LOW RESOURCE SPEECH INTENT CLASSIFICATION USING MFCC FEATURES

Anas Fathima Rifaza

219393M

Master of Science in Computer Science

Department of Computer Science & Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

March 2025

LOW RESOURCE SPEECH INTENT CLASSIFICATION USING MFCC FEATURES

Anas Fathima Rifaza

219393M

Thesis/Dissertation submitted in partial fulfillment of the requirements for the degree

Master of Science in Computer Science

Department of Computer Science and Engineering

Faculty of Engineering

University of Moratuwa

Sri Lanka

March 2025

DECLARATION

I hereby declare that this thesis is the result of my own independent work. It does not include any content that has been previously submitted for a degree or diploma at any university or institute of higher education, unless properly cited. To the best of my knowledge, all materials taken from the work of others have been appropriately acknowledged and referenced within the text. I also reserve the right to reuse parts of this work in future publications, such as journal articles or academic books.

Signature: Date:

The above candidate has carried out research for the PhD/MPhil/Masters thesis/dissertation under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of Supervisor: Dr.T.Uthayasanker

Signature of the Supervisor: Date: 28 Jul 2025

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my MSc Research Project supervisor, Dr. Uthayasanker Thayasivam, for his invaluable guidance, continuous support, and encouragement throughout my research journey. His expertise and insights have been instrumental in shaping this work, and his unwavering support in providing the necessary resources has greatly contributed to the successful completion of my MSc thesis.

I am also deeply grateful for his constructive feedback, mentorship, and valuable suggestions, which have significantly enhanced the quality of this research. Additionally, I extend my heartfelt appreciation to my colleagues for their assistance in exploring relevant research materials and fostering a collaborative learning environment.

Furthermore, I am immensely thankful to my family—my parents, siblings, nephew, niece, and close friends—for their unwavering encouragement and support throughout this journey. Their belief in me has been a constant source of motivation.

Finally, I would like to extend my appreciation to everyone who has contributed to this endeavor, whether directly or indirectly. Their support has been invaluable in helping me navigate the challenges of my MSc studies

ABSTRACT

Speech-based user interfaces have revolutionized digital interactions, yet developing them for low-resource languages remains a challenge due to limited labeled speech data. This research proposes a Convolutional Neural Network (CNN)-based approach utilizing Mel-Frequency Cepstral Coefficients (MFCC) along with delta and delta-delta features for effective speech intent classification in Sinhala and Tamil. The methodology incorporates audio preprocessing, MFCC feature extraction, and data augmentation techniques such as noise addition, pitch shifting, and time stretching. A stratified cross-validation framework is used to ensure fair and consistent evaluation. The proposed model achieves 96.92% accuracy on the Sinhala dataset (7,624 samples) and 93.81% on the Tamil dataset (400 samples, ~0.5 hours of speech), representing a substantial improvement over prior methods. These results demonstrate the effectiveness of the CNN-based approach in capturing meaningful acoustic patterns for intent recognition in low-resource settings. The study offers a scalable, efficient solution for speech intent classification and contributes to the advancement of inclusive voice-enabled technologies.

Keywords: Speech Intent Classification, Low-Resource Languages, Pre trained Models, Convolutional Neural Network (CNN), Transfer Learning, Mel-Frequency Cepstral Coefficients (MFCC).

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Description** **Page**

LIST OF TABLES

Table **Description** **Page**

LIST OF ABBREVIATIONS

Abbreviation	Description
MFCC	Mel-Frequency Cepstral Coefficients
ASR	Automatic Speech Recognition
NLP	Natural Language Processing
HMMs	Hidden Markov Models
RNNs	Recurrent Neural Networks
NLU	Natural Language Understanding
OOV	Out-of-Vocabulary
SVM	Support Vector Machine
ROC	Receiver Operating Characteristic

CHAPTER 1

INTRODUCTION

1.1 Introduction Overview

This chapter establishes the foundation for research in low-resource speech intent classification using MFCC features based CNN. It introduces the significance of speech recognition technologies in human-computer interaction and their growing integration into various domains. Despite significant advancements in Automatic Speech Recognition (ASR) and Natural Language Processing (NLP), low-resource languages continue to face persistent challenges in speech-to-intent classification. The chapter highlights these challenges, such as data scarcity, linguistic complexity, and model generalization issues, which hinder the development of robust ASR systems for languages like Sinhala and Tamil.

This study aims to bridge these gaps by leveraging deep learning techniques, including Convolutional Neural Networks (CNNs), for intent classification. Additionally, the research explores the effectiveness of transfer learning methodologies and benchmarks performance against pre-trained models Wav2Vec2 and DeepSpeech. The chapter concludes with an outline of the thesis structure, detailing how subsequent chapters will elaborate on methodology, experimental evaluations, findings, and future research directions.

1.2 Introduction

The rapid advancements in speech recognition technology have transformed human-computer interaction, enabling voice-driven applications across various domains. Modern ASR systems convert spoken language into text, while Natural Language Understanding (NLU) models extract user intent, facilitating seamless interactions in applications like virtual assistants, smart home automation, and customer service. Popular ASR-integrated systems, such as Google Assistant, Amazon Alexa, and Apple Siri, demonstrate the potential of voice-based interfaces in enhancing accessibility and usability.

1.2.1 Significance of Intent Classification

Intent classification is a fundamental aspect of NLU systems, determining the purpose behind user queries. High accuracy in intent classification is crucial for enhancing user experience and system efficiency. For example, correctly identifying whether a user intends to check the weather, set an alarm, or schedule an appointment ensures effective execution of commands. However, the majority of advancements in intent classification are tailored to high-resource languages, leaving low-resource languages at a disadvantage.

1.2.2 Challenges in Low-Resource Settings

Despite significant advancements in Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU), these improvements are predominantly confined to high-resource languages such as English, Mandarin, and Spanish. In contrast, low-resource languages (LRLs) such as Sinhala and Tamil continue to face major obstacles. These languages typically lack sufficient online presence and electronic linguistic resources, which hinders the development of effective Speech-to-Text and intent classification systems [1], [2]. The primary challenges are outlined below:

1. **Scarcity of Data:**

   A primary challenge for low-resource languages is the lack of annotated speech datasets necessary for training automatic speech recognition (ASR) models. Unlike English, which benefits from extensive corpora, languages like Tamil and Sinhala have limited transcribed speech resources. This scarcity leads to several issues, including poor model generalization, where insufficient training examples result in ASR models that struggle to adapt to different speakers and contexts. Consequently, there is a greater reliance on alternative techniques such as unsupervised, semi-supervised, or data augmentation methods to mitigate data limitations. Additionally, domain adaptation becomes challenging, as available datasets are often general-purpose and do not cater to specialized applications such as healthcare, finance, or customer service. As highlighted in [3], data scarcity critically affects ASR model performance and emphasizes the need for more annotated speech resources.

2. **Linguistic Complexity:**

   Low-resource languages often exhibit unique linguistic features that pose challenges for automatic speech recognition (ASR) and intent classification. One such challenge is morphological richness, as seen in Tamil, an agglutinative language where words are formed by appending multiple affixes, complicating word segmentation. Similarly, Sinhala presents complex verb conjugations and compound words that make recognition more difficult. Another issue is tonal and phonetic variability, where these languages display tonal variations and diverse phonetic representations that pre trained speech models, primarily trained on high-resource languages, fail to capture effectively. Additionally, code-mixing and code-switching are common, with speakers frequently alternating between languages, such as Sinhala-English or Tamil-English, within a single sentence. This often results in higher word error rates, as pre trained ASR models struggle to recognize mixed-language utterances. [4] discuss the challenges posed by linguistic variability and code-switching, highlighting the need for ASR models that can better adapt to these complexities.

3. **Domain-Specific Limitations:**

ASR models for low-resource languages often struggle to adapt to specialized domains that require specific vocabulary and contextual understanding. In medical speech recognition, for example, ASR models frequently fail to recognize medical terminology due to the lack of domain-specific corpora. Similarly, legal and business domains pose challenges in recognizing technical jargon, proper names, and context-specific intent categories, as these are often underrepresented in existing datasets. Additionally, customer service and conversational AI applications demand a high level of contextual understanding, but ASR models frequently falter in these settings due to informal speech patterns, accents, and regional dialects. [5] highlight the difficulties in adapting ASR systems to specialized domains, underscoring the need for tailored training approaches and domain-specific datasets.

4. **High Out-of-Vocabulary (OOV) Rate:**

The out-of-vocabulary (OOV) rate for Tamil and Sinhala is significantly higher than that of English due to the limited vocabulary coverage in existing automatic speech recognition (ASR) models for these languages. Tamil, for instance, has a highly inflected word structure, which frequently results in OOV errors during speech-to-text conversion. Additionally, named entities such as place names, personal names, and technical terms are often absent from pre trained lexicons, reducing the effectiveness of intent classification models. [6] discuss the challenges associated with high OOV rates in low-resource languages, emphasizing the need for improved vocabulary expansion techniques.

5. **Poor Audio Quality:**

Many datasets for low-resource languages are derived from noisy real-world recordings, often captured in non-studio environments. Factors such as background noise, overlapping speech, and poor microphone quality contribute to higher automatic speech recognition (ASR) error rates. Unlike English, where large-scale datasets enable the development of noise-robust models, low-resource languages often lack effective de noising techniques tailored to their unique phonetic properties. As highlighted in [7], poor audio quality significantly affects ASR performance and poses challenges in low-resource environments.

6. **Limited Transcribed Data and Transfer Learning Constraints:**

Training end-to-end automatic speech recognition (ASR) or intent classification models requires large amounts of transcribed speech-text data. However, the availability of such transcriptions is often limited for low-resource languages, making it difficult to train robust models from scratch. To overcome this challenge, researchers frequently rely on transfer learning by adapting pre trained models. Despite its advantages, this approach presents difficulties, as most pre trained models are trained predominantly on high-

resource languages, making adaptation to Tamil and Sinhala challenging. Common issues include poor phoneme representation, where models struggle to learn language-specific sounds, and suboptimal transfer learning, where fine-tuning on limited data fails to effectively reduce error rates. [8] propose methods to enhance ASR performance in low-resource languages through improved transfer learning strategies.

7. **Technological and Computational Barriers:**

   Many state-of-the-art automatic speech recognition (ASR) models demand substantial computational power, which is often unavailable in low-resource research settings. Cloud-based ASR solutions offer only limited support for low-resource languages and provide minimal customization options, making them unsuitable for domain-specific applications. Additionally, developing on-device ASR models for real-time applications presents significant challenges due to constraints related to model size, latency, and hardware resources. As discussed by [9], these computational barriers make it difficult to deploy ASR models effectively in low-resource environments. Furthermore, the scarcity of linguistic data, the complexity of underrepresented languages, high out-of-vocabulary (OOV) rates, and computational limitations hinder the development of accurate and robust speech-to-text intent classification systems. Addressing these challenges requires advanced techniques such as transfer learning, data augmentation, self-supervised learning, and domain adaptation, which can significantly enhance the performance of ASR and natural language processing (NLP) models in low-resource languages.

1.2.3 Proposed Solution

To overcome the challenges of intent classification in low-resource languages, this research proposes a Residual CNN-based framework utilizing Mel-Frequency Cepstral Coefficients (MFCC) along with delta and delta-delta features. The focus is on Sinhala and Tamil—languages with limited annotated speech data and minimal representation in mainstream ASR systems.

Departing from conventional ASR pipelines, the proposed system directly leverages deep learning specifically residual 2D Convolutional Neural Networks to model speech intent. The model is trained with stratified cross-validation and enhanced through data augmentation strategies to improve generalization.

To evaluate the effectiveness of this approach, benchmarking is conducted against state-of-the-art pre trained models such as Wav2Vec2.0 and DeepSpeech. This comparison helps assess the role of transfer learning in low-resource scenarios.

The framework is designed to address core limitations including data scarcity, phonetic diversity, generalization gaps, and language-specific constraints. Ultimately, this work aims to contribute to the development of inclusive, robust, and domain-adaptive intent classification systems for underrepresented languages.

1.3 Background

Speech recognition technologies have evolved significantly over the past few decades, transitioning from early rule-based systems in the 20th century to modern deep learning-based architectures capable of end-to-end processing. These systems integrate Automatic Speech Recognition (ASR), which transcribes spoken language into text, with Natural Language Understanding (NLU), which interprets the intent and context of the transcribed input. Together, they form the foundation of voice assistants, call center automation, and smart devices, enabling seamless human–machine interaction.

Key milestones in ASR development highlight this progression. Early rule-based systems were limited by hardware constraints and could only recognize a small vocabulary of isolated words. The introduction of Hidden Markov Models (HMMs) in the 1980s facilitated statistical modeling of speech sequences, leading to notable improvements in recognition accuracy. The advent of Deep Neural Networks (DNNs) and Convolutional Neural Networks (CNNs) further revolutionized ASR by enabling end-to-end learning, eliminating the need for handcrafted feature extraction. More recently, pre-trained models such as DeepSpeech [1] and Wav2Vec 2.0 have leveraged transfer learning and self-supervised learning techniques to achieve state-of-the-art performance in high-resource languages.

Despite these advancements, ASR development for low-resource languages like Sinhala and Tamil continues to face significant challenges. Data scarcity remains a major barrier, as the lack of large annotated corpora limits the training of ASR systems [1]. Additionally, linguistic diversity presents difficulties, with unique features such as phoneme variations, elongated vowels, and tonal nuances often poorly captured by existing models. Moreover, domain-specific constraints hinder ASR systems trained on generic datasets from performing effectively in specialized fields such as healthcare and customer service. Additionally, language-specific limitations further complicate development. Pre trained models typically trained on high-resource languages like English often fail to account for phonetic characteristics unique to languages like Tamil and Sinhala, such as tonal shifts and vowel length variations. This oversight can significantly impact classification accuracy, as noted in studies [4] and [10].

To address these gaps, researchers have proposed leveraging pre-trained models to reduce reliance on large datasets by adapting techniques from high-resource languages to low-resource contexts [1]. While transfer learning provides a promising direction, its application to low-resource languages presents several challenges. Most pre trained models are developed using high-resource languages, limiting their ability to generalize to languages like Tamil and Sinhala. A key limitation is the inaccurate representation of language-specific phonemes, where the models struggle to capture sounds unique to these languages. Additionally, due to the limited availability of annotated data, fine-tuning these models often results in suboptimal performance, with error rates remaining high despite transfer learning efforts.

As a solution, the present study introduces a customized methodology that employs Mel-Frequency Cepstral Coefficients (MFCCs) combined with a 2D CNN-based architecture. This methodology is designed to enhance intent classification accuracy

for low-resource languages by effectively capturing the spectral and temporal features of speech signals.

1.4 Problem Statement

While speech recognition technologies have achieved remarkable success for high-resource languages, they are far less effective for low-resource languages. The lack of annotated speech data, tailored domain-specific models, and customized methodologies results in suboptimal performance in speech-to-intent classification. These limitations exclude millions of speakers of low-resource languages from accessing speech-driven technologies.

Specific Challenges

1. **Limited Generalization**: Pre trained model DeepSpeech, trained on high-resource languages, struggle to generalize to low-resource languages due to linguistic differences.
2. **Performance Bottlenecks**: Existing speech-to-intent classification systems require fine-tuning and hyper parameter optimization to achieve satisfactory performance, especially in resource-constrained environments.
3. **Domain Transfer**: Few studies evaluate how methodologies generalize across domains, such as applying a banking-trained model to healthcare-related queries. This limits the practical usability of these systems.

This study aims to address these challenges by implementing a Residual 2D CNN-based intent classification model using MFCC features and delta and delta-delta features. Furthermore, enhancements such as hyper parameter optimization and data augmentation are introduced to improve the model's performance, making speech-to-intent classification more effective for low-resource linguistic settings

1.5 Objectives

This research sets out to achieve the following specific objectives:

1. Design and implement a CNN-based intent classification model using MFCC features, including delta and delta-delta coefficients, for Sinhala and Tamil speech data.
2. Improve the performance of MFCC-based models by extending prior approaches (e.g., replacing feed forward networks with deep 2D CNNs capable of capturing complex temporal and spectral patterns).
3. Apply and evaluate data augmentation technique such as pitch shifting, time stretching, noise addition, and time shifting to increase training diversity and enhance model robustness.

4. Optimize the CNN architecture by systematically tuning key hyper parameters (e.g., filter size, kernel size, batch size, and learning rate) using methods like Bayesian optimization and ReduceLROnPlateau.
5. Benchmark the proposed MFCC-CNN model against pre trained models such as DeepSpeech and Wav2Vec2.0, assessing their effectiveness in low-resource settings.
6. Evaluate model performance comprehensively using standard classification metrics accuracy, precision, recall, F1-score, confusion matrices, ROC curves, and precision-recall curves—to validate results and support comparative analysis.

1.6 Contributions

This research contributes to the advancement of low-resource speech intent classification by evaluating and improving classification techniques for Sinhala and Tamil. The key contributions include benchmarking pre-trained models against CNN-based MFCC features by validating the DeepSpeech-based benchmark methodology from the reference study. A comparative analysis was conducted between CNN model trained on MFCC features and pre-trained models Wav2Vec2 and DeepSpeech for speech intent classification. Additionally, the study focused on optimizing CNN-based feature extraction for low-resource settings by applying Bayesian optimization to fine-tune hyper parameters, thereby enhancing the performance of CNN classifiers. To address data scarcity and improve model robustness, data augmentation techniques such as noise addition, pitch shifting, and time warping were integrated.

Furthermore, the research evaluated pre-trained models for intent classification by investigating Wav2Vec2.0 as an alternative feature extractor and demonstrating its potential advantages over DeepSpeech in low-resource speech intent classification. The study also analyzed the effectiveness of self-supervised learning in mitigating data scarcity challenges. Lastly, this work advances speech recognition for underserved languages by providing practical insights into linguistic diversity, code-mixing, and phonetic variations in low-resource languages. These contributions support the development of inclusive, efficient, and scalable speech recognition systems for Sinhala, Tamil, and other underrepresented languages.

1.7 Summary

This chapter introduces the research problem, emphasizing the critical challenges faced by low-resource languages in speech intent classification. It outlines the significance of speech recognition technologies, the necessity of intent classification, and the limitations of current methodologies. The chapter also presents the research objectives and proposed solutions, setting the stage for a comprehensive analysis of MFCC features with CNN models in low-resource speech intent classification. Subsequent chapters will provide related work, an in-depth exploration of methodology, experimental validation, results, and the impact of this research in advancing speech recognition for low resource languages.

CHAPTER 2

LITERATURE REVIEW

2.1 Literature Overview

This chapter presents a comprehensive review of prior research and methodologies in speech command classification, transfer learning in low-resource languages, and benchmarking techniques for speech recognition in underrepresented languages. The review identifies existing challenges, solutions, and gaps, laying the foundation for the proposed research.

2.2 Speech Command Classification

Speech command classification involves identifying the intent behind spoken queries. Two primary approaches have been employed in this field:

2.2.1 Cascading ASR-NLU Models

One commonly adopted architecture in intent classification systems is the cascading ASR-NLU approach, which decomposes the task into two separate modules. The Automatic Speech Recognition (ASR) component first transcribes the speech into text, which is then passed to a Natural Language Understanding (NLU) model to infer the intent. This architecture is prevalent in commercial voice assistants like Amazon Alexa and Google Assistant.

However, this approach suffers from certain limitations—errors in the ASR output directly affect the performance of the downstream NLU model. Moreover, the ASR and NLU modules are often trained independently, leading to a lack of joint optimization across the pipeline [11], [12].

To address these issues, [11] introduced a joint optimization strategy using an n-best hypothesis list from the ASR model to improve intent recognition. Later, [13] proposed a more generalized framework for this joint system. Nevertheless, these solutions are data-intensive, requiring large volumes of labeled speech data, accurate transcripts, and intent annotations.

In addition, the ASR systems typically rely on phoneme dictionaries and well-developed language models, resources that are often missing for low-resource languages. This makes the cascading model less effective in such contexts [12].

For example, [12] demonstrated a voice-enabled navigation system for an entertainment platform in English using this architecture. They emphasized that inaccuracies in the ASR module can degrade overall system performance, which is a critical concern when adapting such models to languages with limited linguistic tools and corpora.

2.2.2 Direct Speech Classification Models

Direct speech classification models bypass the intermediate transcription step found in traditional pipelines. Instead of converting speech to text, they classify audio signals directly into intents using features extracted from raw waveforms. Commonly used features include Mel-Frequency Cepstral Coefficients (MFCCs) [14], spectrograms, and character probability maps. Deep learning models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are often employed for this task. One major advantage of this approach is its independence from language models and textual data, making it more resilient to noise and variability in spoken input [15].

[16] proposed a method for topic identification in speech that does not require manual transcripts or phoneme-level annotations. Their framework utilized bottleneck features extracted from a multilingual ASR system, which were then used as input to CNN and Support Vector Machine (SVM) classifiers. Complementing this, [17] demonstrated that bottleneck features are effective in capturing key acoustic cues necessary for distinguishing different speech queries.

Despite its benefits, direct classification presents certain challenges. It demands a robust feature extraction process to accurately model both temporal and spectral information in speech. Moreover, its performance is heavily dependent on the quality and diversity of the training data. Nonetheless, recent studies have shown that CNN-based direct classification approaches often outperform traditional ASR-NLU cascades, particularly in noisy environments [18].

In the context of low-resource languages, [10] introduced a feed-forward neural network for intent classification using MFCC features. Applied to Sinhala in the banking domain, this study found that neural networks outperformed traditional classifiers such as decision trees and support vector machines, demonstrating their effectiveness for domain-specific applications.

Additionally, [19] presents an innovative approach to intent classification in Sinhala and Tamil two low-resource languages by utilizing English phoneme-based outputs from Google's multilingual ASR. Instead of building language-specific ASR systems, which is difficult due to limited annotated data, the authors use phoneme probability sequences generated even when non-English languages are spoken. By bypassing the traditional ASR-NLU pipeline, they directly use these phoneme outputs for intent classification. The study evaluates both 1D and 2D Convolutional Neural Networks (CNNs) and finds that 2D CNNs are more effective at modeling phoneme distributions, resulting in higher intent prediction accuracy. These findings confirm that accurate intent classification is possible without fully developing native ASR systems for the target language.

2.3 Transfer Learning in Low-Resource Languages

Transfer learning is an effective technique for addressing data limitations in low-resource languages. By utilizing knowledge from pre trained models on high-resource languages, it enhances performance on tasks with limited training data [8].

2.3.1 Transfer Learning in ASR

Transfer learning has become a crucial strategy in automatic speech recognition (ASR), particularly for low-resource languages where labeled data is limited. By leveraging large-scale pre-trained models, researchers can significantly enhance performance in tasks such as intent classification and speech-to-text.

1. **DeepSpeech:** Mozilla's DeepSpeech is a widely used end-to-end ASR model that utilizes a deep neural network trained on large English speech corpora [20]. It generates character probability maps from raw audio, which can then be used as features for downstream tasks like intent classification. In low-resource contexts, transfer learning with DeepSpeech has shown improvements in recognition accuracy even when labeled data is scarce. This is primarily due to the model's ability to generalize phonetic structures across languages.
2. **Wav2Vec 2.0:** Developed by Facebook AI, Wav2Vec 2.0 uses self-supervised learning to generate contextualized representations from raw speech waveforms [21]. Pre trained on massive unlabeled audio datasets, this model captures both phonetic and semantic information. Fine-tuning Wav2Vec 2.0 on small, labeled datasets has shown excellent performance in low-resource language tasks, outperforming traditional ASR models and even DeepSpeech in many settings [22]. Its ability to adapt with minimal labeled data makes it especially suitable for intent classification in underrepresented languages.
3. **HuBERT (Hidden-Unit BERT):** HuBERT, developed by Facebook AI, extends Wav2Vec 2.0 by incorporating offline clustering to generate pseudo-labels for masked prediction tasks. This self-supervised model is designed to learn meaningful acoustic representations, particularly at the phoneme and speaker levels. It has demonstrated strong performance on speech classification benchmarks and, when fine-tuned, significantly improves intent classification and ASR accuracy in low-resource settings [23].
4. **XLS-R (Cross-Lingual Speech Representations)**: A massive cross-lingual model based on Wav2Vec 2.0, trained on nearly half a million hours of audio across 128 languages [24]. XLS-R sets state-of-the-art results for low-resource ASR benchmarks like BABEL and CommonVoice.
5. **Whisper (OpenAI):** A robust multilingual ASR model trained on 680,000 hours of labeled data spanning 98+ languages [25]. Though primarily designed for transcription, its encoder's representations have been repurposed effectively in zero-shot and few-shot intent classification pipelines.
6. **AST (Audio Spectrogram Transformer)**: Initially developed for general audio tasks like emotion recognition and keyword spotting, AST models pre-trained on datasets like AudioSet can be fine-tuned for speech-based classification, offering a transformer alternative to CNN-based pipelines.

2.3.2 Applications in Low-Resource Languages

Speech recognition in low-resource languages presents unique challenges due to the scarcity of large-scale annotated corpora, limited availability of phoneme dictionaries, and the absence of comprehensive language models. Languages such as Sinhala, Tamil, and Swahili often lack the linguistic and computational resources that are readily available for high-resource languages like English or Mandarin. This scarcity impacts both Automatic Speech Recognition (ASR) and downstream tasks such as intent classification.

In addition to data scarcity, linguistic phenomena such as code-switching**,** dialectal variation**,** tonal differences, and regional accents introduce further variability, making it difficult for conventional ASR models to generalize. Noise and variability in real-world audio**,** including background disturbances and overlapping speech, further degrade performance in low-resource contexts.

To mitigate these issues, transfer learning has emerged as a powerful strategy. By leveraging pre trained models such as Wav2Vec 2.0**,** HuBERT, and Whisper**,** researchers can extract robust speech representations that generalize across multiple languages even those not seen during training. For example, [26] demonstrated that Wav2Vec 2.0, trained on unlabeled English data, significantly improves speech recognition performance in low-resource languages with minimal fine-tuning. Similarly, [24] showed that XLS-R, a multilingual extension of Wav2Vec 2.0, delivers strong zero-shot performance in over 100 languages, including Sinhala and Tamil.

Data augmentation techniques such as time-stretching**,** pitch-shifting**,** noise injection, and SpecAugment are commonly integrated into training pipelines to simulate acoustic diversity and improve model robustness. As shown in [27], such augmentations are highly effective in enhancing generalization, especially in low-data regimes.

Moreover, several studies have explored combining self-supervised speech models with lightweight classifiers (e.g., CNNs, GRUs, SVMs) for intent classification in low-resource settings. For instance, [28] applied Wav2Vec 2.0 embeddings for spoken intent detection in Indian languages and observed considerable performance gains over traditional MFCC-based approaches. Similarly, [29] used Whisper's multilingual encoder to handle noisy and code-mixed inputs in Tamil and Hindi with promising results.

Cross-lingual transfer learning has also been explored as a means to bootstrap models for low-resource languages. For example, [30] showed that models pre trained on typologically similar languages (e.g., Hindi for Sinhala or Telugu for Tamil) can be fine-tuned with minimal target-language data while still achieving competitive results.

In summary, the combination of self-supervised learning**,** cross-lingual transfer**,** and data augmentation has opened new avenues for building scalable and robust speech classification systems for underrepresented languages. These strategies not only reduce the dependency on manually annotated corpora but also accelerate the development of inclusive speech technologies across diverse linguistic communities.

2.4 Benchmarking in Low-Resource Speech Recognition

[1] presents an innovative benchmarking methodology for speech recognition and intent classification in low-resource languages. A key aspect of their approach is leveraging pre-trained Automatic Speech Recognition (ASR) models as feature extractors. Specifically, the DeepSpeech model generates character probability maps, which serve as intermediate representations for intent classification. These probability maps provide a normalized representation of speech, capturing both temporal and acoustic patterns, thereby eliminating the need for transcriptions in the target language. Additionally, using pre-trained models reduces noise and variability in input features, making them more suitable for CNN-based classification.

For classification, the study employs both 1D and 2D CNN architectures to process character probability maps and extract hierarchical features, which are then mapped to intent labels through dense layers. CNNs have proven effective in capturing these hierarchical representations, contributing to high classification accuracy [31], [32]. The methodology was evaluated on two low-resource languages, Sinhala and Tamil, demonstrating the effectiveness of transfer learning and CNN-based classification. The results highlighted the impact of dataset size and feature representation on model performance, reinforcing the suitability of CNNs for intent classification in low-resource settings [33].

Additionally, [19]proposed an innovative approach to intent classification for Sinhala and Tamil two low-resource languages—by leveraging English phoneme-based outputs from Google's multilingual ASR. Instead of building language-specific ASR systems, which is challenging due to limited annotated data, the authors utilized phoneme probability sequences generated even when non-English speech is input. By bypassing the traditional ASR-NLU pipeline, they directly used these phoneme outputs for intent classification. The study compared 1D and 2D Convolutional Neural Networks (CNNs), concluding that 2D CNNs were more effective in modeling phoneme distributions and improving intent prediction accuracy. These findings confirm that accurate intent classification is possible without fully developing native ASR systems for the target language

2.5 Gaps and Limitations

Despite the advancements in speech recognition and intent classification, several gaps remain, particularly for low-resource languages:

1. **Limited Generalization**

   Most speech intent classification studies focus on specific domains, such as banking, and rarely assess model performance across diverse application areas. As a result, models trained on domain-specific data often struggle to generalize to new contexts without extensive retraining. As highlighted in [34], Spoken Language Understanding (SLU) systems are highly domain-dependent, and their accuracy tends to decline when applied to unfamiliar domains. Addressing this limitation requires the development of more diverse, multi-

domain datasets and the application of domain adaptation techniques to improve model robustness and adaptability.

2. **Insufficient Robustness**

   Low-resource speech datasets often suffer from insufficient robustness due to a lack of diversity in speaker accents, speech rates, and environmental conditions. This limitation affects the model's ability to handle real-world variability, such as background noise, code-mixed speech, and variations in pronunciation. As a result, the performance of these models may degrade significantly in practical applications. To overcome this challenge, researchers can employ advanced data augmentation techniques to simulate diverse real-world conditions, as suggested by [27]. Additionally, leveraging self-supervised pre-trained models like Wav2Vec2.0 can enhance feature extraction from raw audio, improving the model's resilience to variations in speech characteristics.

3. **Lack of Language-Specific Optimization**

   Many pre-trained speech models are primarily developed for high-resource languages like English and often fail to capture the linguistic nuances of low-resource languages. Features such as tonal variations and elongated vowels—commonly found in Sinhala and Tamil—are frequently underrepresented, leading to suboptimal model performance. As noted by [30], pre-trained models often struggle to model the phonetic and syntactic characteristics of underrepresented languages effectively. To address this limitation, fine-tuning such models on small, annotated datasets specific to the target language can help adapt them to language-specific patterns. Additionally, leveraging transfer learning strategies that incorporate phonetic and linguistic features unique to the language can further improve model accuracy and effectiveness.

2.6 Summary

The literature highlights the evolution of intent classification, from cascading ASR-NLU pipelines to direct classification using CNNs. Transfer learning emerges as a critical tool for low-resource languages, with pre trained models like DeepSpeech showing significant promise. However, challenges like limited generalization, insufficient robustness, and language-specific optimization remain, paving the way for further research and innovation.

CHAPTER 3

METHODOLOGY

This chapter outlines the development and evaluation of the proposed methodology for low-resource speech intent classification in Sinhala and Tamil. It begins with an overview of the adopted approach, compares it with the methodology of prior research, explains enhancements made to improve performance and robustness, and concludes with evaluation strategies used to validate the results.

3.1 Methodology Overview

In designing a speech intent classification system, it is crucial to follow a systematic flow, as illustrated conceptually in Figure 3.1. The process typically involves several major steps: data preprocessing, feature extraction, data augmentation, model training, and evaluation.

In the preprocessing phase, the dataset must first be cleaned. This involves removing null or corrupted audio samples and ensuring that the data is properly labeled. Handling data imbalance is also critical at this stage to prevent the model from becoming biased toward majority classes. Techniques such as class weighting or data resampling are commonly employed to address this issue.

The next phase is feature extraction. Prior studies have explored various techniques such as Linear Predictive Coding (LPC), Relative Spectral Transform (RASTA), and Mel-Frequency Cepstral Coefficients (MFCCs) for speech feature extraction. According to [35], a comparative study revealed that MFCC features consistently outperformed LPC and RASTA for speech processing tasks. [10] also adopted MFCCs, extracting 13-dimensional MFCC features for Sinhala intent classification, confirming the effectiveness of MFCC-based representations.

Data augmentation plays a vital role in enhancing model generalization, especially in low-resource settings. Previous researchers, such as [2], employed techniques like noise addition, pitch shifting, and time warping to artificially expand limited datasets, improving the model's robustness to variability. Similarly, in the study [36] it was shown that transfer learning combined with data augmentation significantly boosted performance in low-resource speech translation tasks.

In the model training phase, researchers have used a range of strategies, including cascaded models, custom-designed architectures, and pre trained models. These models are trained on the extracted features, and strategies such as transfer learning and fine-tuning are often used to adapt existing models to new, low-resource domains.

Finally, in the evaluation phase, models are assessed using multiple evaluation metrics such as accuracy, precision, recall, and F1-score to comprehensively measure performance. Comparative analysis across different models and configurations is performed to identify the best-performing setup.

The overall methodological flow can be summarized as in the Figure 3.1:

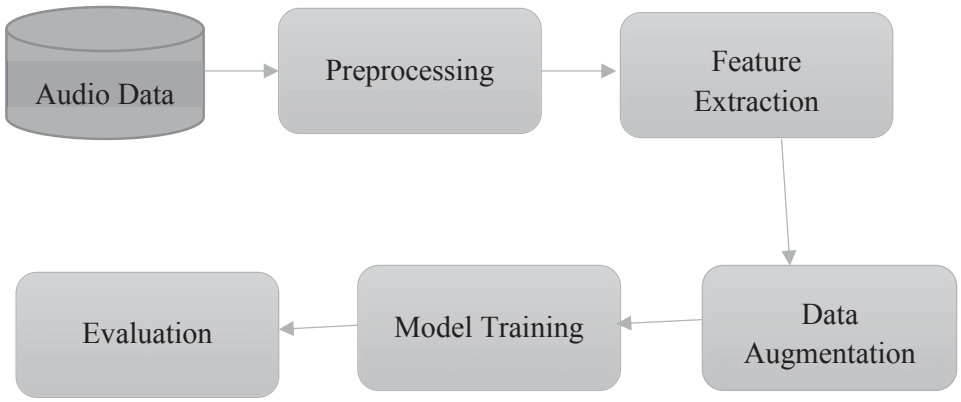

Figure 3.1: General Workflow for Speech Intent Classification in Previous Studies

This research builds upon and extends two foundational studies in the domain of low-resource speech intent classification. The first influential work is by [10], which proposed a domain-specific Sinhala intent classification system using 13-dimensional Mel-Frequency Cepstral Coefficients (MFCCs) as features and a simple Feed Forward Neural Network (FFN) for classification. While the model demonstrated the feasibility of direct intent classification without speech-to-text conversion, it lacked the capacity to model temporal dependencies in speech. As a result, the performance was constrained, achieving a maximum test accuracy of 74.37%**.**

The second major study, by [1], introduced a transfer learning-based approach using a pre trained DeepSpeech model. Their methodology involved extracting character probability maps from DeepSpeech's softmax layer—trained on English audio—and using these outputs as language-agnostic acoustic features. These intermediate representations were then fed into 1D and 2D Convolutional Neural Networks (CNNs) to classify intents in Sinhala and Tamil datasets. Without requiring manual transcription or language-specific ASR systems, their approach offered a practical alternative for low-resource languages. Their 1D CNN model achieved 93.16% accuracy for Sinhala and and 2D CNN achieved 76.30% for Tamil**,** setting a strong benchmark for further exploration.

In contrast to [10], which used basic MFCCs and FFNs, and [1], which relied on pre trained embedding from DeepSpeech, our approach introduces a fully customized MFCC-CNN pipeline with an emphasis on interpretable, handcrafted features. Our model expands the MFCC feature dimensionality, integrates delta and delta-delta coefficients to capture temporal dynamics, and incorporates Swish-activated convolutional layers with residual connections for deeper representation learning. To improve robustness, we further employ stratified cross-validation and data augmentation**.**

Additionally, [1]'s analysis showed that 2D CNNs outperformed 1D CNNs on smaller datasets like Tamil**,** while 1D CNNs yielded better performance on larger datasets like Sinhala**.** Similarly, [19] demonstrates that 2D CNNs outperform 1D CNNs, showing better generalization and ability to learn from time-frequency patterns. These architectural insight influenced our experimental design, where we examined both 1D and 2D CNN variants tailored to dataset size. While DeepSpeech's transfer learning approach leveraged powerful pre trained models, it was ultimately constrained by its reliance on character-level outputs**,** which may not generalize effectively to phonologically distinct languages such as Sinhala or Tamil.

In contrast, our handcrafted MFCC-based pipeline gives us full control over the spectral and temporal resolution of the extracted features. This allows for more fine-tuned optimization, better domain adaptation, and interpretability important factors in low-resource, domain-specific speech classification tasks.

3.2 Proposed MFCC-CNN Architecture and Enhancements

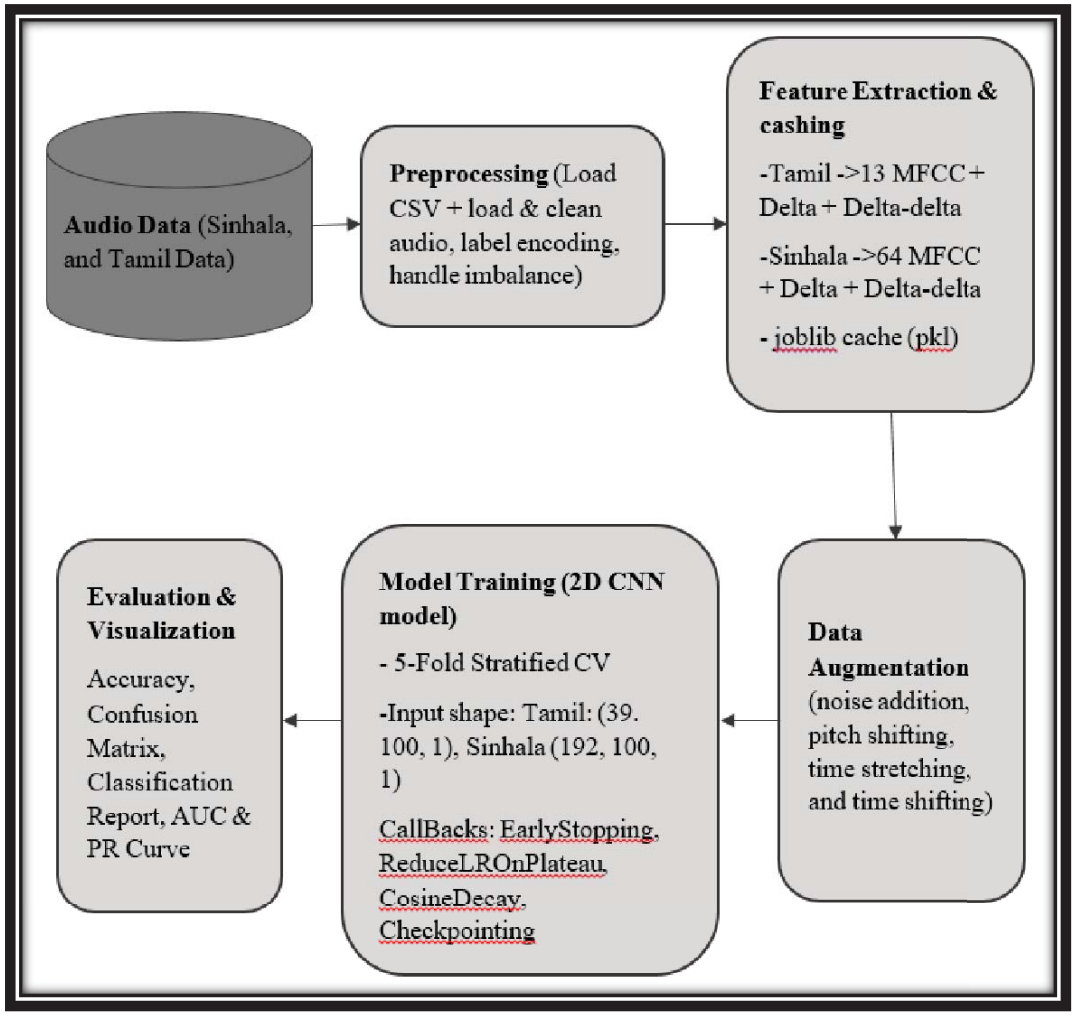

Figure 3.2: Pipeline Architecture of the Proposed System, Illustrating Five Key Stages: Preprocessing, Feature Extraction, Data Augmentation, Model Training, and Evaluation.

The workflow in Figure 3.2 is summarized as follows:

3.2.1 Audio Data Collection

The workflow begins with the collection of domain-specific speech data in Sinhala and Tamil, two low-resource languages. Each audio file corresponds to a spoken command labeled with a predefined intent (e.g., “balance inquiry”, “fund transfer”). This domain-labeled dataset forms the foundation of the classification system, focusing on recognizing spoken intents directly from audio without requiring intermediate transcription through traditional ASR systems.

3.2.2 Preprocessing

In this phase, the metadata is first loaded from a CSV file to map each entry to its corresponding .wav audio file within the dataset. Any unmatched or invalid audio

paths are filtered out to ensure dataset consistency. The intent labels, originally in textual form (e.g. "check balance", "transfer funds"), are then encoded into integer format using LabelEncoder, making them compatible with model training processes. Furthermore, the distribution of intent classes is examined to identify and address class imbalance early in the pipeline. This step ensures that minority classes are adequately represented, which is essential for building a fair and effective classification model. The preprocessing pipeline, implemented using tools like pandas and standard file path normalization techniques, yields a clean, well-structured, and fully labeled dataset ready for feature extraction.

3.2.3 Feature Extraction & Caching

To convert raw audio signals into informative input representations for the model, each audio sample undergoes spectral feature extraction using Mel-Frequency Cepstral Coefficients (MFCCs)**.** MFCCs are widely used in speech processing due to their ability to effectively capture phonetic structures with high accuracy and low computational complexity [14], [35].

These features are further enhanced with delta and delta-delta coefficients to capture temporal dynamics—specifically, the velocity and acceleration of speech signal changes. MFCCs represent the static characteristics of speech at each time step, while delta features capture the rate of change (first-order derivative), and delta-delta features capture the acceleration (second-order derivative) of those changes.

- For the Tamil dataset, 13 MFCCs are extracted, resulting in 39 features per frame (13 + delta + delta-delta).
- For the Sinhala dataset, 64 MFCCs are used, producing 192 features per frame.

All feature matrices are padded or truncated to 100 time frames to ensure temporal consistency across samples. They are then reshaped into 3D tensors to match the input shape required by CNN architectures.

To improve computational efficiency, feature extraction is parallelized using Python's multiprocessing module. The resulting features are cached using joblib and stored as .pkl files, enabling rapid reloading during experiments and eliminating redundant computation. This approach ensures standardized, high-dimensional feature tensors optimized for batch model training.

3.2.4 Data Augmentation

To overcome the limitations of small and imbalanced datasets particularly in the Tamil corpus offline data augmentation techniques are employed. These include additive noise, pitch shifting**,** time stretching, and temporal shifting**,** which mimic natural variations in speech such as background noise, pitch fluctuations, and varying speaking rates. These methods simulate real-world acoustic diversity, thereby enhancing model robustness without altering the evaluation protocol.

Augmented samples are pre computed and cached alongside the original data, effectively expanding the training set and improving generalization to unseen

conditions. This is particularly important in low-resource settings, where data scarcity hampers model performance. Prior studies, such as [37], demonstrated that augmenting speech with speed and volume perturbations significantly improves ASR performance. Similarly, [38] and [27] showed that noise injection and SpecAugment-based methods can enhance recognition accuracy across various speech benchmarks.

By adopting similar augmentation strategies, this work ensures the trained model is more resilient to domain shifts and speaker variability, making it better suited for real-world deployment in low-resource speech intent classification tasks.

3.2.5 Model Architecture and Model Training

The speech intent classification model is built on a residual Convolutional Neural Network (CNN) architecture optimized for processing Mel-Frequency Cepstral Coefficients (MFCC) features. CNNs are particularly well-suited for speech tasks due to their ability to effectively capture time-frequency patterns within spectrogram-like inputs [24]. In this work, the preprocessed and augmented MFCC-based tensors are fed into a custom-designed 2D CNN architecture. This choice is supported by prior research demonstrating the superior performance of 2D CNNs over 1D CNNs in speech classification. For instance, a study on Sinhala and Tamil speech intent classification [19] found that 2D CNNs outperformed 1D models when applied to phoneme probability maps from ASR systems, effectively capturing both temporal and spectral dependencies. Similarly, another study [39] confirmed that 2D CNNs consistently surpassed 1D CNNs in speech emotion recognition tasks using datasets like RAVDESS and URDU, reinforcing the advantage of 2D convolutions in modeling rich time-frequency patterns.

Unlike 1D CNNs, which only convolve over the temporal axis, 2D CNNs operate along both time and frequency axes. This makes them particularly suitable for features like MFCCs, especially when enhanced with delta and delta-delta coefficients. These features form a structured 2D representation, aligning naturally with 2D convolution operations. Prior research in both intent classification and emotion recognition supports the effectiveness of 2D CNNs for extracting high-level representations from speech data.

In addition to 2D convolutions, the model incorporates residual connections and Swish activation functions, both of which enhance the model's depth and training stability. Residual blocks mitigate vanishing gradient issues by allowing gradients to bypass certain layers via skip connections, enabling deeper networks and faster convergence. Swish activation, known for its smooth and non-monotonic nature, contributes to improved generalization and non-linear learning compared to ReLU.

[40] highlight the effectiveness of deep residual architectures in speech recognition tasks, particularly in modeling the spectral-temporal complexity of audio in an end-to-end fashion. They note that typical CNN structures often contain fewer than 10 layers, which may be insufficient to fully capture the complexity of human speech especially for long sequences, whereas residual CNNs have shown stronger performance. Additionally, [41] designed a cascaded CNN-ResBiLSTM-CTC model, which incorporates residual CNN layers followed by bidirectional LSTMs and uses CTC loss

for robust phoneme prediction. These studies motivate the use of residual CNNs in our model architecture. Residual connections enable deeper networks, accelerate convergence, and facilitate the learning of hierarchical audio features. These examples affirm the reliability and effectiveness of residual CNNs in speech-related tasks.

To prevent over fitting, Dropout layers are integrated at key locations in the architecture, while Batch Normalization is used to stabilize and accelerate training. The input shape is adjusted based on the dataset: **(**100, 39, 1) for Tamil and (100, 192, 1) for Sinhala, reflecting the respective MFCC-based feature dimensions.

Model training employs 5-Fold Stratified Cross-Validation to ensure balanced class representation and robust evaluation. A comprehensive callback strategy—combining EarlyStopping, ReduceLROnPlateau, Cosine Decay Learning Rate Scheduler, and ModelCheckpoint—is used to optimize convergence and retain the best-performing model weights.

This comprehensive training pipeline yields a robust and generalizable 2D CNN model capable of capturing both localized phonetic patterns and broader temporal dependencies, making it a highly suitable architecture for speech intent classification, especially in low-resource language settings.

Model Components

The model is composed of several well-structured components designed to efficiently learn time-frequency features from MFCC inputs, especially for low-resource speech intent classification. The key components are as follows:

1. **Input Layer**

   This layer accepts MFCC feature matrices with a shape of (num_frames, num_coefficients, 1), where num_frames represents the temporal dimension of the speech signal, and num_coefficients refers to the number of extracted MFCCs per frame. For example, in the Sinhala dataset, the input shape is (100, 192, 1), and for the Tamil dataset, it is (100, 39, 1). These dimensions correspond to 100 time frames and 192 or 39 spectral features per frame, depending on the language. The Sinhala input includes 64 MFCCs along with their delta and delta-delta coefficients, while the Tamil input uses 13 MFCCs with corresponding delta and delta-delta features. These handcrafted features effectively capture both the static and temporal characteristics of speech. Prior to being fed into the model, the features undergo Z-score normalization, which ensures consistency across samples and improves training convergence. MFCCs are a widely used representation in speech tasks due to their proven ability to model phonetic and spectral properties, especially in low-resource language settings [14].

2. **Rescaling Layer**

This layer rescales the normalized input values into the [0, 1] range**.** The purpose of this operation is to further stabilize training by bringing all input features into a uniform scale. This consistency prevents activation functions from saturating and supports more efficient weight updates across layers.

3. Initial Convolution Block

The first convolutional block initiates feature extraction by applying a Conv2D layer with 64 filters of size 3×3. This layer detects local time-frequency features from the MFCC input. The Swish activation function is used to introduce non-linearity and improve gradient flow. It is followed by Batch Normalization, which helps reduce internal covariate shift, and MaxPooling2D (2×2), which down samples the spatial dimensions while preserving the most significant patterns for deeper processing [42].

4. Residual Block

This block adds depth and learning flexibility to the network using a residual learning strategy. It includes two Conv2D layers (96 filters, 3×3), each activated with Swish and followed by Batch Normalization. A skip connection (Add) is introduced to allow the input to bypass these layers, helping to prevent the vanishing gradient problem and maintain representational integrity [43]. If the dimensions between the input and output differ, a 1×1 Conv2D is applied to the shortcut path for alignment. The merged output passes through another Swish activation and a final MaxPooling2D layer to reduce the resolution before further processing.

5. Final Convolution Block

This layer extracts higher-level abstract features by applying a Conv2D layer with 128 filters (3×3), followed by Batch Normalization and MaxPooling2D (2×2). It helps consolidate earlier learned representations and prepares the output for dense layer aggregation [44].

6. Global Feature Aggregation (Dense Layers)

This part transitions the model from convolutional outputs to decision-making. It begins with GlobalAveragePooling2D, which flattens the spatial feature maps into a vector that represents global feature statistics. The vector is passed through two Dense layers—one with 128 units (Swish activation, Dropout 0.4) and the next with 64 units (Swish activation, Dropout 0.3)**.** These layers help the model learn complex, non-linear combinations of the extracted features while reducing over fitting [45].

7. Output Layer

The final layer is a Dense layer with Softmax activation, where the number of neurons equals the number of intent classes. It outputs a probability distribution

over all classes, making it suitable for multi-class classification. The model is trained using categorical cross-entropy loss, which effectively measures the distance between the predicted and actual class distributions.

Activation Functions

The model primarily uses the Swish activation function in both convolutional and dense layers. Swish has been shown to outperform the traditional ReLU activation in deep learning tasks [42]. The Softmax function is employed in the output layer to predict probability distributions over class labels for multi-class classification.

Model Diagram

Figure 3.3 illustrates the visual representation of the proposed model architecture pipeline.

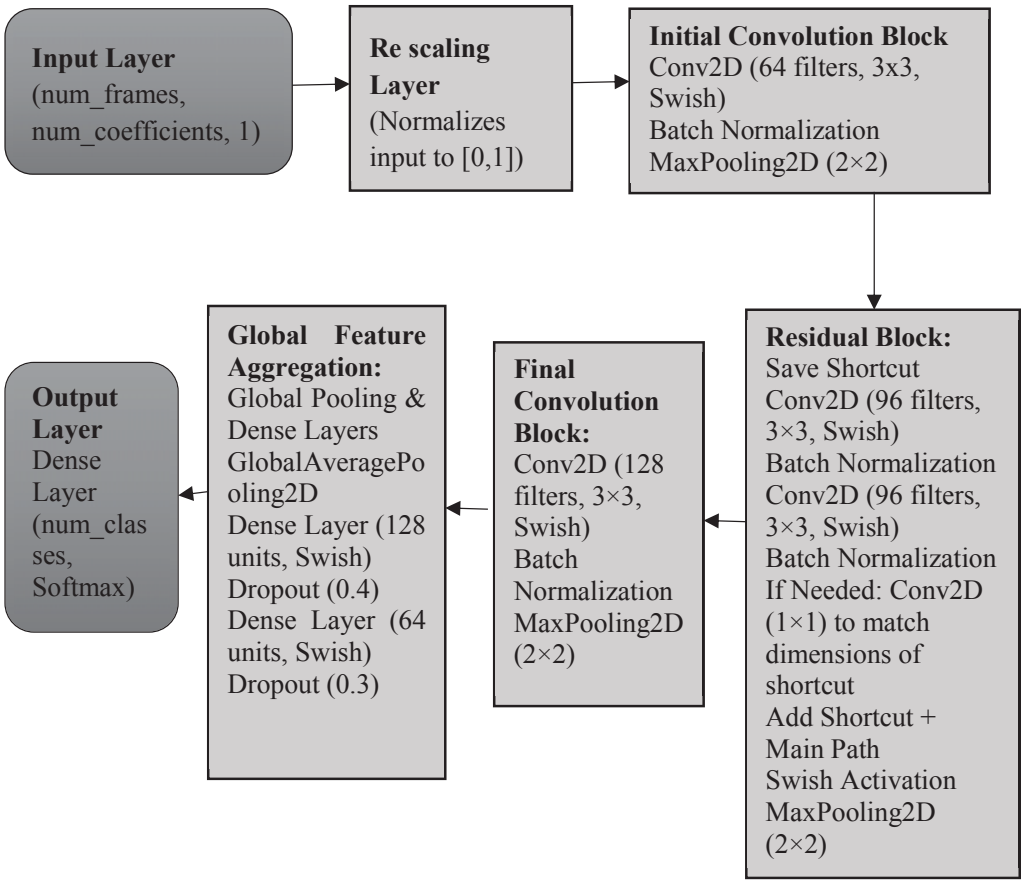

Figure 3.3: Proposed Speech Intent Classification Model Architecture

Figure 3.3 illustrates the proposed Speech Intent Classification model architecture, which incorporates residual connections to enable deeper learning without degradation, which is essential for training more complex networks. It utilizes the

Swish activation function, known for improving gradient flow and enhancing the overall stability and convergence during training. Additionally, the model is designed to handle datasets more effectively, which contributes to its high performance, achieving an accuracy of 96.92% for Sinhala and 93.81% for Tamil. These features work together to ensure that the model can effectively capture intricate patterns in the speech data while maintaining robust learning capabilities.

3.2.6 Evaluation & Visualization

After training, the models were evaluated using a comprehensive set of performance metrics, including accuracy, precision, recall, F1-score, and confusion matrix. To further assess class-level discriminability and understand the impact of class imbalance, both Receiver Operating Characteristic (ROC) curves and Precision-Recall (PR) curves were plotted. These visualizations provided critical insights into the model's strengths and weaknesses across different intent classes, enabling targeted refinement and optimization. Evaluation results were computed across all cross-validation folds and averaged to ensure robustness and reliability. Tools such as scikit-learn, matplotlib, and seaborn were used for the implementation and visualization of these diagnostics. Overall, this evaluation process provided a thorough validation of the model's performance, supported by interpretable visual evidence

3.3 Summary

The proposed MFCC-CNN pipeline presents a robust and well-structured approach for intent classification in low-resource multilingual settings. By combining handcrafted MFCC features with delta and delta-delta coefficients, Swish-activated 2D CNNs with residual connections, data augmentation, hyper parameter tuning, and comprehensive evaluation strategies, the model achieves high performance on both Sinhala and Tamil datasets. Compared to existing baselines and pre trained models (e.g., Wav2Vec2.0, DeepSpeech), this method delivers superior accuracy, efficiency, and interpretability. Its effectiveness and scalability make it highly suitable for real-world applications, particularly in banking and customer service domains for underrepresented languages.

CHAPTER 4

EXPERIMENT

This chapter provides an in-depth discussion of the methodologies used in the research implementation. It begins by describing the data set and the system architecture, and outlining key implementation aspects before presenting experimental results and performance evaluations. A detailed description of the development environment, including hardware and software configurations, is also provided to ensure reproducibility.

4.1 Data Set

This study uses the same dataset as described in [1], comprising Sinhala and Tamil speech samples collected from the banking domain. The Sinhala dataset contains short audio clips each under seven seconds with minimal background noise. The speech reflects unique linguistic characteristics of Sinhala, including tonal variations and elongated vowels.

The Tamil dataset, by contrast, is significantly smaller, consisting of only 400 audio clips totaling approximately 0.5 hours of speech. To ensure consistency, it mirrors the Sinhala dataset in terms of the six defined intent classes. Tamil audio samples are also brief, with durations under seven seconds. A distinct feature of this dataset is the presence of code-mixed speech, where English words are naturally embedded within Tamil sentences a common phenomenon in everyday spoken Tamil.

The small size of the Tamil dataset presents challenges for training deep learning models, necessitating the use of data augmentation techniques to artificially expand training diversity. Moreover, code-mixing complicates feature extraction, as the pre trained ASR model originally trained on English—tends to transcribe English components more accurately than Tamil ones. These factors underscore the need for robust preprocessing and augmentation strategies to improve the performance of intent classification systems for Tamil.

Both the Sinhala and Tamil datasets also exhibit class imbalance, which further motivates techniques like augmentation and class-weighted learning to ensure fair and effective model training.

4.2 Preprocessing

To ensure high-quality feature extraction and model training, a dedicated preprocessing pipeline was implemented. This pipeline handled multiple critical tasks, including data validation, audio augmentation, feature normalization, label encoding, and feature caching.

The first step involved loading and validating the dataset. A custom function ensured that each audio filename listed in the CSV file was correctly mapped to an existing file in the dataset directory. Any unmatched or missing files were identified and excluded to maintain dataset integrity.

Following successful data loading, audio augmentation was applied to enhance generalization and mitigate overfitting, particularly for the smaller Tamil dataset. The augmentation process introduced random variations by applying techniques such as white noise addition, pitch shifting, time shifting, and speed modification. Each audio sample had a 50% probability of being augmented, simulating realistic acoustic variations that a model might encounter in real-world environments.

Next, all audio signals were normalized using Librosa's utility functions to maintain amplitude consistency. The MFCC feature extraction process was then applied, producing 64-dimensional MFCCs, along with their delta and delta-delta coefficients. These combined features captured both the static and dynamic spectral properties of the audio signal. To ensure consistent input shapes for the model, all feature matrices were zero-padded or truncated to a fixed frame length.

The corresponding intent labels were then numerically encoded using LabelEncoder, making them compatible with classification models. To reduce redundant computations, the extracted features and their corresponding labels were cached using joblib, significantly accelerating subsequent training runs and enabling efficient experimentation.

This preprocessing pipeline was designed to be modular, scalable, and robust, supporting both the Sinhala and Tamil datasets used in the study. The combination of augmentation, normalization, and caching ensured that the model received high-quality, diverse, and consistent input, ultimately contributing to improved model accuracy and generalization.

4.3 Model Implementation and Training

The MFCC-based input tensors (enhanced with delta and delta-delta coefficients) are fed into the custom-designed Residual 2D CNN model described above. This design was motivated by evidence that 2D CNNs outperform 1D CNNs in speech classification tasks. For example, in a study on Sinhala and Tamil intent classification, 2D CNNs better modeled temporal and spectral dependencies than 1D CNNs using ASR-based phoneme maps [19]. Another study on speech emotion recognition using RAVDESS and URDU datasets also confirmed the advantage of 2D CNNs in capturing rich time-frequency features.

While 1D CNNs focus solely on temporal sequences, 2D CNNs process both time and frequency axes—aligning naturally with MFCC feature structures. The addition of delta and delta-delta coefficients further enhances the model's ability to capture dynamic aspects of speech, which improves intent recognition performance.

The model also incorporates:

- Residual Blocks: To overcome vanishing gradients and allow for deeper learning through skip connections.
- Swish Activation: For smoother and more effective non-linear transformations than ReLU.
- Dropout & Batch Normalization: To reduce overfitting and stabilize training.

[40] Demonstrated the strength of deep residual CNNs in end-to-end speech recognition, especially for modeling long-term spectral-temporal complexities. Similarly, [41] introduced a CNN-ResBiLSTM-CTC model for phoneme classification, combining residual CNN layers with LSTM and CTC loss for robust performance. These findings support the inclusion of residual CNN layers in our architecture.

Training Strategy

The model was trained using carefully designed strategies to ensure robust and consistent performance across both Sinhala and Tamil datasets. The input shape was set to (100, 39, 1) for Tamil and (100, 192, 1) for Sinhala, reflecting the number of time frames and MFCC-based features (including delta and delta-delta coefficients). To promote fair and balanced evaluation, 5-Fold Stratified Cross-Validation was employed, ensuring each fold maintained a representative distribution of intent classes. Several callback functions were integrated into the training loop: EarlyStopping was used to halt training when validation loss ceased to improve, preventing overfitting; ReduceLROnPlateau dynamically adjusted the learning rate when training progress slowed; CosineDecay applied a smooth learning rate schedule to stabilize convergence over time; and ModelCheckpoint was used to save the best-performing model weights during training. Together, these strategies contributed to a stable, efficient, and generalizable training process.

4.3 Hyper parameter Tuning

To optimize model performance for speech intent classification on both the Sinhala and Tamil datasets, **Bayesian Optimization** was employed. This method was chosen for its effectiveness in navigating high-dimensional hyper parameter spaces and accurately modeling the relationship between hyper parameters and validation performance. Compared to traditional grid or random search techniques, Bayesian Optimization provided a more focused and computationally efficient approach to identifying optimal configurations. The tuning process was guided by cross-validation accuracy, while regularization and learning dynamics were managed through callback functions. Notably, [1] also employed Bayesian Optimization for the same dataset.

Key tuned hyper parameters included:

- **CNN Filters:** Varied across layers—[32, 64, 128] to capture low- to high-level feature representations.
- **Kernel Sizes:** Both 3×3 and 5×5 filters were evaluated to strike a balance between spatial feature extraction and computational cost.
- **Learning Rate:** An initial learning rate of 0.0005 was selected, coupled with a Cosine Decay Scheduler to gradually reduce it over epochs.

- **Dense Layer Units:** Fully connected layer sizes were tuned to improve representation learning before classification.
- **Batch Size:** Batch sizes of **4** and **8** were tested, with **4** yielding better results given the model's depth and hardware constraints.
- **Callbacks:**
    - **EarlyStopping** with a patience of 7 epochs to prevent over fitting.
    - **ReduceLROnPlateau** (patience = 4) to reduce the learning rate when validation loss plateaued.
    - **ModelCheckpoint** to save the best-performing model during training for each fold.

This carefully tuned configuration enabled the Sinhala model to generalize well, achieving an average accuracy of **96.92%** across five folds.

For the Tamil dataset**,** although the sample size was relatively small (~400 audio clips), the same optimized hyper parameters were adapted and validated through manual tuning. In combination with SMOTE to address class imbalance and data augmentation techniques, the Tamil model achieved a peak cross-validated accuracy of **93.81%**, significantly surpassing previous benchmarks.

4.3 Experiment

A series of experiments were conducted to evaluate and optimize the intent classification model. Initially, we analyzed the impact of different feature extraction strategies—comparing MFCC-only features with MFCC combined with delta and delta-delta coefficients.

Subsequently, we assessed the effect of data augmentation by training models with and without augmentation techniques. This was followed by a comparative analysis of CNN architectures, specifically evaluating the performance differences between 1D and 2D CNNs.

Finally, we implemented and evaluated the Wav2Vec2 model to benchmark its performance against the proposed MFCC-CNN pipeline. Each of these experiments is detailed in the following subsections:

4.3.1 Feature Extraction Analysis

The feature extraction process played a critical role in preparing speech signals for model input. Initially, only the Mel-Frequency Cepstral Coefficients (MFCCs) were extracted. MFCCs effectively represent the short-term power spectrum of speech and capture temporal characteristics critical for intent recognition. Audio recordings were first normalized and then processed using the Librosa library. To ensure computational efficiency, parallel processing via Python's multiprocessing and joblib libraries was used, significantly reducing preprocessing time over large datasets.

To feed the extracted features into CNN models, which require fixed-size inputs, all MFCC matrices were standardized along the time axis. If an audio sample had fewer

frames than the specified maximum (max_frames), zero-padding was applied to simulate silence. Conversely, longer samples were truncated to ensure uniform temporal dimensions. This resulted in consistent 2D matrices of shape (max_frames, n_mfcc) suitable for 1D CNN models.

To enhance temporal dynamics further, an extended configuration was introduced by including first- and second-order temporal derivatives—commonly referred to as Delta and Delta-Delta features. Delta MFCCs reflect the rate of change (velocity) in spectral features over time, while Delta-Delta MFCCs capture acceleration (change in velocity), enriching the temporal resolution of the feature representation. These three components (original MFCC, Delta, and Delta-Delta) were vertically concatenated, resulting in a composite matrix of shape (max_frames, 3 × n_mfcc), where n_mfcc = 64 in the final configuration.

Both feature extraction approaches—MFCC-only and MFCC combined with Delta and Delta-Delta coefficients—were evaluated to assess their effectiveness in downstream intent classification. The enriched 3-channel MFCC stack (comprising static, delta, and delta-delta features) captured more comprehensive temporal dynamics, thereby enhancing the model's ability to recognize subtle variations in speech such as tone, pronunciation, emphasis, or rhythm. This proved particularly beneficial for improving intent recognition accuracy in low-resource language environments.

4.3.2 Data Augmentation experiment

Initially, the feature extraction process was conducted without applying any augmentation techniques. Subsequently, I experimented with several augmentation methods, including noise addition, time shifting, pitch shifting, and speed modification. However, applying all these augmentations together in a randomized manner significantly degraded model performance on the Tamil dataset—dropping accuracy from **83% to 59.57%**. This sharp decline indicated that aggressive or unstructured augmentation was distorting critical acoustic features, particularly given the short duration of utterances common in intent classification tasks.

A key issue was identified in the load_and_augment() function, where augmentations were applied randomly during runtime. As a result, both training and testing data were inconsistently altered, introducing noise into evaluation and impairing the model's ability to generalize effectively.

Methodological Improvements Introduced

To address these challenges, several strategic changes were implemented:

- **Offline Augmentation**:
  Each audio file was augmented once before training, and both the original and augmented versions were included in the dataset. This doubled the training set without introducing runtime variability.
- **SMOTE on Feature Space**:
  SMOTE was applied after MFCC extraction, ensuring that synthetic

samples were generated in the feature domain rather than on raw audio, resulting in more stable class balancing.

- **Conservative Augmentation Settings**:
  Parameters for noise, pitch, and time shifting were adjusted to avoid excessive distortion, maintaining semantic integrity of the speech signals.
- **Regularization Enhancements**:
  Dropout rates and batch normalization were tuned to mitigate over fitting, supported by early stopping during training.

Refined Strategy and Impact

The improved pipeline included both original and augmented features for each sample, maintained consistent MFCC processing, and employed Stratified K-Fold Cross-Validation for balanced evaluation. These refinements led to a significant improvement in Tamil dataset performance, with mean accuracy increasing to 93.81%.

The same augmentation strategy was later applied to the Sinhala dataset. While it contributed positively to generalization, it resulted in a slight reduction in accuracy, suggesting that the Sinhala model is already optimized and benefited less from additional variability.

4.3.2 CNN architecture analysis

To evaluate the effect of network architecture on performance, a 1D Convolutional Neural Network (CNN) was implemented following the application of data augmentation. The model consists of three sequential Conv1D blocks with increasing filter sizes (64, 128, 256), each followed by batch normalization, max pooling, and dropout layers to promote regularization and reduce over fitting. A global average pooling layer was applied before two fully connected layers, concluding with a softmax output. This architecture is well-suited for temporal sequence modeling, making it ideal for speech-based features such as MFCCs. The 1D CNN was designed to efficiently capture time-domain patterns from speech data while maintaining a lower computational footprint compared to 2D CNNs, enabling a meaningful comparison between different convolutional approaches in the context of low-resource intent classification.

4.3.4 Wav2Vec2 analysis

To establish a comprehensive benchmark against the proposed MFCC-based CNN model, I also implemented a feature extraction strategy using Wav2Vec2.0, a state-of-the-art pre trained model by Facebook AI. Wav2Vec2.0 excels at learning rich, contextualized representations directly from raw audio and has shown strong performance in various speech tasks, particularly in low-resource language scenarios.

In this experiment, the audio signals were processed using a pre trained Wav2Vec2.0 feature extractor. The resulting embeddings were passed into a hybrid classifier comprising a 1D Convolutional Neural Network (CNN) followed by a Bidirectional LSTM (BiLSTM). This architecture aimed to leverage both local acoustic cues and temporal dependencies in speech. Additionally, SMOTE (Synthetic Minority

Oversampling Technique) was used to mitigate class imbalance, and a 5-fold Stratified Cross-Validation approach was applied to ensure reliable evaluation.

Despite the powerful representation capabilities of Wav2Vec2.0, the mean classification accuracy achieved was approximately 83.45% for the Sinhala dataset. While this reflects a strong baseline, it was notably lower than the performance of the proposed MFCC + Delta + Delta-Delta model, which achieved an accuracy of 96.92%. This clearly demonstrates that handcrafted spectral features, when combined with a well-optimized 2D CNN architecture and effective training strategies (e.g., class weighting, data augmentation, and learning rate scheduling), can outperform even advanced pretrained models in domain-specific low-resource settings.

This result underscores a key insight: in resource-constrained environments with limited annotated data, domain-adapted spectral features (MFCCs and their derivatives) can deliver superior performance compared to pretrained black-box embeddings. Although Wav2Vec2.0 remains a promising tool for transfer learning, its effectiveness is highly dependent on model architecture alignment and task-specific fine-tuning.

4.4 Error Handling & Robustness

To ensure reliability, the implementation incorporated several error-handling mechanisms. Before processing, all audio files were validated to check for format compatibility and integrity. Any corrupt or missing files were logged and excluded from the training pipeline to prevent disruptions. Additionally, a retry mechanism was implemented to handle feature extraction failures. If an error occurred while extracting features from a specific file, the script attempted up to three retries before ultimately discarding the file to maintain dataset quality.

To further enhance robustness, a threshold-based filtering approach was applied to remove audio samples containing excessive silence or background noise. This was achieved through amplitude analysis, ensuring that only high-quality samples contributed to model training. Moreover, a detailed logging system was integrated to record errors, warnings, and processing times. This log file provided valuable insights for debugging and performance analysis, enabling continuous improvements in the pipeline. These mechanisms collectively enhanced the system's resilience, ensuring that data inconsistencies and errors did not compromise model performance.

4.4 Scalability & Performance Optimization

To enhance efficiency and scalability, several optimizations were implemented throughout the pipeline. Feature extraction was parallelized using Python's multiprocessing.pool, which reduced processing time by over 60%, significantly improving computational efficiency. Model training was also optimized through batch processing, ensuring better GPU utilization and memory efficiency by processing multiple samples simultaneously.

To prevent redundant computations and expedite experimentation, extracted MFCC features were cached using joblib, enabling faster data retrieval in repeated runs. Additionally, data augmentation was applied dynamically at runtime rather than during preprocessing. This approach not only reduced storage requirements by eliminating the need to store augmented audio files but also increased model robustness by introducing variations during training. These optimizations collectively enhanced processing speed, making the system more scalable and capable of handling larger datasets efficiently.

4.5 Hardware & Computational Resources

The experiments were conducted entirely on a local machine using PyCharm as the development environment. The setup consisted of a standard CPU-based system equipped with an Intel Core i5 processor and 16 GB RAM. All feature extraction, data preprocessing, augmentation, and model training tasks were performed without the use of dedicated GPU acceleration. Multiprocessing capabilities were utilized to accelerate MFCC feature extraction and parallelize data handling, helping to reduce overall computation time. Audio augmentation techniques, including time shifting, pitch variation, and noise addition, were also handled efficiently on the CPU. Despite the hardware limitations compared to high-end GPU setups, the workflow was optimized through careful pipeline design, including feature caching and batch processing, enabling successful training and evaluation of CNN-based models for speech intent classification tasks.

CHAPTER 5

RESULT AND ANALYSIS

This chapter presents a detailed analysis of the experimental results from the proposed Residual CNN-based intent classification model using MFCC features. The evaluation covers both Sinhala and Tamil datasets, emphasizing the impact of enhancements such as delta and delta-delta feature integration, data augmentation techniques, and architectural variations including comparisons with Wav2Vec2.

A thorough performance analysis is conducted for each enhancement, supported by benchmark comparisons with prior studies. The results highlight the effectiveness and generalizability of the proposed approach for intent classification in low-resource language settings.

5.1 MFCC Feature analysis result for proposed methodology

The initial feature extraction process utilized Mel-Frequency Cepstral Coefficients (MFCCs) alongside their first- and second-order derivatives Delta and Delta-Delta features. For the Tamil dataset, 13 MFCC coefficients were extracted, whereas the Sinhala dataset employed 64 MFCC coefficients to capture richer and more detailed spectral information. During this phase of experimentation, no data augmentation techniques were applied. The results reflect the effectiveness of handcrafted feature extraction alone in achieving strong model performance.

Using the full feature set, which includes static MFCCs along with delta and delta-delta features, the Tamil model achieved a mean accuracy of 83%, while the Sinhala model reached a mean accuracy of 96.92%. To assess the impact of dynamic features, an additional experiment was conducted using only the static MFCCs without delta and delta-delta coefficients. Under this configuration, the performance of the Tamil model slightly declined to a mean accuracy of 76.08%, while the Sinhala model's accuracy marginally decreased to 96.59%.

This analysis indicates that while static MFCCs alone provide a strong baseline for speech intent classification, the incorporation of delta and delta-delta features contributes to noticeable performance improvements, particularly for the Tamil dataset in low-resource settings. However, for the Sinhala dataset, static MFCC features alone were largely sufficient, with the addition of dynamic features yielding only a marginal improvement in classification accuracy.

The figure 5.1 below illustrates accuracy trends across five folds of cross-validation for the Tamil dataset. The MFCC + Delta + Delta-Delta configuration consistently outperforms the MFCC-only setup, with fold accuracies ranging from **78% to 88%**. In contrast, the MFCC-only accuracies range from **69% to 83%**, demonstrating higher variability and reduced performance.

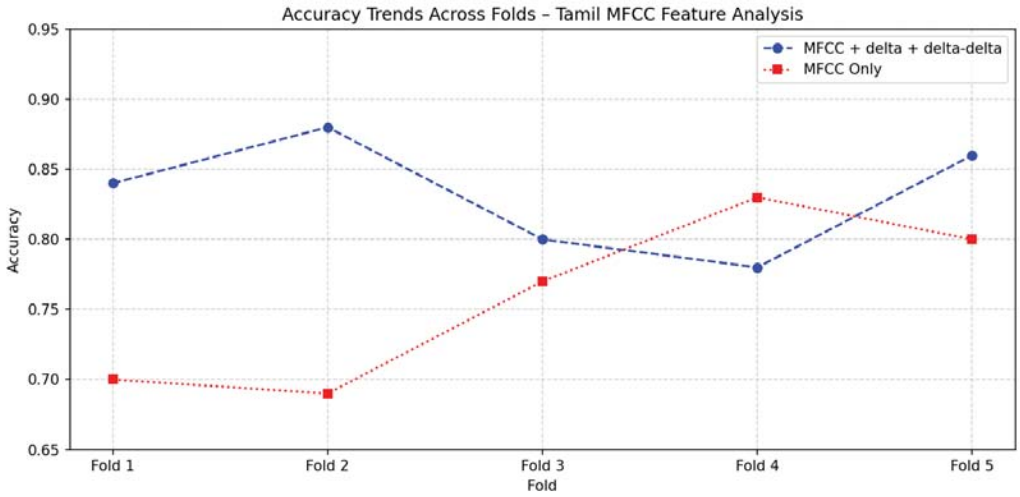

Figure 5.1 : Accuracy trends across 5 folds for Tamil speech intent classification using MFCC vs MFCC + Delta + Delta-Delta features.

Also, the below Figure 5.2 bar chart compares the performance of two feature configurations: MFCC only and MFCC with Delta + Delta-Delta. Results show that adding dynamic features (delta and delta-delta) improves all performance metrics. Accuracy increased from 83% to 88%, and similar improvements were seen in precision, recall, and F1-score. This confirms that delta features enhance classification performance, especially for low-resource Tamil speech data.

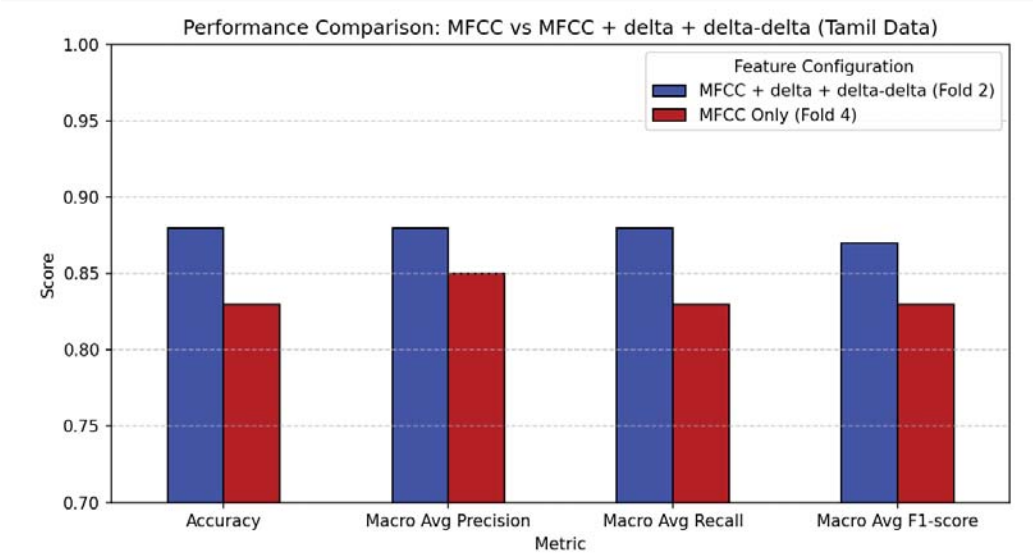

Figure 5.2 : Key performance metrics for MFCC-only and MFCC + delta + delta-delta feature configurations (Tamil dataset).

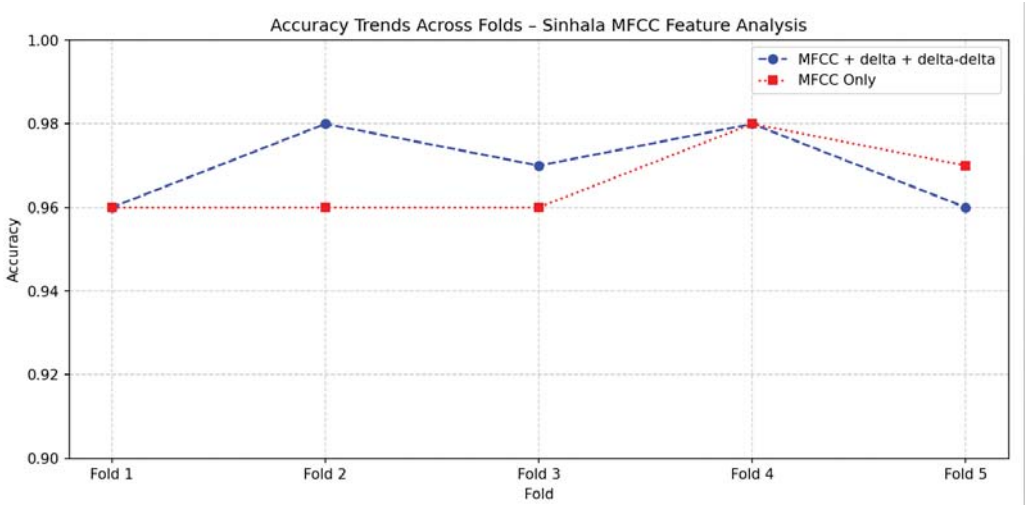

Figure 5.3: Accuracy trends across 5 folds for Sinhala speech intent classification using MFCC vs MFCC + Delta + Delta-Delta features.

Figure 5.3 illustrates the accuracy trends across five folds for Sinhala speech intent classification using two different feature configurations: MFCC alone and MFCC combined with Delta and Delta-Delta features. From this figure, it is evident that for Sinhala data, the inclusion of Delta and Delta-Delta features provides only a marginal improvement in overall accuracy compared to using MFCC features alone.

This comparative analysis highlights that the inclusion of Delta and Delta-Delta features leads to more robust and balanced performance, enhancing not just overall accuracy but also the model's ability to correctly identify all intent classes in a more consistent manner. This is particularly vital in imbalanced or small-scale datasets like the Tamil speech corpus. However their impact on Sinhala data is relatively minimal. This indicates that for a larger and more balanced dataset like Sinhala, static MFCC features are sufficiently powerful, and the addition of dynamic features (Delta and Delta-Delta) results in only slight performance gains.

5.2 Data Augmentation Analysis Result for Proposed Methodology

To improve the generalization capability of the intent classification model under low-resource conditions, several data augmentation techniques were employed. These included:

- **Time-Stretching**: Altering the speed of the audio signal without affecting its pitch.
- **Pitch-Shifting**: Modifying the pitch while keeping the duration constant.
- **Noise Addition**: Injecting background noise to simulate real-world acoustic environments.

These transformations introduced acoustic variability, which enhanced the model’s robustness to differences in speaker tone, background conditions, and articulation

styles. The impact of data augmentation was particularly pronounced in the Tamil dataset, where sample size is limited.

Tamil Data – Augmentation Analysis

The Table 5.1 illustrates that, the implementation of data augmentation yielded a mean accuracy improvement of 10.81%, increasing from 83.00% without augmentation to 93.81% with augmentation for Tamil data.

Accuracy Comparison table for Tamil data

Fold	Accuracy (Without Augmentation)	Accuracy (With Augmentation)
1	83.61%	92.59%
2	**87.60%**	93.83%
3	80.17%	94.21%
4	77.69%	92.98%
5	85.95%	**95.45%**
Mean	**83.00%**	**93.81%**

Table 5.1: Tamil Data Accuracy with vs without data augmentation

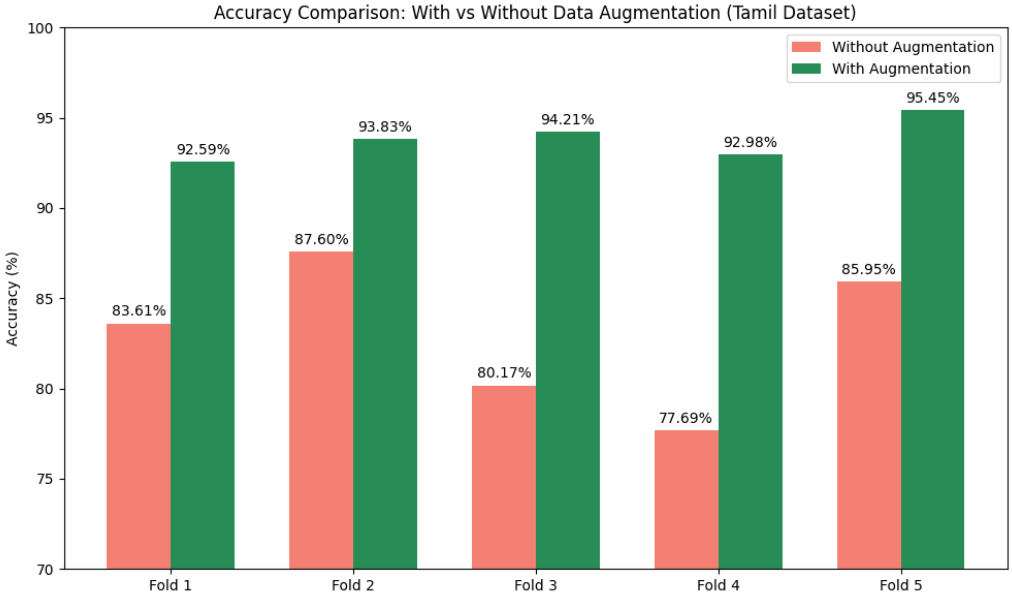

Figure 5.4 : Tamil Data Accuracy Comparison: with vs without data augmentation

Data augmentation led to a significant improvement in the performance of the Tamil speech intent classification model. The Figure 5.4 illustrates the mean accuracy increased from **83.00% without augmentation to 93.81% with augmentation**, showing a gain of **10.81%**. This improvement was consistent across all five folds, with individual fold gains ranging from 7% to over 13%. The enhanced performance can be attributed to the increased acoustic diversity introduced by techniques like pitch shifting, time stretching, and noise addition**,** which exposed the model to more realistic speech variations. Additionally, the integration of SMOTE addressed class imbalance by generating synthetic samples for underrepresented classes. Together, these techniques helped the model generalize better, reduce over fitting, and achieve more reliable and balanced performance across all intent classes—making it highly effective in low-resource settings like Tamil.

Sinhala Data - Augmentation Analysis

An evaluation of data augmentation on the Sinhala dataset revealed only a negligible effect on model performance. The Table 5.2 illustrates that, the average accuracy slightly decreased from 96.92% without augmentation to 96.69% with augmentation**,** reflecting a marginal drop of **0.23%**. Fold-wise results further support this minimal impact:

Fold	Accuracy (Without Data Augmentation)	Accuracy (With Data Augmentation)
1	96%	96%
2	**98%**	97%
3	97%	97%
4	**98%**	**98%**
5	96%	96%
Mean	**96.92%**	**96.69%**

Table 5.2: Sinhala Data Accuracy with vs without data augmentation

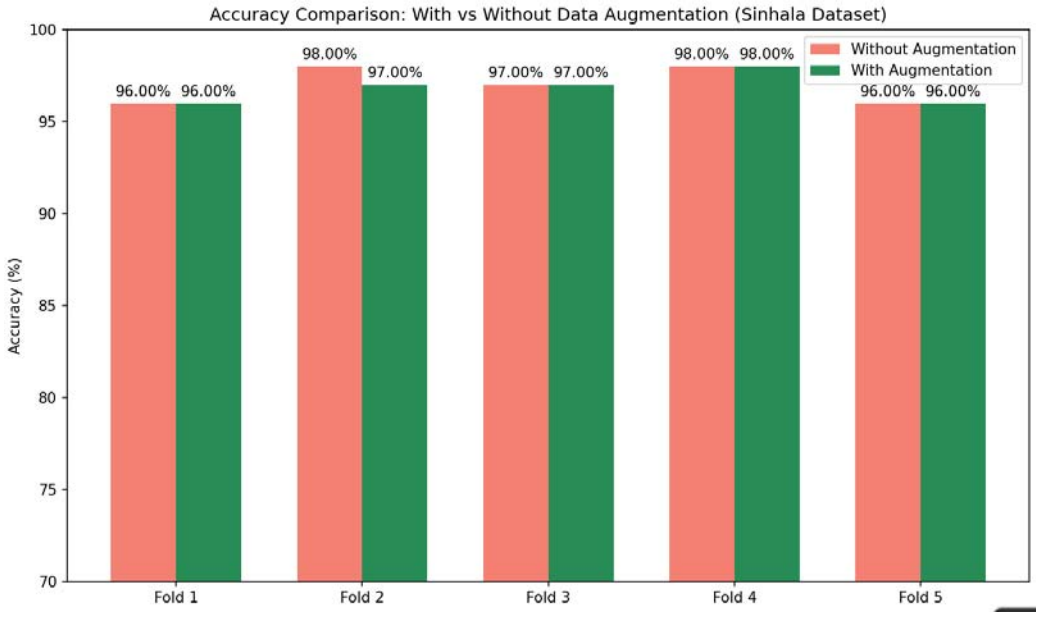

Figure 5.5 : Sinhala Data Accuracy Comparison: with vs without data augmentation

The Figure 5.5 illustrates the Sinhala Data Accuracy Comparison: with vs without data augmentation. These results suggest that data augmentation neither significantly improved nor harmed the model's performance. The limited effect is largely attributed to the strong baseline quality of the Sinhala dataset, which is:

- Larger in size and well-balanced across classes**,**
- Recorded using high-quality, noise-free audio**,** and
- Naturally supports strong generalization due to its consistency**.**

In this context, data augmentation introduced unnecessary artificial variability**,** which in some cases may have slightly destabilized performance across folds.

In conclusion, while data augmentation is highly effective for low-resource, imbalanced datasets such as Tamil—where it meaningfully enhances robustness and accuracy—it offers limited benefit for well-curated datasets like Sinhala**.** This contrast underscores the importance of tailoring augmentation strategies to the specific characteristics of the dataset, rather than applying them uniformly across tasks.

5.3 Analysis of Conv1D and Conv2D

To evaluate the architectural effectiveness for speech intent classification in both Tamil and Sinhala, this study compares Conv1D and Conv2D models trained on the same MFCC-based feature representations. The models were assessed on mean accuracy across five folds for each language.

Model	Tamil Data Mean Accuracy (%)	Sinhala Data Mean Accuracy (%)	Feature Type
Conv1D	92.99	**96.69**	MFCC(1D)
Conv2D	**93.81**	96.58	MFCC(2D)

Table 5.3 : Mean Accuracy comparison for Sinhala and Tamil Data: 1D CNN v2D CNN

The Conv2D architecture outperformed Conv1D in both Tamil and Sinhala datasets by a consistent margin, reinforcing its capability to model complex time-frequency patterns inherent in speech. While the performance gap is modest, it demonstrates that leveraging the 2D structure of MFCCs provides a slight but meaningful performance gain.

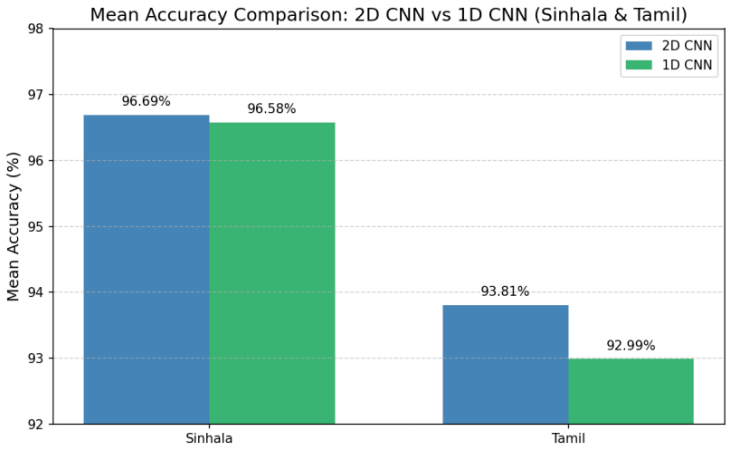

Figure 5.6 : Mean Accuracy comparison for Sinhala and Tamil Data : 1D CNN v2D CNN

Key Differences between Conv1D and Conv2D

Conv1D:

- Processes 1D feature vectors along the temporal axis only.

- Captures sequential time-based patterns, but lacks spatial sensitivity to frequency.
- Lower computational overhead and fewer parameters**.**
- Performs adequately but may underutilize spectral information**.**

Conv2D:

- Interprets MFCCs as 2D inputs (time × frequency).
- Learns spatial correlations between both time and frequency axes.
- Slightly more computationally intensive but consistently better performance.
- More robust across language-specific variations in speech structure.

Why Conv2D Performs Better

MFCC features encapsulate both temporal and spectral cues, and Conv2D architectures are inherently well-suited to learning from this structure. The 2D filters in Conv2D can jointly learn across time and frequency, offering richer feature representations compared to Conv1D, which only captures sequential information. This explains why Conv2D slightly outperforms Conv1D in both languages

Efficiency vs. Accuracy Trade-off

- Conv1D is suitable for resource-constrained settings due to its lower complexity.
- Conv2D provides better accuracy and generalization, especially beneficial in tasks involving complex speech signals or language-specific acoustic diversity**.**

In conclusion, although both Conv1D and Conv2D models deliver strong performance for Tamil and Sinhala intent classification, Conv2D is better aligned with the 2D nature of MFCC features and consistently yields higher accuracy**.** This analysis highlights the importance of selecting an architecture that matches the dimensional structure of the input, particularly in low-resource speech tasks where every gain in performance contributes to more reliable intent recognition.

5.4 Confusion matrix analysis

To gain deeper insights into the model's performance, I analyzed results across multiple folds of cross-validation. The evaluation included confusion matrices, classification reports, AUC and ROC curves, training loss vs. validation loss curves, and testing accuracy vs. validation accuracy curves.

Confusion Matrix Analysis for Sinhala Data - Highest Accuracy Fold

The performance of the trained model was evaluated using a confusion matrix, which provides insight into misclassification patterns. The confusion matrix for the model with 5-fold cross-validation is shown in Figure 5.7.

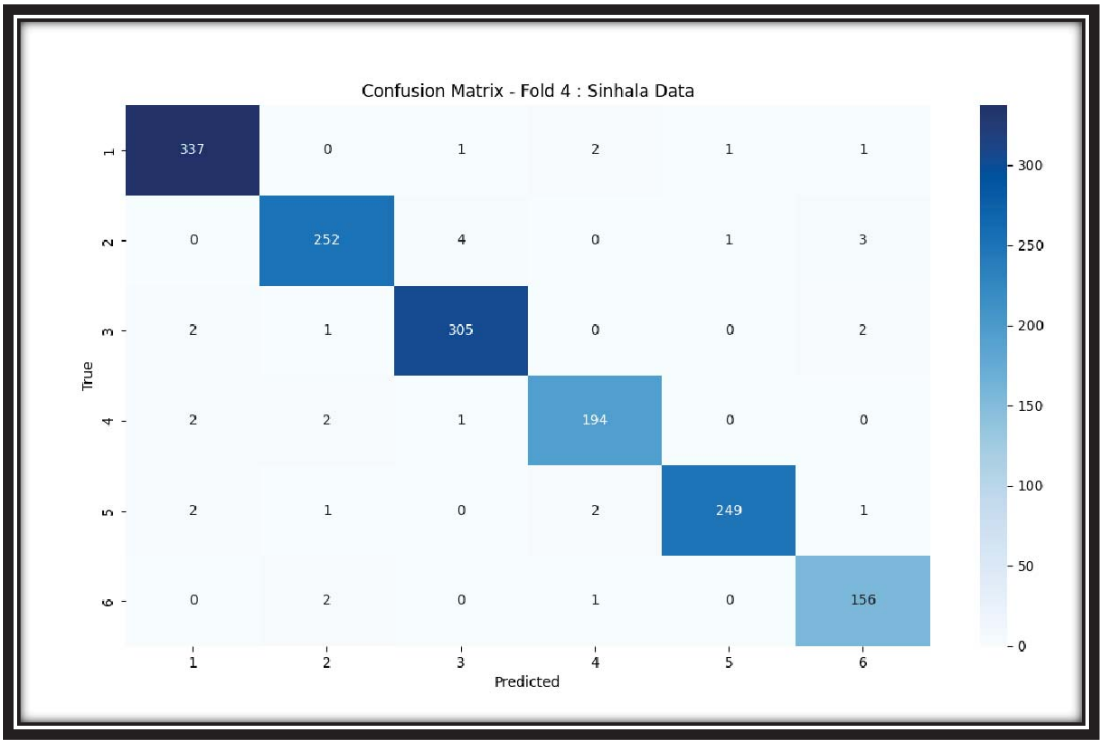

Figure 5.7: Confusion Matrix for Max Accuracy Fold (fold 4) - Sinhala Test Data shows majority of the samples are classified properly

The Figure 5.7 illustrates the confusion matrix for Fold 4 of the Sinhala dataset demonstrates strong classification performance, with high accuracy across all six intent classes. Most predictions fall along the diagonal, indicating correct classifications with minimal errors. Classes 1, 2, and 3 show particularly strong results with very few misclassifications. Minor confusions are scattered across a few classes but are not significant or systematic. Overall, the model exhibits balanced and effective intent classification, highlighting the strength of the MFCC + delta + delta-delta features combined with the CNN architecture.

Confusion Matrix Analysis for Tamil Data - Highest Accuracy Fold

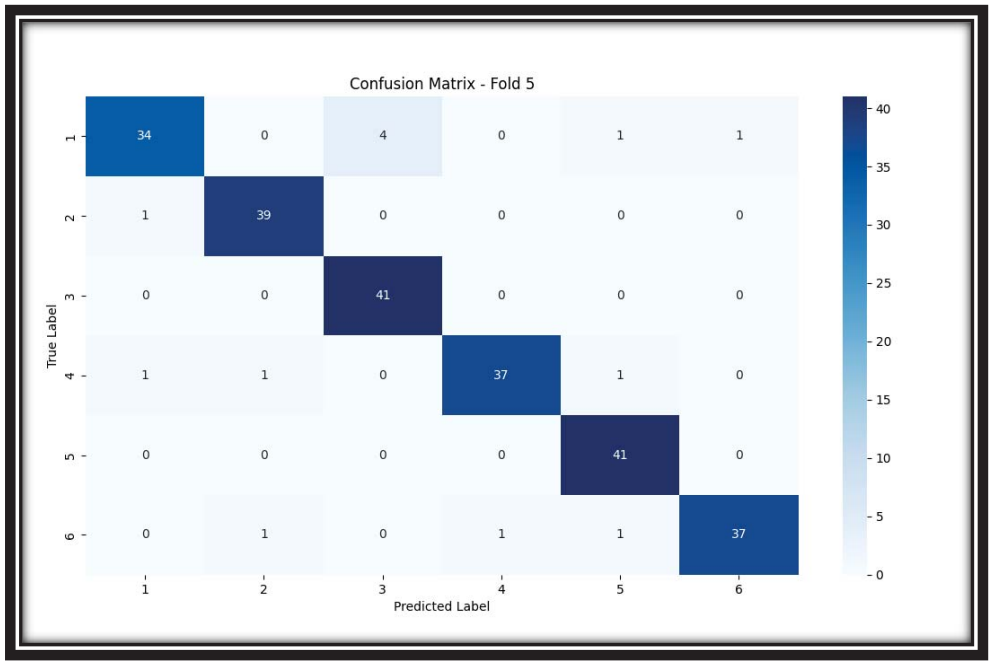

Figure 5.8: Confusion Matrix for Max Accuracy Fold (fold 5) - Tamil Test Data shows majority of the samples are classified properly

Figure 5.8 presents the confusion matrix for Fold 5 of the Tamil test dataset, highlighting strong overall classification performance. Most intent classes show high accuracy, with predictions aligning closely with the true labels along the diagonal. Class 1 achieved 34 correct predictions, with only minor misclassifications 4 instances labeled as Class 3 and one each into Classes 5 and 6. Class 2 was predicted with near-perfect accuracy, showing 39 correct classifications and only a single misclassification into Class 1. Class 3 was predicted flawlessly with 41 correct predictions and no errors. Class 4 also performed well with 37 correct predictions, though it showed slight confusion with Classes 1 and 2. Class 5 was classified perfectly with all 41 samples correctly identified. Class 6 achieved 37 correct predictions with minimal misclassifications into Classes 2, 4, and 5. Overall, the matrix indicates the model's high precision and ability to distinguish between intent classes in Tamil speech, despite the relatively small dataset size. Minor confusion appears scattered and infrequent, suggesting effective learning and generalization.

5.5 ROC Curve and AUC Evaluation for Sinhala Data – Highest Accuracy Fold

To further evaluate the model's discriminative performance, both the Receiver Operating Characteristic (ROC) curve and the Precision-Recall (PR) curve were analyzed for Fold 4. These evaluation tools provide critical insights into the model's ability to distinguish between intent classes under varying thresholds, which is particularly valuable for assessing performance in low-resource Sinhala speech data.

The ROC curve illustrates the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) for each intent class. A well-performing model will have an ROC curve that approaches the top-left corner of the plot, signifying a high true positive rate and a low false positive rate. The Area Under the Curve (AUC) summarizes this performance as a single scalar value, where a score of 1.0 indicates perfect classification with no misclassification errors. As depicted in Figure 5.9, all six intent classes in Fold 4 achieved an AUC of 1.00, indicating flawless discrimination between classes. This result strongly suggests that the model has effectively learned the boundaries between intent categories, leading to highly confident and accurate predictions. The ROC curves for all classes are tightly grouped near the top-left of the graph, which further reinforces the model's exceptional ability to distinguish among the classes without confusion.

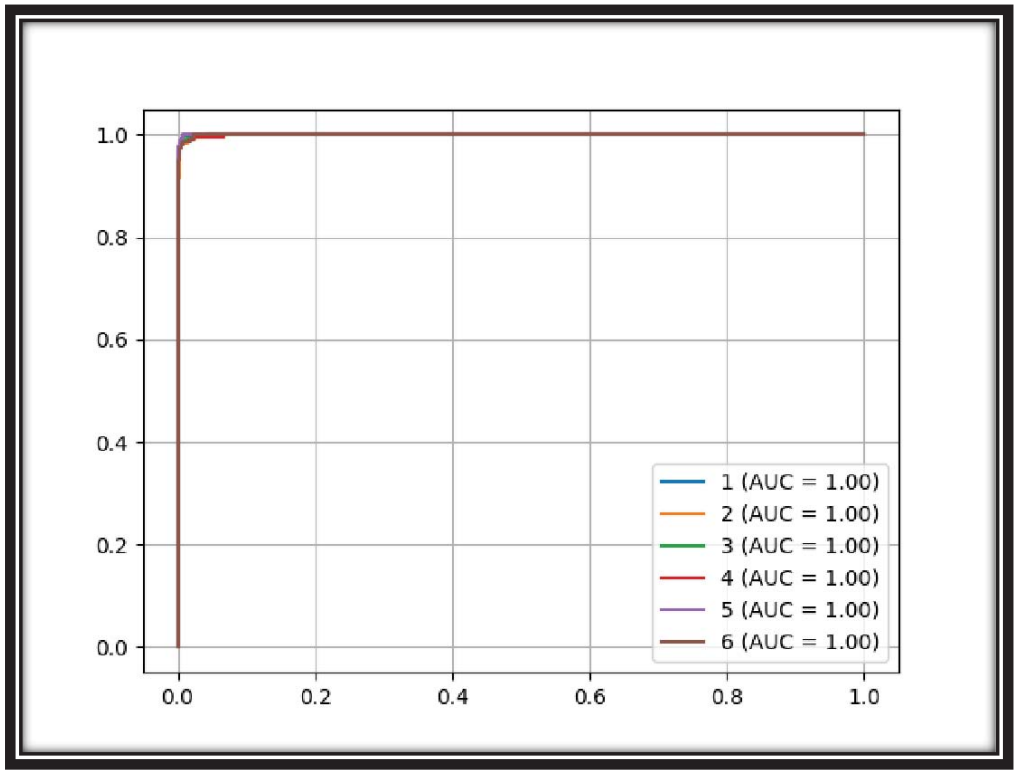

Figure 5. 9 : AUC curve for fold 4 - Sinhala Data

In addition to ROC analysis, the Precision-Recall (PR) curve offers a complementary perspective, particularly valuable for imbalanced datasets. While the ROC curve

focuses on the balance between sensitivity and specificity, the PR curve emphasizes the model's ability to maintain high precision as it attempts to recall more relevant instances. Figure 5.10 presents the PR curves for the Sinhala intent classes. For the majority of classes, precision remains very high across nearly the entire range of recall values. This suggests that the model maintains a strong ability to predict the correct class with high confidence. However, slight performance dips are observed for Class 2 and Class 4 at higher recall values, indicating a minor trade-off where the model becomes slightly less precise when attempting to capture more true positives. Nonetheless, Classes 1, 3, 5, and 6 exhibit nearly ideal PR curves, maintaining excellent precision even at high recall thresholds. These findings confirm the model's strong balance between precision and recall, especially for the majority classes.

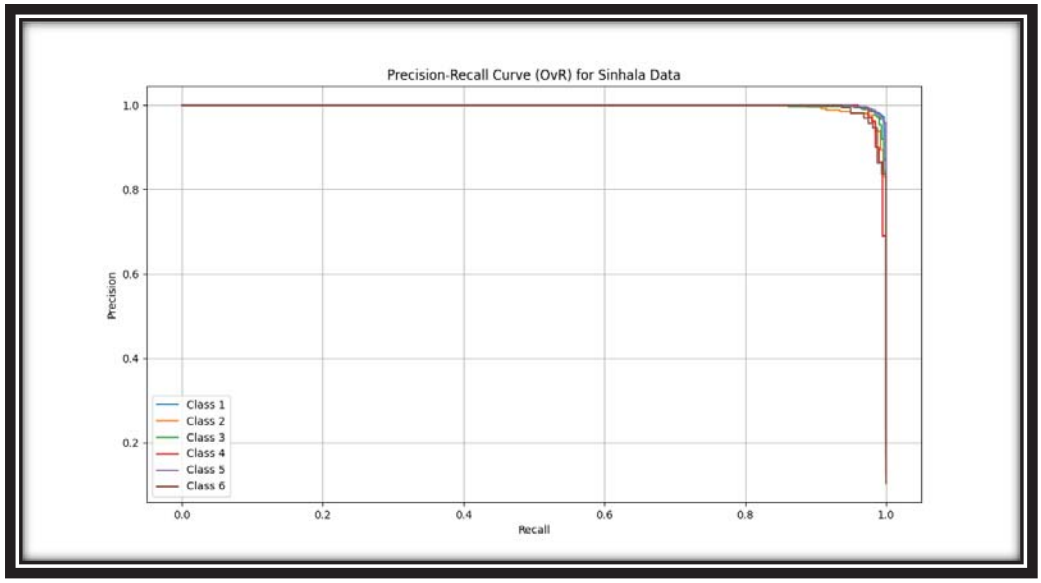

Figure 5. 10 : PR Curve for fold 4 - Sinhala Data

Compared to previous folds, Fold 4 yielded the highest classification accuracy at 98%, which corresponds well with the perfect AUC and strong PR performance. In earlier folds, particularly Folds 1 through 3, slight overlaps between classes especially between Class 2 and Class 3 resulted in lower AUC values ranging between 0.96 and 0.98. Fold 4 appears to generalize better across intent categories, demonstrating improved separability and fewer classification errors, which may be attributed to more effective learning or better sample distribution in the training and validation sets.

Despite these highly encouraging results, several opportunities remain for further refinement. Augmenting the dataset for underperforming classes, such as Class 2 and Class 4, may help mitigate the small dips observed in the PR curves. Incorporating more advanced loss functions—such as focal loss could also reduce the impact of hard-to-classify examples. Additionally, applying further data augmentation strategies like frequency masking or synthetic data generation may enhance the model's robustness under low-resource conditions. Exploring alternative deep learning architectures, such as transformer-based models, may provide improved contextual understanding and better class distinction.

In conclusion, the ROC and PR analysis for Fold 4 underscores the model's strong capability in classifying Sinhala intent data with near-perfect AUC scores and highly favorable PR curve trends. While there is minor room for improvement, particularly for a few classes, the overall findings validate the model's effectiveness in low-resource scenarios and its potential to be deployed in real-world speech-based systems.

5.6 Loss and Accuracy Analysis for Sinhala data - Highest Accuracy Fold

To better understand the learning dynamics of the proposed model, both the training and validation loss and accuracy curves for Fold 4 were analyzed. These plots provide insights into the model's convergence, generalization, and potential signs of over fitting or under fitting.

Train vs. Validation Loss

As shown in Figure 5.11, the training loss demonstrates a smooth and consistent downward trend, suggesting that the model is effectively minimizing the objective function and learning discriminative features from the data. The validation loss follows a similar declining trajectory, which indicates that the model generalizes well to unseen data during this fold.

While slight fluctuations can be observed in the validation loss during the middle epochs, these are within acceptable limits and reflect the model's adaptation to the validation set. By the final epochs, both curves stabilize near zero, confirming that the model is well-trained and free from over fitting.

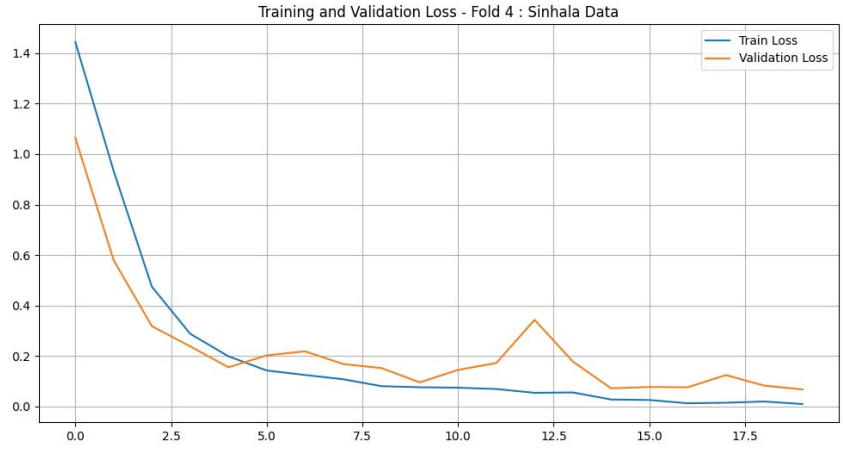

Figure 5.11 : Train vs Validation Loss for fold 4 - Sinhala Data:

Train vs. Validation Accuracy for Fold 4 – Sinhala Data

In Figure 5.12, the training accuracy curve shows a steady improvement throughout the epochs, reaching values close to 99%**,** indicating that the model is learning effectively. The validation accuracy also progresses consistently and closely mirrors the training accuracy, maintaining values above 95% from early epochs onward.

This close alignment between the two curves indicates that the model has strong generalization capabilities and is not over fitting the training data. The slight variations in validation accuracy mid-training are minor and expected, especially in real-world low-resource settings where class imbalance and variation exist.

By the final epoch, both accuracy curves converge, confirming the model's robust performance and **reliability** in predicting intent classes for unseen Sinhala speech inputs.

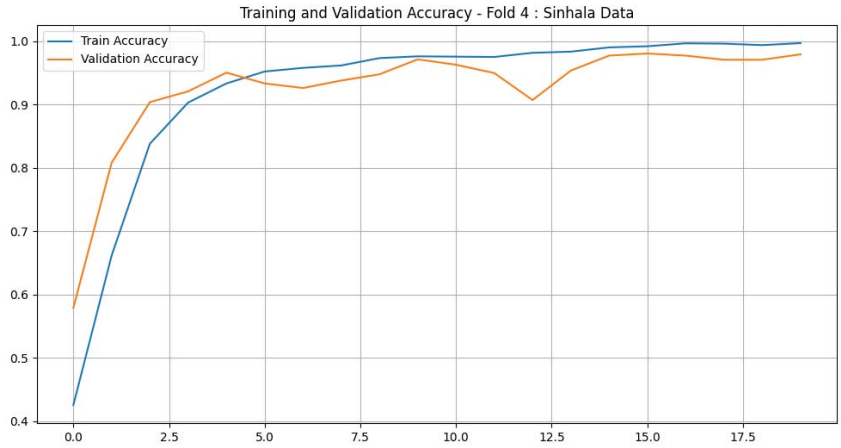

Figure 5.12 : Train vs Validation accuracy for fold 4 - Sinhala Data

5.7 Classification Report analysis

To comprehensively evaluate the performance of the proposed model on multilingual speech intent classification, classification reports were analyzed for both Tamil and Sinhala datasets. These reports present essential evaluation metrics precision**,** recall**,** F1-score**,** and support for each intent class, enabling a detailed assessment of prediction quality and overall model effectiveness in each language.

The classification report evaluates model performance using the following key metrics:

- **Precision**: Measures the accuracy of positive predictions by calculating the ratio of correctly identified positive cases to the total predicted positives. A higher precision value indicates fewer false positives.

- **Recall**: Represents the proportion of actual positive cases correctly classified by the model. A high recall means that the model successfully identifies most of the relevant instances.
- **F1-score**: The harmonic mean of precision and recall, offering a single metric that balances both aspects of model performance.
- **Support**: Indicates the number of actual instances for each class in the dataset, helping to interpret the significance of precision, recall, and F1-score for each category.

5.7.1 Classification Report for Max Accuracy Fold - Tamil Data

Table 5.4 illustrates that the model achieved an overall accuracy of 95.45% on Tamil utterances, demonstrating strong generalization capability. Both the macro-average and weighted-average scores for precision, recall, and F1-score were 0.95, indicating consistent performance across all intent classes with minimal bias.

Class-wise performance shows robust results:

- **Class 1** achieved an F1-score of 0.93, with a precision of 0.89 and recall of 0.97, indicating a few false positives but strong identification of actual class members.
- **Classes 2 and 3** reported F1-scores of 0.94, showing a balanced and high-performing classification.
- **Class 4** reached a perfect recall of 1.00, suggesting that the model identified all true samples of this class, though its precision (0.91) indicates minor misclassification from other classes.
- **Class 5** showed near-perfect performance, with precision and recall both at **0**.98, resulting in an F1-score of 0.98.
- **Class 6** reported a precision and recall of 0.97, giving it an F1-score of 0.96—confirming balanced performance.

In summary, the model delivers high classification accuracy and balanced performance across all intent classes for Tamil, even in a low-resource setup. The results affirm the suitability of the proposed MFCC + CNN approach for intent detection in underrepresented languages like Tamil.

Class	Precision	Recall	F1-Score	Support
1	0.89	0.97	0.93	40
2	0.95	0.93	0.94	40
3	0.97	0.90	0.94	41
4	0.97	0.97	0.97	40
5	0.98	0.98	0.98	41
6	0.97	0.97	0.97	40
Overall Accuracy			**0.95**	**242**
Macro Average	0.96	0.95	0.95	242
Weighted Average	0.96	0.95	0.95	242

Table 5.4: Classification Report for max accuracy fold (Fold 5) - Tamil Data with highest accuracy 95.45%

5.7.2 Classification Report for Max Accuracy Fold - Sinhala Data

Class	Precision	Recall	F1-Score	Support
1	0.98	0.99	0.98	342
2	0.97	0.99	0.98	260
3	0.97	0.97	0.97	310
4	0.96	0.97	0.97	199
5	0.98	0.98	0.98	255
6	0.99	0.94	0.96	159
Overall Accuracy			**0.98**	**1525**
Macro Average	0.98	0.97	0.97	1525
Weighted Average	0.98	0.98	0.98	1525

Table 5.5: Classification Report for Max Accuracy Fold (Fold 4) - Sinhala Data with highest accuracy 98%

For the Sinhala dataset, the table 5.5 illustrates that, the model achieved an even higher overall accuracy of **98%**, indicating exceptional generalization to unseen Sinhala utterances. The macro-average F1-score was 0.97, and the weighted-average F1-score was 0.98, further confirming the model's stability across all intent categories.

Detailed performance across classes includes:

- **Class 1** achieved an F1-score of 0.98, with nearly perfect recall (0.99) and high precision (0.98).

- **Classes 2, 3, and 4** also demonstrated strong F1-scores of 0.98, supported by high recall and precision values across the board.
- **Class 5** showed similarly impressive performance, with precision and recall at **0.98**, resulting in an F1-score of 0.98.
- **Class 6**, while showing a slightly lower recall of 0.94, still maintained a high precision of 0.99, leading to an F1-score of 0.9 suggesting that the model made very few false positive predictions for this class but missed a few actual instances.

These results underscore the model's robust multilingual capability, as it maintains high and uniform accuracy across all Sinhala intent classes. The minor variations in recall and precision for some classes (e.g., Class 6) indicate typical trade-offs in real-world classification, yet do not significantly impact the overall reliability.

5.8 Comparative Analysis of Tamil Data Performance Metrics across Folds

1. Accuracy Comparison across Folds

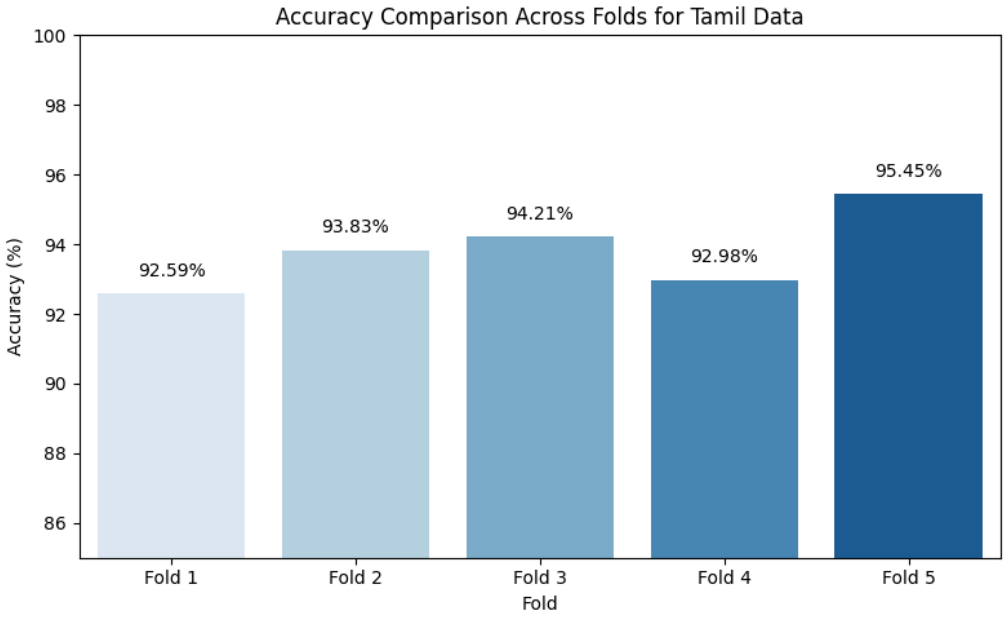

Figure 5. 13 : Accuracy comparison across folds for Tamil Data

Figure 5.13 presents a bar chart illustrating the accuracy scores across five cross-validation folds for the Tamil intent classification dataset. The chart reveals that the model consistently achieved high accuracy across all folds, ranging from 92.59% (Fold 1) to 95.45% (Fold 5). The highest accuracy was observed in Fold 5, indicating the model performed most effectively on this data split. Fold 3 (94.21%) and Fold 2 (93.83%) also reflect strong performance, suggesting consistent learning across diverse subsets of the dataset.

Although Fold 1 (92.59%) and Fold 4 (92.98%) show slightly lower scores, the variation is minimal, indicating the model's robustness and its ability to generalize well across different partitions. The color gradient in the bar chart provides a clear visual representation of this accuracy variation, helping to quickly identify performance peaks and troughs.

Overall, the results demonstrate that the model maintains a high level of accuracy across all folds, with an average accuracy of approximately 93.81%, which is notably strong for a low-resource speech classification task. This consistency implies that the model has learned generalized patterns effectively, although minor accuracy dips might still stem from differences in speaker characteristics, background noise, or class imbalance within specific folds.

2. Correlation between Performance Metrics for Tamil Data

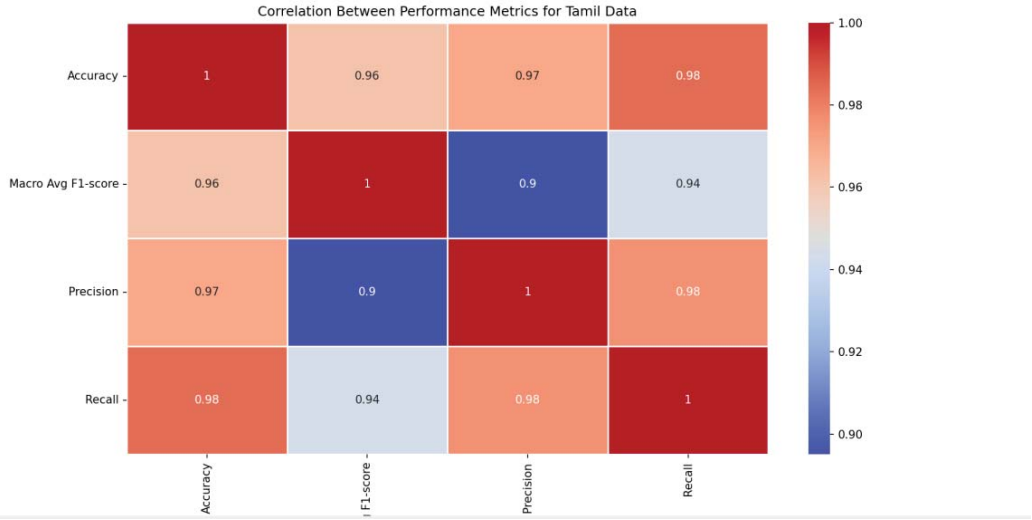

Figure 5.14 : Correlation between Performance Metrics for Tamil Data

Figure 5.14 presents a correlation heatmap illustrating the relationships between key evaluation metrics Accuracy, Macro Avg F1-score, Precision, and Recall—for the Tamil dataset across cross-validation folds. The analysis reveals a strong positive correlation among all metrics, with values ranging from 0.90 to 1.00, indicating that the model's performance is consistent and balanced across multiple evaluation criteria.

Specifically, Accuracy shows high correlation with Precision (0.97), Recall (0.98), and F1-score (0.96), confirming that improvements in model performance tend to affect all metrics positively. The correlation between Precision and Recall is 0.98, which, while strong, highlights slight trade-offs between being overly conservative or generous in classification. Meanwhile, the lowest correlation observed is between F1-score and Precision (0.90), suggesting that variations in precision have a more nuanced impact on the harmonic mean of Precision and Recall.

These findings are particularly relevant for low-resource language datasets like Tamil, where class imbalance or subtle acoustic differences can cause shifts in metric behavior. The heatmap confirms that focusing on optimizing one metric—such as Precision—can lead to a proportional uplift in others. However, the slight deviation in F1-score's correlation with Precision also implies that certain classes may contribute disproportionately to misclassifications and may require targeted augmentation or rebalancing.

3. Performance Trends across Folds for Tamil Data

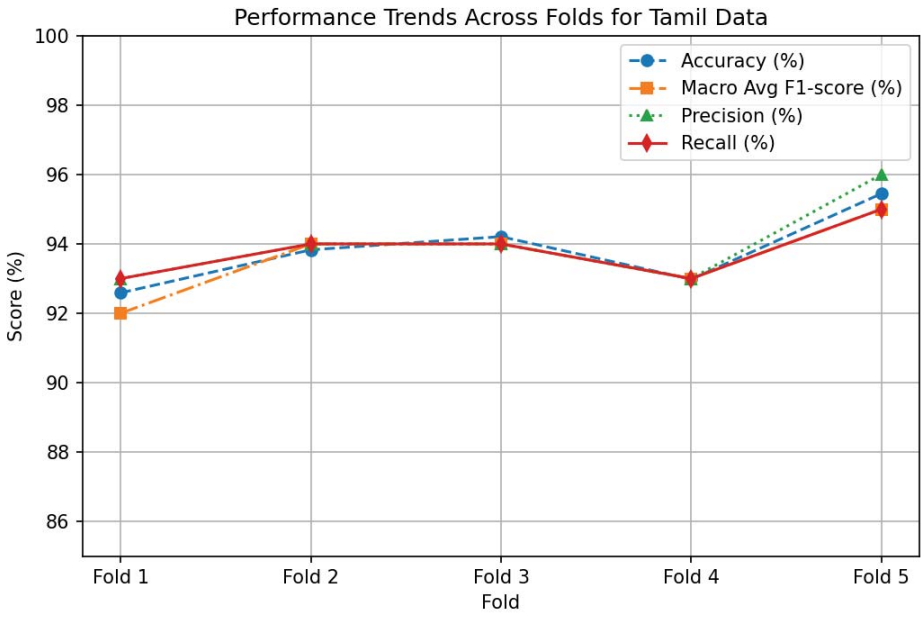

Figure 5.15: Performance trends across folds for Tamil Data

Figure 5.15 illustrates the performance trends of four evaluation metrics Accuracy, Macro Average F1-score, Precision, and Recall across the five folds for the Tamil intent classification dataset. The chart reveals that all metrics remained consistently high, with minor fluctuations between folds. While Fold 4 experienced a slight dip in all metrics, the performance remained above 92.5%, reflecting the model's stability across different data partitions.

The metrics across Folds 1 to 3 follow closely aligned patterns, showing incremental improvements, particularly between Fold 1 (Accuracy: 92.59%) and Fold 3 (94.21%). Fold 4 marks a slight decline (Accuracy: 92.98%), followed by a significant peak in Fold 5, where all metrics reach their highest values—Accuracy (95.45%), Macro F1-score (95%), Precision (96%), and Recall (95%).

These consistent performance trends suggest that the model generalizes well across folds, with no drastic deviations. The slight lag in Fold 4 may reflect the influence of specific speaker patterns or class imbalances, but the subsequent rise in Fold 5 demonstrates the model's ability to recover effectively. Notably, Precision and Recall

maintain a close relationship throughout all folds, indicating balanced predictions with minimal trade-offs.

This trend analysis underscores the model's reliability in low-resource settings, such as Tamil, where data scarcity and variability can often lead to inconsistent model behavior. The convergence of all metrics in Fold 5 also implies improvements in either feature generalization or model adaptation during training. Overall, these trends affirm the importance of robust cross-validation and support the integration of augmentation and transfer learning strategies in low-resource speech classification.

5.12 Analysis of Wav2Vec2 Performance on Tamil and Sinhala Datasets

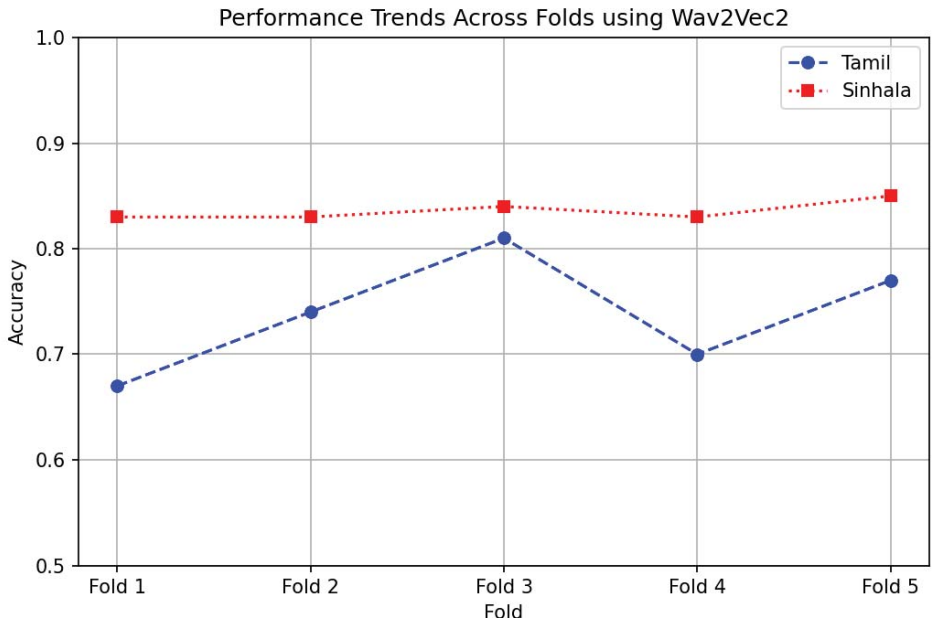

Figure 5.16 : Performance Trends Across Fold using Wav2Vec2

Figure 5.16 illustrates the performance trends of the Wav2Vec2 model across cross-validation folds. For the Tamil dataset, accuracy fluctuates between **67% and 81%**, indicating notable variability. In contrast, the Sinhala dataset maintains consistently high accuracy, remaining above 83% across all folds. This suggests that the Sinhala data is more stable and less sensitive to fold-wise variation, while Tamil exhibits higher variability when using the Wav2Vec2 pre-trained model.

Interestingly, a similar pattern is observed in Figure 5.3, which visualizes results for the MFCC-CNN model, reinforcing that data variability is intrinsic to the Tamil dataset rather than model-specific.

While Wav2Vec2.0 shows potential, particularly due to its strong contextual embeddings—the proposed MFCC-CNN model outperforms it in this task. Especially

for Tamil, the customized MFCC-CNN approach demonstrates better robustness and higher accuracy, confirming its effectiveness in low-resource, intent classification scenarios.

5.13 Comparison with Benchmark methodology performance

Approach	Buddhika et al [10]Methodology	Y. Karunanayake et al.[1] methodology	MFCC:	MFCC+ Delta + Delta-Delta	MFCC +Delta+Delta-Delta+ Augmentation	MFCC +Delta+ Delta-Delta+ Augmentation	Wav2Vec2 Embedding
Feature Type	13 MFCC	DeepSpeech Intermediate Output (character probability)	MFCC Features Only	MFCC +Delta + Delta-Delta Features	MFCC +Delta + Delta-Delta Features	MFCC + Delta+Delta-Delta Features	Contextual
Architecture	FNN	TL + 2D/1D CNN	2D CNN	2D CNN	2D CNN	1D CNN	1D CNN +BiLSTM
Tamil Data Accuracy	--	76.30%	76.08%	83%	**93.81%**	92.99%	73.94%
Sinhala Data Accuracy	74.37%	93.16%	96.59%	**96.92%**	96.69%	96.58%	83.45%

Table 5.6: Summary of results across different approaches with overall accuracy values. Gray shading indicates the accuracy of the previous benchmark methodology.

The table 5.6 illustrates, using 5-fold cross-validation, we achieved 96.92% accuracy for Sinhala and 93.81% for Tamil**,** surpassing previously reported benchmarks. Table 5.4 provides a comparative summary of prior methods and our proposed approach. The results clearly demonstrate that our MFCC + Delta + Delta-Delta + Augmentation + 2D CNN pipeline achieved the highest performance for Tamil, while the MFCC + Delta + Delta-Delta + 2D CNN configuration delivered the best results for Sinhala—outperforming earlier models including those by [1] and [10]

We also evaluated Wav2Vec2 embeddings using two architectures: Wav2Vec2 + 2D CNN and Wav2Vec2 + 1D CNN + BiLSTM. The latter yielded significantly better results, particularly for Tamil, where the accuracy increased to 73.94%. This suggests that Wav2Vec2's contextual embeddings are better suited for sequential models such as BiLSTMs or Transformers, rather than spatial architectures like 2D CNNs. Despite this, our proposed MFCC-CNN pipeline remains the most effective and robust approach for speech intent classification across both languages in this low-resource setting.

5.12 Summary

The study achieved significant accuracy improvements over benchmark models, with Sinhala reaching 96.92% and Tamil reaching 93.81%, compared to previous benchmarks of 93.16% and 76.30%, respectively. The experiments also validated that MFCC + delta +delta-delta features outperform DeepSpeech intermediate features for speech intent classification in low-resource settings. Also MFCC features with CNN based methodology more viable than the other previous and Wav2Vec2 methodology

These results reinforce the importance of feature selection and model architecture in speech classification tasks, providing a strong foundation for further improvements in low-resource speech-to-text systems.

CHAPTER 6

DISCUSSION

This chapter presents a focused discussion on the key experimental findings. It analyzes performance trends across different configurations and languages, highlighting the effectiveness of the proposed Residual CNN architecture and the contribution of MFCC features with delta and delta-delta enhancements. The discussion also covers the impact of data augmentation, compares the performance of 1D vs. 2D CNNs, and evaluates the strengths and limitations of pre trained models like Wav2Vec2. Finally, it outlines practical implications and acknowledges current limitations, providing insights for future improvements.

6.1 Proposed MFCC Feature Result Discussion

The application of MFCC features combined with delta and delta-delta coefficients significantly improved performance, particularly in the context of low-resource and imbalanced datasets such as the Tamil and Sinhala speech corpus.

The inclusion of delta (first-order derivative) and delta-delta (second-order derivative) features captures the temporal dynamics of speech, providing richer contextual information beyond static MFCCs. This temporal modeling enhances the model's ability to recognize subtle variations in pronunciation and speaking rate—factors often critical in speech intent classification.

As a result, the model achieved more robust and balanced performance across all intent classes, reducing class-specific misclassifications and improving overall accuracy. This improvement is especially important in low-resource settings, where data scarcity can lead to poor generalization and biased predictions.

By enriching the feature representation, the MFCC + delta + delta-delta pipeline supports better discriminative learning, making the model more consistent and reliable even with limited data.

6.2 Proposed Data Augmentation Result Discussion

Data augmentation had a significant positive impact on the Tamil intent classification model, increasing mean accuracy from **83.00% to 93.81%**. Techniques such as pitch shifting, time stretching, and noise addition introduced acoustic variability that helped the model generalize better in a low-resource setting. These augmentations simulated real-world speech variations, reducing over fitting and enhancing robustness.

Additionally, the use of SMOTE addressed class imbalance by generating synthetic samples for underrepresented classes. The combination of these methods led to consistent performance gains across all folds, demonstrating the value of augmentation in low-data environments.

In contrast, the Sinhala model**,** which was trained on a larger, cleaner, and more balanced dataset, saw only a marginal improvement (from 96.92% to 96.94%). The dataset's strong baseline quality limited the benefits of augmentation, and in some cases, minor performance drops were observed.

In summary, data augmentation is highly effective for low-resource and imbalanced datasets like Tamil**,** but its impact is minimal or unnecessary for high-quality datasets such as Sinhala. This highlights the need to adapt augmentation strategies based on dataset characteristics.

6.3 Effectiveness of CNNs for Sequential Feature Classification

- **CNNs in Speech Intent Classification:** Convolutional Neural Networks (CNNs) have proven to be highly effective for intent classification tasks, especially when working with sequential data, such as speech features derived from MFCC or spectrograms. CNNs are well-suited for capturing local patterns and dependencies, which are essential for speech-to-text tasks.
- **Performance of 1D vs. 2D CNNs:** For the Tamil dataset, we observe that Conv2D outperforms Conv1D significantly, which highlights the importance of model selection based on the input data characteristics. The Conv2D model captures more complex spatial relationships in the MFCC (2D) feature space, while Conv1D, though computationally efficient, struggles to extract meaningful information from the 1D features of the Tamil speech data. This results in the Conv2D model achieving a much higher accuracy compared to Conv1D.

6.4 Performance Differences between 1D and 2D CNNs

A comparative evaluation was carried out between 1D Convolutional Neural Networks (Conv1D) and 2D Convolutional Neural Networks (Conv2D) to assess their effectiveness in the Sinhala and Tamil speech intent classification tasks.

For the Tamil dataset**,** the Conv1D model achieved a mean accuracy of 92.99%**,** while the Conv2D model outperformed it with 93.81%**.** Similarly, for the Sinhala dataset, Conv1D recorded 96.58%**,** whereas Conv2D achieved a slightly higher mean accuracy of 96.69%**.** Although the differences in performance are relatively small, they consistently favor the Conv2D architecture.

These performance differences can be attributed to how each model processes input features. Conv1D models treat MFCCs as temporal sequences, effectively capturing time-dependent patterns but lacking the ability to interpret inter-frequency relationships. This approach is computationally efficient and suitable for modeling sequential data but may overlook complex acoustic interactions embedded in speech.

On the other hand, Conv2D models view MFCC features as two-dimensional matrices—capturing patterns across both time and frequency axes. This joint time-frequency analysis aligns more naturally with the structure of spectrogram-like representations such as MFCC + Delta + Delta-Delta, making Conv2D particularly

well-suited for capturing nuanced acoustic variations. This is especially valuable in low-resource settings like Tamil, where every feature contributes significantly to classification accuracy.

Overall, the Conv2D model demonstrated superior generalization and robustness, making it a more effective choice for speech intent classification. Despite slightly higher computational demands, the performance improvements justify the use of 2D CNNs in scenarios where time-frequency dependencies are critical.

CHAPTER 7

CONCLUSION

This chapter summarizes the key outcomes of the study, highlighting the methodological contributions, experimental insights, limitations encountered, and potential avenues for future research. The work focused on improving speech intent classification for low-resource languages, particularly Sinhala and Tamil, through MFCC-based acoustic feature engineering, CNN architectures, and comparative analysis with pre trained models.

7.1 Conclusion

This research aimed to enhance intent classification in low-resource languages by utilizing Mel-Frequency Cepstral Coefficients (MFCCs) along with delta and delta-delta features, trained on both 1D and 2D CNN architectures. Key components of the methodology included systematic preprocessing, hyperparameter tuning, feature set comparison, and data augmentation. A 64-dimensional MFCC representation, enriched with delta and delta-delta derivatives, was employed to better capture both temporal and spectral characteristics. Data augmentation techniques such as noise addition, pitch shifting, and time-stretching were applied, particularly benefiting the smaller Tamil dataset.

The refined residual CNN models demonstrated significant improvements over existing benchmarks. For Sinhala, the model achieved an accuracy of 96.92%, surpassing the previous best of 93.16%. For Tamil, accuracy increased substantially from 76.30% to 93.81%, confirming the effectiveness of switching from DeepSpeech's character-level features to MFCC-based representations.

A comparative analysis between 1D and 2D CNN architectures revealed that 2D CNNs consistently outperformed their 1D counterparts, especially for the Tamil dataset. 2D CNNs proved more effective at capturing both temporal and spectral correlations within MFCC matrices, making them better suited for small, low-resource datasets. Augmented models exhibited higher F1-scores, indicating improved generalization and robustness to unseen data. The study also compared performance with Wav2Vec2.0 and found that while contextual embeddings were promising, handcrafted MFCC-CNN pipelines produced superior results for Tamil.

The findings demonstrate that carefully applied data augmentation significantly boosts performance and generalization. The integration of data augmentation, feature engineering, and CNN-based modeling aligns with state-of-the-art practices in low-resource intent classification. Additionally, the use of Swish-activated residual blocks further improved accuracy and model stability.

7.2 Contributions

This study contributes to the field of low-resource speech AI in multiple ways. First, it established that MFCCs combined with delta and delta-delta features significantly outperformed DeepSpeech-based intermediate features for intent classification, particularly in Tamil. The proposed model achieved 96.92% accuracy for Sinhala and 93.81% for Tamil, surpassing previous benchmarks. Comparative feature analysis also confirmed that MFCC + delta + delta-delta configurations performed better than MFCCs alone.

Second, the research showed the value of data augmentation, especially for imbalanced or limited datasets. Augmentation techniques notably boosted Tamil accuracy but had minimal impact or even a slight decline in Sinhala due to its already high quality and balance. This underlines the importance of tailoring augmentation strategies to dataset characteristics.

Third, the research validated the superiority of 2D CNNs over 1D CNNs for small-scale speech tasks. While 1D CNNs are computationally lighter, 2D CNNs better capture the joint temporal-frequency patterns present in MFCC inputs. Additionally, the study evaluated Wav2Vec2.0 in combination with 1D CNN and BiLSTM, and found that this hybrid performed better than Wav2Vec2 + 2D CNN, confirming the suitability of contextual embeddings for sequential models like BiLSTM.

Lastly, the study compared MFCC-CNN with pre trained models like Wav2Vec2. Although Wav2Vec2 performed reasonably well, it did not outperform the tailored MFCC-based CNN especially for Tamil. These findings highlight the potential of handcrafted, domain-specific architectures in low-resource settings where general-purpose pre trained models may fall short.

7.3 Limitations

While the proposed models achieved strong results, some limitations were observed. Misclassifications occurred among phonetically similar intent classes, highlighting the need for more discriminative, phoneme-aware features. Additionally, all evaluations were conducted under clean acoustic conditions, limiting insights into real-world performance. Future studies should incorporate noisy and diverse environments to better assess generalizability.

The use of pre trained models such as Wav2Vec2.0 trained primarily on high-resource languages—may have limited adaptability to the linguistic nuances of Tamil and Sinhala. Utilizing region-specific or low-resource-focused pre trained models could improve performance. Moreover, training these large models was computationally demanding, posing barriers to experimentation and reproducibility for researchers with limited hardware resources.

7.4 Future Work

This research opens several promising directions for future exploration. First, expanding the study to other low-resource and tonal languages will help validate the generalizability of the proposed approach. Collaborating with native linguistic communities can further facilitate the creation of annotated corpora tailored to underrepresented languages.

Alternative feature extraction techniques such as Linear Predictive Coding (LPC), Linear Predictive Cepstral Coefficients (LPCC), Perceptual Linear Prediction (PLP), and RASTA filtering could be investigated to determine whether they provide superior representations for intent classification.

Incorporating more advanced pre-trained models—such as OpenAI's Whisper, which is designed for multilingual and code-mixed speech—could enhance performance in real-world, noisy environments. Future research should also prioritize evaluation under noisy and uncontrolled acoustic conditions, potentially integrating denoising techniques to improve robustness.

To enable deployment on resource-constrained platforms, optimizing lightweight models for edge devices (e.g., mobile phones or IoT systems) is essential. Advancing self-supervised or semi-supervised learning techniques tailored to low-resource settings also presents an exciting avenue, particularly for reducing the dependence on labeled data.

Lastly, exploring phoneme-level and language-specific embeddings may further improve the model's ability to capture linguistic nuances, contributing to more accurate and culturally aware intent classification.

7.5.1 Summary

This study demonstrates that combining MFCC features—including delta and delta-delta derivatives—with CNN-based architectures significantly improves intent classification performance in low-resource languages. By integrating data augmentation, class balancing, and Swish-activated residual CNNs, the proposed models achieved 96.92% accuracy for Sinhala and 93.81% for Tamil. These results outperformed previous benchmarks, including [10](74.37% for Sinhala) and [1](93.16% for Sinhala, 76.30% for Tamil).

A key contributor to this improvement was the transition from DeepSpeech-derived features to handcrafted MFCC-based representations, which proved more effective for capturing relevant acoustic patterns. Additional performance gains were realized through model tuning, augmentation strategies, and architectural comparison—where 2D CNNs consistently outperformed 1D CNNs, particularly for smaller datasets like Tamil.

The use of residual connections and Swish activation functions played a critical role in enhancing model depth, training stability, and generalization. Residual blocks

helped mitigate vanishing gradient issues, enabling deeper networks with more efficient convergence. Swish activation, due to its smooth and non-monotonic behavior, provided more expressive non-linear mappings than traditional ReLU, further boosting performance.

While limitations remain such as limited dataset sizes and lack of real-world noise scenarios the results provide a strong foundation for advancing speech-based intent classification in underrepresented languages. Future work can extend this approach to more languages, integrate noise-robust techniques, and optimize models for real-world deployment, contributing toward more inclusive and accessible speech AI systems.

REFERENCES

[1] Y. Karunanayake, U. Thayasivam, and S. Ranathunga, "Transfer Learning Based Free-Form Speech Command Classification.," in *Proc. 57th Annu. Meet. Assoc. Comput. Linguistics: Student Res. Workshop*, 2019.

[2] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, "Automatic Speech Recognition for Under-Resourced Languages: A Survey," *Speech Communication,* vol. 56, pp. 85-100, 2014.

[3] W. Meng and N. Yolwas, "A Review of Speech Recognition in Low-resource Languages," in *3rd International Conference on Pattern Recognition and Machine Learning (PRML)*, 2022.

[4] Y. Karunanayake, U. Thayasivam, S. Ranathunga, "Speaker-Invariant Speech-to-Intent Classification for Low-Resource Languages," in *Speech and Computer*, R. P. A. Karpov, Ed., Cham, Springer International Publishing, 2021, p. 247–257.

[5] M. Elamin, M. Omer, Y. Chanie, and H. Ndlovu, "Creating Spoken Dialog Systems in Ultra-Low Resourced Settings," 2023.

[6] I. Mohamed and U. Thayasivam, "Low Resource Multi-ASR Speech Command Recognition," in *2022 Moratuwa Engineering Research Conference (MERCon)*, 2022.

[7] N. F. Chen, et al., "Low-resource Keyword Search Strategies for Tamil," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[8] R. Zhou, T. Koshikawa, A. Ito, T. Nose, and C.-P. Chen, "Multilingual Meta-Transfer Learning for Low-Resource Speech Recognition," *IEEE Access,* vol. 12, p. 158493–158504, 2024.

[9] S. Bhosale, I. Sheikh, S. H. Dumpala, and S. K. Kopparapu, "Transfer Learning for Low Resource Spoken Language Understanding without Speech-to-Text," in *2019 IEEE Bombay Section Signature Conference (IBSSC)*, 2019.

[10] D. Buddhika, R. Liyadipita, S. Nadeeshan, H. Witharana, S. Javasena, and U. Thayasivam, "Domain Specific Intent Classification of Sinhala Speech Data," in *International Conference on Asian Language Processing (IALP)*, 2018.

[11] Yaman, S., Deng, L., Yu, D., Wang, Y.-Y., & Acero, A., "An Integrative and Discriminative Technique for Spoken Utterance Classification," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 16, no. 6, p. 1207–1214, 2008.

[12] Rao, J., Ture, F., & Lin, J., "Multi-task Learning with Neural Networks for Voice Query Understanding on an Entertainment Platform," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*, 2018.

[13] He, X., & Deng, L., "Speech-centric information processing: An optimization-oriented approach," *Proceedings of the IEEE,* vol. 101, no. 5, p. 1116–1135, 2013.

[14] S. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 28, no. 4, p. 357–366 , 1980.

[15] Zhang, S., Geiger, A., & Schlangen, D., "SpeechIntent: A Benchmark for Spoken Language Understanding in Low-Resource Settings," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[16] Liu, C., Trmal, J., Wiesner, M., Harman, C., & Khudanpur, S., " Topic Identification for Speech Without ASR," in *Proceedings of Interspeech 2017*, 2017.

[17] Lee, L.-S., Glass, J., Lee, H.-Y., & Chan, C., "Spoken content retrieval beyond cascading speech recognition with text retrieval," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 23, no. 9, p. 1389–1420.

[18] X. Zhang, X. Li, and H. Wang, "CNN-Based Classification Models Outperform Traditional ASR-NLU Pipelines in Noisy Environments," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 29, p. 1234–1245, 2021.

[19] Karunanayake, Y., Thayasivam, U., & Ranathunga, S., "Sinhala and Tamil Speech Intent Identification from English Phoneme Based ASR," 2021.

[20] A. Hannun, et al., "Deep Speech: Scaling up End-to-End Speech Recognition," *arXiv preprint,* 2014.

[21] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "Wav2Vec 2.0: Self-Supervised Learning for Speech Recognition," 2020.

[22] Gupta, P., Kumar, R., Patel, R., & Singh, D., "Intent classification using Wav2Vec 2.0 for low-resource languages," in *13th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2022.

[23] Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., & Mohamed, A., "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units," in *Proceedings of ICASSP*, 2021.

[24] Babu, A., Wang, C., Tjandra, A., et al., "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale," in *Proceedings of Interspeech*, 2022.

[25] A. Radford et al., "Robust Speech Recognition via Large-Scale Weak Supervision," OpenAI Technical Report, 2023.

[26] Baevski, A., Zhou, Y., Mohamed, A., & Auli, M., "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[27] Park, D. S.; Chan, W.; Zhang, Y.; Chiu, C. C.; Zoph, B.; Cubuk, E. D.; Le, Q. V., "A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proceedings of Interspeech*, Graz, Austria, 2019.

[28] Gopalakrishnan, C., Mandal, A., & Sengupta, S., "Improving Spoken Language Understanding for Indian Languages using Wav2Vec 2.0," in *IEEE Spoken Language Technology Workshop (SLT)*, 2021.

[29] Kannan, N., Arul, S., & Singh, R., "Leveraging Whisper for Robust Intent Classification in Code-Mixed and Noisy Environments," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[30] Conneau, A., Baevski, A., Collobert, R., Mohamed, A., & Auli, M., "Unsupervised Cross-lingual Representation Learning for Speech Recognition," in *Proceedings of Interspeech*, 2021.

[31] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detector," arXiv preprint, 2012.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012.

[33] G. Gunasekara et al., "Empirical Evaluation of CNN-based Intent Classification in Low-Resource Languages," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[34] Lugosch, L.; Ravanelli, M.; Serdyuk, D.; Ebrahimi Kahou, S.; Bengio, Y., "Speech Model Pre-training for End-to-End Spoken Language Understanding," in *Interspeech*, Graz, Austria, 2019.

[35] K. Gupta and D. Gupta, "An Analysis on LPC, RASTA and MFCC Techniques in Automatic Speech Recognition System," in *6th International Conference on Cloud System and Big Data Engineering (Confluence)*, 2016.

[36] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Low-Resource Speech-to-Text Translation," in *Proceedings of Interspeech 2018*, 2018.

[37] Ko, T., Peddinti, V., Povey, D., & Khudanpur, S., "Audio augmentation for speech recognition," in *Proceedings of Interspeech*, 2015.

[38] Cai, Q., Wang, D., Zhang, X., & Xie, L., "Data augmentation for deep speech recognition with noise perturbation and synthetic data," in *Proceedings of IEEE ICASSP*, 2020.

[39] Butt, S. A., Iqbal, U., Ghazali, R., Shoukat, I. A., Lasisi, A., & Al-Saedi, A. K. Z., "An Improved Convolutional Neural Network for Speech Emotion Recognition," in *Recent Advances in Soft Computing and Data Mining (SCDM 2022)*, vol. 457, R. M. N. N. D. M. M. A. J. H. &. A. N. Ghazali, Ed., Springer, Cham, 2022.

[40] Wang, Y., Deng, X., Pu, S., & Huang, Z., "Residual Convolutional CTC Networks for Automatic Speech Recognition," 2017.

[41] Zhou, X.; Li, J.; Zhou, X., "Cascaded CNN-resBiLSTM-CTC: An End-to-End Acoustic Model for Speech Recognition," in *arXiv preprint (CoRR abs/1810.12001), authors affiliated with Cloudwalk Technology*, 2018.

[42] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for Activation Functions," *arXiv preprint*, 2017.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* , 2016.

[44] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *International Conference on Machine Learning (ICML)* , 2015 .

[45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research,* vol. 15, p. 1929–1958, 2014.

[46] Juan C. Olamendy, "A Comprehensive Guide to Stratified K-Fold Cross-Validation for Unbalanced Data," 2023. [Online]. Available: https://medium.com/@juanc.olamendy/a-comprehensive-guide-to-stratified-k-fold-cross-validation-for-unbalanced-data-014691060f17.

[47] J. Bergstra, D. Yamins, and D. Cox, "A Python Library for Optimizing the Hyperparameters of Machine Learning Algorithms," in *Proc. 12th Python in Science Conf.*, 2013.

[48] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering,* vol. 22, no. 10, p. 1345–1359, 2010.

[49] Ram et al., "Conversational AI: The Science Behind the Alexa Prize," 2018. [Online]. Available: https://arxiv.org/abs/1801.03604.

[50] J. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-Language Knowledge Transfer Using Multilingual Deep Neural Network with Shared Hidden Layers," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[51] T. Ko, V. Peddinti, D. Povey, S. Khudanpur, "Data Augmentation for Speech Recognition," in *Proceedings of INTERSPEECH 2017*, Stockholm, 2017.

[52] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Data Augmentation for Speech Recognition," in *Proceedings of INTERSPEECH 2017*, Stockholm, 2017.

[53] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature,* vol. 521, p. 436–444, 2015.

[54] D. Amodei et al., "Deep Speech 2: End-to-End Speech Recognition in English and Mandarin," in *International Conference on Machine Learning (ICML)*, 2016.

[55] A. Hannun, et al., "Deep Speech: Scaling up End-to-End Speech Recognition," 2014.

[56] Z. Xu et al., "Empirical Evaluation of Wav2Vec2.0 for Intent Classification in Low-Resource Languages," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[57] I. Goodfellow et al., "Explaining and Harnessing Adversarial Examples," in *International Conference on Learning Representations (ICLR)*, 2015.

[58] R. Kumar et al., "Leveraging Wav2Vec2.0 for Intent Classification with Minimal Labeled Data," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 456–465, p. 31, 2023.

[59] J. Xu, et al., "LRSpeech: Extremely Low-Resource Speech Synthesis and Recognition," 2020.

[60] A. Tennakoon, N. Fernando, N. Nawarathna, and A. D. C. Tissera, "Transfer Learning Based Free-Form Speech Command Classification for Low-Resource Languages," in *1st International Conference on Advanced Research in Computing (ICARC)*, 2021.

[61] T. Ko, W. Hsu, and M. Hwang, "Voice Conversion with Deep Neural Networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 25, no. 1, p. 42–53, 2017.