.99997
LSTM+CW+E+S	12.3462	47.8538	0	-10.4338	25.0738	0.9888

Table B.4: Significance values for each model in relation to the baseline detector (LSTM+CW+S) for Frame Level F1 Performance on the SMC Using ASR Approaches. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. S: smoothing. C: confidence-based alteration. E: feature vector extension. U: undersampling.

Model Type	Confidence Interval		p-value
	Lower	Upper	
FFN+CW	-0.1765	15.9165	0.06286
FFN+CW+S	-2.9665	13.1265	0.70792
FFN+CW+U	-14.8465	1.2465	0.21086
FFN+CW+U+S	-22.6665	-6.5735	0
FFN+CW+C	-5.5165	10.5765	0.99941
FFN+CW+C+S	-12.5565	3.5365	0.85557
FFN+CW+E	-9.9065	6.1865	0.99999
FFN+CW+E+S	-16.4465	-0.3535	0.0309
LSTM+CW	-6.1565	9.9365	0.99998
LSTM+CW+S	x	x	x
LSTM+CW+U	0.5335	16.6265	0.02392
LSTM+CW+U+S	7.6735	23.7665	0
LSTM+CW+C	-19.0965	-3.0035	0.00037
LSTM+CW+C+S	-26.8965	-10.8035	0
LSTM+CW+E	-8.6665	7.4265	1
LSTM+CW+E+S	-11.8465	4.2465	0.96086

Table B.5: Significance values for each model in relation to the best performing detector (LSTM+CW+C+S) for Frame Level F1 Performance on the SMC Using ASR Approaches. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. S: smoothing. C: confidence-based alteration. E: feature vector extension. U: undersampling.

Model Type	Confidence Interval		p-value
	Lower	Upper	
FFN+CW	18.6735	34.7665	0
FFN+CW+S	15.8835	31.9765	0
FFN+CW+U	4.0035	20.0965	0.00005
FFN+CW+U+S	-3.8165	12.2765	0.9079
FFN+CW+C	13.3335	29.4265	0
FFN+CW+C+S	6.2935	22.3865	0
FFN+CW+E	8.9435	25.0365	0
FFN+CW+E+S	2.4035	18.4965	0.00112
LSTM+CW	12.6935	28.7865	0
LSTM+CW+S	-26.8965	-10.8035	0
LSTM+CW+U	-18.3165	-2.2235	0.00155
LSTM+CW+U+S	-11.1765	4.9165	0.99377
LSTM+CW+C	-0.2465	15.8465	0.06868
LSTM+CW+C+S	x	x	x
LSTM+CW+E	-26.2765	-10.1835	0
LSTM+CW+E+S	-23.0965	-7.0035	0

Table B.6: Significance values for each model in relation to the baseline detector (LSTM+CW+S) for Event Level Precision Performance on the SMC Using ASR Approaches. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. S: smoothing. C: confidence-based alteration. E: feature vector extension. U: undersampling.

Model Type	Confidence Interval		p-value
	Lower	Upper	
FFN+CW	-5.1954	20.8354	0.77744
FFN+CW+S	-9.6054	16.4254	0.99994
FFN+CW+U	-10.5354	15.4954	1
FFN+CW+U+S	-27.7454	-1.7146	0.01085
FFN+CW+C	-15.1354	10.8954	1
FFN+CW+C+S	-20.2254	5.8054	0.8665
FFN+CW+E	-8.7154	17.3154	0.99896
FFN+CW+E+S	-20.8354	5.1954	0.77744
LSTM+CW	-6.8354	19.1954	0.95901
LSTM+CW+S	x	x	x
LSTM+CW+U	-6.0054	20.0254	0.8902
LSTM+CW+U+S	5.1146	31.1454	0.00027
LSTM+CW+C	-23.8654	2.1654	0.23049
LSTM+CW+C+S	-35.0054	-8.9746	0
LSTM+CW+E	-10.4154	15.6154	1
LSTM+CW+E+S	-15.6254	10.4054	1

Table B.7: Significance values for each model in relation to the baseline detector (LSTM+CW) for Event Level Recall Performance on the SMC Using ASR Approaches. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. S: smoothing. C: confidence-based alteration. E: feature vector extension. U: undersampling.

Model Type	Confidence Interval		p-value
	Lower	Upper	
FFN+CW	0.0073	33.8327	0.04977
FFN+CW+S	30.3173	64.1427	0
FFN+CW+U	-18.8627	14.9627	1
FFN+CW+U+S	2.5373	36.3627	0.00861
FFN+CW+C	0.4173	34.2427	0.0383
FFN+CW+C+S	31.2273	65.0527	0
FFN+CW+E	-21.8927	11.9327	0.99973
FFN+CW+E+S	-5.6927	28.1327	0.62819
LSTM+CW	x	x	x
LSTM+CW+S	-33.2327	0.5927	0.0718
LSTM+CW+U	-28.5427	5.2827	0.5656
LSTM+CW+U+S	-43.0627	-9.2373	0.00002
LSTM+CW+C	-25.1627	8.6627	0.94865
LSTM+CW+C+S	-43.6027	-9.7773	0.00001
LSTM+CW+E	-17.1127	16.7127	1
LSTM+CW+E+S	-28.7227	5.1027	0.53796

Table B.8: Significance values for each model in relation to the baseline detector (LSTM+CW+S) for Event Level F1 Performance on the SMC Using ASR Approaches. FFN: feed forward neural network. LSTM: long short-term memory network. CW: class weight. S: smoothing. C: confidence-based alteration. E: feature vector extension. U: undersampling.

Model Type	Confidence Interval		p-value
	Lower	Upper	
FFN+CW	5.8439	26.6361	0.00002
FFN+CW+S	3.3839	24.1761	0.00075
FFN+CW+U	-7.9361	12.8561	0.99998
FFN+CW+U+S	-26.2161	-5.4239	0.00003
FFN+CW+C	-1.2961	19.4961	0.1647
FFN+CW+C+S	-6.0961	14.6961	0.98844
FFN+CW+E	-5.2661	15.5261	0.94359
FFN+CW+E+S	-20.0061	0.7861	0.1064
LSTM+CW	-2.2761	18.5161	0.33525
LSTM+CW+S	x	x	x
LSTM+CW+U	-2.4461	18.3461	0.37236
LSTM+CW+U+S	4.8939	25.6861	0.00008
LSTM+CW+C	-23.9961	-3.2039	0.00098
LSTM+CW+C+S	-29.9861	-9.1939	0
LSTM+CW+E	-7.4161	13.3761	0.99981
LSTM+CW+E+S	-13.5961	7.1961	0.99954

Table B.9: Significance values for t-tests comparing performance of LSTM+CW+C+S detector by gender.

Level	Metric	t-statistic	p-value
Frame	Precision	t(358) = 4.05	0.0001
	Recall	t(358) = 3.58	0.00038
	F1	t(358) = 4.66	0.0001
	AUC	t(358) = 3.86	0.00013
Event	Precision	t(358) = 2.56	0.12
	Recall	t(358) = 3.60	0.00047
	F1	t(358) = 3.22	0.0017

Table B.10: Significance values for one-way ANOVAs and associated post-hoc Tukey tests comparing performance of LSTM+CW+C+S detector on each metric by gender pairing in a conversation.

Level	Metric	Gender pairing	one-way ANOVA	Post-Hoc Tukey Confidence Interval		p-value
				Lower	Upper	
Frame	Precision	MM-FF	F(2, 177) = 3.51, $p = 0.032$	0.35	18	0.04
		FF-MF		0.67	14.27	0.082
		MM-MF		5.58	10.32	0.76
	Recall	MM-FF	F(2, 177) = 5.63, $p = 0.0043$	3.83	22.03	0.0028
		FF-MF		1.96	13.46	0.19
		MM-MF		1.03	15.39	0.1
	F1	MM-FF	F(2, 177) = 5.35, $p = 0.0056$	2.75	17.73	0.0042
		FF-MF		0.024	13.54	0.049
		MM-MF		2.88	9.8	0.4
	AUC	MM-FF	F(2, 177) = 4.45, $p = 0.013$	-0.16	6.38	0.067
		FF-MF		0.58	6.12	0.013
		MM-MF		-3.19	2.71	0.98
Event	Precision	MM-FF	F(2, 177) = 8.66, $p = 0.0003$	9.3	33.76	0.0001
		FF-MF		-1.19	19.51	0.095
		MM-MF		1.34	23.4	0.024
	Recall	MM-FF	F(2, 177) = 8.73, $p = 0.0002$	7.53	27.47	0.0002
		FF-MF		-2.06	14.81	0.18
		MM-MF		2.13	20.11	0.011
	F1	MM-FF	F(2, 177) = 11.40, $p < 0.0001$	8.98	28.82	0.0001
		FF-MF		-4.65	12.14	0.54
		MM-MF		6.2	24.1	0.0003

Table B.11: Significance values for t-tests comparing FFN and LSTM based architectures for multi-cue detection.

Level	Metric	T-test	
		t-statistic	p-value
Frame	Precision	t(34) = 4.96	0.033
	Recall	t(34) = 4.37	0.044
	F1	t(34) = 41.35	0.0001
	AUC	t(34) = 89.08	0.0001
Event	Precision	t(34) = 12.25	0.0013
	Recall	t(34) = 9.69	0.0037
	F1	t(34) = 23.36	0.0001

Table B.12: Significance values for AUC performance by the multi-label detection system compared to the Confidence Based Alteration system.

	Confidence Interval		
Detector	Lower	Upper	p-value
Back-channel	7.21	13.11	0.0001
Laugh	4.63	10.53	0.0001
Filler	11.36	17.26	0.0001
Pause	1.86	7.76	0.0001
Speech	27.34	33.24	0.0001
Average	8.56	14.45	0.0001

Table B.13: Significance values for frame level precision performance by the multi-label detection system compared to the Confidence Based Alteration system.

	Confidence Interval		
Detector	Lower	Upper	p-value
Back-channel	7.21	13.11	0.0001
Laugh	4.63	10.53	0.0001
Filler	11.36	17.26	0.0001
Pause	1.86	7.76	0.0001
Speech	27.34	33.24	0.0001
Average	8.56	14.45	0.0001

Table B.14: Significance values for frame level recall performance by the multi-label detection system compared to the Confidence Based Alteration system.

	Confidence Interval		
Detector	Lower	Upper	p-value
Back-channel	40.57	55.09	0.0001
Laugh	11.74	26.26	0.0001
Filler	18.91	33.43	0.0001
Pause	28.85	43.37	0.0001
Speech	38.34	52.86	0.0001
Average	-5	9.51	0.97

Table B.15: Significance values for frame level F1 performance by the multi-label detection system compared to the Confidence Based Alteration system.

	Confidence Interval		
Detector	Lower	Upper	p-value
Back-channel	25.61	33.99	0.0001
Laugh	9.04	17.42	0.0001
Filler	14.66	23.04	0.0001
Pause	15.48	23.86	0.0001
Speech	23.39	31.77	0.0001
Average	1.26	7.12	0.36

Table B.16: Significance values for event level precision performance by the multi-label detection system compared to the Confidence Based Alteration system.

	Confidence Interval		
Detector	Lower	Upper	p-value
Back-channel	30.55	39.59	0.0001
Laugh	26.06	35.1	0.0001
Filler	27.59	36.63	0.0001
Pause	8.1	17.14	0.0001
Speech	2.03	11.07	0.00057
Average	16.25	25.29	0.0001

Table B.17: Significance values for event level recall performance by the multi-label detection system compared to the Confidence Based Alteration system.

	Confidence Interval		
Detector	Lower	Upper	p-value
Back-channel	24.48	41.6	0.0001
Laugh	2.69	14.43	0.38
Filler	17.39	34.51	0.0001
Pause	0.039	17.08	0.052
Speech	19.24	36.36	0.0001
Average	-2.18	8.9	0.23

Table B.18: Significance values for event level F1 performance by the multi-label detection system compared to the Confidence Based Alteration system.

	Confidence Interval		
Detector	Lower	Upper	p-value
Back-channel	35.73	44.55	0.0001
Laugh	27.56	36.38	0.0001
Filler	30.94	39.76	0.0001
Pause	4.26	13.08	0.0001
Speech	9.09	17.91	0.0001
Average	15.2	24.03	0.0001

Table B.19: Significance values for comparing different threshold values for NVC super-class detection system for precision.

Threshold Values		p-value	95% Confidence Interval	
			Lower Bound	Upper Bound
.1	.2	0.96	-0.14	0.08
	.3	0.51	-0.17	0.05
	.4	0.02	-0.23	-0.01
	.5	0.00	-0.26	-0.04
.2	.1	0.96	-0.08	0.14
	.3	0.90	-0.14	0.07
	.4	0.13	-0.20	0.02
	.5	0.02	-0.23	-0.01
.3	.1	0.51	-0.05	0.17
	.2	0.90	-0.07	0.14
	.4	0.57	-0.17	0.05
	.5	0.19	-0.20	0.02
.4	.1	0.02	0.01	0.23
	.2	0.13	-0.02	0.20
	.3	0.57	-0.05	0.17
	.5	0.96	-0.14	0.08
.5	.1	0.00	0.04	0.26
	.2	0.02	0.01	0.23
	.3	0.19	-0.02	0.20
	.4	0.96	-0.08	0.14

Table B.20: Significance values for comparing different threshold values for NVC super-class detection system for recall.

Threshold Values		p-value	95% Confidence Interval	
			Lower Bound	Upper Bound
.1	.2	0.01	0.02	0.21
	.3	0.00	0.09	0.28
	.4	0.00	0.18	0.37
	.5	0.00	0.32	0.51
.2	.1	0.01	-0.21	-0.02
	.3	0.27	-0.03	0.16
	.4	0.00	0.07	0.25
	.5	0.00	0.21	0.39
.3	.1	0.00	-0.28	-0.09
	.2	0.27	-0.16	0.03
	.4	0.06	0.00	0.19
	.5	0.00	0.14	0.32
.4	.1	0.00	-0.37	-0.18
	.2	0.00	-0.25	-0.07
	.3	0.06	-0.19	0.00
	.5	0.00	0.05	0.23
.5	.1	0.00	-0.51	-0.32
	.2	0.00	-0.39	-0.21
	.3	0.00	-0.32	-0.14
	.4	0.00	-0.23	-0.05

Table B.21: Significance values for comparing different threshold values for NVC super-class detection system for F1 using only results obtained with FFN architecture.

Threshold Values		p-value	95% Confidence Interval	
			Lower Bound	Upper Bound
0.1	0.2	0.72	-0.04	0.02
	0.3	0.03	-0.06	0.00
	0.4	0.00	-0.10	-0.03
	0.5	0.00	-0.14	-0.08
0.2	0.1	0.72	-0.02	0.04
	0.3	0.42	-0.05	0.01
	0.4	0.00	-0.08	-0.02
	0.5	0.00	-0.12	-0.06
0.3	0.1	0.03	0.00	0.06
	0.2	0.42	-0.01	0.05
	0.4	0.03	-0.06	0.00
	0.5	0.00	-0.10	-0.04
0.4	0.1	0.00	0.03	0.10
	0.2	0.00	0.02	0.08
	0.3	0.03	0.00	0.06
	0.5	0.00	-0.07	-0.01
0.5	0.1	0.00	0.08	0.14
	0.2	0.00	0.06	0.12
	0.3	0.00	0.04	0.10
	0.4	0.00	0.01	0.07

Table B.22: Significance values for comparing different threshold values for NVC super-class detection system for percentage of data removed using only results obtained with LSTM architecture.

Threshold Values		p-value	95% Confidence Interval	
			Lower Bound	Upper Bound
0.1	0.2	0.05	-0.42	0.00
	0.3	0.00	-0.50	-0.08
	0.4	0.00	-0.54	-0.12
	0.5	0.00	-0.50	-0.09
0.2	0.1	0.05	0.00	0.42
	0.3	0.83	-0.29	0.13
	0.4	0.52	-0.33	0.09
	0.5	0.80	-0.29	0.13
0.3	0.1	0.00	0.08	0.50
	0.2	0.83	-0.13	0.29
	0.4	0.98	-0.25	0.17
	0.5	1.00	-0.21	0.20
0.4	0.1	0.00	0.12	0.54
	0.2	0.52	-0.09	0.33
	0.3	0.98	-0.17	0.25
	0.5	0.99	-0.17	0.24
0.5	0.1	0.00	0.09	0.50
	0.2	0.80	-0.13	0.29
	0.3	1.00	-0.20	0.21
	0.4	0.99	-0.24	0.17

Table B.23: Significance values for comparing different threshold values for NVC super-class detection system for percentage of data removed using only results obtained with FFN architecture.

Threshold Values		p-value	95% Confidence Interval	
			Lower Bound	Upper Bound
0.1	0.2	0.00	-0.16	-0.04
	0.3	0.00	-0.30	-0.18
	0.4	0.00	-0.52	-0.40
	0.5	0.00	-0.77	-0.65
0.2	0.1	0.00	0.04	0.16
	0.3	0.00	-0.20	-0.08
	0.4	0.00	-0.42	-0.30
	0.5	0.00	-0.67	-0.55
0.3	0.1	0.00	0.18	0.30
	0.2	0.00	0.08	0.20
	0.4	0.00	-0.28	-0.16
	0.5	0.00	-0.53	-0.41
0.4	0.1	0.00	0.40	0.52
	0.2	0.00	0.30	0.42
	0.3	0.00	0.16	0.28
	0.5	0.00	-0.31	-0.19
0.5	0.1	0.00	0.65	0.77
	0.2	0.00	0.55	0.67
	0.3	0.00	0.41	0.53
	0.4	0.00	0.19	0.31

Table B.24: Significance values for LSTM vs FFN performance on Back-channel cue detection.

Level	Metric	t statistic	p-value
Frame	Precision	20.75	<0.0001
	Recall	9.27	0.0045
	F1	33.18	<0.0001
	AUC	108.02	<0.0001
Event	Precision	33.79	<0.0001
	Recall	35.63	<0.0001
	F1	42.84	<0.0001

Table B.25: Significance values for LSTM vs FFN performance on Filler cue detection.

Level	Metric	t statistic	p-value
Frame	Precision	20.75	<0.0001
	Recall	9.27	0.0045
	F1	33.18	<0.0001
	AUC	108.02	<0.0001
Event	Precision	33.79	<0.0001
	Recall	35.63	<0.0001
	F1	42.84	<0.0001

Table B.26: Significance values for LSTM vs FFN performance on Laughter cue detection.

Level	Metric	t statistic	p-value
Frame	Precision	0.11	0.74
	Recall	15.06	0.0005
	F1	8.42	0.0065
	AUC	22.4	<0.0001
Event	Precision	0.84	0.37
	Recall	25.48	<0.0001
	F1	31.03	<0.0001

Table B.27: Significance values for each metric comparing performance at Back Channel detection between baseline and two-stage detector using individual detector.

Level	Metric	t statistic	p-value
Frame	Precision	20.66	<0.0001
	Recall	68.05	<0.0001
	F1	17.42	0.0002
	AUC	1.02	0.32
Event	Precision	0.12	0.73
	Recall	0.043	0.84
	F1	0.093	0.76

Table B.28: Significance values for each metric comparing performance at filler detection between baseline and two-stage detector using individual detector.

Level	Metric	t statistic	p-value
Frame	Precision	1.16	0.28
	Recall	5.8	0.022
	F1	0.18	0.68
	AUC	0.004	0.95
Event	Precision	0.32	0.58
	Recall	0.51	0.48
	F1	0.38	0.54

Table B.29: Significance values for each metric comparing performance at laugh detection between baseline and two-stage detector using individual detector.

Level	Metric	t statistic	p-value
Frame	Precision	28.05	<0.0001
	Recall	652.07	<0.0001
	F1	135.14	<0.0001
	AUC	782.09	<0.0001
Event	Precision	2.84	0.1
	Recall	124.13	<0.0001
	F1	33.99	<0.0001

Table B.30: Significance values for each metric comparing performance at back channel detection by architecture for the paralanguage distinguisher.

Level	Metric	t statistic	p-value
Frame	Precision	24.98	<0.0001
	Recall	0.083	0.78
	F1	16.15	0.0003
	AUC	70.73	<0.0001
Event	Precision	26.35	<0.0001
	Recall	36.36	0.074
	F1	32.3	<0.0001

Table B.31: Significance values for each metric comparing performance at filler detection by architecture for the paralanguage distinguisher.

Level	Metric	t statistic	p-value
Frame	Precision	5.15	0.03
	Recall	1.82	0.19
	F1	7.25	0.011
	AUC	36.19	0.0001
Event	Precision	20.43	0.0001
	Recall	3.14	0.085
	F1	19.4	0.0001

Table B.32: Significance values for each metric comparing performance at laughter detection by architecture for the paralanguage distinguisher.

Level	Metric	t statistic	p-value
Frame	Precision	0.32	0.57
	Recall	16.41	0.0003
	F1	5.15	<0.0001
	AUC	72.15	<0.0001
Event	Precision	4.11	0.051
	Recall	1.13	0.03
	F1	6.2	0.018

Table B.33: Significance values for each metric comparing the macro average across all three paralinguistic cues detection performance by architecture.

Level	Metric	t statistic	p-value
Frame	Precision	0.0021	0.96
	Recall	2.96	0.094
	F1	16.17	0.0003
	AUC	58.83	<0.0001
Event	Precision	13.41	0.0008
	Recall	3.26	0.08
	F1	17.52	0.0002

Table B.34: Significance values for each metric comparing performance at back channel detection between baseline and two-stage detector using paralanguage distinguisher.

Level	Metric	t statistic	p-value
Frame	Precision	30.45	<0.0001
	Recall	566.21	<0.0001
	F1	10.16	0.0031
	AUC	0.55	0.46
Event	Precision	2.054	0.16
	Recall	18.065	0.0002
	F1	0.39	0.54

Table B.35: Significance values for each metric comparing performance at filler detection between baseline and two-stage detector using paralanguage distinguisher.

Level	Metric	t statistic	p-value
Frame	Precision	0.92	0.34
	Recall	65.92	<0.0001
	F1	0.028	0.87
	AUC	630.82	<0.0001
Event	Precision	77.47	<0.0001
	Recall	174.99	<0.0001
	F1	106.45	<0.0001

Table B.36: Significance values for each metric comparing performance at laughter detection between baseline and two-stage detector using paralanguage distinguisher.

Level	Metric	t statistic	p-value
Frame	Precision	0.21	0.65
	Recall	32.96	<0.0001
	F1	1.57	0.22
	AUC	189.93	<0.0001
Event	Precision	0.88	0.36
	Recall	374.56	<0.0001
	F1	5.48	0.025

Appendix C

Significance Statistics Tables for Chapter 6

Table C.1: Significance values for each model in relation to the baseline detector (Left) and best performing detector (right) for frame level AUC Performance on the SMC using transformer based detectors.

Model Type	Confidence Interval		p-value	Confidence Interval		p-value
	Lower	Upper		Lower	Upper	
Baseline	x	x	x	15.13	18.79	<0.0001
Wav2Vec S	13.78	17.44	<0.0001	-3.18	0.48	0.3
Wav2Vec L	15.13	18.79	<0.0001	x	x	x
HuBERT S	13.39	17.05	<0.0001	-0.093	3.57	0.075
HuBERT L	14.73	18.39	<0.0001	-1.43	2.23	0.99
Whisper S	12.79	16.45	<0.0001	0.51	4.17	<0.0001
Whisper L	11.47	15.13	<0.0001	1.83	5.49	<0.0001

Table C.2: Significance values for each model in relation to the baseline detector (Left) and best performing detector (right) for frame level Recall Performance on the SMC using transformer based detectors.

Model Type	Confidence Interval		p-value	Confidence Interval		p-value
	Lower	Upper		Lower	Upper	
Baseline	x	x	x	-1.36	19	0.14
Wav2Vec S	-16.14	4.22	0.58	4.6	24.96	0.0006
Wav2Vec L	-1.36	19	0.14	x	x	x
HuBERT S	0.16	1.58	0.16	7.24	27.6	<0.0001
HuBERT L	-2.34	18.02	0.25	-9.2	11.16	0.99
Whisper S	-5.5	14.86	0.81	-14.32	6.04	0.89
Whisper L	-7.61	12.75	0.99	-16.43	3.93	0.52

Table C.3: Significance values for each model in relation to the baseline detector (Left) and best performing detector (right) for frame level Precision Performance on the SMC using transformer based detectors.

Model Type	Confidence Interval		p-value	Confidence Interval		p-value
	Lower	Upper		Lower	Upper	
Baseline	x	x	x	63.34	80.48	<0.0001
Wav2Vec S	56.04	73.18	<0.0001	-1.27	15.87	0.8
Wav2Vec L	59.36	76.5	<0.0001	-4.59	12.55	0.8
HuBERT S	55.18	72.32	<0.0001	-0.41	16.73	0.073
HuBERT L	51.79	68.93	<0.0001	2.98	20.12	0.0018
Whisper S	63.29	80.43	<0.0001	-8.62	8.52	1
Whisper L	63.34	80.48	<0.0001	x	x	x

Table C.4: Significance values for each model in relation to the baseline detector (Left) and best performing detector (right) for frame level F1 Performance on the SMC using transformer based detectors.

Model Type	Confidence Interval		p-value	Confidence Interval		p-value
	Lower	Upper		Lower	Upper	
Baseline	x	x	x	40.5	52.28	<0.0001
Wav2Vec S	27.73	39.51	<0.0001	6.88	18.66	<0.0001
Wav2Vec L	40.5	52.28	<0.0001	x	x	x
HuBERT S	25.07	36.85	<0.0001	9.54	21.32	<0.0001
HuBERT L	35.52	47.3	<0.0001	-0.91	10.87	0.16
Whisper S	38.99	50.77	<0.0001	-7.4	4.38	0.99
Whisper L	36.93	48.71	<0.0001	-9.46	2.32	0.53

Table C.5: Significance values for each model in relation to the baseline detector (Left) and best performing detector (right) for event level precision Performance on the SMC using transformer based detectors.

Model Type	Confidence Interval		p-value	Confidence Interval		p-value
	Lower	Upper		Lower	Upper	
Baseline	x	x	x	54.22	77.38	<0.0001
Wav2Vec S	27.57	50.73	<0.0001	15.07	38.23	<0.0001
Wav2Vec L	31.48	54.64	<0.0001	11.16	34.32	<0.0001
HuBERT S	23.73	46.89	<0.0001	18.91	42.07	<0.0001
HuBERT L	27.88	51.04	<0.0001	14.76	37.92	<0.0001
Whisper S	33.86	57.02	<0.0001	8.78	31.94	<0.0001
Whisper L	54.22	77.38	<0.0001	x	x	x

Table C.6: Significance values for each model in relation to the baseline detector (Left) and best performing detector (right) for event level recall Performance on the SMC using transformer based detectors.

Model Type	Confidence Interval		p-value	Confidence Interval		p-value
	Lower	Upper		Lower	Upper	
Baseline	x	x	x	31.91	50.37	<0.0001
Wav2Vec S	28.14	46.6	<0.0001	-13	5.46	0.88
Wav2Vec L	31.91	50.37	<0.0001	x	x	x
HuBERT S	29.11	47.57	<0.0001	-6.43	12.03	0.97
HuBERT L	27.34	45.8	<0.0001	-4.666	13.8	0.75
Whisper S	30.8	49.26	<0.0001	-10.34	8.12	0.96
Whisper L	26.95	45.41	<0.0001	-14.92	4.27	0.68

Table C.7: Significance values for each model in relation to the baseline detector (Left) and best performing detector (right) for event level F1 Performance on the SMC using transformer based detectors.

Model Type	Confidence Interval		p-value	Confidence Interval		p-value
	Lower	Upper		Lower	Upper	
Baseline	x	x	x	50.23	68.41	<0.0001
Wav2Vec S	33.38	51.56	<0.0001	7.76	25.94	<0.0001
Wav2Vec L	37.4	55.58	<0.0001	3.74	21.92	0.0088
HuBERT S	30.99	49.17	<0.0001	10.15	28.33	<0.0001
HuBERT L	32.29	50.47	<0.0001	8.85	27.03	<0.0001
Whisper S	39.41	57.59	<0.0001	1.73	19.91	0.0091
Whisper L	50.23	68.41	<0.0001	x	x	x

Table C.8: Significance values for the effect of each post-processing method in relation to the Whisper L detector for AUC Performance on the SMC. All models used the FFN base architecture. CW: class weight. D: delta. S: smoothing. U: undersampling. C: confidence-based alteration. E: feature vector extension.

Detector	Confidence Interval		p-value
	Lower	Upper	
Whisper L	x	x	x
Whisper L+CW	1.95	1.97	1
Whisper L+D	6.69	10.61	<0.0001
Whisper L+S	2.21	6.13	<0.0001
Whisper L+CW+D	4.17	8.09	<0.0001
Whisper L+CW+S	2.24	6.16	<0.0001
Whisper L+D+S	1.05	2.87	0.95
Whisper L+CW+D+S	0.08	5.05	0.032
Whisper L+U	9.5	13.42	<0.0001
Whisper L+U+S	-1.06	2.86	0.95
Whisper L+C	2.23	6.15	<0.0001
Whisper L+C+S	1.13	5.05	<0.0001
Whisper L+E	1.96	1.96	1
Whisper L+E+S	2.02	5.94	<0.0001

Table C.9: Significance values comparing the best overall score for Whisper L detectors with post-processing. All models used the FFN base architecture. CW: class weight. D: delta. S: smoothing. C: confidence-based alteration. E: feature vector extension.

Detector	Confidence Interval		p-value
	Lower	Upper	
Whisper L+S	-1.93	1.99	1
Whisper L+CW+S	x	x	x
Whisper L+CW+D+S	0.08	4	0.032
Whisper L+C+S	0.2	4.12	0.016
Whisper L+E+S	-1.74	2.18	1

Table C.10: Significance values for the effect of each post-processing method in relation to the Whisper L detector for frame level precision performance on the SMC. All models used the FFN base architecture. CW: class weight. D: delta. S: smoothing. U: undersampling. C: confidence-based alteration. E: feature vector extension.

Detector	Confidence Interval		p-value
	Lower	Upper	
Whisper L	x	x	x
Whisper L+CW	-6.16	27.74	0.66
Whisper L+D	4.57	38.47	0.002
Whisper L+S	-8.64	25.26	0.93
Whisper L+CW+D	6.27	40.17	0.0005
Whisper L+CW+S	-13.86	20.04	1
Whisper L+D+S	49.73	83.63	<0.0001
Whisper L+CW+D+S	-6.36	27.54	0.69
Whisper L+U	-16.05	17.85	1
Whisper L+U+S	-1.06	2.86	0.95
Whisper L+C	-14.8	19.1	1
Whisper L+C+S	-8.04	25.86	0.88
Whisper L+E	-16.28	17.62	1
Whisper L+E+S	-9.45	24.45	0.97

Table C.11: Significance values for the effect of each post-processing method in relation to the Whisper L detector for frame level recall performance on the SMC. All models used the FFN base architecture. CW: class weight. D: delta. S: smoothing. U: undersampling. C: confidence-based alteration. E: feature vector extension.

Detector	Confidence Interval		p-value
	Lower	Upper	
Whisper L	x	x	x
Whisper L+CW	1.61	18.49	0.0055
Whisper L+D	35.95	52.83	<0.0001
Whisper L+S	-4.09	12.79	0.9
Whisper L+CW+D	16.17	33.05	<0.0001
Whisper L+CW+S	-1.77	15.11	0.3
Whisper L+D+S	37.72	54.6	<0.0001
Whisper L+CW+D+S	26.97	43.85	<0.0001
Whisper L+U	-5.08	11.8	0.99
Whisper L+U+S	-0.4	16.48	0.079
Whisper L+C	-4.65	12.23	0.96
Whisper L+C+S	-0.43	16.45	0.82
Whisper L+E	-7.47	9.4	1
Whisper L+E+S	-5.4	11.48	0.99

Table C.12: Significance values for the effect of each post-processing method in relation to the Whisper L detector for frame level F1 performance on the SMC. All models used the FFN base architecture. CW: class weight. D: delta. S: smoothing. U: undersampling. C: confidence-based alteration. E: feature vector extension.

Detector	Confidence Interval		p-value
	Lower	Upper	
Whisper L	x	x	x
Whisper L+CW	-3.98	11.76	0.92
Whisper L+D	47.55	63.29	<0.0001
Whisper L+S	-5.87	9.87	1
Whisper L+CW+D	21.33	37.07	<0.0001
Whisper L+CW+S	-1.64	14.1	0.3
Whisper L+D+S	50.97	66.71	<0.0001
Whisper L+CW+D+S	32.79	48.53	<0.0001
Whisper L+U	-4.77	10.97	0.99
Whisper L+U+S	-2.22	13.52	0.46
Whisper L+C	-5.11	10.63	1
Whisper L+C+S	-2.44	13.3	0.53
Whisper L+E	-7.32	8.42	1
Whisper L+E+S	-6.83	8.91	1

Table C.13: Significance values for the effect of each post-processing method in relation to the Whisper L detector for event level precision performance on the SMC. All models used the FFN base architecture. CW: class weight. D: delta. S: smoothing. U: undersampling. C: confidence-based alteration. E: feature vector extension.

Detector	Confidence Interval		p-value
	Lower	Upper	
Whisper L	x	x	x
Whisper L+CW	-4.66	23	0.6
Whisper L+D	11.47	39.13	<0.0001
Whisper L+S	-2.84	24.82	0.29
Whisper L+CW+D	15.31	42.97	<0.0001
Whisper L+CW+S	-7.69	19.97	0.97
Whisper L+D+S	-8.36	19.3	0.99
Whisper L+CW+D+S	-12.53	15.13	1
Whisper L+U	-11.98	15.68	1
Whisper L+U+S	-2.69	24.97	0.27
Whisper L+C	-11.43	16.23	1
Whisper L+C+S	-2.29	25.37	0.22
Whisper L+E	-12.82	14.84	1
Whisper L+E+S	-2.94	24.72	0.3

Table C.14: Significance values for the effect of each post-processing method in relation to the Whisper L detector for event level recall performance on the SMC. All models used the FFN base architecture. CW: class weight. D: delta. S: smoothing. U: undersampling. C: confidence-based alteration. E: feature vector extension.

Detector	Confidence Interval		p-value
	Lower	Upper	
Whisper L	x	x	x
Whisper L+CW	-6.24	17.58	0.94
Whisper L+D	61.05	84.87	<0.0001
Whisper L+S	15.5	39.32	<0.0001
Whisper L+CW+D	9.49	33.31	<0.0001
Whisper L+CW+S	5.91	29.73	<0.0001
Whisper L+D+S	67.14	90.96	<0.0001
Whisper L+CW+D+S	32.87	56.69	<0.0001
Whisper L+U	-6.54	17.28	0.96
Whisper L+U+S	17.54	41.36	<0.0001
Whisper L+C	-8.8	15.02	1
Whisper L+C+S	18.88	42.7	<0.0001
Whisper L+E	-11.5	12.32	1
Whisper L+E+S	14.91	38.74	<0.0001

Table C.15: Significance values for the effect of each post-processing method in relation to the Whisper L detector for event level F1 performance on the SMC. All models used the FFN base architecture. CW: class weight. D: delta. S: smoothing. U: undersampling. C: confidence-based alteration. E: feature vector extension.

Detector	Confidence Interval		p-value
	Lower	Upper	
Whisper L	x	x	x
Whisper L+CW	-7.6	13.14	1
Whisper L+D	53.48	74.22	<0.0001
Whisper L+S	2.205	22.99	0.004
Whisper L+CW+D	23.1	43.84	<0.0001
Whisper L+CW+S	-2.95	17.79	0.47
Whisper L+D+S	62.54	83.28	<0.0001
Whisper L+CW+D+S	22.57	43.31	<0.0001
Whisper L+U	-8.46	12.28	0.96
Whisper L+U+S	3.83	24.57	0.0005
Whisper L+C	-9.94	10.8	1
Whisper L+C+S	4.84	25.58	0.0001
Whisper L+E	-10	10.74	1
Whisper L+E+S	1.85	22.59	0.0065

Table C.16: Significance values for t-tests comparing Whisper L laughter detection by role.

Level	Metric	t statistic	p-value
Frame	AUC	4.42	<0.0001
	Precision	2.3	0.022
	Recall	3.48	0.0006
	F1	3.66	0.0003
Event	Precision	2.39	0.017
	Recall	1.44	0.15
	F1	1.98	0.048

Table C.17: Significance values for t-tests comparing Whisper L laughter detection by Gender.

Level	Metric	t statistic	p-value
Frame	AUC	0.19	0.85
	Precision	3.96	<0.0001
	Recall	1.63	0.11
	F1	2.44	0.015
Event	Precision	3.98	<0.0001
	Recall	4.45	<0.0001
	F1	4.59	<0.0001

Table C.18: Significance values for post hoc Tukey tests comparing Whisper L laughter detection performance by gender pairing of a conversation.

		MM-MF			MM-FF			FF-MF		
Level		Confidence Interval			Confidence Interval			Confidence Interval		
	Metric	Lower	Upper	p-value	Lower	Upper	p-value	Lower	Upper	p-value
Frame	AUC	No difference								
	Precision	7.83	21.29	<0.0001	5.11	20.03	0.0003	-4.32	8.3	0.74
	Recall	No difference								
	F1	0.3	13.06	0.038	-0.11	14.05	0.055	-6.28	5.7	0.99
Event	Precision	7.14	21	<0.0001	4.37	19.75	0.0008	-4.5	8.52	0.75
	Recall	1.02	14.98	0.02	1.85	17.33	0.011	-5.03	7.55	0.74
	F1	5.21	17.83	0.0001	4.42	18.42	0.0005	-5.83	6.027	0.99

Table C.19: Significance values for post hoc Tukey tests comparing different filtering approaches with the Hamming window.

		Hamming vs MinMax			Hamming vs Median		
Level	Metric	Confidence Interval		p value	Confidence Interval		p value
		Lower	Upper		Lower	Upper	
Frame	Precision	22.29	30.87	<0.0001	-2.71	5.87	0.77
	Recall	24.91	38.13	<0.0001	-3.09	10.13	0.5
	F1	6.68	16.4	<0.0001	-2.01	7.71	0.42
Event	Precision	9.2	19.76	<0.0001	-4.22	6.34	0.95
	Recall	7.54	18.3	<0.0001	0.46	11.22	0.028
	F1	1.13	9.34	0.005	0.21	-0.92	6.28

Table C.20: Significance values comparing Whisper L with Baseline performance on the SVC dataset.

Level	Metric	t statistic	p-value
Frame	AUC	22.83	<0.0001
	Precision	37.13	<0.0001
	Recall	10.82	0.0023
	F1	24.06	<0.0001
Event	Precision	7.78	0.0086
	Recall	54.79	<0.0001
	F1	22.57	<0.0001

Table C.21: Significance values comparing Whisper L with Baseline performance on filler detection.

Level	Metric	t statistic	p-value
Frame	AUC	2950.16	<0.0001
	Precision	8492.59	<0.0001
	Recall	72.72	<0.0001
	F1	3672.79	<0.0001
Event	Precision	9069.92	<0.0001
	Recall	1332.86	<0.0001
	F1	6959.51	<0.0001

Table C.22: Significance values comparing Whisper L with Baseline performance on back channel detection.

Level	Metric	t statistic	p-value
Frame	AUC	2950.16	<0.0001
	Precision	642.54	<0.0001
	Recall	1036.31	<0.0001
	F1	256.81	<0.0001
Event	Precision	446.36	<0.0001
	Recall	34.08	<0.0001
	F1	633.4	<0.0001

Table C.23: Significance values for the effect of each post-processing method in relation to the Whisper L detector for frame level precision Performance on back channel detection in the SMC. All models used the FFN base architecture. CW: class weight. D: delta. S: smoothing.

Detector	Confidence Interval		p-value
	Lower	Upper	
Whisper L	x	x	x
Whisper L+CW	20.1	33.51	<0.0001
Whisper L+D	-5.96	7.34	1
Whisper L+S	46.12	59.42	<0.0001
Whisper L+CW+D	25.5	38.8	<0.0001
Whisper L+CW+S	2.39	15.69	0.0013
Whisper L+D+S	7.1	20.4	<0.0001
Whisper L+CW+D+S	12.48	25.78	<0.0001

Table C.24: Significance values for the effect of each post-processing method in relation to the Whisper L detector for frame level recall Performance on back channel detection in the SMC. All models used the FFN base architecture. CW: class weight. D: delta. S: smoothing.

Detector	Confidence Interval		p-value
	Lower	Upper	
Whisper L	x	x	x
Whisper L+CW	14.74	26.14	<0.0001
Whisper L+D	-0.49	10.91	0.1
Whisper L+S	55.1	66.5	<0.0001
Whisper L+CW+D	35.41	46.82	<0.0001
Whisper L+CW+S	10.56	21.96	<0.0001
Whisper L+D+S	-4.59	6.81	1
Whisper L+CW+D+S	36.68	48.08	<0.0001

Table C.25: Significance values for the effect of each post-processing method in relation to the Whisper L detector for frame level F1 Performance on back channel detection in the SMC. All models used the FFN base architecture. CW: class weight. D: delta. S: smoothing.

Detector	Confidence Interval		p-value
	Lower	Upper	
Whisper L	x	x	x
Whisper L+CW	2.45	12.03	0.0002
Whisper L+D	1.2	10.78	0.0043
Whisper L+S	15.13	24.71	<0.0001
Whisper L+CW+D	3.64	13.22	<0.0001
Whisper L+CW+S	8.43	18.01	<0.0001
Whisper L+D+S	-2.35	7.22	0.77
Whisper L+CW+D+S	15.32	24.9	<0.0001

Table C.26: Significance values for the effect of each post-processing method in relation to the Whisper L detector for frame level AUC Performance on back channel detection in the SMC. All models used the FFN base architecture. CW: class weight. D: delta. S: smoothing.

Detector	Confidence Interval		p-value
	Lower	Upper	
Whisper L	x	x	x
Whisper L+CW	-1.49	3.03	0.97
Whisper L+D	0.35	4.87	0.012
Whisper L+S	6.84	11.36	<0.0001
Whisper L+CW+D	3.51	8.03	<0.0001
Whisper L+CW+S	2.86	7.38	<0.0001
Whisper L+D+S	5.23	9.75	<0.0001
Whisper L+CW+D+S	7.05	11.57	<0.0001

Table C.27: Significance values for the effect of each post-processing method in relation to the Whisper L detector for event level precision Performance on back channel detection in the SMC. All models used the FFN base architecture. CW: class weight. D: delta. S: smoothing.

Detector	Confidence Interval		p-value
	Lower	Upper	
Whisper L	x	x	x
Whisper L+CW	22.96	35.06	<0.0001
Whisper L+D	-5.77	6.33	1
Whisper L+S	35.41	42.86	<0.0001
Whisper L+CW+D	28.57	40.67	<0.0001
Whisper L+CW+S	9	21.1	<0.0001
Whisper L+D+S	10.77	22.87	<0.0001
Whisper L+CW+D+S	16	28.1	<0.0001

Table C.28: Significance values for the effect of each post-processing method in relation to the Whisper L detector for event level recall Performance on back channel detection in the SMC. All models used the FFN base architecture. CW: class weight. D: delta. S: smoothing.

Detector	Confidence Interval		p-value
	Lower	Upper	
Whisper L	x	x	x
Whisper L+CW	16.47	31.43	<0.0001
Whisper L+D	1.38	16.34	0.0088
Whisper L+S	5.64	20.6	<0.0001
Whisper L+CW+D	19.3	34.26	<0.0001
Whisper L+CW+S	5.39	20.35	<0.0001
Whisper L+D+S	2.56	17.52	0.0016
Whisper L+CW+D+S	18.14	33.1	<0.0001

Table C.29: Significance values for the effect of each post-processing method in relation to the Whisper L detector for event level F1 Performance on back channel detection in the SMC. All models used the FFN base architecture. CW: class weight. D: delta. S: smoothing.

Detector	Confidence Interval		p-value
	Lower	Upper	
Whisper L	x	x	x
Whisper L+CW	13.01	23.81	<0.0001
Whisper L+D	-1.43	9.37	0.32
Whisper L+S	35.69	46.49	<0.0001
Whisper L+CW+D	20.46	31.26	<0.0001
Whisper L+CW+S	0.095	10.89	<0.0001
Whisper L+D+S	-5.28	5.52	1
Whisper L+CW+D+S	4.47	15.27	<0.0001

Bibliography

[1] B. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, K. R. Scherer, F. Ringeval, M. Chetouani, F. Weninger, F. Eyben, E. Marchi, M. Mortillaro, H. Salamin, A. Polychroniou, F. Valente, and S. Kim, "The INTERSPEECH 2013 Computational Paralinguistics Challenge: Social Signals, Conflict, Emotion, Autism," in *Proceedings of Interspeech*, 2013, pp. 148–152.

[2] B. B. Dash and K. Davis, "Significance of nonverbal communication and paralinguistic features in communication: A critical analysis," *International Journal for Innovative Research in Multidisciplinary Field*, vol. 8, no. 4, pp. 172–179, 2022.

[3] J. Vettin and D. Todt, "Laughter in conversation: Features of occurrence and acoustic structure," *Journal of Nonverbal Behavior*, vol. 28, no. 2, pp. 93–115, 2004.

[4] J. Raclaw and C. E. Ford, "Laughter and the management of divergent positions in peer review interactions," *Journal of pragmatics*, vol. 113, pp. 1–15, 2017.

[5] H. Kangasharju and T. Nikko, "Emotions in organizations: Joint laughter in workplace meetings," *The Journal of Business Communication*, vol. 46, no. 1, pp. 100–119, 2009.

[6] J. Navarro, R. Del Moral, M. Alonso, P. Loste, J. Garcia-Campayo, R. Lahoz-Beltra, and P. Marijuán, "Validation of laughter for diagnosis and evaluation of depression," *Journal of affective disorders*, vol. 160, pp. 43–49, 2014.

[7] J. Navarro, M. Fernández Rosell, A. Castellanos, R. Del Moral, R. Lahoz-Beltra, and P. C. Marijuán, "Plausibility of a neural network classifier-based neuroprosthesis for depression detection via laughter records," *Frontiers in neuroscience*, vol. 13, p. 267, 2019.

[8] W. J. Hudenko, W. Stone, and J.-A. Bachorowski, "Laughter differs in children with autism: An acoustic analysis of laughs produced by children with and without the disorder," *Journal of autism and developmental disorders*, vol. 39, pp. 1392–1400, 2009.

[9] F. d. A. A. Gondim, F. P. Thomas, S. Cruz-Flores, H. Nasrallah, and J. B. Selhorst, "Pathological laughter and crying: a case series and proposal for a new classification," *Annals of Clinical Psychiatry*, vol. 28, no. 1, pp. 11–21, 2016.

[10] M. Terriza, J. Navarro, I. Retuerta, N. Alfageme, R. San-Segundo, G. Kontaxakis, E. Garcia-Martin, P. C. Marijuan, and F. Panetsos, "Use of laughter for the detection of parkinson's disease: Feasibility study for clinical decision support systems, based on speech recognition and automatic classification techniques," *International journal of environmental research and public health*, vol. 19, no. 17, p. 10884, 2022.

[11] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. MüLler, and S. Narayanan, "Paralinguistics in speech and language—state-of-the-art and the challenge," *Computer Speech & Language*, vol. 27, no. 1, pp. 4–39, 2013.

[12] B. Schuller, S. Steidl, A. Batliner, P. B. Marschik, H. Baumeister, F. Dong, S. Hantke, F. B. Pokorny, E.-M. Rathner, K. D. Bartl-Pokorny, C. Einspieler, D. Zhang, A. Baird, S. Amiriparian, K. Qian, Z. Ren, M. Schmitt, P. Tzirakis, and S. Zafeiriou, "The interspeech 2018 computational paralinguistics challenge: Atypical self-assessed affect, crying heart beats," in *Proceedings of Interspeech*, 2018, pp. 122–126.

[13] S. Petridis and M. Pantic, "Audiovisual discrimination between speech and laughter: Why and when visual information might help," *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 216–234, 2010.

[14] J.-A. Bachorowski, M. J. Smoski, and M. J. Owren, "The acoustic features of human laughter," *The journal of the Acoustical Society of America*, vol. 110, no. 3, pp. 1581–1597, 2001.

[15] M. J. Owren, M. Philipp, E. Vanman, N. Trivedi, A. Schulman, and J.-A. Bachorowski, "Understanding spontaneous human laughter: The role of voicing in inducing positive emotion," *Evolution of emotional communication: From sounds in nonhuman mammals to speech and music in man*, pp. 176–190, 2013.

[16] J.-A. Bachorowski and M. J. Owren, "Not all laughs are alike: Voiced but not unvoiced laughter readily elicits positive affect," *Psychological science*, vol. 12, no. 3, pp. 252–257, 2001.

[17] H. Tanaka and N. Campbell, "Classification of social laughter in natural conversational speech," *Computer Speech & Language*, vol. 28, no. 1, pp. 314–325, 2014.

[18] J. Dennis, T. Dat, and E. Chng, "Analysis of spectrogram image methods for sound event classification," in *Proceedings of Interspeech*, 2014, pp. 2533–2537.

[19] J. Gillick, W. Deng, K. Ryokai, and D. Bamman, "Robust laughter detection in noisy environments." in *Proceedings of Interspeech*, 2021, pp. 2481–2485.

[20] L. Dong, S. Xu, and B. Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *Proceedings of ICASSP*, 2018, pp. 5884–5888.

[21] J. Huang, J. Tao, B. Liu, Z. Lian, and M. Niu, "Multimodal transformer fusion for continuous emotion recognition," in *Proceedings of ICASSP*, 2020, pp. 3507–3511.

[22] Y. Yu, D. Park, and H. K. Kim, "Auxiliary loss of transformer with residual connection for end-to-end speaker diarization," in *Proceedings of ICASSP*, 2022, pp. 8377–8381.

[23] H. V. Koay, J. H. Chuah, and C.-O. Chow, "Convolutional neural network or vision transformer? benchmarking various machine learning models for distracted driver detection," in *Proceedings of TENCON*, 2021, pp. 417–422.

[24] M. L. Hutchinson, E. Antono, B. M. Gibbons, S. Paradiso, J. Ling, and B. Meredig, "Overcoming data scarcity with transfer learning," *arXiv:1711.05099*, 2017.

[25] K. Laskowski, "Contrasting emotion-bearing laughter types in multiparticipant vocal activity detection for meetings," in *Proceedings of ICASSP*, 2009, pp. 4765–4768.

[26] G. Gosztolya, B. András, T. Neuberger, and T. László, "Laughter classification using deep rectifier neural networks with a minimal feature subset," *Archives of Acoustics*, vol. 41, no. 4, pp. 669–682, 2016.

[27] L. Kaushik, A. Sangwan, and J. H. Hansen, "Laughter and filler detection in naturalistic audio," in *Proceedings of Interspeech*, 2015.

[28] G. Gosztolya, A. Beke, and T. Neuberger, "Differentiating laughter types via hmm/dnn and probabilistic sampling," in *Proceedings of SPECOM*, 2019, pp. 122–132.

[29] T. F. Krikke and K. P. Truong, "Detection of nonverbal vocalizations using gaussian mixture models: looking for fillers and laughter in conversational speech." in *Proceedings of Interspeech*, 2013, pp. 163–167.

[30] R. Cai, L. Lu, H.-J. Zhang, and L.-H. Cai, "Highlight sound effects detection in audio stream," in *Proceedings of ICME*, vol. 3, 2003, pp. III–37.

[31] S. Petridis and M. Pantic, "Fusion of audio and visual cues for laughter detection," in *Proceedings of CIVR*, 2008, pp. 329–338.

[32] ——, "Audiovisual laughter detection based on temporal features," in *Proceedings of ICMI*, 2008, pp. 37–44.

[33] K. Laskowski and T. Schultz, "Detection of laughter-in-interaction in multichannel close-talk microphone recordings of meetings," in *Proceedings of MLMI*, 2008, pp. 149–160.

[34] S. Scherer, F. Schwenker, N. Campbell, and G. Palm, "Multimodal laughter detection in natural discourses," *Human centered robot systems: Cognition, interaction, technology*, pp. 111–120, 2009.

[35] M. Glodek, S. Scherer, and F. Schwenker, "Conditioned hidden markov model fusion for multimodal classification," in *Proceedings of Interspeech*, 2011, pp. 2269–2272.

[36] S. Scherer, M. Glodek, F. Schwenker, N. Campbell, and G. Palm, "Spotting laughter in natural multiparty conversations: A comparison of automatic online and offline approaches using audiovisual data," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 2, no. 1, pp. 1–31, 2012.

[37] S. Pammi, H. Khemiri, and G. Chollet, "Laughter detection using alisp-based n-gram models," in *ISCA workshop on laughter and other non-verbal vocalisations*, 2012, pp. 16–17.

[38] R. Brueckner and B. Schuller, "Hierarchical neural networks and enhanced class posteriors for social signal classification," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 362–367.

[39] T. Neuberger and A. Beke, "Automatic laughter detection in spontaneous speech using gmm–svm method," in *Proceedings of TSD*, 2013, pp. 113–120.

[40] S. Petridis, B. Martinez, and M. Pantic, "The mahnob laughter database," *Image and Vision Computing*, vol. 31, no. 2, pp. 186–202, 2013.

[41] H. Salamin, A. Polychroniou, and A. Vinciarelli, "Automatic detection of laughter and fillers in spontaneous mobile phone conversations," in *International Conference on Systems, Man, and Cybernetics*, 2013, pp. 4282–4287.

[42] R. Gupta, K. Audhkhasi, S. Lee, and S. S. Narayanan, "Paralinguistic event detection from speech using probabilistic time-series smoothing and masking." in *Proceedings of Interspeech*, 2013, pp. 173–177.

[43] R. Brueckner and B. Schulter, "Social signal classification using deep blstm recurrent neural networks," in *Proceedings of ICASSP*, 2014, pp. 4823–4827.

[44] H. Rao, Z. Ye, Y. Li, M. A. Clements, A. Rozga, and J. M. Rehg, "Combining acoustic and visual features to detect laughter in adults' speech." in *Proceedings of AVSP*, 2015, pp. 153–156.

[45] M. Tahon, M. A. Sehili, and L. Devillers, "Cross-corpus experiments on laughter and emotion detection in hri with elderly people," in *Proceedings of International Conference on Social Robotics*, 2015, pp. 633–642.

[46] F. Yang, M. A. Sehili, C. Barras, and L. Devillers, "Smile and laughter detection for elderly people-robot interaction," in *Proceedings of ICSR*, 2015, pp. 694–703.

[47] S. Bedoya and T. H. Falk, "Laughter detection based on the fusion of local binary patterns, spectral and prosodic features," in *Proceedings of MMSP*, 2016, pp. 1–5.

[48] R. Gupta, K. Audhkhasi, S. Lee, and S. Narayanan, "Detecting paralinguistic events in audio stream using context in features and probabilistic decisions," *Computer speech & language*, vol. 36, pp. 72–92, 2016.

[49] B. B. Turker, Y. Yemez, T. M. Sezgin, and E. Erzin, "Audio-facial laughter detection in naturalistic dyadic conversations," *Transactions on Affective Computing*, vol. 8, no. 4, pp. 534–545, 2017.

[50] G. Hagerer, N. Cummins, F. Eyben, and B. W. Schuller, "Did you laugh enough today?"-deep neural networks for mobile and wearable laughter trackers." in *Proceedings of Interspeech*, 2017, pp. 2044–2045.

[51] H. Inaguma, K. Inoue, M. Mimura, and T. Kawahara, "Social signal detection in spontaneous dialogue using bidirectional lstm-ctc." in *Proceedings of Interspeech*, 2017, pp. 1691–1695.

[52] H. Bohy, K. El Haddad, and T. Dutoit, "A new perspective on smiling and laughter detection: Intensity levels matter," in *Proceedings of ACII*, 2022, pp. 1–8.

[53] P. Tzirakis, A. Baird, J. Brooks, C. Gagne, L. Kim, M. Opara, C. Gregory, J. Metrick, G. Boseck, V. Tiruvadi *et al.*, "Large-scale nonverbal vocalization detection using transformers," in *Proceedings of ICASSP*, 2023, pp. 1–5.

[54] N. Lavan, S. K. Scott, and C. McGettigan, "Laugh like you mean it: Authenticity modulates acoustic, physiological and perceptual properties of laughter," *Journal of Nonverbal Behavior*, vol. 40, pp. 133–149, 2016.

[55] A. Vinciarelli, P. Chatziioannou, and A. Esposito, "When the words are not everything: The use of laughter, fillers, back-channel, silence, and overlapping speech in phone calls," *Frontiers in ICT*, vol. 2, 2015.

[56] R. J. Van Son, D. Binnenpoorte, H. Heuvel, and L. Pols, "The IFA corpus: a phonemically segmented dutch "open source" speech database," *Proceedings of Eurospeech*, 2001.

[57] J. J. Godfrey and E. Holliman, "Switchboard-1 release 2," *Linguistic Data Consortium, Philadelphia*, vol. 926, p. 927, 1997.

[58] F. Ataollahi and M. T. Suarez, "Laughter classification using 3D convolutional neural networks," in *Proceedings of ICAAI*, 2019, pp. 47–51.

[59] M. Tanaka and N. Sumitsuji, "Electromyographic study of facial expressions during pathological laughing and crying." *Electromyography and clinical neurophysiology*, vol. 31, no. 7, pp. 399–406, 1991.

[60] J. Volkema, G. Roger and H. Ronald, "The influence of cognitive-based group composition on decision-making process and outcome," *Journal of Management Studies*, vol. 35, no. 1, pp. 105–121, 1998.

[61] R. A. Fisher, "Methods for research workers," 1950.

[62] D. Lala, K. Inoue, and T. Kawahara, "Prediction of shared laughter for human-robot dialogue," in *Proceedings of ICMI*, 2020, pp. 62–66.

[63] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke *et al.*, "The ICSI meeting corpus," in *Proceedings of ICASSP*, vol. 1, 2003, pp. I–I.

[64] R. Lenain, J. Weston, A. Shivkumar, and E. Fristed, "Surfboard: Audio feature extraction for modern machine learning," *arXiv preprint arXiv:2005.08848*, 2020.

[65] L. Muda, M. Begam, and I. Elamvazuthi, "Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques," *arXiv preprint arXiv:1003.4083*, 2010.

[66] S. Tirronen, S. R. Kadiri, and P. Alku, "The effect of the MFCC frame length in automatic voice pathology detection," *Journal of Voice*, 2022.

[67] S. Alghowinem, R. Goecke, M. Wagner, J. Epps, M. Breakspear, and G. Parker, "From joyous to clinically depressed: Mood detection using spontaneous speech." in *Proceedings of FLAIRS-25*, 2012.

[68] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[69] S. Skansi, *Introduction to Deep Learning: from logical calculus to artificial intelligence*. Springer, 2018.

[70] H. Phan, M. Krawczyk-Becker, T. Gerkmann, and A. Mertins, "Weighted and multi-task loss for rare audio event detection," in *Proceedings of ICASSP*, 2018, pp. 336–340.

[71] G. Gosztolya, "Optimizing class priors to improve the detection of social signals in audio data," *Engineering Applications of Artificial Intelligence*, vol. 107, p. 104541, 2022.

[72] R. Gupta, K. Audhkhasi, S. Lee, and S. Narayanan, "Detecting paralinguistic events in audio stream using context in features and probabilistic decisions," *Computer speech & language*, vol. 36, pp. 72–92, 2016.

[73] L. S. Kennedy and D. P. Ellis, "Laughter detection in meetings," in *Proceedings of NIST*, 2004.

[74] M. T. Knox and N. Mirghafori, "Automatic laughter detection using neural networks." in *Proceedings of Interspeech*, 2007, pp. 2973–2976.

[75] M. T. Knox, N. Morgan, and N. Mirghafori, "Getting the last laugh: Automatic laughter segmentation in meetings," in *Proceedings of ISCA*, 2008.

[76] K. Ryokai, E. Duran Lopez, N. Howell, J. Gillick, and D. Bamman, "Capturing, representing, and interacting with laughter," in *Proceedings of CHI*, 2018, pp. 1–12.

[77] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[78] G. Rennie, O. Perepelkina, and A. Vinciarelli, "Which Model is Best: Comparing Methods and Metrics for Automatic Laughter Detection in a Naturalistic Conversational Dataset," in *Proceedings of Interspeech*, 2022, pp. 4008–4012.

[79] S. Wehrli, C. Hertweck, M. Amirian, S. Glüge, and T. Stadelmann, "Bias, awareness, and ignorance in deep-learning-based face recognition," *AI and Ethics*, vol. 2, no. 3, pp. 509–522, 2022.

[80] B. F. Klare, M. J. Burge, J. C. Klontz, R. W. V. Bruegge, and A. K. Jain, "Face recognition performance: Role of demographic information," *IEEE Transactions on information forensics and security*, vol. 7, no. 6, pp. 1789–1801, 2012.

[81] M. Atay, H. Gipson, T. Gwyn, and K. Roy, "Evaluation of gender bias in facial recognition with traditional machine learning algorithms," in *Proceedings of SSCI*, 2021, pp. 1–7.

[82] K. Ghosh, C. Bellinger, R. Corizzo, P. Branco, B. Krawczyk, and N. Japkowicz, "The class imbalance problem in deep learning," *Machine Learning*, pp. 1–57, 2022.

[83] N. V. Shmyrev and other contributors, "Vosk speech recognition toolkit: Offline speech recognition api for android, ios, raspberry pi and servers with python, java, c and node." https://github.com/alphacep/vosk-api, 2020.

[84] S. Team, "Silero VAD: pre-trained enterprise-grade voice activity detector (VAD), number detector and language classifier," https://github.com/snakers4/silero-vad, 2021.

[85] B. Zellner, "Pauses and the temporal structure of speech," in *E. Keller Fundamentals of speech synthesis and speech recognition*. John Wiley, 1994, pp. 41–62.

[86] V. L. Smith and H. H. Clark, "On the course of answering questions," *Journal of memory and language*, vol. 32, no. 1, pp. 25–38, 1993.

[87] G. Kjellmer, "Hesitation. in defence of er and erm," *English Studies*, vol. 84, no. 2, pp. 170–198, 2003.

[88] M. Swerts, "Filled pauses as markers of discourse structure," *Journal of pragmatics*, vol. 30, no. 4, pp. 485–496, 1998.

[89] R. Bertrand, G. Ferré, P. Blache, R. Espesser, and S. Rauzy, "Backchannels revisited from a multimodal perspective," in *Auditory-visual Speech Processing*, 2007, pp. 1–5.

[90] C. D. Manning, *Introduction to information retrieval*. Syngress Publishing,, 2008.

[91] A. Mohamed, D. Okhonko, and L. Zettlemoyer, "Transformers with convolutional context for ASR," *arXiv preprint arXiv:1904.11660*, 2019.

[92] E. Tsunoo, Y. Kashiwagi, T. Kumakura, and S. Watanabe, "Transformer ASR with contextual block processing," in *Proceedings of ASRU*, 2019, pp. 427–433.

[93] N. Kanda, G. Ye, Y. Gaur, X. Wang, Z. Meng, Z. Chen, and T. Yoshioka, "End-to-end speaker-attributed ASR with transformer," *arXiv preprint arXiv:2104.02128*, 2021.

[94] D. Zhu and D. Wang, "Transformers and their application to medical image processing: A review," *Journal of Radiation Research and Applied Sciences*, p. 100680, 2023.

[95] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," in *Proceedings of CVF*, 2021, pp. 12 299–12 310.

[96] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM computing surveys (CSUR)*, vol. 54, no. 10s, pp. 1–41, 2022.

[97] D. Shome, T. Kar, S. N. Mohanty, P. Tiwari, K. Muhammad, A. AlTameem, Y. Zhang, and A. K. J. Saudagar, "Covid-transformer: Interpretable covid-19 detection using vision transformer for healthcare," *International Journal of Environmental Research and Public Health*, vol. 18, no. 21, p. 11086, 2021.

[98] K. K. Podder, S. Tabassum, L. E. Khan, K. M. A. Salam, R. I. Maruf, and A. Ahmed, "Design of a sign language transformer to enable the participation of persons with disabilities in remote healthcare systems for ensuring universal healthcare coverage," in *Proceedings of TEMSCON-EUR*, 2021, pp. 1–6.

[99] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, "A comparative study on transformer vs rnn in speech applications," in *Proceedings of ASRU*. IEEE, 2019, pp. 449–456.

[100] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of NIPS*, 2017, p. 6000–6010.

[101] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5929–5955, 2020.

[102] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *Transactions on audio, speech, and language processing*, vol. 29, pp. 3451–3460, 2021.

[103] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[104] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proceedings of ICML*, 2023, pp. 28 492–28 518.

[105] I. Pitas and A. N. Venetsanopoulos, "Median filters," in *Nonlinear Digital Filters: Principles and Applications*, 1990, pp. 63–116.

[106] C.-C. Chang, J.-Y. Hsiao, and C.-P. Hsieh, "An adaptive median filter for image denoising," in *Proceedings of Second international symposium on intelligent information technology application*, vol. 2, 2008, pp. 346–350.

[107] G. Gupta *et al.*, "Algorithm for image processing using improved median filter and comparison of mean, median and improved median filter," *Proceedings of IJSCE*, vol. 1, no. 5, pp. 304–311, 2011.

[108] B. Justusson, "Median filtering: Statistical properties," *Two-dimensional digital signal prcessing II: transforms and median filters*, pp. 161–196, 2006.

[109] R. Provine, "Laughter punctuates speech: Linguistic, social and gender contexts of laughter," *Ethology*, vol. 95, no. 4, pp. 291–298, 1993.