

IAMDIVA
INFERENTIAL AFFECTIVE MOOD DETECTION FOR
INTERACTIVE VIRTUAL ASSISTANTS

By

Weston Nelson

July, 2026

Director of Thesis: Nic Herndon, PhD

Major Department: Computer Science

ABSTRACT

The widespread adoption of conversational AI assistants has revealed an emerging challenge: their static, uniformly cheerful delivery can clash with users' moods, potentially worsening negative emotional states during routine interactions. While substantial research has focused on making AI assistants more empathetic and emotionally supportive, less attention has been paid to the simpler, but crucial, goal of preventing these assistants from inadvertently compounding a user's existing distress during routine daily tasks.

This research develops a modular, real-time mood detection system for conversational virtual assistants, combining vocal landmark analysis, voice fundamental frequency (F0) analysis, and text-based sentiment analysis to classify a user's emotional state from speech. The three detection methodologies and their fusion are implemented as components that can each be used independently or with a response modulation module that has been reserved for later work. The system is built to operate within the latency and hardware constraints of mobile and IoT voice assistant platforms, and is evaluated independently of any specific assistant integration. The resulting mood detection module, and the accuracy and latency benchmarks established for it, are intended to provide a replicable foundation for response modulation research that respects users' emotional states rather than inadvertently amplifying their discomfort.

IAMDIVA
INFERENTIAL AFFECTIVE MOOD DETECTION FOR
INTERACTIVE VIRTUAL ASSISTANTS

A Thesis

Presented to The Faculty of the Department of Computer Science
East Carolina University

In Partial Fulfillment of the Requirements for the Degree
Master of Science in Data Science

By

Weston Nelson

July, 2026

Director of Thesis: Nic Herndon, PhD

Thesis Committee Members:

Qin Ding, PhD

David Hart, PhD

©Weston Nelson, 2026

ACKNOWLEDGEMENTS

I would like to thank my father, Chris Nelson, for his support throughout my life and academic journey. None of this would have been possible without him. I would also like to thank my mother, friends, and support system, as well as the faculty of the East Carolina University Department of Computer Science. Jennifer Andriot, Alex Vilkomir, Qin Ding, Nic Herndon, and several others have been trusted advisors to me and guided me through many important decisions.

Table of Contents

List of Tables	viii
List of Figures	ix
Chapter 1: Introduction	1
1.1 Topic Definition	1
1.2 Research Area	1
1.3 Motivation & Importance	2
1.4 Problem Statement	4
1.5 Objectives	5
Chapter 2: Related Work	6
2.1 Empathy in Virtual Assistants	6
2.2 How Interactive Virtual Assistants Work	7
2.3 Acoustic Features & Analysis	10
2.4 Emotion Representation Frameworks	15
2.5 Technical Advances	16
2.6 Technical Challenges	17
2.7 Research Gaps	20
2.8 User Experience	21
2.9 Conclusion	22
Chapter 3: Methodology	24
3.1 System Architecture & Design	24
3.2 Algorithms & Theoretical Frameworks	27
3.3 Feature Selection & Classifier Comparison Experiments	39

3.4	Data Collection & Preparation	41
3.5	Experimental Setup & Environment	44
3.6	Evaluation Metrics, Testing, and Baselines	46
Chapter 4: Results and Discussion		48
4.1	Individual Detector Results	48
4.2	Fusion Results	53
4.3	Feature Vector Selection	59
4.4	Latency Analysis	68
4.5	Discussion	69
Chapter 5: Conclusion and Future Work		73
5.1	Summary of the Research	73
5.2	Conclusions	73
5.3	Future Work	74
5.4	Closing Statement	77
Bibliography		79

List of Tables

3.1	Source Dataset Sizes	42
3.2	Primary Software Dependencies by Pipeline Stage	45
4.1	Vocal Landmark Detector Performance Statistics	49
4.2	Tone (F0) Detector Performance Statistics	50
4.3	Sentiment Detector Performance Statistics	52
4.4	Rule-Based Fusion Performance Statistics	54
4.5	Stacked Fusion Performance Statistics	54
4.6	Bimodal Stacked Fusion Performance Statistics	56

List of Figures

4.1	Vocal Landmark Detector Recall Heatmap	49
4.2	Tone (F0) Detector Recall Heatmap	51
4.3	Sentiment Detector Recall Heatmap	52
4.4	Rule-Based Fusion Recall Heatmap	54
4.5	Stacked Fusion Recall Heatmap	55
4.6	Bimodal Stacked Fusion Recall Heatmap	56
4.7	Overall Accuracy by System	57
4.8	Overall Latency by System	58
4.9	Phase 1 Vocal Landmark Classifier Comparison	60
4.10	Phase 1 Tone (F0) Classifier Comparison	60
4.11	Phase 3 Vocal Landmark Grouping Scheme Comparison	65
4.12	Phase 3 Tone (F0) Grouping Scheme Comparison	65

Chapter 1

Introduction

1.1 Topic Definition

This research develops IAMDIVA, a modular, real-time mood detection system for conversational virtual assistants on mobile and Internet of Things (IoT) devices. The system combines three independent methodologies – acoustic vocal landmark analysis, voice fundamental frequency (F0) analysis, and text-based sentiment analysis – to classify the emotional valence of user speech into one of three states (angry, sad, or neutral), with the goal of signaling when an assistant’s default cheerful tone should give way to a more emotionally appropriate response. The mood detection module developed here is intentionally separated from a response modulation module, which is left for future research. This work concerns only the detection of emotional state, not the generation or adjustment of assistant responses.

1.2 Research Area

The convergence of Speech Emotion Recognition (SER) and conversational Artificial Intelligence (AI) represents a rapidly evolving field with significant implications for human-computer interaction [1]. Voice assistants currently dominate the consumer technology landscape, with market valuations reaching approximately \$6-7 billion in 2024 and projected explosive growth as high as \$79 billion by 2034 across various industry reports [2, 3]. However, these systems predominantly maintain uniform, optimistic response patterns regardless of user emotional context. This can create potential misalignment between a user’s emotional needs and the system’s responses [3, 4].

The majority of industry research in emotion detection is focused in the areas of Healthcare and Customer Service [1]. Mental health researchers are utilizing SER in combination

with Computer Vision to detect depression in psychiatric patients [5]. Where most health-care research is concerned in this area, work is mainly focusing on depression detection. For customer service, many companies are guiding their research toward detecting negative customer moods in phone or chatbot conversations and finding ways for the digital assistant to enhance the mood of the customer [6]. This research is generally aimed at putting customers in a better mood by dissolving their anger and/or frustration with the situation. This area of research is typically spearheaded by corporations because happy customers are better for business.

Most current research in emotional artificial intelligence falls into the following two categories [1]:

- Emotion detection systems that classify user emotions but don't connect this to response generation.
- Empathetic response systems designed specifically for emotional support conversations.

What's missing is the middle ground: Systems that detect emotional states during everyday interactions and make subtle adjustments to their responses to prevent the interaction from worsening mood.

The architectures used to extract and classify emotion from these features have evolved substantially in recent years, moving from handcrafted statistical descriptors toward learned representations such as Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) systems, and self-supervised speech encoders. These advances are reviewed in detail in Section 2.5.

1.3 Motivation & Importance

The motivation for this research stems from observed limitations in current virtual assistant emotional intelligence. Existing systems often respond with blind cheerfulness to users experiencing distress, frustration, or grief – potentially alienating users during emotionally

vulnerable moments [4]. This represents both a user experience problem and an untapped opportunity for creating more empathetic human-computer interactions.

From an industry perspective, emotion-aware virtual assistants could significantly enhance user engagement and satisfaction across multiple domains including healthcare support, customer service, educational applications, and mental health monitoring [3]. The global emphasis on mental health awareness, particularly following the COVID-19 pandemic, has increased demand for technology that can sensitively respond to emotional contexts. There are also positive implications for cellular device hardware and software companies, as users could associate their products with positive or neutral emotions, rather than anger and frustration. This has the potential to influence sales of these products.

This research addresses critical gaps in real time emotion processing while creating a foundation for appropriate emotional response generation – areas of substantial commercial and academic interest. Companies investing heavily in conversational AI – including Amazon, Google, Microsoft, and Apple – would benefit from advances in contextually appropriate emotional intelligence that respects user autonomy while providing supportive interaction experiences [3].

Arguments for research need include:

- **Prevention of emotional escalation:** Research shows that inappropriate responses can create “disgust” and interrupt user interactions entirely [7].
- **Trust maintenance:** Emotional appropriateness is essential for maintaining user trust, which directly impacts continued usage [8].
- **User satisfaction:** Studies demonstrate that emotional mismatches lead to user dissatisfaction and negative perceptions of virtual assistant performance [4].
- **Practical business impact:** The research on empathic voice assistants suggests this is an active area of commercial interest because current systems lack appropriate emotional responsiveness [3].

1.4 Problem Statement

Users of conversational virtual assistants such as Siri and Alexa interact with these systems across the full range of human emotional experience – including moments of stress, grief, frustration, and exhaustion – yet these systems are architecturally blind to emotional context [4]. They produce responses calibrated for a neutral or positive user regardless of the affective state the user is actually in. When a user experiencing distress receives a cheerful, upbeat response to a routine request, the mismatch is not merely aesthetically unpleasant – it can be perceived as dismissive, deepen frustration, and cause the user to abandon the interaction or disengage from the assistant altogether.

The core problem is an absence of a lightweight, privacy-preserving mechanism that can detect the emotional valence of a user’s speech in real time and use that signal to modulate the tone, not the content, of an assistant’s response. Existing emotion recognition research either produces systems that are too computationally heavy for edge deployment, relies on visual input that is generally unavailable in typical voice assistant usage contexts, targets clinical or customer service applications rather than everyday conversational interaction, or detects emotion without connecting the detection to any form of response adjustment. No existing system addresses the everyday, non-clinical use case: a conversational assistant that can recognize when a user is in a negative emotional state and respond in a tonally appropriate way – not attempting to fix the user’s mood, but avoiding the compounding effect of emotional mismatch.

This research addresses that gap by developing a multimodal, real-time mood detection system grounded in acoustic vocal landmarks, voice fundamental frequency, and text-based sentiment analysis, designed to operate within the latency and hardware constraints of mobile and IoT voice assistant platforms.

1.5 Objectives

This research accomplishes the following:

- **Develops a multimodal, real-time mood detection tool** that integrates acoustic vocal landmarks, F0, and text-based sentiment analysis to classify the emotional valence of user speech, with end-to-end latency compatible with conversational assistant usage (less than 100 ms per utterance).
- **Validates vocal landmarks as a viable feature representation for general mood detection**, extending their established use in depression detection to the broader challenge of real-time emotional state classification across positive, negative, and neutral affective states.
- **Evaluates system performance and establishes baseline performance benchmarks** for the combined vocal landmark, F0, and sentiment feature fusion approach, providing a replicable foundation for future research extending this work to additional languages, demographic groups, or assistant platforms.

Chapter 2

Related Work

The field of emotion detection in virtual assistants and chatbots represents a crucial intersection of affective computing, natural language processing, and human-computer interaction. As [9] observes, virtual assistants have revolutionized the digital landscape by seamlessly integrating into our lives through smartphones, smart home devices, and various online platforms. As conversational AI systems become increasingly integrated into daily life, we see the emergence of a critical limitation: their inability to evoke human emotions during interactions. This creates a gap between technological capabilities and user experience expectations. Throughout this work, the focus will be on conversational virtual assistants that are voice activated.

2.1 Empathy in Virtual Assistants

Virtual Assistants (VAs), particularly Intelligent Virtual Assistants (IVAs), have evolved from simple command-execution systems to sophisticated conversational agents capable of managing complex interactions. Modern voice assistants like Amazon’s Alexa, Apple’s Siri, and Google Assistant represent the mainstream implementation of this technology, serving as the interface through which users interact with AI systems via voice commands [8].

The concept of empathy in voice assistants has gained considerable attention as researchers recognize that emotionally responsive systems can significantly enhance user experience. [2] defines empathy in conversational agents as “the ability to relate and respond to someone else’s emotional state,” which consists of emotional communication competence, emotion regulation, and cognitive mechanisms resulting in a spectrum of appropriate behaviors.

Research distinguishes between two primary forms of empathy [2]:

- Other-oriented empathy - driven by altruistic motives and care for others, and
- Self-oriented empathy - relates to one’s own performance and emotions.

Studies have demonstrated that users express strong support for empathetic virtual assistants [2]. Approximately 79% of survey participants indicated interest in using mood-improving VAs in the future, suggesting significant market demand for emotionally intelligent systems. The benefits of empathetic voice assistants extend to multiple domains, including mood elevation, communication training, education, emotional intelligence development, companionship, and storytelling applications. However, researchers emphasize that such systems should complement rather than replace human interaction.

2.2 How Interactive Virtual Assistants Work

2.2.1 Activation

Voice-activated virtual assistants remain in a low-power listening state until they detect a predefined *trigger word* (also called a wake word), such as “Hey Siri,” “Alexa,” or “OK Google” [8]. The device continuously monitors incoming audio using a lightweight on-device model dedicated to recognizing this keyword, designed to reduce computational overhead and preserve battery life while remaining sensitive enough to reliably distinguish the trigger phrase from ambient speech and background noise. Upon detection of the wake word, the assistant transitions to an active listening state and begins capturing the user’s full spoken utterance for subsequent processing.

2.2.2 Automatic Speech Recognition

Once activated, the captured audio signal is passed to an Automatic Speech Recognition (ASR) module, which transcribes the spoken input into text. A typical spoken dialogue system follows a cascaded pipeline structure in which the ASR module serves as the first stage,

converting spoken language into a textual representation that downstream components can process [10]. Traditional ASR systems rely on acoustic models, pronunciation dictionaries, and language models to decode the most probable word sequence from the acoustic signal. More recent approaches leverage end-to-end deep learning architectures, including models pre-trained through self-supervised learning on large speech corpora such as WavLM and HuBERT, which jointly encode both linguistic content and paralinguistic information from the raw waveform [10]. The accuracy of the ASR transcription is critical, as any errors at this stage propagate through the rest of the pipeline. Research has shown that emotional states can affect the acoustic characteristics of speech in ways that challenge conventional ASR models, with recognition errors increasing under conditions of high valence, high activation, or low dominance [11].

2.2.3 Natural Language Understanding

The textual output of the ASR module is passed to a Natural Language Understanding (NLU) component, which is responsible for determining the user’s intent and extracting relevant entities or parameters from the transcribed utterance. NLU typically involves two core tasks: *intent classification*, which identifies what the user wants to accomplish (e.g., playing music, setting a reminder, or asking a question), and *entity extraction*, which identifies specific pieces of information within the request (e.g., a song title, a time, or a location). Modern NLU systems are often implemented using large pre-trained language models fine-tuned on task-specific datasets, enabling robust understanding across diverse phrasings [8]. In emotionally-aware systems, this stage may also incorporate sentiment or emotion classification, analyzing the lexical content of the utterance to infer the user’s emotional state in complement to the acoustic features extracted earlier in the pipeline [11].

2.2.4 Execution

Once the user’s intent has been identified, a *dialogue manager* orchestrates the system’s response, maintaining the state of the conversation across turns, resolving ambiguities, and determining the most appropriate action to take. Depending on the intent, this may involve querying a knowledge base, calling an external API (e.g., retrieving weather data or playing a media item), controlling a connected device via the IoT, or managing calendar and email services [8]. In more sophisticated systems, the dialogue manager may also track emotional context across conversation turns, informing downstream response generation with information about the user’s inferred affective state [12]. End-to-end architectures that jointly optimize understanding and response generation represent an emerging paradigm, with models combining WavLM-based speech encoding with Large Language Models (LLMs) such as LLaMA, enabling seamless integration of emotion-aware processing within the execution stage [10].

2.2.5 Response

The final stage of the pipeline involves generating a natural-sounding response and delivering it to the user. This stage typically comprises two components [10]:

- *Natural Language Generation (NLG)*, which produces the textual content of the response, and
- *Text-to-Speech (TTS) synthesis*, which converts the generated text into spoken audio.

NLG systems – increasingly implemented using large pre-trained language models that are fine-tuned on dialogue data – aim to produce responses that are contextually relevant, coherent, and conversationally appropriate. In empathetically responsive systems, the NLG component may additionally modulate the emotional tone of the response content based on the user’s detected emotional state [12]. The TTS module then renders this content as speech, with prosodic parameters – including pitch, speech rate, and volume – adjusted via

Speech Synthesis Markup Language (SSML) to convey the intended emotional tone. For example, a calm and reassuring response may employ a reduced speech rate and lower pitch, while an upbeat response uses elevated pitch and increased volume [2]. Together, the NLG and TTS stages determine both what the assistant says and how it sounds, making them critical points of intervention for systems aimed at emotional appropriateness in human-VA interaction.

2.3 Acoustic Features & Analysis

2.3.1 Mel-Frequency Cepstral Coefficients

The standard representation for virtual voice analysis is the Mel-Frequency Cepstral Coefficient (MFCC). MFCCs were originally developed at MIT in the late 1960s for seismic audio analysis and later adapted to model human vocal-tract characteristics [5]. They capture changes in vocal-tract spectral energy by mapping the power spectrum of a speech signal onto the Mel scale – a perceptual frequency scale that approximates human auditory response. The extraction pipeline divides the audio into short overlapping frames (typically 20–25 ms with a 10 ms step), computes a Fourier transform for each frame, filters the spectrum through a bank of triangular Mel-spaced filters, takes the logarithm of the filter-bank energies, and finally applies a Discrete Cosine Transform to obtain the cepstral coefficients. Usually 13 coefficients are retained per frame, together with their first-order (delta) and second-order (delta-delta) derivatives, bringing the feature count to 39 for a typical MFCC feature vector.

MFCCs are highly discriminative for emotion recognition and are commonly combined with other acoustic parameters such as Linear Predictive Coding, Basilar Membrane Frequency Band Cepstral Coefficients, Human Factor Cepstral Coefficients, Perceptual Linear Prediction, formants (F1–F3), and signal energy [13]. Specific MFCC components have been linked to affective states. For instance, MFCC5 and MFCC7 have shown promise as behavioral indicators of depression, and the trajectory concentration of MFCC feature spaces varies

with depression severity [5].

While they are widely used and considered the standard for virtual voice representation, MFCCs do have limitations. Their continuous, high-dimensional nature can impose computational overhead in real-time systems and may not efficiently encode discrete, articulatory-level pronunciation patterns relevant to speaker variation and cross-cultural emotional expression [1]. Nonetheless, MFCCs remain a key baseline in SER research, where they are used with classifiers such as Support Vector Machines (SVMs), CNNs, LSTM networks, and Bidirectional LSTM architectures.

2.3.2 Voice Fundamental Frequency

The Fundamental Frequency of Voice (F0), often colloquially referred to as pitch or tone, is one of the most studied and foundational acoustic correlates of human emotion. Phonation is the production of vocal sounds [14] and F0 corresponds to the rate of vibration of the vocal folds during phonation. It is perceived by listeners as the “base tone” or melody of speech and has been recognized as a primary carrier of emotional information since early psychoacoustic work by Klaus Scherer and Rainer Banse in the 1980s and 1990s. F0 continues to appear prominently in virtually all contemporary speech emotion recognition feature sets. [13] demonstrated that F0 alone can provide sufficient information to identify emotional states by using various machine learning algorithms – including K -Nearest Neighbors, Random Forest, and Random Subspace methods – on the AGH Emotional Speech Corpus. In their research, the use of F0 and its derived features achieved classification accuracy as high as 89.74% for two-emotion discrimination and 76.14% for three-emotion sets, establishing it as a strong stand-alone feature with potential for further improvement when combined with complementary modalities.

Research has consistently demonstrated that distinct emotional states produce characteristic patterns in F0. Emotions associated with high arousal – such as anger, excitement, and fear – typically manifest as elevated F0 mean values, increased F0 range, and greater variabil-

ity across an utterance [5]. Conversely, low-arousal states such as sadness and depression are associated with reduced pitch variation, lower mean F0, longer inter-utterance pauses, and a slower overall speech rate. Depressed individuals in particular have been found to exhibit quiet, low-pitched, and monotonous speech, with a more concentrated distribution of acoustic features in MFCC space. In clinical studies, metrics such as jitter (cycle-to-cycle pitch perturbation) and shimmer (amplitude perturbation) have also been found to differentiate depressed from non-depressed speakers.

For time-varying analysis – which is essential in conversational contexts – Fast Fourier Transformation (FFT) based approaches construct a Fundamental Frequency Series by computing F0 over short, overlapping frames (typically 20 ms frames with 10 ms step), yielding a sequence of pitch estimates across the utterance [13]. Statistical functionals computed from this series – including mean, standard deviation, range, minimum, maximum, and slope – serve as compact, utterance-level descriptors for downstream classifiers. F0 can be extracted using several computational methods, including autocorrelation, zero-crossing rate, complex cepstrum analysis, and FFT. Importantly, F0 extraction is computationally inexpensive compared to many deep learning feature representations and is relatively robust to environmental noise, making it particularly well suited to deployment on edge devices such as smart speakers and mobile phones.

2.3.3 Vocal Landmarks

Vocal landmarks represent a relatively novel approach to speech analysis that focuses on discrete, abrupt acoustic changes at the sub-segmental level – changes that correspond to specific articulatory events during speech production [15]. The concept originates from research on distinctive phonetic features, with foundational contributions from [16] and [17], who proposed that listeners rely on these acoustic events to gather the cues necessary to decode the underlying phonetic structure of speech. Rather than representing speech as a continuous spectral envelope, vocal landmarks discretize the acoustic signal into a compact

set of binary symbols that carry direct linguistic and articulatory significance [15].

Six primary landmark types are recognized, each of which corresponds to a specific phonetic event [15]:

- **Glottal landmarks (g)**: Indicate the onset (+) or cessation (−) of vocal fold vibration.
- **Burst landmarks (b)**: Mark the onset (+) or offset (−) of turbulent noise, characteristic of obstruent consonants.
- **Syllabic landmarks (s)**: Capture the releases (+) or closures (−) of nasal sounds.
- **Periodicity landmarks (p)**: Mark the beginning (+) or end (−) of periodic vocal activity.
- **Frication landmarks (f)**: Capture the onset (+) or offset (−) of unvoiced fricative noise.
- **Voiced frication landmarks (v)**: Indicate the onset (+) or offset (−) of voiced fricative regions.

Detection is performed by dividing the spectrogram into six frequency bands and identifying significant energy changes within those bands using a two-pass strategy [15, 18]. Glottal, burst, and syllabic landmarks are determined primarily from low-frequency band energy changes, while frication and voiced frication landmarks require a 6 dB power increase within the three high-frequency bands (bands 4–6) combined with a decrease in low-frequency bands. Periodicity landmarks are identified via autocorrelation to detect repetitive patterns in the audio frame.

An important finding in the landmark literature is that the sequential patterns in which landmarks appear carry more diagnostic value than the individual landmarks themselves [15]. This characteristic makes landmarks particularly well-suited to integration with LLMs, as landmark sequences can be treated similarly to token streams in natural language processing. For this research, vocal landmarks offer three key advantages [15]:

1. Their binary, discrete nature dramatically reduces the computational footprint relative to continuous spectral features. This enables real-time processing of approximately 5–10 ms per 160 ms audio chunk.
2. They maintain a direct relationship with articulatory events – making them robust to hardware variation and environmental noise.
3. Their compatibility with language model architectures opens a pathway for integrating speech and text modalities without requiring visual input.

While their primary use in the literature has been in depression detection tasks, this research explores their application to the broader challenge of real-time mood classification in conversational contexts.

2.3.4 Sentiment Analysis

Sentiment analysis refers to the computational identification of subjective emotional tone from text, typically classifying content along a valence axis of positive, negative, or neutral polarity, though more granular emotion taxonomies are also employed. In the broader SER literature, sentiment analysis occupies a complementary role to acoustic feature extraction. Where acoustic features such as F0 and MFCCs are well-suited to capturing arousal and dominance dimensions of emotion, the valence dimension – the positive versus negative axis of emotional experience – has consistently proven to be the most difficult to decode from acoustic signals alone [19, 11]. Research has shown that valence is strongly correlated with the lexical content of speech, making text-based sentiment analysis a critical complement to acoustic-only approaches.

Modern sentiment analysis leverages transformer-based architectures pre-trained on large text corpora, most notably Bidirectional Encoder Representations from Transformers (BERT) and its derivatives (e.g., DistilBERT, RoBERTa, ALBERT, and more) [20]. RoBERTa-based models have demonstrated strong performance on sentiment classification

benchmarks – with accuracy exceeding 84% on sentiment analysis of product reviews – while offering practical inference speeds of approximately 20–30 ms per utterance, compatible with real-time system constraints.

The functional integration of sentiment analysis with speech emotion recognition proceeds through the ASR pipeline: user audio is transcribed to text in real time, and the resulting transcript is analyzed for sentiment polarity. Research by [11] using the Interactive Emotional Dyadic Motion Capture Database (IEMOCAP) corpus demonstrated a meaningful correlation between text sentiment and speech emotion labels. Utterances classified as negative in sentiment were predominantly associated with negative speech emotion labels (sadness, anger, frustration), while positive-sentiment utterances corresponded largely to positive emotion labels (happiness). This relationship motivates the use of sentiment as a “weak signal” that, while imperfect, provides complementary information to acoustic features – particularly for the valence dimension where acoustic cues are less reliable.

It is important to note that text-based sentiment does not capture the full complexity of emotional state. Cases exist where neutral semantic content is spoken with strong emotional tone – for instance, a brief response such as “Nothing” carries neutral sentiment but may convey frustration through acoustic delivery. This highlights why the multimodal integration of sentiment with vocal landmarks and F0 is necessary: each modality provides information the others do not, and their combination yields a more robust and accurate picture of the user’s emotional state than any single modality alone.

2.4 Emotion Representation Frameworks

Two primary frameworks dominate emotion representation in speech: discrete emotion categories and dimensional emotion models [11]. Discrete emotion frameworks, such as Ekman’s six basic emotions (happiness, sadness, anger, disgust, fear, and surprise), provide intuitive labels but suffer from inconsistencies in defining standard emotion sets. Cowen and Keltner’s more comprehensive framework identifies 27 distinct emotion categories bridged by

continuous gradients, offering greater granularity but increased complexity.

Dimensional emotion models, particularly the three-dimensional framework of valence, arousal, and dominance, provide an alternative representation where emotions are characterized as points within a continuous space [21]. Valence indicates the positiveness or negativeness of an emotion, arousal reflects the energy level in voice, and dominance specifies how strong or submissive one sounds [11]. Research has demonstrated that arousal and dominance are more readily detectable from acoustic features alone, while valence estimation proves more challenging and benefits significantly from lexical information.

2.5 Technical Advances

The evolution of deep learning has revolutionized speech emotion recognition. LSTM networks have emerged as the foundation for many emotion recognition systems due to their ability to capture temporal dependencies in speech signals. [21] demonstrated that single-layer LSTM networks with 128 neurons in recurrent and embedding layers can effectively detect primitive emotion attributes (valence, arousal, dominance) from speech, achieving a 30% improvement in concordance correlation coefficient for valence when trained with multiple datasets and robust features. Time-convolutional LSTM (TC-LSTM) architectures, which incorporate convolutional layers before LSTM processing with skip connections, have shown enhanced performance over standard LSTM approaches by capturing local temporal patterns before sequential dependencies are modeled [21].

CNNs have proven particularly effective when working with spectral representations of speech, excelling at extracting spatio-temporal characteristics from spectrogram features. Attention-based CNN models have demonstrated state-of-the-art performance on standard datasets [1]. CNN-LSTM combinations have achieved accuracies exceeding 98% on benchmark datasets like AMIGOS and DREAMER for emotion classification from physiological signals.

Self-supervised pretrained models represent a further advance. The WavLM model, pre-

trained on large-scale speech corpora, encodes both linguistic and paralinguistic features from its 6th transformer layer, transforming continuous speech features into discrete representations compatible with language models through techniques like k -means clustering [10]. The HuBERT model similarly learns robust speech representations through masked prediction of hidden units, proving effective for downstream emotion recognition tasks. Sentiment-aware ASR pretraining has also shown promise, with [11] demonstrating that ASR systems pretrained with sentiment awareness can improve speech emotion recognition by learning representations that better capture emotional content. Collectively, these self-supervised approaches reduce dependency on labeled emotion data, which is often expensive and difficult to obtain with high inter-annotator agreement.

Multi-modal fusion approaches that combine acoustic and lexical information have consistently outperformed unimodal systems, since acoustic features effectively capture arousal and dominance while lexical information is crucial for accurate valence estimation [11]. Hierarchical multimodal models that align lexical and acoustic representations at the word level have achieved state-of-the-art performance on benchmarks like IEMOCAP, and modality-based attention mechanisms such as the Gated Multimodal Unit (GMU) allow systems to dynamically prioritize acoustic or lexical features depending on which modality carries more emotional information for a given utterance. [22] demonstrated that GMU-based models combined with context-based attention achieve unweighted accuracies exceeding 73% on IEMOCAP.

2.6 Technical Challenges

Despite the aforementioned advances, several challenges constrain real-world application. One of those challenges concerns dataset quality and annotation. Early studies relied on acted or elicited emotions, where actors were recorded while speaking with specified emotions [11]. Models trained on such data often fail to generalize to spontaneous, subtle emotions encountered in natural conversation. The shift toward datasets with spontaneous emotion,

such as MSP-Podcast – which contains over 50,000 utterances of natural emotional speech from podcast recordings – represents progress on this front, though obtaining ground-truth labels for spontaneous speech remains difficult [23]. [21] found that concordance correlation coefficients (CCCs) for valence increase consistently with higher annotator agreement, and performance on MSP-Podcast data varied substantially depending on whether utterances were graded by 5 or 16 annotators.

A second, related challenge is domain mismatch and generalization. [21] demonstrated an approximately 40% reduction in CCCs when models trained on one dataset were evaluated on a mismatched dataset, though training on multiple diverse datasets followed by fine-tuning substantially improves cross-domain performance. This generalization gap extends across languages: training on linguistically diverse datasets (e.g., English and Mandarin) improves emotion recognition accuracy on unseen languages compared to English-only training, with models trained on multi-lingual data exhibiting lower mutual information between lexical content and predicted emotion when evaluated on Spanish language data (the MESD dataset), indicating better extraction of language-independent paralinguistic features. The dominant training datasets in SER – including IEMOCAP, RAVDESS, CREMA-D, and EmoDB – consist primarily of acted or elicited speech in which emotions are often exaggerated, and models trained on such data can fail to generalize to the subtle, mixed, and ambiguous emotions typical of spontaneous conversational speech [24]. State-of-the-art SER models evaluated on naturalistic conversational corpora such as Switchboard-Affect show variable and often poor performance across emotion categories, with particularly poor generalization for anger – underscoring that the gap between laboratory benchmarks and real-world deployment remains a persistent, unresolved problem for any system intended for use in virtual assistant contexts [25].

A third challenge is real-time processing. Real-time emotion recognition demands computational efficiency alongside accuracy. Traditional perturbation-based methods for attribute control, such as Plug-and-Play Language Models (PPLM), suffer from generation

times averaging 16.43 seconds per response, making them unsuitable for interactive applications, while more efficient frameworks such as Plug-and-Play with Rational Speech Act (PPRSA) reduce this to 7.07 seconds while maintaining or improving response quality [12]. Integrating emotion recognition into end-to-end spoken dialogue systems compounds this challenge, since audio capture, feature extraction, emotion classification, response generation, and speech synthesis must all complete within the 200-300ms latency budget typically allowed for natural conversation. Traditional cascaded architectures, in which these stages occur sequentially, struggle to meet this constraint, though [10] demonstrated that efficient architectures combining WavLM-based speech encoding with LLaMA-based language models can achieve real-time performance while maintaining emotion recognition accuracy above 90% on standard benchmarks. Streaming approaches that classify emotion incrementally as speech continues, rather than waiting for a complete utterance, offer one potential mitigation – though they introduce their own challenge of prediction stability since early segments may lack sufficient emotional information while later segments may contradict earlier predictions. Hardware acceleration and model optimization techniques (quantization, pruning, knowledge distillation) further help meet real-time constraints, but typically involve accuracy-latency tradeoffs whose appropriate operating point depends on the demands of the application.

A fourth, closely related challenge is valence detection specifically. Among the three primary emotion dimensions, valence is consistently the most difficult to infer from acoustic signals alone, since it correlates more strongly with lexical content than with acoustic features [19]. This means acoustic-only systems fundamentally underperform on the dimension most relevant to distinguishing positive from negative user states, and addressing it requires the multimodal fusion described above – itself a non-trivial real-time engineering challenge, since acoustic features and linguistic content operate at different temporal resolutions (10-20ms frames versus word- or utterance-level boundaries) and must be aligned, typically via forced alignment, before fusion [22]. This challenge extends to coordinating emotion recognition with the broader dialogue pipeline – ASR, NLU, dialogue management, response generation,

and speech synthesis – where each additional component introduces latency and potential error propagation. [10] demonstrated that end-to-end architectures jointly optimizing these components, such as WavLM-based emotion recognition combined with LLaMA-based language models, can achieve this integration while maintaining real-time performance.

2.7 Research Gaps

As demonstrated in Sections 2.5 and 2.6, contemporary research exhibits clear strengths alongside persistent limitations. While significant advances have been made in the field, several gaps still exist, leaving space for further research and exploration. First, while acoustic landmarks have shown promise in depression detection [15, 26], their application to broader emotional state recognition in conversational settings remains unexplored. The discrete nature of landmarks offers computational advantages, but their effectiveness for real-time mood detection across diverse emotional states has not been established.

Second, the integration of vocal landmarks with text-based sentiment analysis – without reliance on computer vision – represents an underexplored approach to multimodal emotion detection [11]. Computer vision has been prominent in this field, particularly in depression detection, but is deliberately not utilized in this research. Not all devices on which an IVA might run have cameras, and even when a camera is present, a user may be out of frame or in another room while speaking, meaning a system dependent on visual input would behave very differently in deployment than in a research setting. Camera-based input also carries a substantially larger memory and storage footprint, which conflicts with this research’s goal of a lightweight system.

Third, the field lacks clear frameworks for determining when and how virtual assistants should modulate emotional tone. [2] confirmed that users unanimously want assistants to default to neutral when prediction confidence is low rather than applying emotional modulation indiscriminately, framing the challenge not as emotion detection in isolation but as contextually appropriate and bounded response adjustment.

Finally, most SER systems classify individual utterances in isolation, without regard for the emotional trajectory of an ongoing conversation [27]. In practice a user’s emotional state evolves across a dialogue, and systems that ignore this temporal context risk responding to momentary acoustic artifacts rather than genuine emotional states.

2.8 User Experience

User perception of empathy in voice assistants depends on both verbal content and prosodic delivery. [2] found that participants could reliably distinguish empathetic from non-empathetic responses when both aspects were properly implemented, using a perceived empathy scale incorporating factors from standardized instruments such as the Consultation and Relational Empathy (CARE) measure and the Toronto Empathy Questionnaire (TEQ). Authenticity – “the feeling of genuineness, trueness, and realness as perceived by the customer or user” – and personalization, whether implicit or explicit, both significantly affect trust and continued use [8]. As personalization increases, users develop stronger self-brand connections with their IVAs, which mediates the relationship between trust and reusage intentions.

Although general enthusiasm persists, users impose important boundaries on emotionally responsive systems. Participants in [2] unanimously agreed that assistants should not feel and emote continuously – preferring selective emotional responsiveness – and many worried that excessive reliance on empathetic AI could reduce human-to-human interaction. The accuracy of emotion recognition is a related concern: users indicated that emotion-perception accuracy below 100% could lead to inappropriate responses with undesirable consequences, leading to suggestions that empathetic assistants should default to neutral expression whenever prediction confidence falls below a threshold, rather than reach mass deployment only once recognition is near-perfect. This concern connects directly to a broader ethical concern: systems capable of detecting and responding to emotional states could, in principle, be designed to exploit emotional vulnerability rather than support it, which is why researchers

emphasize that voice assistants must avoid expressing opinions, making judgments, or exhibiting negativity beyond what would naturally occur in supportive human interaction. Relatedly, users should understand when they are interacting with a system capable of emotion recognition and response modulation, though excessive disclosure risks diminishing the natural, fluid quality of effective empathetic interaction – a tension between transparency and usability that remains unresolved.

These concerns about exploitation are closely tied to privacy. Voice recordings inherently contain rich emotional content, and any system analyzing them necessarily extracts intimate information about a user’s psychological state, with the permanent storage of such emotional profiles creating potential for misuse, discrimination, or unauthorized disclosure [2]. [1] highlight a fundamental tension between model performance and privacy protection: high-quality emotion recognition typically requires substantial training data from diverse individuals, yet collecting it raises privacy concerns, and while federated learning offers a potential way to train on distributed data without centralizing it, doing so introduces technical challenges and may compromise performance. Cloud-based processing, the typical deployment architecture for voice assistants, exposes emotional data to interception risk and requires trust in a provider’s data handling practices. On-device processing offers better privacy but currently demands computational resources beyond most consumer hardware. Taken together, these concerns mean that any system handling emotional data – including the one developed in this thesis – should be designed with restraint regarding what is stored, for how long, and under what user control.

2.9 Conclusion

The development of multi-modal voice-based mood detection systems for emotionally appropriate responses in generic AI voice assistants represents a rapidly advancing research area with significant potential benefits and important challenges. Current technical capabilities enable accurate emotion recognition from speech in many contexts, with state-of-the-art

systems achieving >90% accuracy on standard benchmarks through sophisticated neural architectures combining acoustic and lexical information.

The transition from emotion recognition to appropriate empathetic response generation proves more challenging than recognition alone. Research demonstrates the feasibility of prosodic manipulation and response content selection to convey appropriate emotional tones, with user studies confirming that well-designed systems can achieve perceptible empathy. However, users express important concerns about accuracy requirements, potential for manipulation, privacy protection, and appropriate boundaries for AI empathy.

Significant research gaps persist in understanding cross-cultural emotion variation, long-term effects of interaction with empathetic systems, optimal evaluation frameworks, and appropriate deployment contexts. Technical challenges including real-time processing, multi-modal integration, robustness to diverse acoustic conditions, and privacy-preserving architectures require continued research attention.

As these systems move toward practical deployment, careful attention to ethical considerations, privacy protection, and appropriate boundaries becomes essential. Emotionally intelligent voice assistants offer genuine potential to improve human-computer interaction, support mental health, enhance education, and assist vulnerable populations. Realizing this potential while avoiding pitfalls requires ongoing collaboration among technologists, social scientists, ethicists, policymakers, and users. The goal should not be to create AI systems that replicate human empathy but rather to develop tools that appropriately support human needs, maintain emotional sensitivity, and complement rather than replace authentic human connection.

Chapter 3

Methodology

IAMDIVA is organized into two top-level modules that are intentionally separated from one another: a *mood detection module* and a *response modulation module*. This research concerns only the mood detection module. The response modulation module will be the subject of future work and is not addressed here. The mood detection module is composed of three independent emotion-detection methodologies – vocal landmark analysis, F0 analysis, and text-based sentiment analysis – together with a fusion layer that combines their outputs into a single estimate of the user’s emotional state.

3.1 System Architecture & Design

3.1.1 Modularity Principle

A guiding design constraint for this project is that every component must be usable independently, outside of this specific system, so that any individual detector can be extracted and reused as a plug-and-play component in other research. This is enforced architecturally in two ways. First, the project separates low-level feature extraction (computing a numeric feature vector from raw audio or text) from the mood detection logic (mapping a feature vector to an emotion label and a probability distribution over emotions). Feature extraction occupies its own logical layer, independent of the mood detection logic built on top of it. Each feature extractor can be used on its own, with no dependency on the rest of the system. Second, each of the three mood-detection wrappers implements the same minimal interface: a `detect()` operation that accepts audio and/or a transcript and returns a shared result object. Any one of the three can therefore be removed, replaced, or reused without modifying the other two or the fusion layer that sits above them.

3.1.2 Module Layout

The mood detection module is organized into three logical layers – audio processing, feature extraction, and mood detection. Audio processing handles microphone capture and signal conditioning upstream of feature extraction. This layer is responsible for producing a clean, normalized, 16 kHz mono float32 audio buffer regardless of what hardware or sample rate the host assistant captures audio at. Utterance boundaries are determined by WebRTC voice activity detection (VAD) operating on 30 ms frames. A candidate utterance begins once at least 90% of frames in a 300 ms ring buffer are classified as speech, and ends once at least 90% of frames in the trailing 300 ms window are classified as silence. A 300 ms padding window of surrounding audio is retained on each side of the detected speech, and any resulting segment shorter than 250 ms is discarded as noise rather than passed downstream. Feature extraction consists of three independent feature extractors – one per acoustic or textual modality – described in Section 3.2.

Mood detection handles the detection and fusion logic. This is the layer that a host system actually invokes. Within the mood detection layer, every detector and the fusion layer return a common result object, `MoodResult`, which is the single output contract for the entire module. `MoodResult` carries the emotion label (one of *angry*, *sad*, or *neutral*), a confidence value, continuous valence and arousal scores in $[0, 1]$, a boolean `is_distressed` flag, and the per-method probability breakdown for auditability. A `to_dict()` operation serializes the full result to a JSON-compatible dictionary, which is the intended hand-off point to the response modulation module, or to any other downstream consumer, without requiring any change to the detection pipeline itself.

3.1.3 Input & Output Contract

The mood detection module is designed to accept exactly the kind of information a conversational virtual assistant already has available mid-pipeline, and nothing more. The primary entry point accepts:

- **audio** – a one-dimensional NumPy array of float32 PCM samples (the captured user utterance), or **None** if audio is unavailable
- **sample_rate** – an integer sample rate (number of samples per second taken from the audio clip), defaulting to 16 kHz, and a
- **transcript** – the output of the ASR component utilized by the host assistant

Both audio and transcript are optional, and either may be supplied alone. This was a deliberate design decision. A host assistant that has already discarded raw audio after running its own STT, or one that does not perform STT in a form this module can use, should still be able to obtain an emotion estimate from whichever input it does have. Internally, any detector that receives no usable input for a given call (for example, the sentiment detector when no transcript is available) returns a uniform probability distribution (1/3 for each of the three classes) with zero confidence rather than failing. This signals to the fusion layer that the detector should not influence the final decision for that utterance.

3.1.4 Label Space

The system targets a three-class label space: angry, sad, and neutral. This label space was deliberately narrowed from the larger emotion taxonomies offered by the source datasets, as accuracy tends to have an inverse relationship with the length of the list of emotions – as more emotions are added, accuracy generally declines. The research problem, as defined in Chapter 1, is to detect when a user’s emotional state calls for a shift away from a default cheerful tone, not to classify the full space of human emotion. Calm, happy, and similar non-distressed states are functionally identical from the standpoint of response modulation. In all of these cases, the assistant should respond normally. Because of this, early iterations of this research placed those emotions in the same class. However, they vary widely from an acoustic standpoint, which made it difficult to classify them into a single space, since the

model would struggle with what characteristics define the class. Because of this, a neutral state was used as a baseline against the distressed emotions.

3.1.5 Fusion Architecture

Once received, audio is passed in parallel to the vocal landmark detector and the tone detector. If a transcript is available, it is passed to the sentiment detector. Each of the three detectors produces an independent probability distribution over the three emotion labels. A fusion layer then combines these three distributions into the single `MoodResult` that is returned to the caller. The primary fusion mechanism implemented and evaluated in this research is a rule-based arbitration scheme, described in Section 3.2. Two alternative, exploratory fusion mechanisms were also implemented: a stacked meta-classifier and a joint single-model approach that fuses raw features rather than detector opinions. For completeness, these alternatives are described in Section 3.2 as well, since all three share the same upstream feature-extraction and detection logic.

3.2 Algorithms & Theoretical Frameworks

3.2.1 Vocal Landmark Detection

Adopted Framework & This Project’s Contribution

The vocal landmark extractor implements the acoustic landmark detection framework originally described by [16] and [17], and operationalized for depression detection by [15]. The six-band frequency decomposition, the six landmark types, the dB-threshold detection criteria, and the use of landmark bigrams as features are all adopted from [15] and [18]. This project’s contribution is the adaptation of that depression-detection framework to three-class mood classification, the specific 18-dimensional feature vector assembled from the detected landmarks (Section 3.2.1), and the removal of frication-rate and voiced-frication-rate features from the final classifier input, described at the end of this subsection.

Band Decomposition & Event Detection

To enable acoustic landmark detection, audio is first converted to a short-time power spectrogram using a 512-point FFT with a 128-sample hop, at a 16 kHz sample rate. This corresponds to a frame interval of 8 ms. Let $S(f, t)$ denote the power at frequency bin f and frame t . The spectrogram is partitioned into six frequency bands:

- Band 1: 0–400 Hz (glottal / voiced periodicity)
- Band 2: 800–1500 Hz (first-formant region)
- Band 3: 1200–2000 Hz (first-formant region)
- Band 4: 2000–3500 Hz (second-formant / obstruent region)
- Band 5: 3500–5000 Hz (frication onset)
- Band 6: 5000–8000 Hz (high frication / unvoiced)

For each band b and frame t , a mean band power $P_b(t)$ is computed by averaging $S(f, t)$ over all frequency bins falling inside that band’s edges, and converted to decibels as

$$P_b^{\text{dB}}(t) = 10 \log_{10}(\max(P_b(t), \epsilon))$$

where $\epsilon = 10^{-12}$ prevents the logarithm of zero. Each per-band dB signal is smoothed with a three-frame moving average before transition detection.

Six landmark types are detected from energy transitions within and across these bands, each occurring as an onset (+) or offset (−) event. For a smoothed signal $x(t)$, a rising transition is registered at frame t if

$$x(t) - x(t - 1) \geq \theta_{\text{rise}}$$

and a falling transition is registered if

$$x(t) - x(t - 1) \leq -\theta_{\text{fall}}$$

The specific bands and thresholds used for each landmark type are as follows.

- **Glottal (g)** landmarks are detected from Band 1 (0–400 Hz) alone, with $\theta_{\text{rise}} = \theta_{\text{fall}} = 5$ dB. Detected onsets and offsets are then paired by a dynamic-programming step – each onset is matched to the nearest subsequent offset, producing alternating (g+, g–) pairs [15, 16].
- **Burst (b)** landmarks are detected from the mean of Bands 4 through 6 (2000–8000 Hz combined), with $\theta_{\text{rise}} = \theta_{\text{fall}} = 8$ dB.
- **Syllabic/sonorant (s)** landmarks are detected from the mean of Bands 2 and 3 (800–2000 Hz combined), with $\theta_{\text{rise}} = \theta_{\text{fall}} = 6$ dB, and are only retained if they fall within a voiced span, defined as the interval between a paired glottal onset and its matched offset.
- **Frication (f) and voiced frication (v)** landmarks are detected from the same combined Band 4-6 signal used for burst detection, with the same 6 dB threshold as the syllabic detector. A detected transition is labeled *voiced frication* (v) if it falls within a voiced span and *frication* (f) otherwise [15].
- **Periodicity (p)** landmarks are detected separately from the other five types, using short-window autocorrelation rather than band energy. For each analysis window of length equal to four hop intervals (512 samples, 32 ms), the normalized autocorrelation $r(\tau)$ is computed (as shown below), and the maximum value of $r(\tau)$ over lags τ corresponding to the 80–400 Hz pitch range is taken as a periodicity score. A frame is classified as periodic if this score exceeds 0.30. A periodicity onset (p+) is registered at the first frame where the periodic/non-periodic classification switches from false to

true, and a periodicity offset (p-) is registered at the corresponding switch from true to false.

$$r(\tau) = \frac{\sum_n x(n)x(n+\tau)}{\sum_n x(n)^2}$$

Feature Vector

From the resulting sequence of landmark events, an 18-dimensional feature vector is assembled per utterance, grouped into six categories:

- **Rate features (5)** – The per-second event rate for glottal, burst, syllabic, and periodicity landmarks, computed as $\frac{\text{event count}}{\text{utterance duration in seconds}}$, plus the combined rate of all detected landmarks of any type.
- **N-gram features (4)** – Landmark events are first converted into a sequence of tokens ordered by time of occurrence (for example: g+, b-, s+). The two most frequent two-token sequences (bigrams) and the two most frequent three-token sequences (trigrams) in that token sequence are identified, and their counts are normalized by the total number of bigrams (or trigrams, respectively) in the utterance.
- **Band-energy-at-landmark features (4)** – The mean spectral power, in a specific frequency range, evaluated only at the frames where a specific landmark type occurs: mean power in 0–400 Hz at glottal-landmark frames, mean power in 800–2000 Hz at syllabic-landmark frames, mean power in 2000–3500 Hz at burst-landmark frames, and mean power in 3500–8000 Hz at frication and voiced frication landmark frames combined.
- **Energy-variability feature (1)** – The coefficient of variation (σ/μ) of frame-level root mean squared (RMS) energy across the whole utterance, where RMS energy at frame t is $\sqrt{S(f,t)}$, the square root of the mean spectral power across all frequency bins at that frame.

- **Inter-landmark interval statistics (2)** – The mean and standard deviation of the time, in seconds, between consecutive landmark events of any type, after sorting all events by their time of occurrence.
- **RMS-at-landmark features (2)** – The mean RMS energy, using the same per-frame RMS definition above, evaluated specifically at burst-landmark frames and at periodicity-landmark frames.

This feature set is intentionally restricted to discrete, low-dimensional descriptors rather than the full landmark event sequence, in order to keep the per-utterance computational cost low and to keep the resulting classifier interpretable. Not all detected landmark types are represented by a rate feature in the final vector. Frication and voiced-frication landmark counts proved to be among the least useful features during feature selection. The underlying frication and voiced-frication events are still detected and retained internally, since they feed the n-gram and band-energy-at-landmark features, but their raw counts do not appear in the final 18-dimensional vector.

3.2.2 Fundamental Frequency Detection

Adopted Framework & This Project’s Contribution

The F0 mood detector is restricted by design to F0-derived and time-domain features only. This restriction keeps the module’s computational footprint low and keeps its feature set conceptually distinct from both the vocal landmark detector and conventional MFCC-based approaches. The eight core statistical functionals of the F0 contour (mean, geometric mean, median, skewness, variance, harmonic mean, standard deviation, and kurtosis) and the 20 ms frames with 10 ms hops used to compute them are adopted directly from [13]. Every other feature in the 26-dimensional vector described below, including the contour-shape features, the voice-quality proxies, the temporal/rhythmic features, and the speaker-normalized F0 block, is this project’s own extension of that base feature set, motivated by the wider F0-

emotion literature rather than by [13] itself.

F0 Extraction

The F0 extraction path implemented by this research uses the YIN algorithm [13], as implemented in the `librosa` library, bounded to a pitch search range of 80–800 Hz for the tone detector specifically. A frame is classified as voiced if its estimated F0 falls strictly inside this range. A secondary, project-implemented extractor reproduces the FFT-based largest-magnitude-spectrum-peak method described by [13] and serves both as a fallback when `librosa` is unavailable and as a standalone, paper-faithful F0 extractor in its own right.

From the resulting voiced F0 values $f_1, \dots, f_n$ (in Hz), the eight statistical functionals specified by [13] are computed as follows, where $\bar{f} = \frac{1}{n} \sum_i f_i$ is the arithmetic mean:

$$\text{arithmetic mean} = \bar{f}$$

$$\text{geometric mean} = \exp \left(\frac{1}{n} \sum_i \ln f_i \right)$$

$$\text{harmonic mean} = n / \sum_i \frac{1}{f_i}$$

$$\text{variance} = \frac{1}{n} \sum_i (f_i - \bar{f})^2$$

$$\text{standard deviation} = \sqrt{\text{variance}}$$

$$\text{median} = \text{the middle value of } f_1, \dots, f_n \text{ when sorted}$$

$$\text{skewness} = \frac{\frac{1}{n} \sum_i (f_i - \bar{f})^3}{\left(\frac{1}{n} \sum_i (f_i - \bar{f})^2 \right)^{3/2}}$$

$$\text{kurtosis (excess)} = \frac{\frac{1}{n} \sum_i (f_i - \bar{f})^4}{\left(\frac{1}{n} \sum_i (f_i - \bar{f})^2 \right)^2} - 3$$

Feature Vector

The tone detector computes a 26-dimensional feature vector that extends beyond the eight statistics above. The additional 18 features were developed for this project and are organized into five groups.

- **F0 distributional statistics (10)** – Six of the eight functionals defined above (arithmetic mean, geometric mean, median, standard deviation, skewness, kurtosis), plus the maximum voiced F0 value, the entropy of a 20-bin histogram of the voiced F0 values, the interquartile range (IQR) of voiced F0, and the fraction of all frames classified as voiced.
- **F0 contour-shape features (3)** – A quadratic polynomial $ax^2 + bx + c$ is fit, by least squares, to the voiced F0 values against a normalized time axis running from 0 to 1 across the utterance. The leading coefficient a (curvature) and the mean absolute residual of the fit are retained as two features. A separate linear fit $mx + c$ to the same data yields the slope m as another feature. Positive curvature corresponds to a pitch contour that rises then falls. Negative curvature corresponds to a contour that falls then rises. A large residual indicates an irregular, unpredictable contour, while a small residual indicates a smooth one.
- **Time-domain energy features (3)** – The mean, standard deviation, and coefficient of variation of per-frame RMS energy, computed directly in the time domain with a hop of 512 samples and without any spectrogram computation.
- **Voice-quality proxies (3)** – Shimmer is the mean absolute difference between consecutive RMS-energy frames, normalized by the mean RMS energy. Jitter is the mean absolute difference between consecutive F0 values, normalized by the mean F0. A harmonics-to-noise ratio (HNR) proxy is computed by taking a 40 ms window centered on the utterance, removing its mean, computing its normalized autocorrelation

$r(\tau)$ as defined in the periodicity-landmark description above, and reporting the maximum value of $r(\tau)$ over the lag range corresponding to 80–400 Hz, clipped to $[0, 1]$.

- **Temporal/rhythmic features (3)** – The mean zero-crossing rate of the time-domain signal, the fraction of frames whose RMS energy falls below the 20th percentile of RMS energy across the utterance (a pause-rate proxy), and the number of voiced-segment onsets per second, where a voiced-segment onset is a frame at which the per-frame voicing decision switches from unvoiced to voiced.
- **Speaker-normalized F0 features (4)** – The voiced F0 contour is z -scored within the utterance: $\hat{f}_i = (f_i - \bar{f})/\sigma_f$, where σ_f is the standard deviation of $f_1, \dots, f_n$. The mean, standard deviation, interquartile range, and linear-fit slope of $\hat{f}_1, \dots, \hat{f}_n$ are then computed using the same definitions given above, applied to the normalized contour instead of the raw one. This removes the speaker’s own baseline pitch and isolates emotion-driven pitch *pattern* rather than speaker identity, intended to improve generalization across speakers with different natural pitch ranges.

All 26 features above are computed for every utterance, but only 22 are passed to the tone classifier. The interquartile range and linear-fit slope of the raw (non-normalized) F0 contour, and the mean and standard deviation of the z -scored F0 contour, are computed and retained for diagnostic and feature-selection purposes (Section 3.3) but excluded from the active classifier input. This mirrors the vocal landmark detector’s separation of feature *computation* from feature *selection*, except that the vocal landmark detector currently uses its full 18-dimensional computed vector with no features deactivated.

3.2.3 Sentiment Analysis

Adopted Framework & This Project’s Contribution

The sentiment detector operates exclusively on the text transcript of a user utterance and is the only one of the three detectors with no acoustic component. Unlike the vocal land-

mark and tone detectors, no classifier is trained from scratch for this component. Instead, a pretrained transformer model is used directly, without fine-tuning, and this project’s contribution is the wrapper interface that lets this detector participate in the shared fusion contract.

Two separate sentiment detectors exist within the code-base. A low-level feature extractor wraps a HuggingFace transformer pipeline and exposes a generic three-dimensional positive/negative/neutral feature vector, defaulting to the binary model `distilbert/distilbert-base-uncased-finetuned-sst-2-english`. This class is reusable on its own, independent of the project’s three-class label space, and is intended as back-up method, as well as a general-purpose, plug-and-play sentiment component. The mood-detection-facing wrapper defaults to a richer pretrained model, `j-hartmann/emotion-english-distilroberta-base`, a DistilRoBERTa-based classifier producing seven output classes: joy, sadness, anger, fear, disgust, surprise, and neutral. It is this wrapper’s output that feeds the fusion layer.

Label Mapping

Let $p_{\text{ang}}, p_{\text{sad}}, p_{\text{neu}}$ denote the model’s raw probabilities for its anger, sadness, and neutral classes, respectively. The seven raw classes are mapped onto the project’s three-class label space by direct correspondence, with $P(\text{angry}) = p_{\text{ang}}$, $P(\text{sad}) = p_{\text{sad}}$, and $P(\text{neutral}) = p_{\text{neu}}$, renormalized to sum to 1. The model’s remaining four classes – joy, fear, disgust, and surprise – are discarded entirely rather than redistributed across the three target classes. This was decided so as not to add potential confusion to the classifier, in keeping with the conservative, single-purpose mapping used elsewhere in this label space (Section 3).

If the underlying model is instead configured (via an environment variable) to a binary SST-2-style model, the project’s negative class probability is split evenly between angry and sad, and the positive and any neutral mass is assigned to neutral. An analogous, explicit remapping function is defined for a three-class (positive/negative/neutral) model. If

no transcript is supplied, the sentiment detector returns a uniform probability distribution with zero confidence, signaling to the fusion layer that it has nothing to contribute for that utterance, rather than guessing.

3.2.4 Fusion Strategies

Primary System: Rule-Based Arbitration

This rule set, and its specific decision thresholds, are an original contribution of this project, motivated by detector-reliability patterns observed during development rather than adopted from prior fusion literature.

The primary fusion mechanism is a hand-specified arbitration rule set, chosen for this system over a purely statistical weighted average because the vocal landmark and tone detectors were found, during development, to be of broadly comparable and substantially higher standalone reliability than the sentiment detector. The sentiment detector exhibited a strong bias toward over-predicting *neutral* on transcripts of acted emotional speech that carry little explicit emotional vocabulary. A simple global weighted average of all three detectors' probabilities was found to let this weak signal degrade cases the two acoustic detectors had already resolved correctly. The rule set instead confines sentiment's influence to specific, well-defined situations. Let $\hat{y}_{vl}$, $\hat{y}_{tone}$, and $\hat{y}_{sent}$ denote the most probable class predicted by the vocal landmark, tone, and sentiment detectors, respectively, and let c_{sent} denote the sentiment detector's confidence in its own prediction (its maximum class probability).

1. If $\hat{y}_{vl} = \hat{y}_{tone}$, that class wins outright, unless $\hat{y}_{sent} \in \{\text{angry}, \text{sad}\}$, $\hat{y}_{sent} \neq \hat{y}_{vl}$, and $c_{sent} \geq 0.85$, in which case the sentiment prediction overrides instead. Sentiment is never permitted to override a vocal-landmark/tone agreement with a confident *neutral* prediction, since over-predicting neutral is sentiment's dominant failure mode. This override exists as an infrequent safety valve rather than a routine decision path.
2. If $\hat{y}_{vl} \neq \hat{y}_{tone}$ and $\{\hat{y}_{vl}, \hat{y}_{tone}\} = \{\text{sad}, \text{neutral}\}$, the most confusable disagreement

pair for both acoustic detectors, sentiment acts as a tiebreaker, provided $\hat{y}_{\text{sent}} \in \{\text{sad}, \text{neutral}\}$ as well. If sentiment instead predicts angry, which is out of scope for this tiebreak, the decision falls through to weighted fusion (rule 4).

3. If $\hat{y}_{\text{vl}} \neq \hat{y}_{\text{tone}}$ and angry is involved on either side, sentiment is ignored entirely, since both acoustic detectors are independently reliable at detecting anger. The decision falls through to the weighted-fusion resolver.
4. Any case not resolved by rules 1 through 3 is resolved by weighted fusion: a weighted average of all three detectors' probability distributions, $P_{\text{fused}}(\ell) = \sum_d w_d P_d(\ell)$ for each class ℓ , with weights w_d derived from each detector's measured per-class F1 score (Section 3.6) where available, and a global weight otherwise. A per-clip confidence gate excludes any detector whose own maximum class probability falls below a threshold (0.60 in normal operation) for that specific utterance, redistributing its weight to the remaining detectors, so an uninformative prediction from one detector cannot dilute a confident prediction from another.

Adaptive Weight Calculation

Global and per-class fusion weights are not hand-tuned. They are derived from measured detector accuracy. Let a_d denote a detector's accuracy (global weighting) or per-class F1 score (per-class weighting), for $d \in \{\text{vocal_landmarks}, \text{tone}, \text{sentiment}\}$. Three weighting strategies are implemented. The direct accuracy-proportional strategy sets $w_d \propto a_d$. The power-law strategy sets $w_d \propto a_d^p$ for a configurable exponent p (default $p = 2$). The softmax strategy, used by default, sets

$$w_d = \frac{\exp(a_d/T)}{\sum_{d'} \exp(a_{d'}/T)}$$

with temperature $T = 0.15$ controlling how sharply better-performing detectors are favored. After any of the three strategies produces raw weights, a floor of 0.05 is applied (any weight below the floor is raised to it) and the full set is renormalized to sum to 1, so no detector

is ever reduced to zero baseline influence. If a detector has no measured accuracy at all, or no measured F1 for a specific class, its score is set to a fixed chance-level constant (0.1, deliberately below the true three-class chance rate of $1/3$) rather than falling back to an above-chance guess or to another detector’s value.

Per-class weights allow, for example, the tone detector to be weighted more heavily than the vocal landmark detector specifically for the *angry* class if its measured F1 score on that class is higher, even where the two detectors’ overall (global) weights are nearly equal. This reflects the empirical finding that the two acoustic detectors do not fail uniformly across classes.

At inference time, a per-clip confidence gate is applied before the weighted average above is computed. Any detector whose maximum class probability for the current utterance falls below a gate threshold of 0.60 has its weight set to zero for that utterance only, and the remaining weights are renormalized to sum to 1. A detector silenced by the gate on one utterance retains its full weight on the next.

Exploratory Alternative: Learned Fusion

As an exploratory alternative to the hand-specified rule set above, a learned fusion back-end was also implemented and evaluated, though it is not the primary system reported in Chapter 4. Rather than encoding fusion logic by hand, this approach trains a small XGBoost meta-classifier directly on a 13-dimensional meta-feature vector: the nine class probabilities produced by the three base detectors (three classes each), each detector’s own maximum-probability confidence score, and a binary flag indicating whether a transcript was available for that utterance, so the meta-model can learn how much to trust the sentiment columns conditional on whether they reflect real input or a zero-confidence placeholder. This meta-classifier uses fewer estimators and a comparable tree depth to the base detectors’ own XGBoost classifiers (500 estimators, maximum depth 10, learning rate 0.01), since a 9- to 13-feature meta-problem requires less capacity than the 18- or 26-dimensional raw-

feature classification problems solved by the base detectors. A configuration toggle allows this vector to be reduced to 12 features, dropping the transcript flag, or to the raw nine-class probabilities alone, for comparison.

A companion variant trains an analogous meta-classifier using only the vocal landmark and tone detectors’ outputs, a six-dimensional meta-feature vector (the six class probabilities) by default, to evaluate the acoustic portion of the system in isolation from any transcript dependency. This is relevant to deployment contexts where ASR output is unavailable, unreliable, or deliberately excluded for privacy reasons. Both stacked variants require that the meta-classifier be trained on base-detector outputs the base detectors never saw during their own training, to avoid stacking leakage. This requirement is what motivates the four-way data split described in Section 3.4.

Exploratory Alternative: Joint Feature-Level Model

A second exploratory alternative departs from late fusion entirely. Instead of combining three independently trained detectors’ opinions (probability distributions), this standalone experiment concatenates the three detectors’ raw features, the 18-dimensional vocal landmark vector, the 26-dimensional tone vector, and the three-dimensional sentiment probability vector, for 47 raw columns total, into a single feature space, and trains one XGBoost classifier directly on the combined vector. Of these 47 raw columns, 27 are retained as the active, hand-pruned feature set actually passed to the classifier, following the same selection-without-deletion pattern used by the tone detector. This model is entirely self-contained, meaning it does not load any of the trained checkpoints produced for the primary system and it computes every feature itself from the same underlying feature-extraction functions.

3.3 Feature Selection & Classifier Comparison Experiments

Before the final architecture described above was settled on, a separate, self-contained experimentation procedure was used to make three of the project’s central design decisions

empirically rather than by assumption. These design decisions were: which classifier family to use, which candidate features within the vocal landmark and tone feature pools were worth retaining, and how coarse or fine the target label space should be. This procedure does not directly modify or otherwise affect the production detectors. It operates on one detector module at a time (vocal landmark or tone) and writes a timestamped report on each run, so that earlier comparisons are never overwritten.

The experiment proceeds in four phases.

Phase 1 trains XGBoost, Random Forest, and a Support Vector Classifier on the target module’s full current feature vector and compares accuracy, per-class precision/recall/F1, and the confusion matrix across all three. XGBoost was selected as the production classifier for both the vocal landmark and tone detectors on the basis of this comparison, motivating its use as the default classifier type described in Section 3.2, with Random Forest, SVM, and logistic regression retained as configurable fallbacks.

Phase 2 takes the winning classifier from Phase 1 and runs it against an expanded candidate feature pool, broader than the feature set ultimately shipped, through forward stepwise selection, L1 (Lasso) logistic regression, recursive feature elimination, and principal component analysis, reporting the selected features, their importance scores, and the resulting accuracy for each method. The active feature subsets described in Section 3.2 are the product of this phase.

Phase 3 re-trains the winning classifier on a deliberately expanded label space loaded from all three source corpora, including emotions the production system discards, and evaluates five candidate grouping schemes:

- G1: neutral, sad / happy, angry / fear, disgust
- G2: neutral / sad, fearful / angry, disgust
- G3: neutral / sad, fearful / angry, disgust / happy
- G4: neutral / sad / angry / happy

- G5: neutral / sad / angry

Clips whose source label does not fit a given scheme’s buckets are filtered out and the count logged, rather than aborting the whole comparison, since not every grouping scheme defines a bucket for every emotion IEMOCAP, RAVDESS, and CREMA-D jointly provide. This phase is the empirical basis for the label-space decision discussed in Section 3.2.

Phase 4 narrows to a binary *sad*-versus-*neutral* classifier specifically, since this pair was identified as the most confusable for both acoustic detectors (Section 3.2), and re-runs each feature-selection method from Phase 2 on this harder, two-class subproblem in isolation, reporting per-sample feature values and feature importances alongside the usual metrics, as a targeted diagnostic for the project’s most persistent failure mode.

3.4 Data Collection & Preparation

3.4.1 Datasets

Three labeled emotional speech corpora were combined for training and evaluation: RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song), CREMA-D (Crowd-sourced Emotional Multimodal Actors Dataset), and IEMOCAP (Interactive Emotional Dyadic Motion Capture) [28, 29, 30]. RAVDESS and CREMA-D consist of short, acted utterances recorded by trained actors reading fixed, lexically matched sentence sets across a range of target emotions. IEMOCAP consists of longer dyadic, partly improvised dialogue sessions with utterance-level emotion labels assigned by majority vote among multiple human annotators. Using all three together was intended to offset, to some degree, the acted-speech homogeneity of RAVDESS and CREMA-D with IEMOCAP’s more naturalistic, conversational delivery, while still allowing every clip to be loaded through the same pipeline, resampled to a common 16 kHz mono representation. Table 3.1 shows the overall amounts of usable clips from the databases and how many files were actually utilized.

Table 3.1: Source Dataset Sizes

Dataset	Total	Angry	Sad	Neutral	A+S+N	Used (balanced)
RAVDESS	1,440	192	192	96	480	—
CREMA-D	7,442	1,271	1,271	1,087	3,629	—
IEMOCAP	10,039	1,103	1,084	1,708	3,895	—
Total	24,837	2,566	2,547	2,801	8,004	7,914

3.4.2 Label Mapping

Each source corpus uses its own emotion taxonomy and its own filename or annotation-file convention. RAVDESS encodes emotion as a numeric code embedded in the filename (01, 04, 05). CREMA-D encodes emotion as a three-letter code in the filename (NEU, SAD, ANG). IEMOCAP supplies majority-vote emotion codes in separate annotation files keyed to each utterance ID, with utterances having no clear annotator majority excluded.

Only clips whose source-corpus label maps to one of the project’s three target classes are loaded. Every other source emotion is skipped entirely. This conservative inclusion rule was adopted specifically so that the *neutral* class remains a clean estimate of genuinely neutral speech, rather than an unintentional grab-bag of every emotion the project does not explicitly target.

3.4.3 Class Balancing

Across all three corpora, source-label frequency is uneven, and *neutral* clips are typically overrepresented relative to *angry* and *sad*. Before any train/test split is performed, the combined dataset is balanced by undersampling. Let $n_{\min}$ be the count of the smallest of the three classes after the label mapping above. Each class is capped at $1.1 \times n_{\min}$ clips, with the specific clips to discard within an oversized class chosen uniformly at random (seeded for reproducibility). This is applied once, globally, prior to splitting, so that every downstream split inherits the same approximately balanced class distribution. This is apparent in the balanced/used counts of Table 3.1.

3.4.4 Data Splitting

A four-way, stratified split is used, replacing an earlier three-way split used during initial development. The three-way split was created before stacked-fusion models were used to train the ensemble approach. Each split is stratified on emotion label so that class balance is preserved across all four partitions:

- **Train (40%)** – Used to fit the vocal landmark and tone detectors’ classifiers, and, where applicable, the sentiment detector’s lightweight per-class calibration weights.
- **Individual test (15%)** – A held-out set on which all three base detectors are evaluated independently and identically, so that their standalone accuracy figures are measured on exactly the same clips.
- **Ensemble train (30%)** – Used exclusively by the stacked-fusion alternatives. The already-trained vocal landmark, tone, and sentiment analysis modules are run on this split to produce the meta-feature vectors used to fit the XGBoost meta-classifier.
- **Ensemble test (15%)** – A final, fully held-out set used to evaluate whichever fusion backend or backends are being assessed, so that the primary rule-based system and either exploratory alternative can be compared on identical data.

The four-way split exists specifically to give any learned meta-classifier a training set that is disjoint from both the base detectors’ own training data and the final evaluation set, preventing stacking leakage.

For text transcripts, since RAVDESS, CREMA-D, and IEMOCAP are audio-only corpora with no associated transcript, the OpenAI Whisper *tiny* model is run once over the individual test set and the ensemble test set to generate transcripts for the sentiment detector and for any fusion approach that includes a text modality, with results cached to disk so repeated evaluation runs do not require re-transcription. In a real-world scenario, the text transcript

is generally passed to the modules by the IVA, eliminating the need for the Whisper model in production.

3.5 Experimental Setup & Environment

3.5.1 Hardware Specifications

Development and evaluation were carried out on a Mac Mini (Apple Silicon) with 48GB of unified memory. All training, feature extraction, and evaluation described in this chapter were run on this machine’s CPU. No discrete GPU or CUDA device was used. PyTorch’s Metal Performance Shaders (MPS) backend was available on this hardware for any torch-based component, such as the HuggingFace sentiment model and Whisper.

It is important to note that this is a development and evaluation machine rather than a deployment target. The hardware profile of an actual conversational assistant device, a smart speaker, mobile phone, or other IoT endpoint, is considerably more constrained, and is the implicit benchmark against which the project’s real-time and footprint design goals (Chapter 1) are framed throughout. As a point of reference for what a downstream integration would require: LEON, the open-source virtual assistant this research was originally built to work with, requires 5GB of free RAM at runtime and is recommended to be stored on local disk rather than a cloud-synced folder, since LEON installs several gigabytes of model files. LEON’s own LLM, ASR, and TTS components run as a separate process from the mood detection module described here. LEON’s resource requirements are reported here only to characterize the kind of host environment this module is intended to run alongside. They are not used in any of the evaluation reported in Chapter 4, since this research’s training, testing, and latency measurements were conducted independently of any running instance of LEON.

3.5.2 Software Stack

The project is implemented in Python 3.13 and developed using Visual Studio Code on macOS. Table 3.2 summarizes the primary libraries used by each stage of the pipeline.

Table 3.2: Primary Software Dependencies by Pipeline Stage

Pipeline Stage	Library / Tool
Audio capture & VAD	PyAudio, <code>webrtcvad-wheels</code> on Apple Silicon
Audio preprocessing	NumPy, SciPy
Acoustic feature extraction	<code>librosa</code> , NumPy
Classical ML / classifiers	scikit-learn, XGBoost
Sentiment analysis	HuggingFace <code>transformers</code>
Transcript generation	OpenAI Whisper (<i>tiny</i> model)

Audio capture relies on PyAudio together with WebRTC voice activity detection to segment continuous microphone input into discrete speech utterances, padding each segment with a short buffer of surrounding silence and discarding segments shorter than a minimum duration, as detailed in Section 3.2’s module layout discussion above. The audio preprocessing stage applies, in order, mono conversion, resampling to a common 16 kHz target rate, spectral-subtraction noise reduction with an exponentially updated noise-floor estimate, a pre-emphasis high-pass filter with coefficient 0.97, and RMS-based amplitude normalization. Every stage is independently toggleable so the pipeline can be adapted or simplified for testing.

The primary classifier used by both the vocal landmark and tone detectors is XGBoost (1000 estimators, maximum depth 10, learning rate 0.01), wrapped in scikit-learn’s isotonic-calibration cross-validation utility (three-fold) so that the probability outputs from each detector are well-calibrated before being combined in fusion. All randomized procedures, including dataset balancing, train/test splitting, and classifier initialization, use a fixed random seed so that results are reproducible across runs.

3.6 Evaluation Metrics, Testing, and Baselines

Each of the three base detectors, and each fusion strategy, is evaluated using the same set of metrics on its corresponding held-out test split from Section 3.4:

- Per-class accuracy: $\frac{\text{Correct Predictions}}{\text{Total Predictions}}$
- Per-class precision: $\frac{\text{Correct Positive Predictions}}{\text{All Positive Predictions}}$
- Per-class recall: $\frac{\text{Correct Positive Predictions}}{\text{Actual Positive Instances}}$
- Per-class F1 score: $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
- 3x3 confusion matrix
- Average inference latency (per-utterance)

Latency is measured end-to-end from raw audio (and transcript, where applicable) input to emotion label output. Per-class F1 scores additionally serve as the input to the adaptive ensemble weight calculation described in Section 3.2.

Latency is budgeted against the real-time constraint discussed in Chapter 1. Each individual detector is required to perform well within the time available between the end of a user utterance and the point at which a virtual assistant would otherwise begin generating its response, and the fused pipeline as a whole is held to a combined latency budget appropriate for conversational use. The goal for this research was to keep total latency, summed across all modules, under 100 ms.

Because all three base detectors and all three fusion strategies are evaluated on exactly the same held-out clips under the same metric definitions, their accuracy, per-class F1, and latency figures are directly comparable to one another, and the primary rule-based fusion system can be directly compared against each exploratory alternative on identical data. The sentiment detector and any fusion approach incorporating it are additionally evaluated specifically on Whisper-generated transcripts of the same audio test clips, rather than on

hand-written or idealized text, so that its reported performance reflects the same speech-to-text uncertainty a deployed system would actually face.

Two distinct testing mechanisms exist alongside the training pipeline. The first is a unit and integration test suite covering each detector in isolation, the full fusion pipeline end to end, the shared output contract, per-utterance latency against the real-time budget, and the audio-processing layer, with hardware dependencies such as PyAudio and `webrtcvad` left out so the suite runs without a physical microphone. The second is the accuracy benchmark produced as a byproduct of the training procedure itself, covered in Chapter 4.

Chapter 4

Results and Discussion

Two notes on scope and reporting practice apply throughout this chapter. First, where multiple training runs were performed over the course of development – which was frequently, since the feature vectors of both acoustic detectors changed several times – this chapter cites the smallest number of reports necessary to support a claim, and always treats the most recent run of a given experiment as authoritative for any number reported as a current or final result. Earlier runs are referenced only where the change between runs is itself the finding being reported. Second, the dataset, splitting procedure, balancing procedure, and evaluation metrics referenced below are exactly the ones specified in Chapter 3. This chapter does not re-derive them – it will simply report the numbers they produced.

4.1 Individual Detector Results

Each of the three base detectors was evaluated independently and identically on the same held-out individual test split: 1,187 clips drawn from the combined, class-balanced corpus of RAVDESS, CREMA-D, and IEMOCAP (7,914 clips after balancing at a balance ratio of 1.1), stratified by emotion label. Sentiment was evaluated on Whisper-*tiny* transcripts of the same audio clips rather than on hand-written text, so that its reported accuracy reflects the same speech-to-text uncertainty a deployed system would face. 1,182 of the 1,187 clips produced a usable transcript.

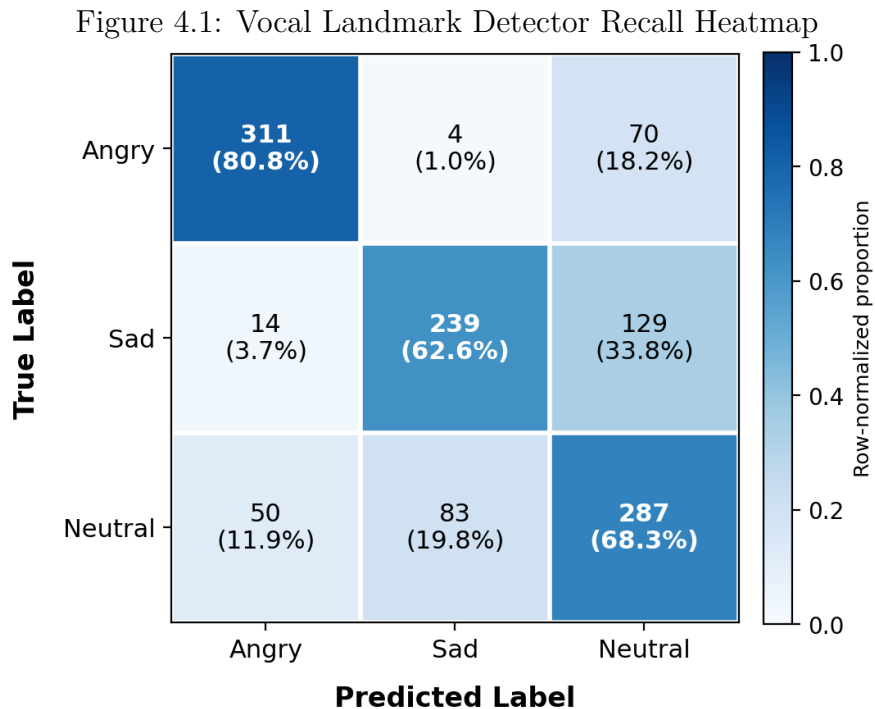
4.1.1 Vocal Landmark Detector

The vocal landmark detector reached 70.51% accuracy on the individual test split, with a macro-averaged F1 of 70.90% and an average per-clip latency of 21.3 ms. As seen in

Table 4.1, per-class performance was uneven: *angry* was detected reliably ($F1 = 81.84\%$), while *sad* ($F1 = 67.51\%$) and *neutral* ($F1 = 63.36\%$) were both substantially weaker and were frequently confused with one another. The heatmap in Figure 4.1 shows that of 382 true *sad* clips, 129 (33.8%) were misclassified as *neutral*; of 420 true *neutral* clips, 83 (19.8%) were misclassified as *sad*. By contrast, confusion between *angry* and either other class was rare in both directions.

Table 4.1: Vocal Landmark Detector Performance Statistics

Emotion	Precision	Recall	F1 Score
Angry	0.829	0.808	0.818
Sad	0.733	0.626	0.675
Neutral	0.591	0.683	0.634
Average	0.718	0.706	0.709



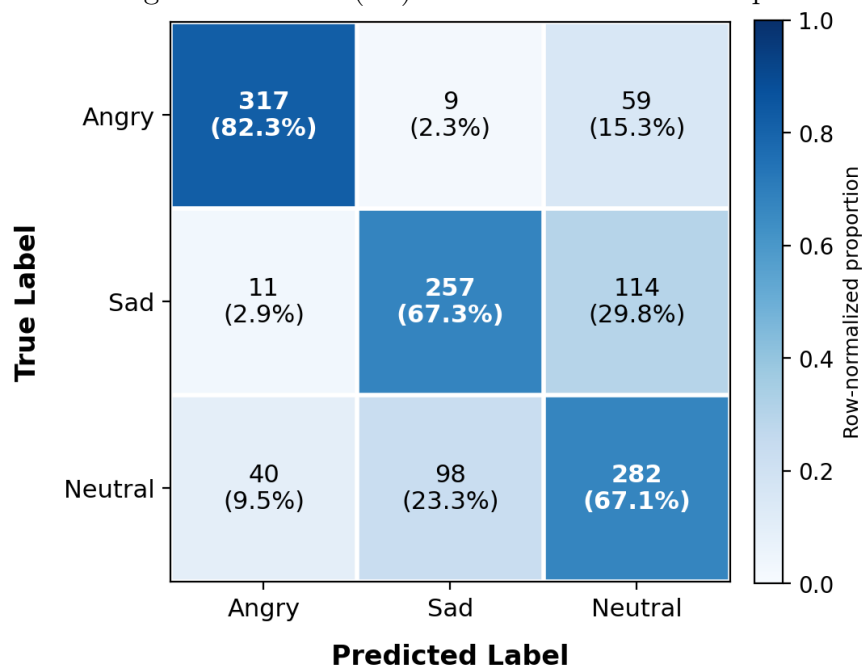
4.1.2 Tone (F0) Detector

The tone detector reached 72.11% accuracy, the highest of the three base detectors, with a macro-averaged F1 of 72.52% and an average per-clip latency of 7.4 ms – by a wide margin the fastest of the three, and a direct consequence of its restriction to F0-derived and time-domain features with no short-time Fourier transform beyond what F0 extraction itself requires. The overall precision, recall, and F1 scores can be seen in Table 4.2. The same sad/neutral confusion pattern observed for vocal landmarks recurs here, somewhat less severely: 114 of 382 true *sad* clips (29.8%) were predicted *neutral*, and 98 of 420 true *neutral* clips (23.3%) were predicted *sad*, as seen in Figure 4.2. *Angry* was again the most reliably detected class, with an F1 score of 84.20%).

Table 4.2: Tone (F0) Detector Performance Statistics

Emotion	Precision	Recall	F1 Score
Angry	0.861	0.823	0.842
Sad	0.706	0.673	0.689
Neutral	0.620	0.671	0.645
Average	0.729	0.723	0.725

Figure 4.2: Tone (F0) Detector Recall Heatmap



4.1.3 Sentiment Detector

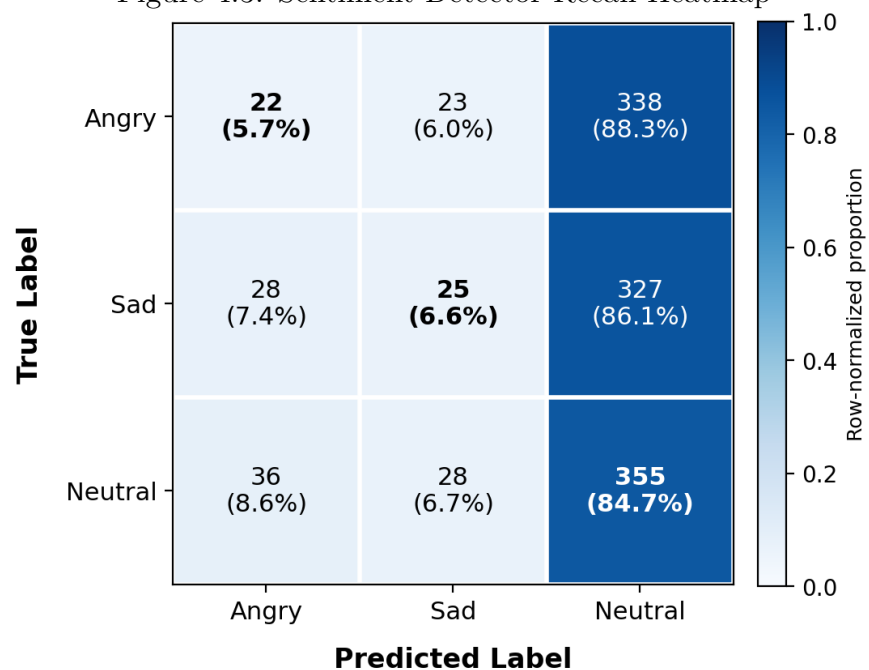
The sentiment detector, evaluated on Whisper-generated transcripts of the 1,182 individual-test clips that produced usable transcripts, reached only 34.01% accuracy – close to the 33.3% chance level for a three-class problem – with a macro-averaged F1 of only 23.23% (as seen in Table 4.3). Figure 4.3 shows why: the detector predicted *neutral* for 1,020 of the 1,182 evaluated clips (86.3% of all predictions) regardless of the clip’s true label, and recall for the two distressed classes was correspondingly poor (5.7% for *angry*, 6.6% for *sad*). This is the expected behavior of a pre-trained, non-fine-tuned sentiment model applied to short, often single-sentence acted utterances from datasets such as RAVDESS and CREMA-D, whose lexical content is frequently emotion-neutral even when the speaker’s vocal delivery is not. The model has no acoustic information to draw on and therefore has little lexical signal to work with on this kind of data. Average latency was 14.0 ms. It is noteworthy to point out that this excludes the cost of the Whisper transcription step itself, which was run once over the test set and cached rather than timed as part of the per-utterance budget.

This was done to reflect the production environment, where the transcript would be passed to the sentiment detector from the IVA.

Table 4.3: Sentiment Detector Performance Statistics

Emotion	Precision	Recall	F1 Score
Angry	0.256	0.057	0.094
Sad	0.329	0.066	0.110
Neutral	0.348	0.847	0.493
Average	0.311	0.323	0.232

Figure 4.3: Sentiment Detector Recall Heatmap



This result is consistent with, and motivates, the narrow role sentiment plays in the rule-based fusion scheme described in Chapter 3: rather than contributing a uniformly weighted vote alongside the two acoustic detectors, sentiment is confined to specific arbitration cases, and its near-chance standalone accuracy is the empirical justification for that design choice rather than an incidental finding.

4.2 Fusion Results

All three fusion strategies were evaluated on the same fully held-out ensemble test split (1,188 clips), distinct from the individual test split and from the data used to train any meta-classifier, by the four-way data split described in Chapter 3.

4.2.1 Three-Way Fusion: Rule-Based vs. Stacked

The primary, rule-based fusion system reached 73.06% accuracy with an F1 score of 73.25% and a combined latency of 48.9 ms per clip. The learned, stacked three-way alternative reached a closely comparable 72.90% accuracy with an F1 score of 73.23% and a latency of 49.7 ms. The two systems are, in practice, statistically indistinguishable on this test set, and their performance statistics in Tables 4.4 and 4.5 show the same qualitative pattern as the base detectors. *Angry* is detected with high reliability (F1 of 85.83% for rule-based fusion and 84.49% for stacked fusion) while the *sad/neutral* boundary remains the dominant source of error for both. Figure 4.4 shows that 155 of 382 true *sad* clips (40.6%) were predicted *neutral* under the rule-base fusion approach – a higher *sad*→*neutral* confusion rate than either acoustic detector showed standing alone, partially offset by a corresponding improvement in *neutral* recall (333 of 421, 79.1%, versus 287/420 for vocal landmarks and 282/420 for tone). The *sad/neutral* confusion boundary for the stacked fusion system can be seen in Figure 4.5. While it’s recall for *sad* clips rose 8.4% above that of the rule-based fusion model, its *neutral* recall dropped by 7.6% when compared to the same model.

Table 4.4: Rule-Based Fusion Performance Statistics

Emotion	Precision	Recall	F1 Score
Angry	0.893	0.826	0.858
Sad	0.767	0.568	0.653
Neutral	0.607	0.791	0.687
Average	0.756	0.728	0.733

Table 4.5: Stacked Fusion Performance Statistics

Emotion	Precision	Recall	F1 Score
Angry	0.871	0.821	0.845
Sad	0.709	0.652	0.679
Neutral	0.635	0.715	0.673
Average	0.738	0.729	0.732

Figure 4.4: Rule-Based Fusion Recall Heatmap

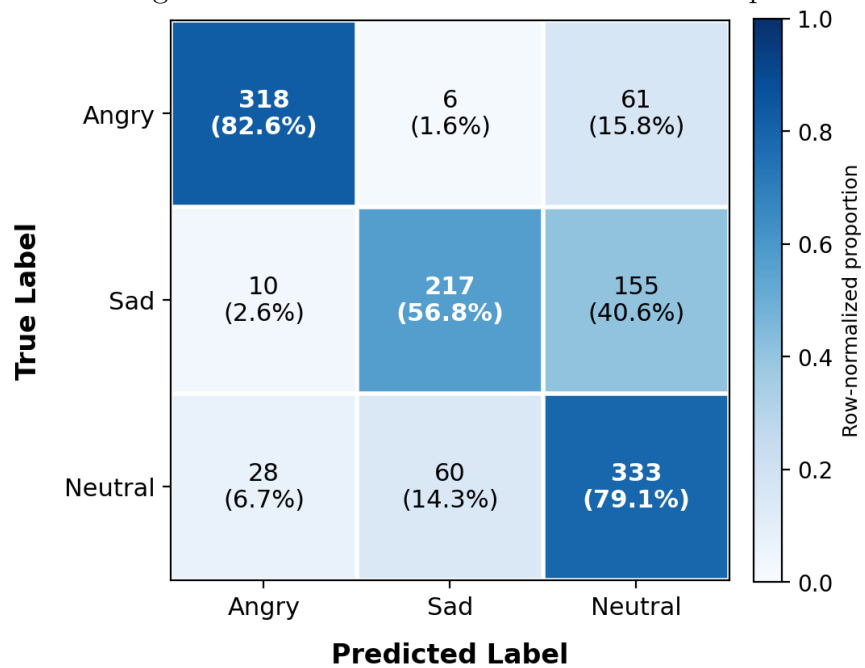
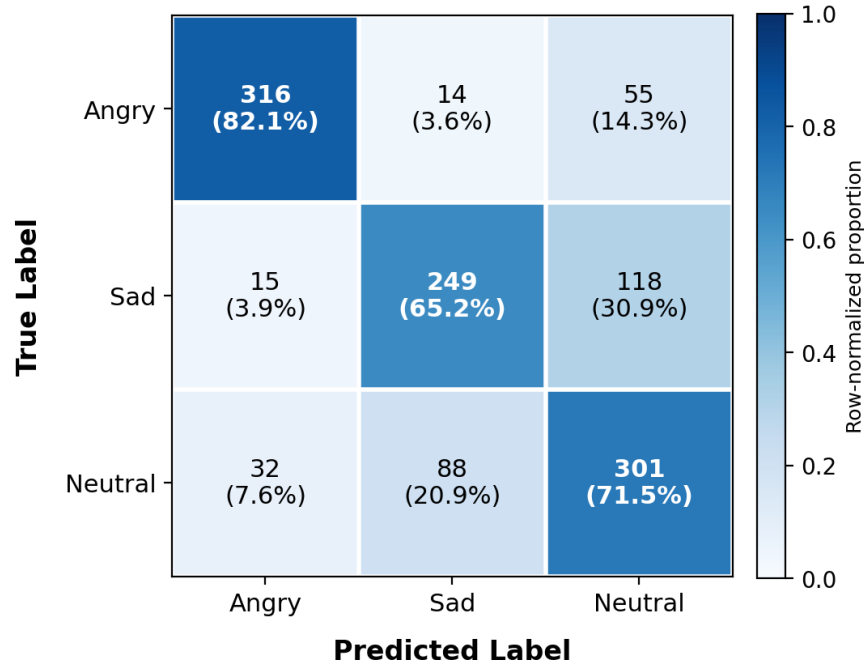


Figure 4.5: Stacked Fusion Recall Heatmap



Both fusion strategies see modest accuracy improvement over either acoustic detector in isolation (70.51% for vocal landmarks, 72.11% for tone) and substantial improvement over sentiment in isolation (34.01%). Each models sees roughly double the latency of the slower acoustic detector alone, since both base detectors and, for transcribed clips, the sentiment detector must all run before fusion can occur. Of the 1,188 ensemble-test clips, 1,182 (99.5%) had a usable Whisper transcript and therefore an active sentiment contribution. The remaining six were fused from vocal landmarks and tone alone.

4.2.2 Bimodal Stacked Fusion (Acoustic-Only)

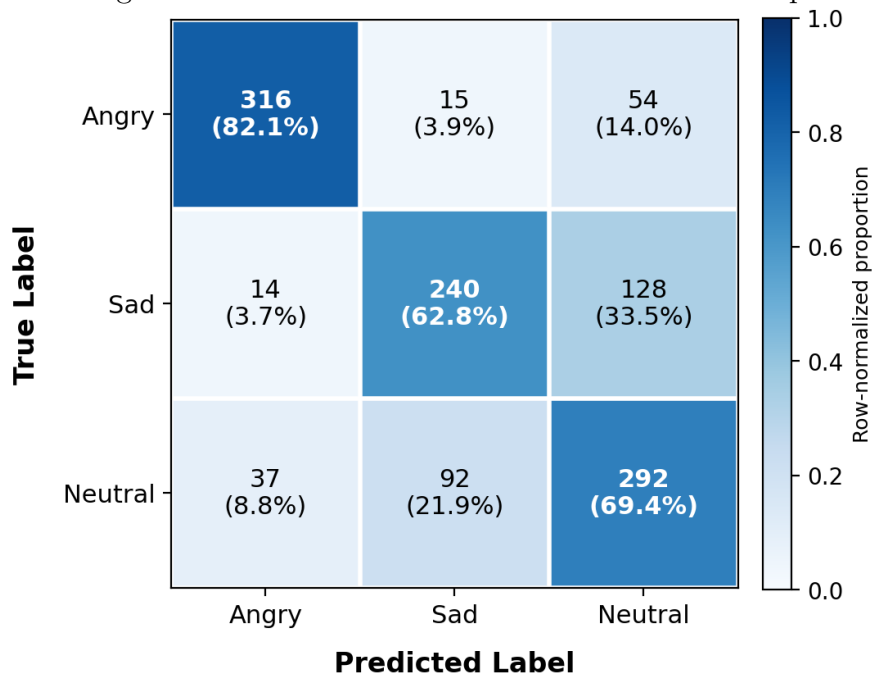
The bimodal stacked ensemble, which fuses only the vocal landmark and tone detectors' outputs with no sentiment or transcript dependency, reached 71.38% accuracy ($F1 \approx 71.7\%$) at approximately 30 ms per clip – between the standalone accuracy of the two acoustic detectors and the full three-way fusion results, and at roughly 60% of the three-way systems' latency. This result is relevant to deployment contexts in which ASR output is unavailable, unreliable, or deliberately excluded for privacy reasons (Chapter 3): it shows that removing

the sentiment branch costs a small but measurable amount of accuracy (1.5–1.7 percentage points relative to the three-way systems) while reducing both latency and the system’s dependency surface. The performance statistics and recall heatmap for the bimodal stacked ensemble model can be seen in Table 4.6 and Figure 4.6

Table 4.6: Bimodal Stacked Fusion Performance Statistics

Emotion	Precision	Recall	F1 Score
Angry	0.860	0.820	0.840
Sad	0.690	0.630	0.660
Neutral	0.620	0.690	0.650
Average	0.720	0.710	0.720

Figure 4.6: Bimodal Stacked Fusion Recall Heatmap



4.2.3 Overall Model Comparison

Figures 4.7 and 4.8 sum up the overall performance of the predictors and models covered in this section. Of the individual detectors, sentiment scored the lowest and F0 scored the highest. When it comes to the fusion models, they all scored within 1.7 points of each other. Rule based fusion achieved the highest accuracy, at 73.1%, but stacked fusion was not far behind. Bimodal stacked fusion scored just below the accuracy of the F0 detector on it's own. The latency figures show what would be expected from the preceding discussion. Tone is by far the lowest, given it's minimal calculations. The stacked and rule based fusion models have latency figures that are roughly equivalent to the sum of the latencies of the individual predictors while, of course, the bimodal stacked fusion model has the lowest latency of the ensemble models, due to excluding one of the individual detectors. In the end, rule based fusion comes out on tope with the highest accuracy percentage and a more than acceptable latency, at less than half the pre-determined maximum latency rate.

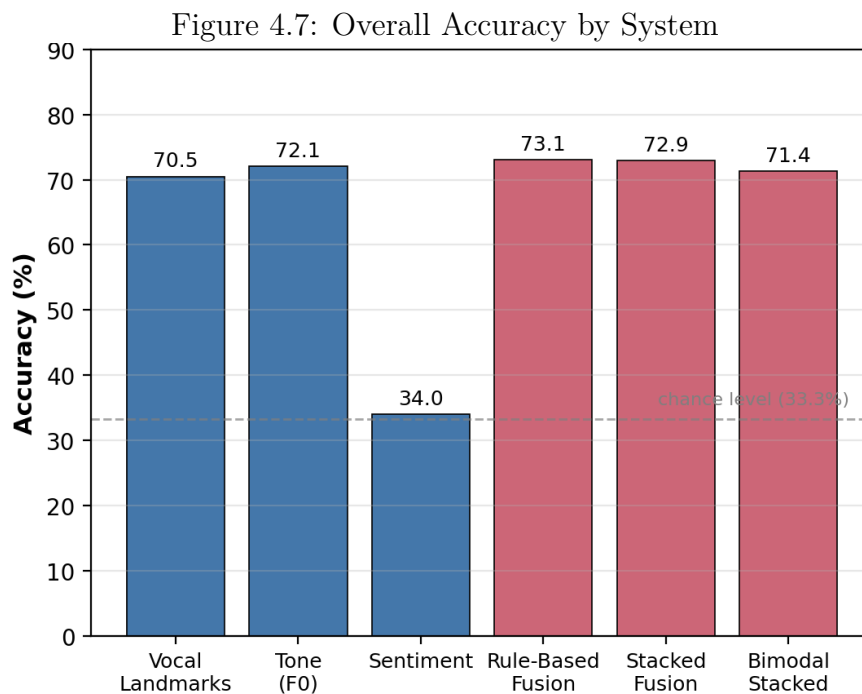
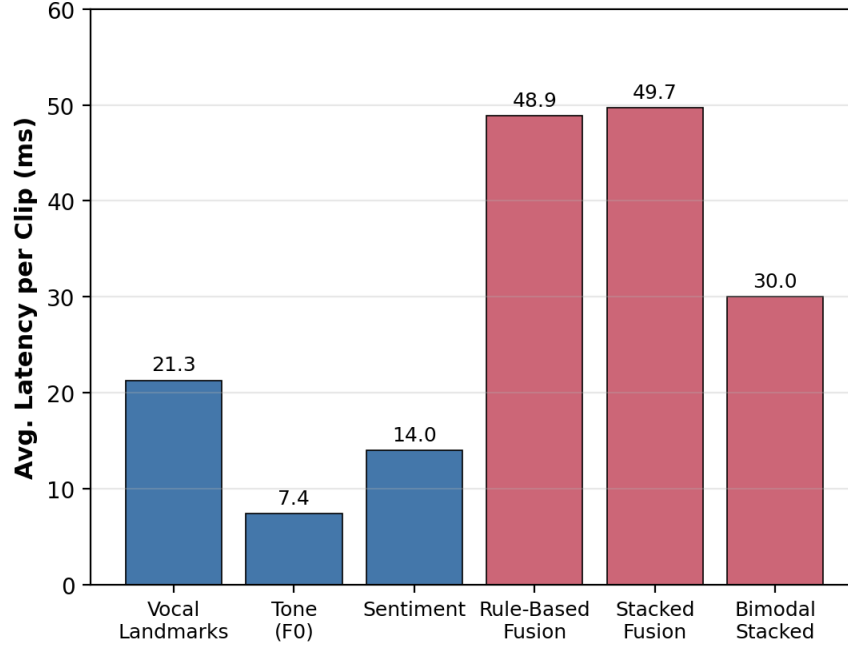


Figure 4.8: Overall Latency by System



4.2.4 A Weighting Defect in Rule-Based Fusion

The rule-based fusion system’s per-detector weights are derived from each detector’s own measured per-class F1 scores (Chapter 3, Section 3.2), so that a detector contributes to the fused decision in proportion to how well it has actually been shown to perform. Two training reports from early in the evaluation process (2026-06-15) show this derivation behaving incorrectly: the weights actually applied to the fused decision assigned vocal landmarks 22.5%, tone 25.3%, and **sentiment 52.1%** – despite sentiment’s standalone accuracy of roughly 34–40% on those same runs, well below either acoustic detector. The resulting fused accuracy on those runs was 52.4% and 50.3% respectively: *worse than either acoustic detector achieved on its own*, and barely above chance.

By the 2026-06-17 training run, the defect was corrected: the erroneous duplicate weight-derivation path was removed, leaving a single, correctly normalized weight calculation. The corrected weights for that run were 48.3% vocal landmarks, 46.7% tone, and 5.0% sentiment – closely tracking the detectors’ actual relative standalone accuracies – and fused accuracy rose to 72.7%, finally exceeding both acoustic detectors individually as the fusion mechanism

was intended to. The latest training run reproduces this corrected behavior (45.0% / 50.1% / 4.9%, with 73.1% fused accuracy) and is the run reported as the system’s current performance throughout this chapter.

This episode is reported here in some detail because it is methodologically informative beyond the specific bug: a fusion layer is not guaranteed to improve on its inputs, and an incorrectly calibrated fusion layer can perform substantially *worse* than the weaker of its own inputs by over-trusting a low-accuracy detector. The corrected weighting scheme’s success is therefore better understood as having been verified empirically across multiple runs, rather than assumed from the rule-based design alone.

4.3 Feature Vector Selection

The feature vectors ultimately used by the vocal landmark and tone detectors (Chapter 3, Section 3.2) were not fixed in advance; they were arrived at over a sequence of training and feature-selection runs using the `test_model_training.py` experiment script. Each subsection below corresponds to one of the four phases of that script. Where more than one run of a given phase exists, the most recent run is reported as authoritative, and the trend across runs is reported only when the change between runs is the relevant finding.

4.3.1 Phase 1: Classifier Comparison

XGBoost was compared against Random Forest and a Support Vector Classifier on each module’s then-current feature vector. In the most recent run of each, XGBoost outperformed both alternatives: 72.08% vs. 70.42% (Random Forest) and 69.40% (SVC) for vocal landmarks, and 74.23% vs. 73.21% (Random Forest) and 72.62% (SVC) for tone. This margin was consistent across every Phase 1 run performed over the course of development, not just the most recent one, and is the basis for XGBoost’s selection as the production classifier for both detectors (Chapter 3, Section 3.2).

Figure 4.9: Phase 1 Vocal Landmark Classifier Comparison

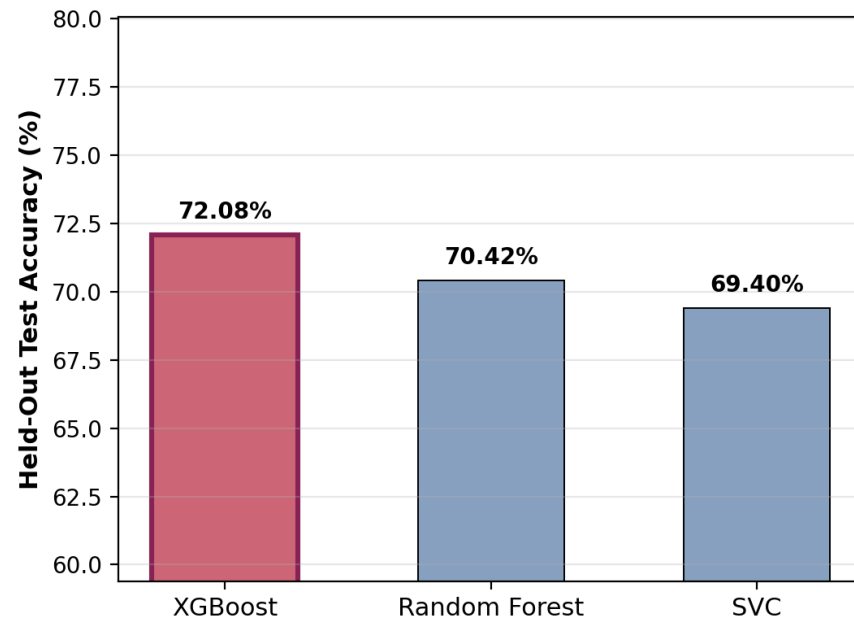
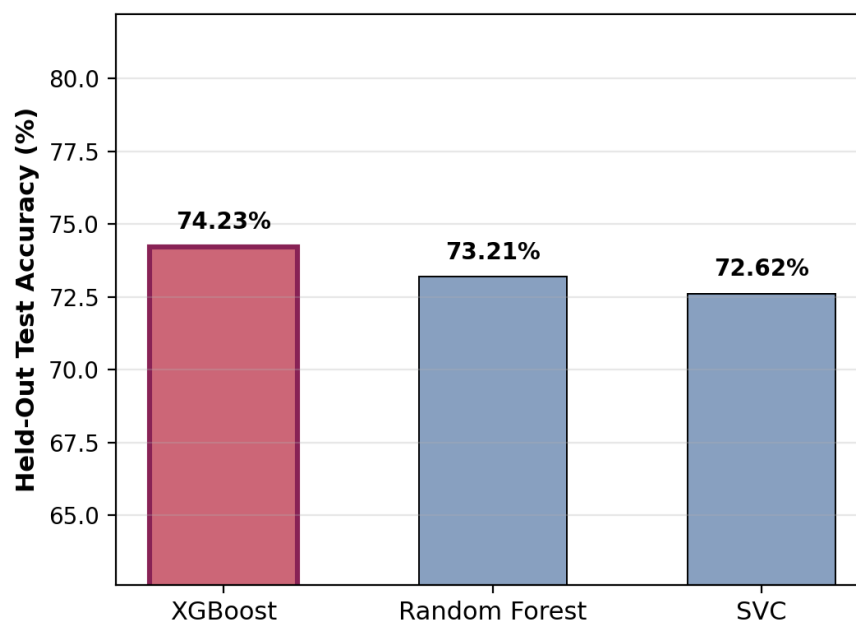


Figure 4.10: Phase 1 Tone (F0) Classifier Comparison



4.3.2 Phase 2: Feature Selection

Phase 2 ran the Phase 1 winner (XGBoost) against an expanded candidate feature pool through four selection methods: forward stepwise selection, L1 (Lasso) logistic regression, recursive feature elimination, and principal component analysis. It then reported each method’s cross-validated accuracy. In the most recent run, Lasso was the nominal winner for both modules: 72.06% (16 of 18 candidate features) for vocal landmarks and 74.49% (15 of 16 candidate features) for tone, both narrowly ahead of forward stepwise selection (72.44% and 73.89% respectively). PCA was consistently the weakest method for both modules across every run performed (68.75% and 69.85% in the most recent run), which is unsurprising given that several of the project’s hand-engineered features – the landmark bigram/trigram counts in particular – are categorical-like counts whose discriminative content does not survive projection onto continuous, variance-maximizing components as cleanly as it survives explicit retention.

Neither module’s production feature set is the literal output of Phase 2’s nominal winner, and the reasons differ for each module.

Vocal Landmarks: From 33 Candidate Features to 18

The vocal landmark candidate pool changed more than three times over the course of development, and the production 18-feature vector described in Chapter 3 is the result of these many changes, not of a single feature-selection run.

The earliest candidate pool (33 features, evaluated only on RAVDESS and CREMA-D before IEMOCAP was added to the corpus) included four spectral-shape descriptors – `spectral_flux`, `spectral_centroid_mean`, `spectral_rolloff`, and `spectral_bandwidth` – alongside the landmark-specific features. These spectral-shape values were added for comparison and ranked among the most important features by every selection method in that run. Their contributions were paramount in achieving a respectable 76.12% XGBoost test accuracy on that smaller, 4,301-clip dataset. They were nonetheless removed in the very

next run, before any larger feature-selection comparison was run on them, and replaced with four band-energy-at-landmark features (Chapter 3, Section 3.2) computed only at the frames where a specific landmark event occurs. This was a deliberate, principle-driven substitution rather than a performance-driven one: spectral-shape descriptors of this kind are conceptually adjacent to MFCCs, which the vocal landmark detector is specifically intended not to depend on, since the entire premise of landmark-based analysis is that discrete articulatory events, not a continuous spectral envelope, are doing the discriminative work (Chapter 3, Section 3.2; [15]). The substitution cost a small amount of measured accuracy on the same dataset (76.12% $\rightarrow$ 73.98%) and was accepted regardless.

The candidate pool was reduced a second time, from 31 to 24 features, coinciding with IEMOCAP being added to the training corpus (4,301 $\rightarrow$ 8,196 clips after balancing). Several inter-landmark-interval statistics and density-trend features (`ili_skewness`, `ili_cv`, `density_slope`, `density_end_vs_start`, `max_local_density`) were dropped at this stage for consistently ranking near the bottom of every feature-importance and selection-method report up to that point. Both bigram and trigram feature counts were reduced, as these features can become redundant in short audio clips where only a certain amount of combinations exist. A final reduction from 24 to 18 features removed the remaining redundant n -gram variant and several of the lowest-ranked rate and interval statistics, arriving at the feature vector specified in Chapter 3.

It is worth reiterating that many of the vocal landmark derived features performed better than the raw information from the landmarks themselves, revealing that the order of landmarks and notation of what is happening at the landmarks creates a more complete emotional picture. Throughout this entire trajectory, one pattern was stable: the landmark *n-gram* features – normalized counts of the two most frequent landmark bigrams and trigrams – consistently outranked the individual landmark *rate* features (`g_rate`, `b_rate`, `s_rate`, `p_rate` considered separately) in every Phase 1 and Phase 2 importance ranking produced. In the most recent 18-feature run, for example, `bigram_0`, `bigram_1`, `trigram_0`,

and `trigram_1` together account for roughly 23% of total XGBoost feature importance, more than double the combined importance of all four individual rate features (`g_rate`, `b_rate`, `s_rate`, `p_rate` together account for under 11%). This is consistent with the finding from the landmark literature motivating the n-gram features’ inclusion in the first place (Chapter 3, Section 3.2): the *sequence* in which landmark types occur carries more diagnostic information than how frequently any single landmark type occurs on its own [15]. The retention of both bigram and trigram features, rather than only one or the other, despite the resulting redundancy that later selection rounds had to prune around, follows directly from this finding, and the persistence of the pattern across every dataset size and feature-pool configuration tested gives it more weight than a single run would.

Tone: A Spectral-Feature Regression and Its Correction

The tone detector’s candidate pool shows a different, and less flattering, trajectory: a period during which the F0/time-domain-only design principle stated in Chapter 3 was not actually being followed in the experiment script, followed by a correction.

The earliest tone candidate pool (29 features) included five unrelated descriptors that had leaked in from adjacent testing. These were `spectral_centroid_mean`, `spectral_centroid_std`, `spectral_bandwidth`, `spectral_rolloff` and `loudness_std`, none of which are F0-derived or simple time domain statistics. On a randomized test dataset, `loudness_std` was in fact the single most important feature by XGBoost’s own ranking in that run, ahead of every F0 statistic, and the full 29-feature pool reached 76.82% accuracy. A subsequent attempt to prune this pool to 18 features retained `spectral_centroid_mean`, `spectral_centroid_std`, `spectral_bandwidth`, and `loudness_std` rather than removing them, and reached a comparable 76.96–77.16% across two runs – in other words, the spectral-shape and loudness features were never actually excluded during this phase of development, despite the tone module’s stated design intent of using only F0-derived and time-domain features with no spectral-shape content (Chapter 3, Section 3.2).

This was corrected in the next candidate pool (24 features), which removed every `spectral_*` and `loudness_*` term entirely, restoring the pool to F0, energy, and time-domain descriptors only, and reached 75.01% accuracy on the same randomized test dataset – a roughly two-percentage-point reduction relative to the spectral-inclusive 18-feature pool it replaced. A final pruning step to 16 features, removing several lower-importance F0-contour statistics, reached 74.23%. The production tone feature vector specified in Chapter 3 (26 computed features, 22 active) is a separately curated superset of this corrected, spectral-free candidate pool rather than its literal output – it restores a small number of additional F0-contour and normalized-F0 statistics that the smaller experimental pool excluded for tractability, while preserving the same exclusion of spectral-shape and loudness terms established at this stage.

The roughly two-point accuracy cost of removing the spectral and loudness features is reported plainly because it is a real trade-off, not a null result: those features were measurably useful to a generic XGBoost classifier on this dataset, and removing them was a decision made on the grounds of what the tone detector is supposed to represent – prosodic, F0-centered information conceptually distinct from the vocal landmark detector’s features and from conventional MFCC-style spectral representations – rather than on the grounds of what scored highest. It is also worth noting here that the 4-phase testing framework utilizes smaller training and testing groups, which can lead to different accuracy percentages when run on the entire dataset.

4.3.3 Phase 3: Label-Space Grouping

Phase 3 evaluated five candidate label-grouping schemes against the project’s eventual three-class angry/sad/neutral baseline (denoted G5), using each module’s Phase 2 feature set retrained on the full extended emotion set available across all three corpora (angry, sad, neutral, happy, fearful, disgust). In the most recent run of each module, the three-class baseline (G5) outperformed every alternative grouping by a wide margin: 70.67% for vocal

landmarks and 75.21% for tone, against a best alternative of 62.87% (vocal landmarks, G1) and 64.99% (tone, G1). The two four-class schemes that kept *happy* as a separate category (G3, G4) performed worst of all (51–60% across both modules), consistent with *happy* clips being acoustically heterogeneous across the three source corpora and difficult to separate from *angry* in particular, since both are high-arousal states.

Figure 4.11: Phase 3 Vocal Landmark Grouping Scheme Comparison

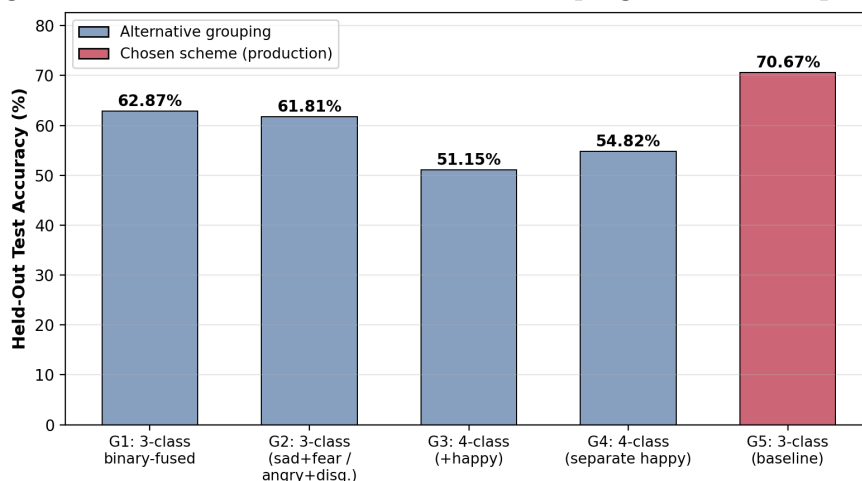
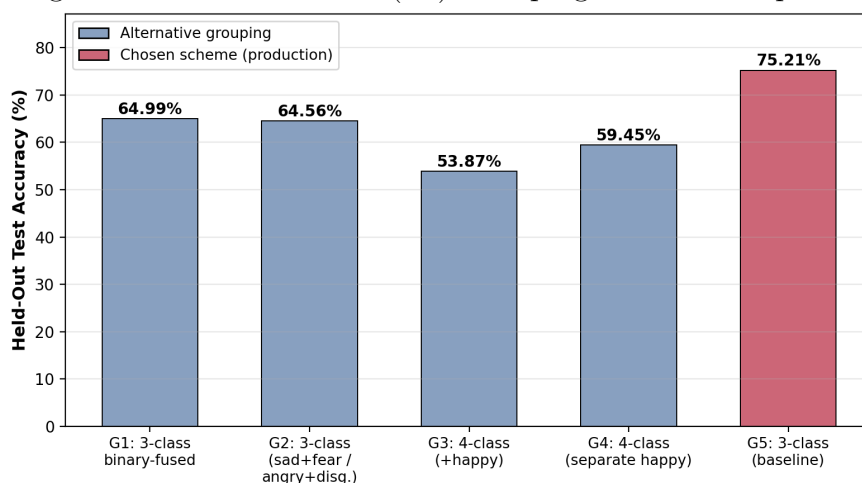


Figure 4.12: Phase 3 Tone (F0) Grouping Scheme Comparison



4.3.4 Phase 4: Sad-vs-Neutral Deep Dive

Because *sad* and *neutral* were identified, in every result reported above, as the most persistently confusable pair for both acoustic detectors, Phase 4 isolated this pair as a binary classification problem and re-ran all four Phase 2 selection methods on it specifically. In the most recent run of each module, forward stepwise selection was the best-performing method for vocal landmarks (74.00% binary accuracy, 9 of 18 features), and was narrowly outperformed by Lasso for tone (74.97% vs. 74.94% for forward stepwise, on a 15-of-16-feature candidate pool). PCA was again the weakest method for both modules (71.83% and 72.33% respectively).

These binary accuracies (74–75%) are noticeably higher than the sad-class and neutral-class F1 scores reported for the full three-class detectors in Section 4.1 (63–69%), which is expected: a classifier asked to distinguish only two classes is not penalized for confusing either with *angry*, and removing that third option simplifies the decision boundary. The persistence of a roughly 25% binary error rate even in this simplified setting indicates that the sad/neutral boundary is a genuinely difficult acoustic discrimination on this data, rather than an artifact of the three-class setup, and is consistent with the literature’s broader observation that low-arousal emotional states are acoustically closer to each other than high-arousal states are to anything else (Chapter 1; [5]).

4.3.5 The Joint Feature-Level Model

The exploratory joint feature-level model (Chapter 3, Section 3.2) concatenates all three detectors’ raw features – 18 vocal landmark, 26 tone, and 3 sentiment-probability columns, 47 in total – and trains a single XGBoost classifier on the combined vector, rather than combining three detectors’ opinions as the fusion strategies above do. Its own internal feature-selection experiment (run independently of, and using a different candidate pool than, the per-module Phase 2 experiments above) provides the clearest single illustration in this project of statistically best not meaning production-chosen.

Three configurations of this model were evaluated on an identical 2,402-clip held-out test set. The full, unselected 47-feature vector reached 78.43% accuracy – the highest accuracy recorded anywhere in this chapter for any system, fusion or otherwise. An aggressively pruned 14-feature configuration, built from the feature-selection experiment’s smallest selected subset, reached only 73.23% – a five-percentage-point drop, indicating that this particular cut removed features still carrying real discriminative signal. The configuration actually shipped in production uses 27 of the 47 features and reaches 77.35%: lower than the unselected ceiling, higher than the over-pruned alternative, and reasonably close to both. The 27-feature set is a hand-adjusted superset of the feature-selection script’s own top-ranked forward-stepwise subset (23 features, 75.94% cross-validated), rather than that subset taken verbatim; four additional features with high standalone XGBoost importance, but excluded by forward stepwise selection’s greedy search at the 23-feature point, were added back in by hand. This raised the model from the stepwise result to within roughly one accuracy point of the full 47-feature ceiling while still removing 20 of the 47 raw columns, accepting a small accuracy cost relative to the absolute best score observed in exchange for a meaningfully smaller, more interpretable feature space. This particular model was created to see what could be done with all of the features when combined. However, it was never the intent of this research to use such a large feature vector. A general MFCC feature vector is 39 features and the goal was to keep all feature vectors smaller than that, if possible.

4.3.6 Summary

While automated feature selection techniques are quite helpful in pruning feature vectors, they don't always choose the most appropriate group of features for an individual scenario. For each case in this research, the configuration that scored highest in a Phase 2 cross-validation run, a Phase 4 binary deep-dive, or the joint model's own selection script, was not the configuration that shipped. For vocal landmarks, the divergence is mechanical: production retains every computed feature rather than the 22-of-24 subset Lasso preferred, because the redundant candidates Lasso was choosing among had already been removed from the pool by the time the 18-feature production vector was finalized. For tone, the divergence reflects a design constraint applied irrespective of measured performance: no candidate pool containing spectral-shape or loudness features was ever allowed to become the production set, regardless of how well such features scored, once the spectral-feature regression described above was identified and corrected. For the joint model, the divergence is a direct, quantified accuracy-versus-parsimony trade-off, made visible by evaluating all three configurations on the same held-out data rather than reporting only the winner. None of these three divergences is a case of the experiment script being wrong. In each case the experiment script faithfully reported what it was asked to evaluate, and the production configuration reflects a judgment applied on top of that report, taking other factors like memory and latency into account.

4.4 Latency Analysis

Chapter 1 establishes a real-time constraint of sub-100ms per-utterance latency as a design objective, and Chapter 3 budgets each individual detector to complete well within the time available between the end of a user utterance and the point at which a virtual assistant would otherwise begin generating its response. Every system evaluated in this chapter meets this constraint with substantial margin: the slowest single configuration measured, the three-way stacked fusion system, averages 49.7 ms per clip, under half of the 100ms ceiling.

The latency ordering across systems follows directly from architectural choices already discussed in Chapter 3. Tone (7.4 ms) is the fastest individual detector by a wide margin, a direct consequence of its restriction to F0 and time-domain features with no spectrogram computation. Vocal landmarks (21.3 ms) is slower because it requires a multi-band landmark-detection pass. Sentiment (14.0 ms, excluding the cached Whisper transcription step) sits between the two acoustic detectors. Fusion latency is, as expected, approximately additive: the three-way systems (48.9–49.7 ms) cost roughly the sum of all three base detectors plus a small fusion overhead, while the bimodal acoustic-only system (30 ms) costs roughly the sum of vocal landmarks and tone alone, confirming that the fusion logic itself – whether the hand-specified rule-based arbitration or the learned meta-classifier – contributes negligible latency relative to feature extraction. The latency panel of the accuracy/latency comparison figure introduced in Section 4.2 is the relevant reference for this discussion.

Because the sentiment detector’s reported latency excludes ASR, the latency figures for any fusion strategy that includes sentiment likewise understate the true cost of a system that must perform speech-to-text before fusion can complete, in a deployment context where the host assistant does not already produce a transcript as a byproduct of its own pipeline. Whisper-*tiny* transcription was run once, offline, over the test sets specifically so that evaluation would not be bottlenecked by transcription speed; Chapter 3 notes this caching explicitly, and it is repeated here because it directly affects how the latency numbers in this section should be read in the context of a live deployment.

4.5 Discussion

4.5.1 Relating Results to the Research Objectives

Chapter 1 sets out three objectives for this research. The first – developing a real-time, multimodal mood detection tool with sub-100ms per-utterance latency – is met with considerable margin by every system evaluated, as shown in Section 4.4; even the slowest fused configuration completes in under half the target budget. The second objective, validating

vocal landmarks as a viable feature representation for general mood detection beyond their established use in depression detection, is supported but not unambiguously confirmed by these results. Vocal landmarks reach 70.51% standalone accuracy, which is comparable to but consistently slightly below the tone detector’s 72.11%, despite the tone detector relying on a substantially simpler and cheaper feature set. Vocal landmarks do not establish themselves as a clearly superior representation in this evaluation. However, they do establish themselves as a workable, real-time-compatible one whose primary practical advantage – a small, interpretable, 18-dimensional feature vector built from discrete acoustic events – is conceptual and architectural rather than a raw accuracy advantage over the simpler F0-based alternative evaluated alongside it. The third objective, establishing baseline performance benchmarks for the combined fusion approach to support future replication, is addressed directly by the quantitative results in Sections 4.1 through 4.4: a three-way rule-based fusion accuracy of 73.1% at 48.9 ms, against base detector accuracies of 70.5%, 72.1%, and 34.0%, on a stratified, class-balanced, three-corpus dataset with a fully specified train/test split, gives a concrete and reproducible starting point for any future work that substitutes a different detector, dataset, or fusion mechanism into this architecture.

4.5.2 Relating Results to the Literature Gaps

Chapter 2 identifies poor generalization from acted, exaggerated training corpora to spontaneous conversational speech as a persistent, largely unresolved problem in speech emotion recognition, and notes that systems evaluated on naturalistic corpora such as Switchboard-Affect tend to show particularly poor performance for *anger* specifically [25]. The results in this chapter cannot directly speak to this gap, since IEMOCAP – the one corpus among the three used here with substantial spontaneous, dyadic content – was combined with the more clearly acted RAVDESS and CREMA-D corpora rather than evaluated as a separate held-out generalization test, and *angry* was in fact the most reliably detected class throughout this evaluation rather than the least, which may say more about the acted-speech composi-

tion of the combined training distribution than about the generalizability of the underlying acoustic features.

Chapter 2 also identifies the field’s lack of clear frameworks for determining when and how a virtual assistant should modulate its tone, citing survey evidence that users do not want assistants to emote at all times and that defaulting to neutral under low prediction confidence may be preferable to applying emotional modulation indiscriminately [2]. The present results bear on this indirectly: the *neutral* class is, across every system evaluated, never the best-detected class, and is frequently confused with *sad* in both directions. A response modulation module built on top of this mood detection module (left, per Chapter 3, to future work) would therefore inherit a non-trivial false-positive rate on the *sad* label specifically – cases where a genuinely neutral utterance is classified as *sad* and could trigger an unwarranted tone shift. The `MoodResult` contract’s confidence score and `is_distressed` flag (Chapter 3, Section 3.2) exist specifically to give any downstream response-modulation logic the information needed to default to neutral behavior when this kind of ambiguity is present, consistent with the survey guidance above, but evaluating whether a given confidence threshold actually achieves this in practice is a question for future work rather than for this chapter.

4.5.3 Limitations

Several limitations of the evaluation reported in this chapter should be stated plainly rather than left implicit.

First, as noted above, the training and test data does not isolate acted from spontaneous speech as separate evaluation conditions. All results are reported on a combined, balanced pool of all three corpora, stratified only on emotion label. The strong standalone performance of *angry* reported throughout this chapter may not transfer to a deployment context dominated by spontaneous, non-acted anger, and this evaluation does not contain a held-out condition capable of measuring that gap directly.

Second, several of the comparisons drawn within Section 4.3 – specifically, the accu-

racy changes associated with adding IEMOCAP to the corpus mid-development – are confounded with simultaneous feature-set changes in the surrounding development history, and are flagged as such at the point each comparison is made rather than presented as fully controlled ablations. A cleaner, fully controlled ablation (identical feature set, identical classifier, dataset size as the only varying factor) was not separately run and would be a natural addition to any continuation of this feature-selection work.

Third, the sentiment detector’s reported 34.01% accuracy reflects a pre-trained, non-fine-tuned HuggingFace model applied to short, frequently lexically neutral acted-speech transcripts. This is a meaningful and informative result for the specific question this project asks (whether a generic, off-the-shelf sentiment model contributes useful signal to this fusion architecture without further training), but should not be read as an upper bound on what text-based sentiment analysis can achieve on emotionally expressive speech in general, since a fine-tuned model or a corpus with more emotionally explicit lexical content would likely narrow this gap considerably.

Fourth, the rule-based fusion weighting defect documented in Section 4.2 was caught by this project’s own evaluation process specifically because per-detector and per-fusion-strategy accuracy figures were being compared against one another on every run. A deployment or evaluation methodology that reported only the final fused accuracy, without the base-detector figures to compare it against, would not have surfaced this defect as readily. This is offered not as a criticism specific to this project but as a general argument for why base-detector and fusion-level metrics should be reported and compared together, as Chapter 3 specifies they are designed to be (Section 3.6), rather than fusion accuracy being reported in isolation.

Chapter 5

Conclusion and Future Work

5.1 Summary of the Research

This research set out to address a specific, narrowly scoped gap identified in Chapter 1: conversational virtual assistants have no lightweight, privacy-preserving way to detect a user’s emotional state in real time and use that signal to keep the assistant’s tone from clashing with it. The goal was never to make an assistant that improves a user’s mood – that ambition belongs to a different, already well-populated area of industry research, as discussed in Chapter 2. The goal here was the smaller and, in the authors’ view, more honest problem of simply not making things worse: recognizing when cheerful, upbeat defaults are the wrong response, without pretending to fix what the assistant has no business trying to fix.

To work toward that goal, this research built IAMDIVA’s mood detection module: three independent, swappable detectors – vocal landmarks, F0, and text-based sentiment – combined through a fusion layer that arbitrates between them. Every architectural choice described in Chapter 3, from the three-class label space to the rule-based fusion logic to the deliberate exclusion of computer vision, followed from the same constraint: this needed to be something a real assistant on real consumer hardware could actually run, not a research-only system that only works on a development machine.

5.2 Conclusions

Measured against the three objectives set out in Chapter 1, the results reported in Chapter 4 support three conclusions:

First, the real-time objective was met cleanly, and with room to spare. Every system

evaluated – single detector or full fusion – finished well inside the sub-100ms budget this research set for itself. This was not a close call decided by Chapter 4’s numbers. The margin held consistently, which is a finding in itself. It suggests that discrete, low-dimensional acoustic representations are a genuinely viable alternative to heavier spectrogram- or deep-learning-based pipelines for this kind of task, not merely a workaround forced by hardware limits.

Second, the vocal landmark objective produced a more interesting answer than a simple pass or fail. Landmarks did not prove themselves more accurate than the simpler F0-based detector they were evaluated alongside. What they did prove is that a feature representation built almost entirely around depression detection in the existing literature transfers, intact, to a broader real-time mood classification task, while staying small, interpretable, and fast enough to matter. That is a real result. It is just a different kind of result than “landmarks won.”

Third, this research leaves behind something future work can actually build on: a fully specified, reproducible baseline, measured on a stratified three-corpus dataset with a documented train/test split, rather than a one-off number that cannot be checked or extended. Whatever direction the work takes next, it has somewhere concrete to start from and something concrete to improve against.

None of this amounts to a dramatic discovery. It amounts to a working system, an honestly reported set of trade-offs, and a foundation solid enough to be worthy of future expansion.

5.3 Future Work

The path forward from here falls into three groups, ordered roughly by how directly each one builds on what already exists: first, work that makes the mood detection module itself better; then the response modulation module this research was always intended to lead into; and finally the integration and real-world validation work that would test all of it outside of

a held-out test split.

5.3.1 Bettering the Mood Detection Model

The most immediate opportunities for improvement lie inside the module that already exists, rather than in anything new.

The sentiment detector is the clearest case. Its near-chance standalone performance was informative precisely because it was a generic, off-the-shelf model never fine-tuned on this kind of data, and Chapter 4 is explicit that this result should not be read as a ceiling on what text-based sentiment can contribute. Fine-tuning a sentiment or emotion-classification model directly on transcripts of the same emotional speech corpora used elsewhere in this research – or on a corpus with more emotionally explicit lexical content than the largely neutral, scripted utterances in RAVDESS and CREMA-D – is a natural next step, and one that could let sentiment take on a less narrowly constrained role in fusion than the tiebreaker-only function it currently serves.

The sad/neutral boundary deserves equally direct attention. It was the dominant source of error for every detector and every fusion strategy evaluated in this research, and Chapter 4’s Phase 4 deep dive showed that this is a real acoustic discrimination problem on this data, not an artifact of the three-class setup. Closing this gap further likely means looking past the per-utterance feature vectors this research relied on and looking into vocal markers that were outside the scope of this research. This includes richer voice-quality descriptors and features that draw on context beyond a single isolated utterance – the kind of cross-utterance emotional trajectory that Chapter 1 identifies as missing from most current speech emotion recognition systems, this one included. Either direction is a reasonable continuation of the feature-selection groundwork already laid in Chapter 3.

A third, lower-priority direction worth naming is expanding the label space beyond the current three classes. This research deliberately narrowed to angry/sad/neutral specifically because accuracy degrades as more emotion categories are added, and Chapter 4’s Phase 3

results show that effect clearly. Revisiting that decision only makes sense once the sentiment and sad/neutral improvements above have meaningfully raised the ceiling this research is currently working against. Expanding the taxonomy before that point would likely just reintroduce the same accuracy cost this research deliberately avoided.

5.3.2 The Response Modulation Module

The response modulation module was part of this project’s original scope and was set aside specifically for time, not because it was deemed unimportant. Chapter 3 draws the boundary between the two modules explicitly for this reason, and it remains the most direct continuation of this research available.

The mood detection module was built with this handoff in mind from the start. The `MoodResult` contract’s confidence score and `is_distressed` flag exist specifically to give a future response modulation layer the information it needs to default to neutral, unmodulated behavior when the underlying emotion prediction is uncertain. This is consistent with survey evidence, discussed in Chapter 2, that users do not want assistants emoting indiscriminately. Chapter 4 is explicit that whether a given confidence threshold actually achieves this in practice was left as an open question, and it is the first one a response modulation module would need to answer: at what confidence level should the system choose to do nothing, and how should that threshold be tuned against the specific failure mode most likely to trigger it?

Beyond that threshold question, the module itself would need to define what “appropriate tone” actually means in implementation terms – how an assistant’s response content or delivery should shift, if at all, when the mood detection module reports distress, and how to avoid the kind of emotional overcorrection or manipulation that Chapter 2 flags as a real ethical hazard in this space. The output contract this research produced – a clean, JSON-serializable result requiring no change to the detection pipeline to consume – means this work can begin without needing to revisit anything described in Chapter 3 beforehand.

5.3.3 Integration, Benchmarking, and Real-World Validation

The work above stays inside the same evaluation conditions used throughout this research—held-out clips from acted and partly spontaneous speech corpora, on a development machine, with no live assistant in the loop. Closing that gap is its own line of future work, and a natural escalating sequence.

The first step is the integration this research was originally scoped around but did not pursue: wiring the mood detection module into LEON, or any other IVA, through the kind of clean input/output contract Chapter 3 describes. This research deliberately built that contract to require nothing from a host assistant beyond what it already has available mid-pipeline, specifically so that this step would not require re-architecting anything already built.

Once integrated, the next step is benchmarking on the kind of hardware this research was designed for but never tested on. Chapter 3 is explicit that the Mac Mini used for development and evaluation is not the deployment target – the real test of this research’s real-time and footprint goals is a smart speaker, phone, or comparable IoT device – not a 48GB unified-memory desktop machine.

The last and most demanding step is testing against real human emotional expression rather than acted or elicited speech. Every result in this research was produced on RAVDESS, CREMA-D, and IEMOCAP, and Chapter 4’s limitations are direct that this leaves the system’s performance on genuinely spontaneous, naturalistic emotion an open question rather than a settled one. User studies are the only way to answer that question, and they are the natural endpoint of the sequence this section lays out: from a better model, to a complete two-module system, to a system tested on the people it was built for.

5.4 Closing Statement

This research did not set out to solve speech emotion recognition, and it has not done so. What it has done is build a working, real-time, multimodal mood detection system;

validate, honestly and without overstatement, a feature representation borrowed from a different problem; catch and correct a real defect in its own fusion logic specifically because it was designed to be checked against itself; and leave behind a reproducible baseline and a clearly defined next module for whoever continues this work, including its own author.

Each one of these modest claims is something this research can actually stand behind, and each one points somewhere worth going next. A conversational assistant that simply stops making a bad moment worse may not be an earth-shattering achievement. It is, however, a real and useful one, and the work in this thesis is a genuine step toward it.

Bibliography

- [1] S. K. Khare, V. Blanes-Vidal, E. S. Nadimi, and U. R. Acharya, “Emotion recognition and artificial intelligence: A systematic review (2014–2023) and research recommendations,” *Information Fusion*, 2024.
- [2] T. Aggarwal and J. Goncalves, “Towards empathetically responsive voice assistants,” in *Proceedings of the 35th Australian Computer-Human Interaction Conference*, p. 669–678, Association for Computing Machinery, 2024.
- [3] A. Mari, A. Mandelli, and R. Algesheimer, “Empathic voice assistants: Enhancing consumer responses in voice commerce,” *Journal of Business Research*, p. 114566, Mar. 2024.
- [4] S. Jain, S. Basu, A. Ray, and R. Das, “Impact of irritation and negative emotions on the performance of voice assistants: Netting dissatisfied customers’ perspectives,” *International Journal of Information Management*, vol. 72, p. 102662, 2023.
- [5] L. Wang, H. Zhang, and D. Lin, “Multimodal analysis of vocal emotion and prosody in subclinical depressed individuals,” *Current Psychology*, Apr. 2025.
- [6] K. Machová, M. Szabóová, J. Paralič, and J. Mičko, “Detection of emotion by text analysis using machine learning,” *Frontiers in Psychology*, vol. 14, 2023.
- [7] Y. Zhang, Y. Ma, D. Fu, S. Z. Portales, M. Fjeld, and D. Kragic, “Personalizing emotion-aware conversational agents? exploring user traits-driven conversational strategies for enhanced interaction,” 2025.
- [8] S. Alimamy and M. A. Kuhail, “I will be with you alexa! the impact of intelligent virtual assistant’s authenticity and personalization on user reusage intentions,” *Computers in Human Behavior*, vol. 143, p. 107711, 2023.
- [9] A. Mishra, M. Li, and S. Deo, “Sentinellms: Encrypted input adaptation and fine-tuning of language models for private and secure inference,” in *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, 2023.
- [10] J. Lee, Y. Sim, J. Kim, and Y.-J. Suh, “Emosds: Unified emotionally adaptive spoken dialogue system using self-supervised speech representations,” *Future Internet*, 2025.
- [11] A. Ghriss, B. Yang, V. Rozgic, E. Shriberg, and C. Wang, “Sentiment-aware automatic speech recognition pre-training for enhanced speech emotion recognition,” in *Proceedings of ICASSP*, 2022.
- [12] J. Chang and C.-H. Wu, “Pprsa: A plug-and-play language model with rational speech act inference for generating empathetic and engaging dialogue responses,” *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

- [13] T. Dimitrova-Grekow, A. Klis, and M. Igras-Cybulska, “Speech emotion recognition based on voice fundamental frequency,” *Archives of Acoustics*, p. 277–286, 2019.
- [14] Merriam-Webster, “Phonate,” 2026. [Online; accessed 16-June-2026].
- [15] X. Zhang, H. Liu, K. Xu, Q. Zhang, D. Liu, B. Ahmed, and J. Epps, “When llms meet acoustic landmarks: An efficient approach to integrate speech into large language models for depression detection,” in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, p. 146–158, 2024.
- [16] S. Liu, “Landmark detection for distinctive feature-based speech recognition,” *The Journal of the Acoustical Society of America*, 1996.
- [17] K. Stevens, “Toward a model for lexical access based on acoustic landmarks and distinctive features,” *The Journal of the Acoustical Society of America*, 2002.
- [18] X. Zhang, D. Liu, T. Xiao, C. Xiao, T. Szalay, M. Shahin, B. Ahmed, and J. Epps, “Auto-landmark: Acoustic landmark dataset and open-source toolkit for landmark extraction,” May 2025.
- [19] V. Mitra, H.-Y. S. Chien, V. Kowtha, J. Y. Cheng, and E. Azemi, “Speech emotion: Investigating model representations, multi-task learning and knowledge distillation,” July 2022.
- [20] D. Lim and Y.-G. Cheong, “Integrating plutchik’s theory with mixture of experts for enhancing emotion classification,” in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, p. 857–867, Association for Computational Linguistics, 2024.
- [21] V. Kowtha, V. Mitra, C. Bartels, E. Marchi, S. Booker, W. Caruso, S. Kajarekar, and D. Naik, “Detecting emotion primitives from speech and their use in discerning categorical emotions,” in *Proceedings of ICASSP 2020*, 2020.
- [22] M. Morales and R. Levitan, “Multimodal and multiview models for emotion recognition,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.
- [23] R. Lotfian and C. Busso, “Building naturalistic emotionally balanced speech corpus by retrieving emotional speech from existing podcast recordings,” *IEEE Transactions on Affective Computing*, vol. 10, pp. 471–483, 2019.
- [24] V. Mitra, A. Romana, D. T. Tran, and E. Azemi, “Modeling speech emotion with label variance and analyzing performance across speakers and unseen acoustic conditions,” Mar. 2025.
- [25] A. Romana, J. Narain, T. D. Tran, A. Davis, J. Fong, R. Rasipuram, and V. Mitra, “Switchboard-affect: Emotion perception labels from conversational speech,” Oct. 2025.

- [26] X. Zhang, F. He, and H.-N. Dai, “Achieving effective virtual reality interactions via acoustic gesture recognition based on large language models,” 2025.
- [27] S. Zhang, H. Hu, and S. Xing, “Dual learning for conversational emotion recognition and emotional response generation,” *IEEE TRANSACTIONS ON AFFECTIVE COMPUTING*, 2024.
- [28] S. R. Livingstone and F. A. Russo, “The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english,” *PLoS ONE*, vol. 13, no. 5, p. e0196391, 2018.
- [29] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, “Crema-d: Crowd-sourced emotional multimodal actors dataset,” *IEEE Transactions on Affective Computing*, vol. 5, no. 4, pp. 377–390, 2014.
- [30] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, “Iemocap: Interactive emotional dyadic motion capture database,” *Language Resources and Evaluation*, vol. 42, no. 4, pp. 335–359, 2008.