9-23-2022

A MACHINE LEARNING FRAMEWORK FOR AUTOMATIC SPEECH RECOGNITION IN AIR TRAFFIC CONTROL USING WORD LEVEL BINARY CLASSIFICATION AND TRANSCRIPTION

Fowad Shahid Sohail
Rowan University

A MACHINE LEARNING FRAMEWORK FOR AUTOMATIC SPEECH RECOGNITION IN AIR TRAFFIC CONTROL USING WORD LEVEL BINARY CLASSIFICATION AND TRANSCRIPTION

by

Fowad Shahid Sohail

A Thesis

Submitted to the
Electrical and Computer Engineering Department
College of Engineering
In partial fulfillment of the requirement
For the degree of
Master of Science in Electrical and Computer Engineering
at
Rowan University
August 26, 2022

Thesis Chair: Ravi P. Ramachandran, Ph.D., Professor, Department of Electrical & Computer Engineering

Committee Members:
Parth Bhavsar, Ph.D., Assistant Professor, Department of Civil Engineering, Kennesaw State University
Yusuf Mehta, Ph.D., P.E., Professor, Department of Civil Engineering, Rowan University
John Schmalzel, Ph.D., P.E., Professor, Department of Electrical & Computer Engineering, Rowan University

Acknowledgements

I'd like to thank Dr. Ravi Ramachandran and Dr. Parth Bhavsar for working with me throughout my time at Rowan and for all these months with this research. I would also like to thank Dr. Yusuf Mehta for allowing me to work with CREATES and Dr. John Schmalzel for serving on my Defense Committee. In, addition I'd like to thank my close friends and family for supporting me on my journey through graduate school and broader life.

Abstract

Fowad Shahid Sohail
A MACHINE LEARNING FRAMEWORK FOR AUTOMATIC SPEECH RECOGNITION IN AIR TRAFFIC CONTROL USING WORD LEVEL BINARY CLASSIFICATION AND TRANSCRIPTION
2021-2022
Ravi P. Ramachandran, Ph.D.
Master of Science in Electrical and Computer Engineering

Advances in Artificial Intelligence and Machine learning have enabled a variety of new technologies. One such technology is Automatic Speech Recognition (ASR), where a machine is given audio and transcribes the words that were spoken. ASR can be applied in a variety of domains to improve general usability and safety. One such domain is Air Traffic Control (ATC). ASR in ATC promises to improve safety in a mission critical environment. ASR models have historically required a large amount of clean training data. ATC environments are noisy and acquiring labeled data is a difficult, expertise dependent task. This thesis attempts to solve these problems by presenting a machine learning framework which uses word-by-word audio samples to transcribe ATC speech. Instead of transcribing an entire speech sample, this framework transcribes every word individually. Then, overall transcription is pieced together based on the word sequence. Each stage of the framework is trained and tested independently of one another, and the overall performance is gauged. The overall framework was gauged to be a feasible approach to ASR in ATC.

Table of Contents

Table of Contents (Continued)

Table of Contents (Continued)

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement

In the earliest stages of technology and computers, signals were entirely binary. Binary eventually evolved into instruction sets which allowed for more complex operations. As the architecture and hierarchy grew further and further, there were newer ways to interact with technology which improved the user experience and allowed computers to accomplish greater tasks. For decades, text-based input has been a staple in how humans interact with the computers around them. For humans, however, text is not the most natural form of communication. The past decade has seen advancements In speech recognition which has now allowed for speech-based input to computers – a more natural way for humans to interact with technology in a broad range of fields and applications.

A mission critical application of speech recognition is In air traffic control (ATC). Traditionally, ATC is done through radio communication between air traffic controller (ATCO) and pilot. This is a process that is highly subject to human errors, miscommunication and imposes a heavy workload on the ATCO. As a result, lives are put in danger because of a one-dimensional communication protocol and process. By adding the dimension of transcription, the spoken communication can be automatically written down by machines. Now, instead of just one layer of communication, pilot and ATCO have another way to understand directions. As a result, air traffic safety is improved, and lives can be saved.

1.2 Motivation

This research developed a unique, word-by-word framework for Automatic Speech Recognition (ASR) applications in the ATC domain. ASR has enabled many new technologies found in day-to-day life. In the ATC space, it promises to improve safety by presenting another modality of information for each the pilot and ATCO to communicate with. Speech is the most natural way for humans to communicate. However, as a failsafe and safety mechanism, transcribing conversations allows for a greater degree of confidence and clarity in ATC communications.

The framework takes advantage of word level binary classification and transcription. Instead of attempting to transcribe an entire audio file, each word in the file is transcribed and the entire transcription is pieced together. Using Word Error Rate (WER) calculations, the framework is compared to a state-of-the-art generalized model called Wav2Vec 2.0. The domain uniqueness of ATC presents many challenges for modern ASR systems. As such, this framework was developed to determine the feasibility of a domain specific, distributed machine learning framework for ASR in ATC.

1.3 Objectives and Hypothesis

The primary objective of this thesis is to evaluate the feasibility of a distributed machine learning framework for automatic speech recognition, specifically for the transcription of conversations between air traffic controllers and aircraft pilots. The following key tasks were executed to achieve this objective:

1. *To determine the best speech enhancement method for ATC audio data.*
2. *To develop a word-by-word ATC audio corpus.*

3. *To train a Binary Classifier to distinguish between ATC and ENGLISH words.*
4. *To train ASR models to specifically transcribe ATC and ENGLISH words.*
5. *To transcribe ATC audio on word level basis.*
6. *To analyze performance of the developed framework in comparison to the current state of the art ASR models.*

The hypothesis for this research is below:

Developing a word-by-word audio corpus and distributed machine learning framework for ASR will yield statistically significant performance gains over generalized ASR models.

1.4 Contributions

The following is a list of major contributions of this work:

1. *A unique, word-by-word audio corpus purpose built for ASR in ATC*
2. *A determined optimal speech enhancement method for the audio corpus*
3. *Word level Binary Classification of ATC speech*
4. *Word level transcription of ATC speech*
5. *A wholistic framework for word-by-word ASR in ATC*

1.5 Focus and Organization

The focus of this thesis is to investigate and implement automatic speech recognition techniques in air traffic control. There are a variety of unique constraints and problems for the ATC space which complicate this task. Standard approaches to ASR perform poorly, as the conditions of ATC speech are vastly different than the speech corpora that

modern ASR models are trained on. Because of this, model architectures must vary with these changing conditions. The thesis is organized as follows:

Chapter 1 is an introduction which outlines the motivations and objectives of this work to the field of ASR in ATC.

Chapter 2 is a literature review which highlights the current state of affairs for ASR in ATC. Contained within this chapter is a high-level overview of artificial intelligence, air traffic control, speech transcription and how they all intersect for ASR in ATC.

Chapter 3 outlines the methodology and techniques used to implement the ASR on ATC data. This includes constructing experiments, training and testing, speech enhancement and more.

Chapter 4 presents the results of all the experimentation conducted in this research. It includes tables, figures and diagrams to substantiate the results and claims made in this thesis.

Chapter 5 finally concludes this thesis with an analysis of the results and recommendations for future work.

Chapter 2

Background

Artificial intelligence (AI) is a field of engineering and computer science that focuses on improving performance of machines, specifically in performing tasks that typically require human intelligence [1]. AI has become increasingly important in the past few decades as advancements in technology have allowed researchers to tackle more complex problems by utilizing an increase of computing power [2]. There are several tasks that AI can accomplish. AI applications include but are not limited to: search engines [3], targeted advertisements [4], self-driving cars [5] and speech recognition systems [6]. AI has made its way into our daily lives through technology such as Google [3], Netflix [7], Amazon [8] and smart assistants [9].

Speech transcription is a versatile application of artificial intelligence which has fundamentally changed the way humans interact with technology. At the core, speech transcription is a new way for humans to give directions to computers. This opens a whole new realm of possibilities and use cases. Some of the places where speech transcription has been applied include assistive technology [10], voice assistants [11] and air traffic control [12]. Speech transcription has also played a significant role in improving the quality of life for those living with disabilities by allowing them to use technology with minimal effort, by leveraging their voice [13]. Machine learning is a subset of artificial intelligence and includes the process by which computers take in data to learn how to perform a specific task. Advancement in machine learning technologies has reduced the need for manual transcription and led to the development of automatic speech recognition (ASR) [14]. These advanced algorithms are able to overcome

limitations found during previous generations and have paved the way for applications of ASR in diverse domains, including Air Traffic Control.

This chapter focuses on background and foundational information regarding automatic speech recognition in air traffic control, including some history and challenges in the current state of affairs.

2.1 Air Traffic Control

Air Traffic Control (ATC) comprises communication between Air Traffic Controllers (ATCOs) and pilots. The main purpose of ATC is to prevent collisions between aircrafts, thus making it a mission critical system. The unique conditions presented in aviation means that traffic must be managed by a third-party entity that is on the ground instead of the aircraft. This third-party entity is the ATCO, who is tasked to ensure all flights arrive safely at their destination. The ATCOs and pilots go through specialized training and follow specific communication standards to ensure safe operations [15]. It has been discovered many times over the years that many pilot errors are caused from human error such as distraction or fatigue when performing tasks [16]. This is why ATCOs are trained extensively so they do not make any mistakes which could endanger an entire flight. There are several different types of ATC systems used today but one common feature among them is voice communications.

Because the primary avenue of ATC communication is through spoken language, a number of factors have to be considered to ensure the safety and success. ATC voice communication is done primarily through radio communications, which opens up the door for a variety of environmental factors to impact the quality of audio that is received by either the ATCO or the pilot. Radio channels can be affected via weather interference,

static and dropouts which impacts both parties communicating with each other. Additionally, the radio waves themselves propagate differently due to atmospheric changes, which impacts quality of the transmission [17]. Because of constraints such as bandwidth, sampling rates and number of channels, audio quality obtained using voice radios can be degraded [18]. These issues present problems when using radios for remote operations where distance increases between two points [19]. In addition to communication equipment reliability being lower than computerized modes of operation, it is extremely hard to maintain good audio clarity under these circumstances.

Noise is always present in various environments, and as a result, speech signals cannot be recorded in their purest form [20]. In these cases, we often hear more than what was intended because of a combination of echo effect between speaker and microphones themselves, and interference signals. Since it's difficult to separate sound coming out from speakers and noise coming from various sources, speech enhancement processing can play a significant role in providing better quality during certain critical tasks such as landing procedures. Due to this reason, ATC language requires greater accuracy especially under noisy conditions. During high-speed environment, even a slight increase in transmission time would delay the messages considerably, therefore requiring faster data exchange methods. When comparing different communication formats, the most obvious choice would be text-based messaging/chat that allows less latency due to its constant connection and easy access. However, text-based methods only offer basic features whereas voice provides richer content including more details such as the urgency of information, background noises or weather issues affecting airplane operation. Voice also plays a big part towards effective collaboration amongst ATC personnel, since it

allows interaction between both parties in real-time and allows a much quicker response if needed. In critical situations, such as a landing procedure, a real-time Automatic Speech Recognition (ASR) system that converts voice into text can significantly enhance the quality of communication and save time, resources, and lives.

2.2 Automatic Speech Recognition (ASR)

Automatic speech recognition (ASR) is the process by which machines take audio and transcribe them into text. There are many applications for ASR, especially in the consumer space. Most notably, ASR applications have manifested themselves into our everyday lives through voice assistants such as Siri or Cortana [11]. Text is easier for computers to work with compared to speech, where speech is easier for humans to produce than text into a computer. As such, ASR has enabled a new, more convenient way for humans to interact with technology. A detailed systematic review of recent ASR methods is provided by Alharbi et al. [21]. While there are many applications of ASR, this study focuses on ASR for Air Traffic Control and specifically enhancing communication between an Air Traffic Controller and an aircraft pilot.

An important application for ASR is in the ATC space, where text transcriptions of ATCO speech can improve safety in a mission critical system. An example scenario could be while pilots land at airports who may receive instructions over their headsets on how to approach a runway; they may then follow those instructions by reading back the transcription of the instructions given to them by ATCOs or automated systems on board their airplanes. Having a machine do the interpretation of speech reduces the risk associated with human error in interpreting speech. It should be noted that while it is

unlikely for automation to replace every aspect of ATC, it can complement existing processes and protocols to make them safer, more efficient, and more cost effective.

2.3 ASR in the ATC Domain

Speech communication between ATCO and pilots is highly subject to risks, such as human error. These risks are likely to compromise the safety of aircraft. Thus, there has been significant research into presenting computerized monitoring systems that can improve the safety and reduce the risks of ATC systems [22]. To accomplish this, automatic speech recognition (ASR) has been explored as an interface between humans and machines in the ATC systems even earlier than 2015 [23]. ASR has received a significant amount of attention in the literature worldwide. In the ATC domain, there have been a variety of approaches for the application of ASR. These approaches tackle many subtasks and challenges, including but not limited to: multilingual ASR [24], real-time safety monitoring [22] and end to end models [25], [26]. As a whole, there are a number of similar challenges that are shared for ASR in ATC.

2.4 Challenges in ASR for ATC

ASR in and of itself is a well-researched area with many successful applications not only in academia but also in industry [27]. ASR has made its way into everyday consumer products, perhaps most commonly in voice assistants such as Siri and Cortana. However, despite many advantages that come with ASR techniques, challenges still remain regarding their practical implementation within the ATC context. ASR, that is training a machine to translate audio to text, is traditionally a supervised learning task. As such, training a high performing ASR model will require a high-quality dataset. A high-quality speech corpus for ASR includes characteristics such as dataset size, diverse

vocabulary, varying speakers among others. Given low quality data, an ASR model will yield low quality performance. To that end, there have been many different datasets developed for general ASR research, including LibriSpeech [28] and TED-LIUM3 [29]. These datasets, among others, include hundreds of hours of audio and from domains that have a vast amount of data available. These types of datasets have allowed for many advancements in ASR research. However, the unique conditions and constraints of the ATC domain present unique challenges for ASR that are not present elsewhere.

2.4.1 Scarcity of Data

Unlike LibriSpeech [28], which gathers data from audiobooks, and TED-LIUM3 [29], which gathers data from TED Talks, recording and labeling ATC is significantly more difficult. ATCO and pilot training is highly specialized and requires a special skillset to properly label data. Thus, creating ATC datasets for ASR tasks is a costly and time consuming initiative. Therefore, there are smaller amounts of transcribed recordings compared to other applications of ASR. Current ASR models heavily rely on large annotated corpora of clean data, resulting in potentially poor generalization for the ATC space.

There has been work done to combat the lack of ATC specific data. Lin et. al present a new training approach for this specific problem [30]. Using an unsupervised pretraining strategy, they train the model to learn the distribution of features and then utilize transfer learning to accomplish the ASR task. In another work, the small dataset problem is tackled using speech representation learning in a self-supervised, wave to feature paradigm [25]. A multilingual speech corpus called ATCSpeech has been developed from real ATC systems [24].

2.4.2 Noise

Since ATC communication is accomplished via radio communications on high frequency bands, there is a propensity to a large amount of noise and, consequently, low intelligibility. In addition, the environments of both the pilot and the ATCO will introduce more noise in the communications. The pilot is in an environment where the humming noise of the cabin will effect his speech. The ATCO is typically in an office setting with many other people, which will inevitably introduce noise. All of these factors result in more noise in ATC speech. From a signal processing perspective, the signal to noise ratio (SNR) for ATC speech is lower than speech from clean environments. From an ASR and machine learning perspective, low SNR means that the features in ATC speech are very different from that of common speech. As a result, training ASR models for ATC using common speech corpora will result in poor performance and generalization.

To properly learn features from ATC speech, rather than learning noise, there have been a few different approaches in the literature. One common approach involves using multi-scale convolutional neural networks (MCNN) [31]. As a feature engineering method, many ASR systems use the Mel-frequency cepstral coefficient (MFCC) [32], which relies on Fourier transforms and deals with signals in the frequency domain. The tradeoff in using MFCCs is that temporal feature information is not used. MCNNs solve this issue by analyzing speech in multiple scales [33]. The philosophy of MCNN based approaches is that the noise distribution is generally dispersed throughout the frequency and time domains and the multi-scale nature will discard the overlap with speech. This

will result in more clear features extracted from the noisy data, which will ultimately make for a more robust model.

2.4.3 Multilingual ASR

The International Civil Aviation Organization (ICAO) has set requirements that English should be the universal language of ATC communication [34]. However, practically, especially in domestic flights, pilots are communicating with ATCOs in their common local languages [35]. In a single sentence, there may be one or more languages spoken. As such, any practical ASR solution in the ATC domain must be equipped to deal with multilingual communications.

This requirement has yielded research solutions that include a dedicated multilingual ATC speech corpus, which has proven to generalize well when trained on state-of-the-art ASR models [24]. There have been both end to end [25] and phoneme based [31] approaches that tackle the multilingual ASR in ATC problem. End to end models convert the raw waveform into text directly where phoneme and language-based models have an intermediate vocabulary which is then translated into the text label.

2.4.4 Differing Speech Rates

Speech rate is defined simply as the rate at which words are spoken. Typically, it is calculated in words spoken per minute (wpm) or words spoken per second (wps). Typically, every day spoken speech is between 120 and 150 wpm. However, the speech rate in ATC is higher than this [36]. In addition, ATC speech rates vary based on conditions. The ATC speech datasets gathered in [31] are compared to the speech rates for standard English and Chinese speech corpora. In this comparison, the ATC speech datasets have higher average speech rates and higher standard deviations. This is

indicative of the fact that ATCOs typically speak quicker than everyday speech but will also vary the speed at which they speak, presumably according to the conditions. Intuitively, there are many reasons for this. ATCOs operate in a very active environment where they must monitor and communicate with many flights and pilots at one time. Because of this, the ATCOs can be influenced by their working conditions and, as a result, have higher speech rates in order transmit information quickly. During less busy times, the ATCOs are likely to speak slower, as they are under less stress.

Because of the higher and varying speech rate, feature extraction becomes a difficult and application specific task. To solve this, researchers have taken approaches which vary the model architectures according to the constraints. A feature encoder, multi layered convolutional neural network (MCNN) approach is highlighted in [25]. The different CNN layers cope with the peculiarities of ATC speech, including the unstable speech rate. In [30], Lin et al. introduce another approach to MCNNs in which they vary the kernel sizes according to the speech rate, where smaller kernels are used for higher speech rates.

2.4.5 Imbalanced Features and Code Switching

The International Civil Aviation Organization (ICAO) published a set of guidelines and procedures for communication and pronunciation [37]. By and large, ATC speech adheres to these guidelines. Included in these guidelines is code switching. Code switching is replacing certain words with others to eliminate miscommunication. The ICAO guidelines also include different pronunciations of words used in daily life [31]. An example of this switch is that “nine” becomes “niner”. Common speech corpora do

not include this type of vocabulary. As a result, developing an ASR system for ATC becomes increasingly challenging due to code switching and differing pronunciations.

Even though the ICAO has published guidelines for ATC communication, out of vocabulary words still occur in practice. ATCOs and pilots do not strictly adhere to the rules and terminology. In addition, when the standardized rules are not followed, the speech features and vocabulary are subject to colloquialisms specific to the region of flight or cultures of the ATCOs. This leads to a feature imbalance where the vocabulary is not normally or equally distributed. In fact, there are cases where 40% of the words appear less than ten times and other words are present millions of times [31].

2.5 Summary

This chapter provided a brief introduction about the challenges associated with speech transcription, specifically transcribing conversations between air traffic controllers and pilots. As presented in the chapter, the primary challenges are scarcity of data, noise, multilingual transcription, differing speech rates and code switching. The niche of the ATC realm means that dataset creation requires domain expertise, making it difficult to create large audio corpora. The conditions of recording environments mean that the resulting audio files are very noisy, thus making it difficult to train a machine learning model on the data. A variety of feature imbalances also pose challenges for ASR in ATC, as ATC communication can be done in many languages, can be spoken at varying speeds and be subject to code switching. Considering the challenges presented in this chapter and the mission critical nature of ATC communications, there is a pressing need to investigate speech transcription processes and develop application-specific

solutions that remain consistently accurate. The next chapter describes the methods adopted for this research to accomplish this task.

Chapter 3

Methodology

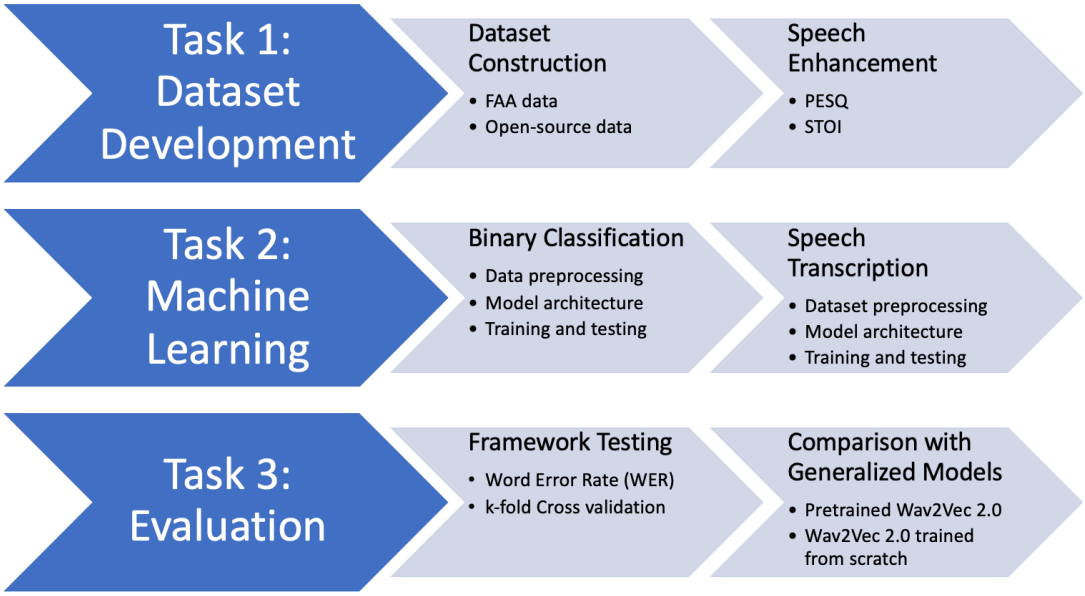
This chapter introduces the research approach and provides an explanation of the proposed architecture. First, the research methodology is introduced in a sequence of three tasks: (1) Dataset Development, (2) Machine Learning, and (3) Evaluation. An explanation of each task and its subtasks is given in the following sections. This chapter sets the stage for Chapters 4 and 5 which present the results of the conducted experimentation and the conclusions drawn therein.

3.1 Research Tasks

The research methodology employed for this work is explained in this section. The research is divided into three tasks. Figure 1 below shows a flowchart and the sequence of these tasks.

Figure 1

Research Tasks Breakdown



As presented in Figure 1, Task 1, titled Dataset Development, encompasses all the data creation, cleaning, labeling and maintenance tasks pertinent to machine learning research and applications. The first subtask under Dataset Development is Dataset Construction. This includes the creation of audio files, transcription labels and class labels – all in a format acceptable to the machine learning models used in Task 2. The second subtask under Dataset Development is Speech Enhancement. Because of the noisy nature of ATC audio, the data had to be cleaned to optimize the performance of the overall transcription outputted from Task 2.

Task 2 is titled Machine Learning. It contains the machine learning framework used to transcribe the ATC audio created in Task 1. The first subtask under Machine Learning is Binary Classification. The role of the Binary Classifier was to label each

word in each audio file in one of two classes: ATC or ENGLISH. Many words used in ATC communication are unique and are not common in every day English. As a result, the nature of this data lends itself to binary classification. The next subtask under Task 2 is Speech Transcription. Once the word-by-word data has been given a class, it is then fed to two purpose-built models that will transcribe them. One of these models is only trained on ATC words and the other only trained on ENGLISH words. The rationale being that only training on one of the two classes will provide a purpose-built model suited for transcribing a particular class. This is by and large very similar to transcription models for different languages, which are only trained on words from a particular language. These transcription models predict and provide a word-by-word sequence of text labels. Given this and the sequence of words, the overall transcription for the entire audio file is constructed.

Task 3 is titled Evaluation. Its purpose is to quantify the performance of the framework. The first subtask under Task 3 is titled Framework Testing. The framework's performance is quantified using a Word Error Rate (WER) metric on the entire dataset. This gives a clearer picture on how well the model performs on its task of transcribing spoken sentences in an ATC environment. The next subtask under Task 3 is titled Comparison with Generalized Models. The framework is compared to a variety of baseline models to gauge its performance and practical feasibility.

3.2 Task 1: Dataset Development

As with any machine learning task, the quality and structure of the dataset is critical. This was no different for this research – especially due to its niche, both in its domain and underlying machine learning architecture. The ASR task in ATC domain

presents unique challenges, as highlighted in Chapter 2. Furthermore, the machine learning framework used to accomplish this task, highlighted in Task 2, is distinct in that it has a variety of machine learning tasks embedded within it. As a result, the dataset needs to be highly catered to this specific application and architecture. This section covers Task 1, the development of the dataset. This includes constructing audio files, feature extraction, cleaning the data, class labels, transcriptions, and data formats.

3.2.1 Dataset Construction

Because of the highly unique nature of this task and framework, constructing a purpose-built dataset was critical. Not only did the data have to be representative of the domain in which the model would be deployed, it also had to be presented in a format which the machine learning models would accept. Contained within this section is an overview of the given FAA data, constructed open-source data and the formats of the dataset at large.

3.2.1.1 FAA Data. The FAA provided us with a dataset of 10 audio files and their respective labels. These audio files were taken from real-life flights and record the communication between pilots and ATCOs. The audio is 8kHz and 16-bit sampled. Many of the dataset specific challenges mentioned in Chapter 2 were observed in this small sampling of data, including code switching and ATC specific vocabulary.

There were already a few challenges with this given dataset, having not even conducted any experimentation. First, there were only 10 given audio samples, totaling to under 40 seconds of audio. Machine learning is heavily data dependent – that is, without a high quality and high-volume dataset, the resulting model will perform very poorly. Presented with little data, there are very few features that the model can learn, yielding

poor performance and generalization. This is very problematic for any machine learning application, let alone one in a specific domain. Further, because this data was recorded in a real-life ATC environment, it is not surprising that it is very noisy. This posed challenges for the transcription task as low signal-to-noise ratios result in features that are harder to differentiate and detect.

3.2.1.2 Open-Source Data. Given the data scarcity issues in the provided FAA dataset, a larger dataset had to be constructed to make for a more robust model and to prove the effectiveness of the proposed framework and experimentation. The open-source dataset Air Traffic Control Complete [42] was used to construct additional audio files. Similar to the FAA data, the open-source dataset is also 8kHz and 16-bit sampled.

The same challenges present in the FAA data were also present with this dataset. The issues specific to ATC data, such as vocabulary and code switching, were also present in this dataset. In addition, the noisy ATC environments also posed an issue for this data. Finally, the small sample issue was not entirely solved with this constructed dataset. In total, 30 additional audio files were constructed from a two hour long recording. The addition of this data brought the total time of the entire dataset to just over 2 minutes. Comparing this to other common speech corpora, it becomes clear just how big an issue the small dataset size is. LibriSpeech [28], for example, is a speech dataset recorded in a clean environment coming in at 1000 hours or 60,000 minutes long. LibriSpeech is 5 orders of magnitude larger than the dataset used in this research. As a result, the purpose of this research was not to achieve state of the art performance. Rather, it was to prove the feasibility of the dataset format and machine learning framework.

3.2.1.3 Data Format. The format of this dataset is unique in that it presents audio files in a word-by-word sequence. Every audio file was manually split word-by-word. A segmentation algorithm was not used to achieve this word-by-word splitting for a variety of reasons. First, a segmentation algorithm would subject this dataset to the algorithm's own accuracy, and no segmentation algorithm is perfect. Second, performance degradation would occur due to the noise and domain of this data. There is no tailor-made segmentation algorithm for ATC data. For all these reasons, the audio files were manually split into word-by-word files by visually examining the waveform and listening for the beginning and end of each word.

Each word was given two labels: a binary class definition and a transcription. The binary class definition corresponds to either ATC or ENGLISH, where ATC is given to words that are specific to ATC speech and ENGLISH is given to words that belong to every day spoken English. The binary class labels are used in Task 2, Chapter 3.3.1. The transcription label is the written word that is spoken in each audio file. These labels are used to train transcription models corresponding to each of the two binary classes. The transcription models are developed in Task 2, Chapter 3.3.2. This highly specific dataset format is what allows for the machine learning framework, highlighted in Task 2, to function properly.

3.2.2 Speech Enhancement

Given the environment that ATC speech is recorded in, any experimentation would have to account for the presence of volatile noise. In this research, experimentation was conducted to determine the best method for speech enhancement on this particular dataset. Only spectral subtractive speech enhancement methods were examined. A deep

learning approach was not selected to maintain an independence on training data, as there is no speech enhancement method tailor made for ATC data. Once the optimal method was determined, the data was enhanced with this method before being used for training on the several models presented in this framework.

There are a total of seven different speech enhancement methods tested in this portion of the work. They are as follows: Martin [43], MCRA [38], MCRA2[38], IMCRA [39], Doblinger [44], Hirsch [44] and Conn_Freq [45]. Only spectral subtractive speech enhancement methods were tested as they have proven to perform well, comparable to other techniques such as Wiener Filtering [46]. All speech enhancement methods were tested using MATLAB code provided in [47]. Every audio file in the given FAA dataset and the constructed open-source dataset was enhanced using all seven aforementioned enhancement methods. Given enhanced audio files, the quality of the enhancement could then be tested.

The metrics selected to measure the enhancement quality are PESQ [40] and STOI [48]. The MATLAB code for the PESQ calculation was taken from [47]. These two metrics were used to compare the original, noisy speech to the enhanced speech, for both given FAA data and crafted open-source data.

3.2.2.1 Determining the Optimal Enhancement Method. The PESQ and STOI scores were determined for all audio files, enhanced with every enhancement method. Given these scores, the next step was to determine which method enhanced the audio the best.

To determine the optimal enhancement method, the 95% confidence intervals were calculated for both PESQ and STOI scores. This was the most accurate way to

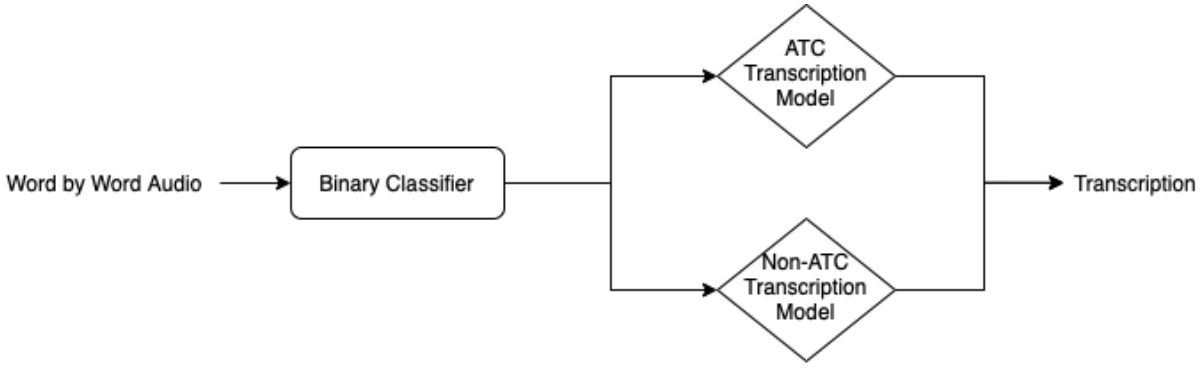
determine the best enhancement method on these datasets – independent of sample size. Because the two datasets were independently recorded, the PESQ and STOI scores and graphs were independently calculated as means to combat discrepancies in recording environments. This also substantiated the effectiveness of the optimal enhancement method. Finally, given the 95% confidence intervals, the optimal enhancement method was determined to be the Martin method. The Martin method was used to enhance all the audio files, which were then used for Task 2: Machine Learning.

3.3 Task 2: Machine Learning

The purpose of Task 2 was to take the cleaned data from Task 1 and develop a framework to transcribe the audio files. The architecture for Task 2 is highlighted in Figure 2 below.

Figure 2

Framework for Task 2: Machine Learning



The architecture is heavily influenced by the word by word, sequential nature of the dataset. On the left side of the figure, the word-by-word audio is fed into the Binary Classifier. The job of the Binary Classifier is to label whether the given word belongs to

an ATC corpus or if it is every day, non-ATC speech. Given this, the entire dataset is split up into ATC words and non-ATC words. These two subsets are then used to train two separate speech transcription models, one for ATC words and the other for non-ATC words. Depending on the class label given by the Binary Classifier, each word is fed through to the respective model corresponding to its class. Finally, given the sequence of words and the transcriptions of the models, a final transcription is outputted.

3.3.1 Binary Classification

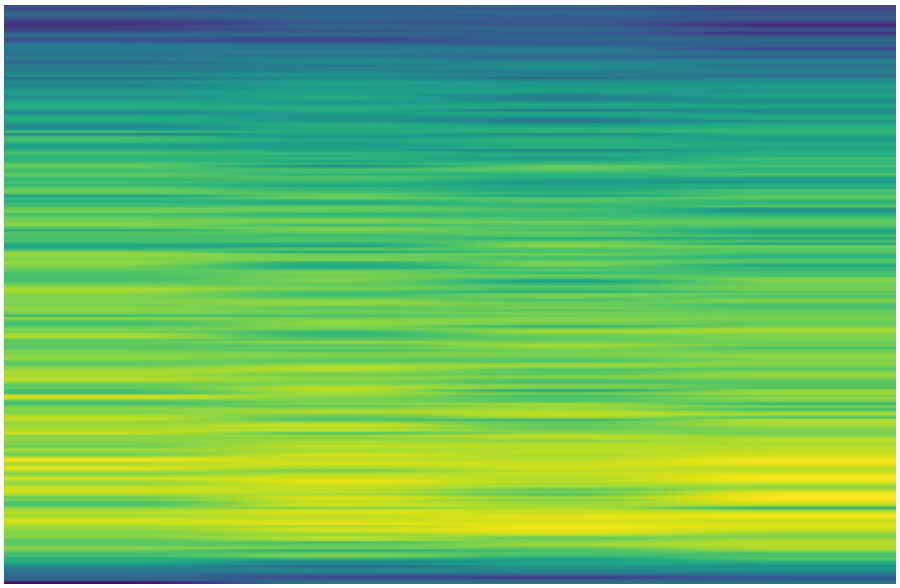
3.3.1.1 Data Preprocessing. The word-by-word audio files went through a preprocessing step before they were ready to be fed into the Binary Classifier for training. Because of the nature of the model architecture, discussed in the next section, the data had to be converted into spectrograms before being used for training and testing. Spectrograms were used because of the convolutional neural network architecture of the Binary Classifier. There have been audio recognition tasks accomplished using spectrograms as a data preprocessing step in CNN models [49]. This work converts the audio recognition task into an image recognition task by way of their data preprocessing. They found an 88.9% accuracy using their own CNN model and an 88.5% accuracy using transfer learning on a VGG19 model. In addition, multi-task audio classification was accomplished using spectrograms and deep learning [50]. They showed that the recognition accuracy was better using their multi-task model compared to multiple task specific models, many of which do not use deep learning but instead use traditional machine learning approaches. Spectrograms have proven their effectiveness as a data preprocessing step in deep learning audio classification tasks. As a result, a deep learning

approach to audio binary classification was chosen, thereby leveraging the effectiveness of spectrograms as a data preprocessing step.

A spectrogram is a way of representing audio visually in an image format. It shows the signal strength over time at the various frequencies that are present in the waveform. Spectrogram conversion is a common feature extraction technique used in speech recognition tasks. Spectrograms are very information dense, meaning they convey a lot of information very concisely. Where a typical time domain signal would only show signal amplitude at any given time, a spectrogram presents information in another dimension - the signal strength, over many frequencies and across time. Figure 3 below shows an example of a spectrogram in the training dataset for the Binary Classifier.

Figure 3

A Sample Spectrogram



This feature extraction and data preprocessing also allows for use of different model architectures. Because the audio is now represented in images, machine learning techniques suited for image data can be used to distinguish between the two binary classes, ATC, and ENGLISH. It is widely known that Convolutional Neural Networks (CNNs) are very effective at image classification [51]. Now that the Binary Classifier effectively became an image classifier, we can take advantage of CNN models to accomplish this task. The CNN model architecture of the Binary Classifier is discussed in the next section.

3.3.1.2 Model Architecture. The chosen binary classifier model architecture was primarily influential in the data preprocessing and feature extraction steps conducted before training. The conversion of each audio file into a spectrogram meant that we were now dealing with image data. In addition, the new type of data also simplified the task. The task was no longer to train a classifier to distinguish audio files into two classes. Instead, the task was to train a classifier to distinguish images into two classes. This change in datatype and task presented many new options when determining the Binary Classifier model architecture.

Convolutional Neural Networks (CNNs) have proven effective at classifying images [51]. As a result, it was wise to use a CNN model to build this Binary Classifier. The Binary Classifier consisted of 14 different layers and a total of almost 14 million trainable parameters. The architecture used a pattern in its design such that a succession of 2D convolutional layers, max pooling layers and dropout layers were repeated three times followed by flatten and dense layers which provided the final class output. Each piece of the repeated pattern has a rationale behind its selection. The convolutional layers

were selected due to their superior performance on the spectrogram image data. Max pooling and dropout layers were selected intuitively based on the initial training performance. Initial testing revealed that model was overfitting the training data and generalizing poorly on the test data. As such, max pooling and dropout layers were employed to combat this issue. A summary of the model architecture is seen in Figure 4 below.

Figure 4

A Summary of the Binary Classifier Model Architecture

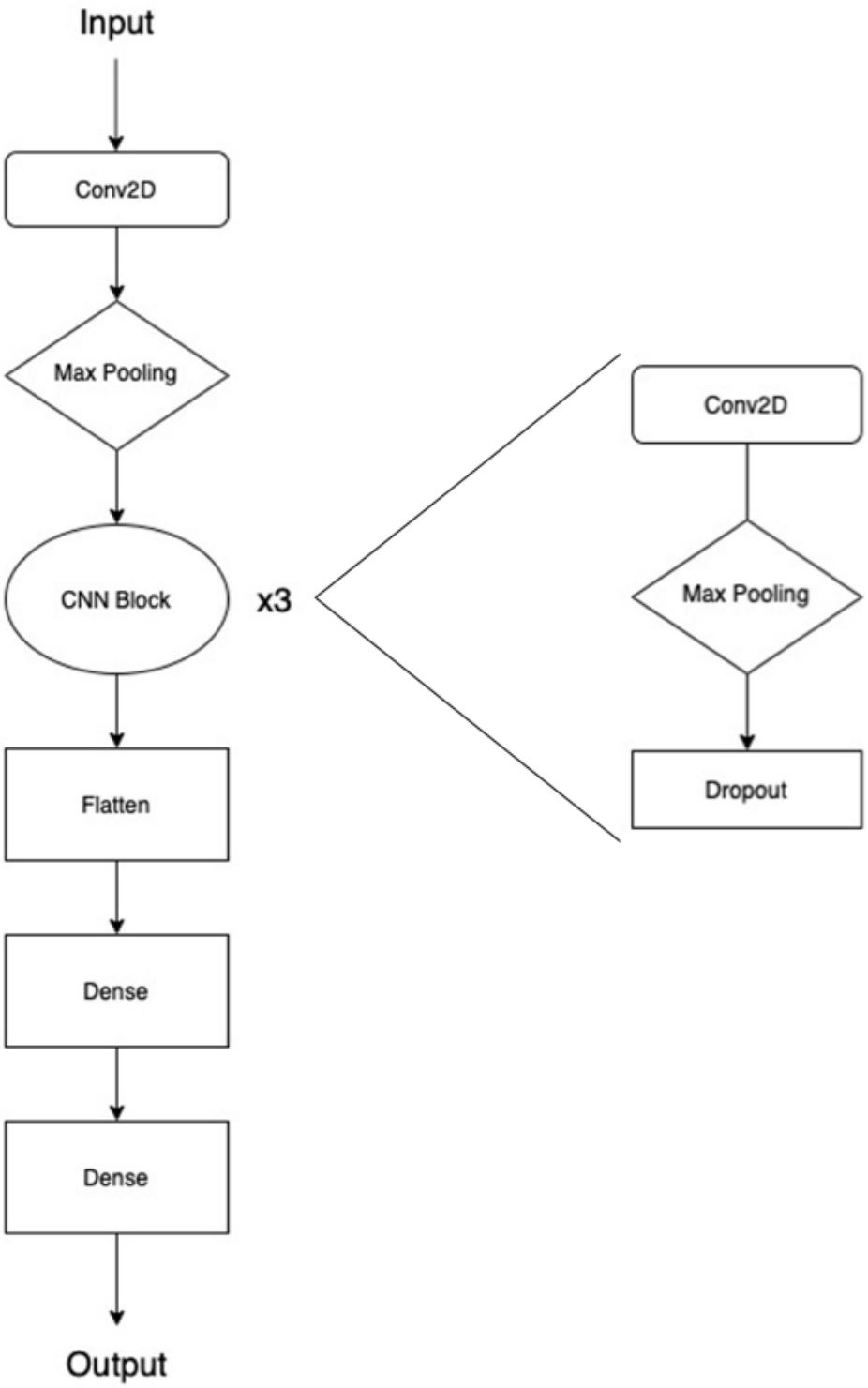
```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_10 (Conv2D)          (None, 332, 215, 32)      896

 max_pooling2d_9 (MaxPooling  (None, 166, 107, 32)     0
 2D)

 conv2d_11 (Conv2D)          (None, 164, 105, 64)      18496

 max_pooling2d_10 (MaxPoolin  (None, 82, 52, 64)       0
 g2D)

 dropout_2 (Dropout)         (None, 82, 52, 64)        0

 conv2d_12 (Conv2D)          (None, 80, 50, 128)       73856

 max_pooling2d_11 (MaxPoolin  (None, 40, 25, 128)      0
 g2D)

 dropout_3 (Dropout)         (None, 40, 25, 128)       0

 conv2d_13 (Conv2D)          (None, 38, 23, 128)       147584

 max_pooling2d_12 (MaxPoolin  (None, 19, 11, 128)      0
 g2D)

 dropout_4 (Dropout)         (None, 19, 11, 128)       0

 flatten_4 (Flatten)         (None, 26752)             0

 dense_8 (Dense)             (None, 512)               13697536

 dense_9 (Dense)             (None, 1)                 513

=================================================================
Total params: 13,938,881
Trainable params: 13,938,881
Non-trainable params: 0
```

As mentioned above and seen in the model summary, there is a sequence of 2D convolution, max pooling and dropout layers that repeat several times in the Binary

Classifier's architecture. A diagram of the entire model architecture is seen in Figure 5 below. The model begins with a 2D convolutional layer, followed by a max pooling layer and three repetitions of a pattern of a 2D convolutional layer, max pooling layer and a dropout layer. Finally, the model output is preceded by a flatten layer and two dense layers which produce the binary class output.

Figure 5

A Diagram of the Binary Classifier Architecture



3.3.1.3 Training and Performance Metrics. As with training any machine learning model, there are a number of decisions to make by way of hyperparameters and performance metrics. The Binary Classifier was trained on 10 epochs, using the Adam optimizer and sparse categorial cross entropy loss function. The model was trained on 70% of the total data and tested on the remaining 30%.

The performance metrics used to prove the model architecture and data format were loss values and validation accuracy. The loss value is a metric which measures how far off the model's prediction was from the actual label. As the model trains, the loss values should decrease because of the model learning the features in the dataset. Validation accuracy is a measure of how well the model performs on the test dataset, i.e. how many of the test samples it classifies correctly. As the model learns, the validation accuracy should increase, meaning the model has learned how to generalize to data it has not seen before. In addition, it is important to examine the accuracy for a class imbalance. If the model is only accurately identifying one of the classes, the overall accuracy metric can be misleading. As such, the overall accuracy was further examined to determine if there was a class imbalance.

3.3.2 Speech Transcription

There were two ASR transcription models trained in Task 2. These models were trained for the two different classes that each word in the dataset belonged to – ATC or ENGLISH. Both models have the same architecture and follow a pretrained, fine-tuning strategy to adjust model parameters in hopes to combat the issue of small sample size and domain uniqueness. The following sections explain the datasets, model architecture and performance metrics in greater detail.

3.3.2.1 Dataset. The word-by-word audio dataset was split into two classes which corresponded to ATC specific vocabulary and everyday English speech. These two classes were analogous to two different languages. As such, there were two different models trained on datasets specific to these classes. Because of the scarcity of samples, there were too few features to train a single model for fear that the model would not be able to learn enough. In addition, the framework was constructed such that the preceding Binary Classifier step would distinguish between the two classes and feed each word into the corresponding transcription model. Much in the same way as the Binary Classifier training, the datasets for these two models followed the same 70/30 train/test split.

3.3.2.2 Model Architecture. The model chosen for the transcription task was Facebook's Wav2Vec 2.0 [52]. As explained by their team, there are thousands of different languages spoken in the world, with different dialects, environments, and accents. This presents many difficulties for speech recognition tasks, as developing high quality datasets for all of the different languages and environments is not realistic. As a result, it becomes critical to investigate methods of speech transcription that can generalize well by training on relatively few samples.

This dataset is marred by many of the problems Wav2Vec 2.0 aims to solve – especially the scarcity of samples. In addition, noise, domain uniqueness and feature imbalances all pose issues for a typical ASR model. The team behind Wav2Vec 2.0 showed that training on a small amount of labeled audio and pretraining on a large amount of unlabeled data resulted in good performance – achieving as low as 5.2% word error rate.

Following much of their example, the transcription models are fine-tuned Wav2Vec 2.0 models. That is, they take the pretrained weights of the existing model and fine-tune them according to the relevant dataset corresponding to the two binary classes, ATC or ENGLISH. This methodology promised to optimize the performance of the transcription, as it closely mirrored the procedure followed in the original work.

3.3.2.3 Performance Metrics and Experimentation. Because the training strategy consisted of fine-tuning an existing model, there were significantly less decisions to make by way of hyperparameters and tuning. An important decision to make in this experimentation was to determine an appropriate metric to calculate model accuracy. In addition, the accuracies of the transcription models were determined both on classes. That is, the ASR transcription model was tested on not just ASR words, but also on ENGLISH words. The same is true for the ENGLISH transcription model – it was tested on both ENGLISH and ATC datasets. The motivation behind testing on the other dataset was to gauge performance in case the preceding Binary Classifier was wrong in its prediction, in which case a word would be given to the wrong transcription model**.** The following sections explain the chosen accuracy metric as well as all the experiments conducted for the transcription models.

3.3.2.3.1 Character Error Rate. The overall task of the transcription models was to take in audio of a single word and output a predicted transcription. The delta between the prediction and the label gives an indication of accuracy. To quantify this discrepancy and gauge model accuracy, the Character Error Rate (CER) was calculated. Equation 1 shows the formula used to calculate CER.

$$CER = \frac{(S + D + I)}{N} \quad (1)$$

In this equation, the numerator holds information about the model prediction and the denominator represents the label. In the numerator, S is the number of substitutions, D the number of deletions and I the number of insertions. In the denominator, N is the number of characters in the label. The CER formula is intuitive in that a more accurate prediction will yield a smaller numerator and, thus, a smaller error rate. The smaller the CER, the better the transcription model performed.

3.3.2.3.2 Determining Accuracies on Specific Datasets. The transcription models were tested on a variety of datasets. Both the ASR and ENGLISH models were tested on their corresponding datasets, as well on the opposite dataset. This resulted in four different CERs:

1. ATC model on ATC data
2. ATC model on ENGLISH data
3. ENGLISH model on ATC data
4. ENGLISH model on ENGLISH data.

The motivation behind these experiments was to determine how the transcription step would perform if the Binary Classifier predicted the word's class both correctly and incorrectly. There is a possibility that the classifier predicts the word's class wrong and ends up sending it to the wrong model for transcription. These four metrics give a greater understanding of the strengths and potential weaknesses of the transcription step.

3.4. Task 3: Evaluation

Finally, with the Binary Classifier and Transcription models individually trained and tested, the entire framework was ready to transcribe audio files. This section explains the flow of testing the framework. It is a unique workflow as it contains several different machine learning models and is dependent on the sequential nature of the word-by-word dataset.

3.4.1. Framework Testing

To test the framework, all audio files were fed through the framework and the final Word Error Rate (WER) was calculated. WER is the same as CER except that the calculations are done on a word-by-word basis instead of character by character. To test the framework, the word-by-word audio files were first fed into the Binary Classifier, transformed into spectrograms, and given a class, ATC or ENGLISH. Then, each word was fed into the transcription model corresponding to its predicted class. The transcription model would then transcribe the word. At this point, every word in the audio file had a transcription. Because the sequence of words was known, a transcription of the entire audio file was pieced together word by word to obtain the entire transcription.

3.4.2. Comparison with Generalized Models

The performance of the developed framework was determined with the audio files of several test datasets. Word error rate (WER), much like CER, is a common metric for measuring the accuracy of a speech-to-text ASR system. Instead of calculating the difference between the prediction and label on a character per character basis, the difference is calculated on a word-by-word basis.

To gauge the framework's feasibility, it was compared to two different Wav2Vec 2.0 models. The first was an off the shelf, pretrained, Wav2Vec 2.0. This was trained on thousands of hours of audio that was recorded in a clean environment. The second was a Wav2Vec 2.0 model trained from scratch using the proprietary ATC dataset, following a 70/30 train/test split – training on under 2 minutes of data. These two points of comparison were chosen to gauge whether the research approach and framework were feasible. Given a large dataset, spanning several hours of audio, a transcription model would be trained with the standard approach that ASR algorithms use. However, ATC corpora are noisy and large datasets are hard to come by. This thesis's framework was developed to combat those issues. As such, even in the presence of few samples, the word-by-word framework should theoretically outperform both Wav2Vec 2.0 models, pretrained and trained on this dataset.

To prove that the framework outperformed other baselines consistently, it's WER was determined using 20-fold cross validation. 20 different test datasets were constructed and a corresponding WER was calculated for each set. Then, the average WERs and 95% confidence intervals were calculated for each of the three models. Finally, the average and confidence intervals were plotted on the same graph to determine whether the framework statistically outperformed the other baselines.

Chapter 4

Results and Discussion

This chapter provides a detailed look into the results of all experimentation done throughout this research. The results are presented in the same order as Chapter 3, the order in which the experiments were performed.

The first experiment done was to determine the optimal speech enhancement method for this dataset. The 95% confidence interval plots for both PESQ and STOI metrics are presented in Figures 6, 7, 8 and 9. These graphs are used to determine the best enhancement method to combat the noisy nature of ATC data.

Next, a Binary Classifier was developed to categorize every word as either ATC or ENGLISH. The Binary Classifier's performance was analyzed with two metrics: validation loss and validation accuracy. The relevant section contains loss and accuracy plots versus training epoch. The final validation accuracy was determined using these graphs.

Then, two transcription models were trained on purpose built datasets to transcribe words belonging to either of the two classes: ATC or ENGLISH. These models were evaluated for their performance using their Character Error Rate (CER). The average CER was calculated for the class that they were trained for as well as the opposing class – as a measure to determine model robustness.

Finally, the overall framework's performance was determined using WER. Given a word-by-word transcription of an audio file, the entire transcription was pieced together. The average WER was calculated and compared to two baseline generalized models: a pretrained Wav2Vec 2.0 and a Wav2Vec 2.0 trained from scratch.

4.1 Speech Enhancement

Four 95% confidence interval plots, corresponding to two datasets and two performance metrics (PESQ and STOI), for the analysis of seven speech enhancement methods follow. The four different experiments done in this section are: FAA PESQ, FAA STOI, non-FAA PESQ and non-FAA STOI. Each experiment yielded a plot of averages and 95% confidence intervals. These four plots were analyzed to determine the optimal enhancement method according to the conducted experiment. Finally, the last subsection in this experiment provides the rationale for determining the best speech enhancement method according to these four experiments.

4.1.1 FAA PESQ Scores

Figure 6 below shows the PESQ scores and 95% confidence intervals for the FAA dataset. Each datapoint represents the average PESQ score for the given enhancement method. The vertical lines extending from the dot represents the 95% confidence interval for the data. For the FAA dataset, the PESQ scores show, with statistical significance, that the Martin method is the most optimal speech enhancement method. FAA speech files enhanced with the Martin method yielded a range of PESQ scores from 3.45 to 3.63, with an average of 3.54. The method that is closest to the Martin method, in terms of PESQ performance, is the Hirsch method. All the other six enhancement methods, however, all have overlapping confidence intervals. This is an indication that, for this experiment, they are not statistically significantly different from each other. The conclusion of this experiment was that the Martin method performed best on the FAA dataset.

Figure 6

PESQ Scores for the FAA Dataset

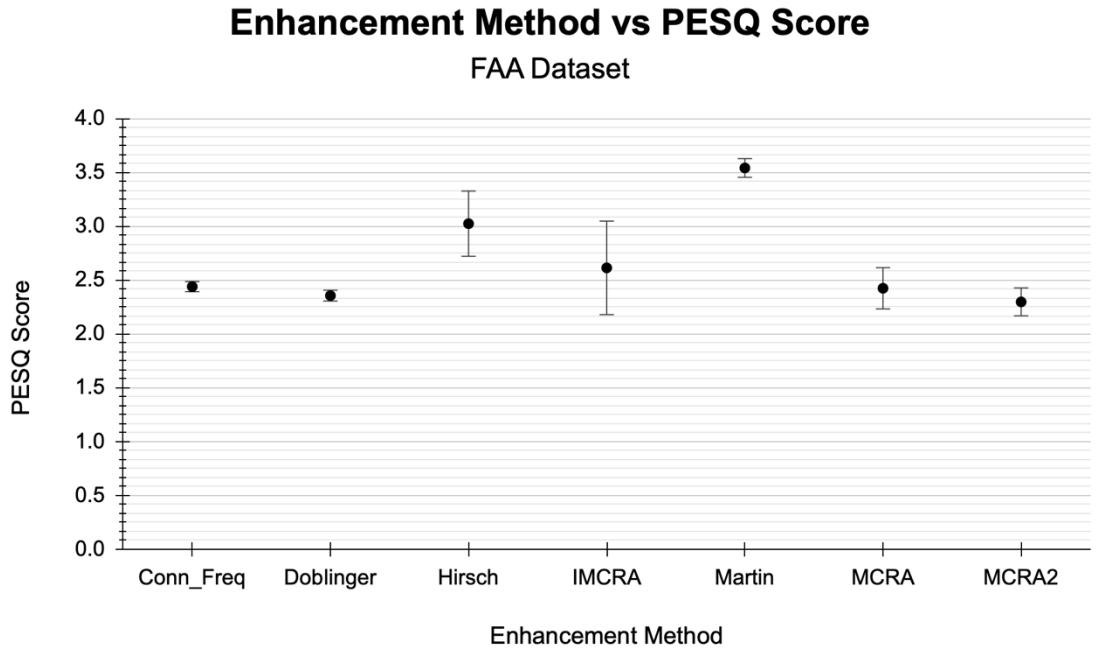


Table 1

PESQ 95% Confidence Intervals for the FAA Dataset

	Average	95% CI
Conn_Freq	2.43	[2.39, 2.47]
Doblinger	2.35	[2.3, 2.4]
Hirsch	3.02	[2.72, 3.32]
IMCRA	2.61	[2.18, 3.04]
Martin	3.53	[3.45, 3.61]
MCRA	2.42	[2.23, 2.61]
MCRA2	2.29	[2.17, 2.41]

4.1.2 FAA STOI Scores

Figure 7 below shows the STOI scores for the FAA dataset. STOI scores have a much lower range compared to PESQ. This is important to note because the data does not have large variation for this reason. The result of this experiment was that the Doblinger enhancement method yielded the highest STOI scores for the FAA dataset. Doblinger did not outperform the rest of the enhancement methods with statistical significance, however. It had a range of STOI scores from 0.61 to 0.63, with an average of 0.62. While it did have the highest average, its confidence interval overlapped with all of the other enhancement methods. This very well may be due to the smaller range of values that STOI scores can take as well as the small sample size of the data, only ten audio files.

Figure 7

STOI Scores for the FAA Dataset

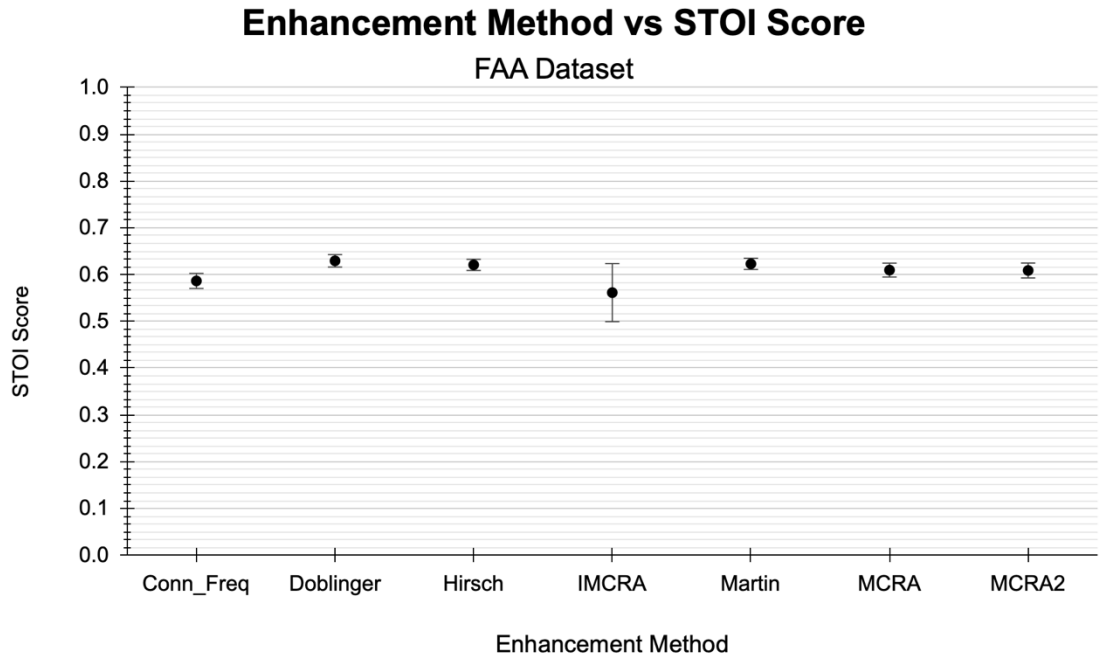


Table 2

STOI 95% Confidence Intervals for the FAA Dataset

	Average	95% CI
Conn_Freq	0.58	[0.57, 0.59]
Doblinger	0.62	[0.61, 0.63]
Hirsch	0.62	[0.61, 0.63]
IMCRA	0.56	[0.5, 0.62]
Martin	0.62	[0.61, 0.63]
MCRA	0.6	[0.59, 0.61]
MCRA2	0.6	[0.59, 0.61]

4.1.3 Non-FAA PESQ Scores

Figure 8 below is a graph of the PESQ scores on the non-FAA dataset. The averages and 95% confidence intervals have a much wider range of values compared to STOI. The results from this experiment show that the Martin method has the highest range PESQ score, ranging from 4.06 to 4.26 with an average of 4.16. Martin scores highest, but without statistical significance, as the error bars overlap with the Hirsch and IMCRA enhancement methods. A potential cause for this is the small sample size of this dataset, coming in at 30 audio files. In addition, this data was recorded in a different environment from the FAA dataset. Upon cursory auditory evaluation, it appears that the non-FAA dataset is not as noisy as the FAA dataset. These observations very well may impact the scores and results of experimentation. The conclusion of this test was that the Martin method was the optimal speech enhancement method for the non-FAA dataset, albeit without statistical significance.

Figure 8

PESQ Scores for the Non-FAA Dataset

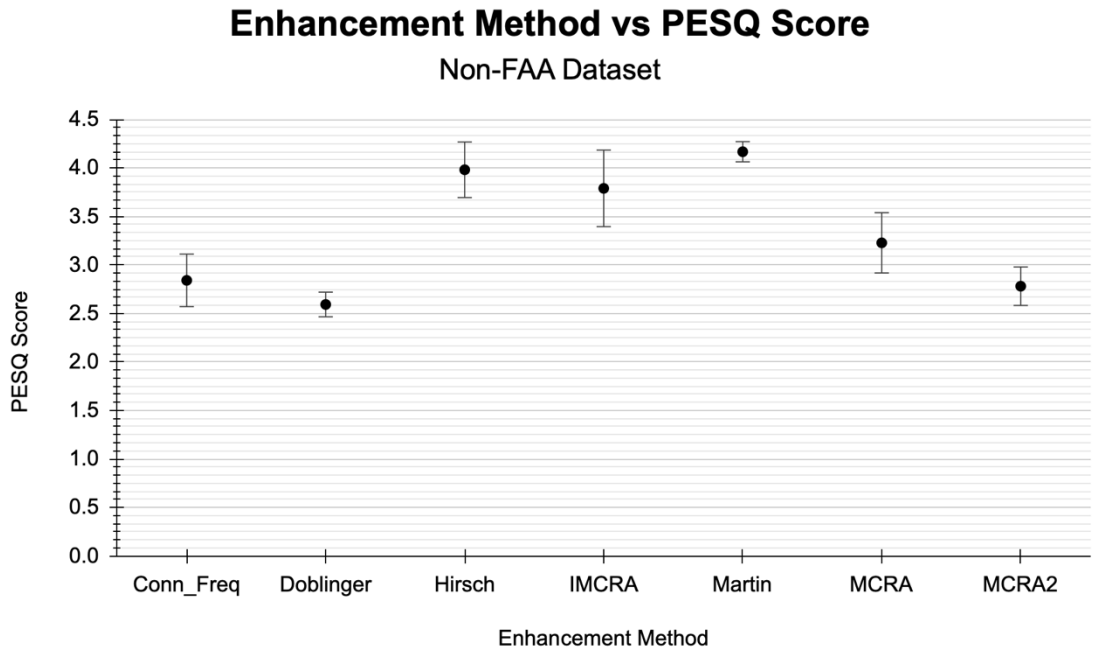


Table 3

PESQ 95% Confidence Intervals for the Non-FAA Dataset

	Average	95% CI
Conn_Freq	2.84	[2.57, 3.11]
Doblinger	2.59	[2.47, 2.71]
Hirsch	3.98	[3.7, 4.26]
IMCRA	3.78	[3.39, 4.17]
Martin	4.16	[4.06, 4.26]
MCRA	3.22	[2.91, 3.53]
MCRA2	2.78	[2.59, 2.97]

4.1.4 Non-FAA STOI Scores

The plot of STOI scores on the non-FAA dataset are shown in Figure 9 below. Again, because of the nature of STOI scores, the data does not vary as much as PESQ scores. The Doblinger enhancement method had the highest average STOI score, at 0.62, with a range of 0.57 to 0.67. However, as with the STOI data on the FAA dataset, it did not outperform the other methods with statistical significance. In fact, the STOI scores on the non-FAA dataset varied even less than the FAA dataset. A potential reason for this is the different, and observed to be less noisy, environment this data was collected in compared to the FAA data. In addition, the larger dataset size may have reduced the variation in STOI scores. There are three times as many audio samples in this dataset compared to the FAA dataset. As such, the averages and confidence intervals are less likely to be swayed by potential outliers. In any case, the conclusion of this experiment was that the Doblinger method was most optimal for speech enhancement on the non-FAA dataset, without statistical significance.

Figure 9

STOI Scores for the Non-FAA Dataset

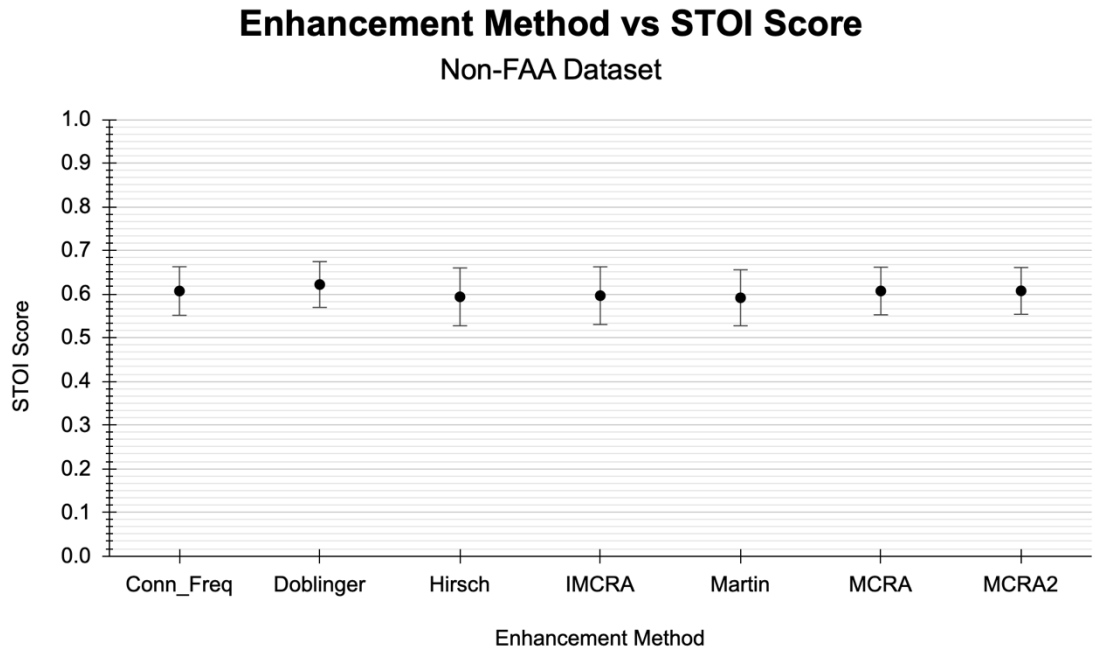


Table 4

STOI 95% Confidence Intervals for the Non-FAA Dataset

	Average	95% CI
Conn_Freq	0.6	[0.55, 0.65]
Doblinger	0.62	[0.57, 0.67]
Hirsch	0.59	[0.53, 0.65]
IMCRA	0.59	[0.53, 0.65]
Martin	0.59	[0.53, 0.65]
MCRA	0.6	[0.55, 0.65]
MCRA2	0.6	[0.55, 0.65]

4.1.5 Determining the Optimal Speech Enhancement Method

The four experiments conducted in this section each yielded an enhancement method which scored the best for the particular metric under examination. FAA PESQ scores said that the Martin method was optimal, with statistical significance. Non-FAA PESQ scores also said the Martin method was optimal, without statistical significance. Both FAA and non-FAA STOI scores said that the Doblinger method was optimal, without statistical significance. Table 5 below shows a concise breakdown of these results. Given these varied results, an optimal method had to be chosen to continue with experimentation. Statistical significance was given importance as deciding factor. In the case of statistically insignificant data, the overlap of error bars was analyzed. If there was a lot of overlap with many other enhancement methods, the statistical insignificance was given less importance.

Table 5

Optimal Enhancement Methods for the Four Conducted Experiments

	FAA	**Non-FAA**
PESQ	Martin (statistically significant)	Martin
STOI	Doblinger	Doblinger

Given this rationale, the STOI plots became less relevant, as the data was statistically insignificant. The vast majority confidence intervals overlapped. In the case of PESQ plots, the only instance of statistically insignificant data was such that

overlapping confidence intervals were small and infrequent. As a result, the optimal speech enhancement method for this research was determined to be the Martin method.

4.2 Binary Classifier

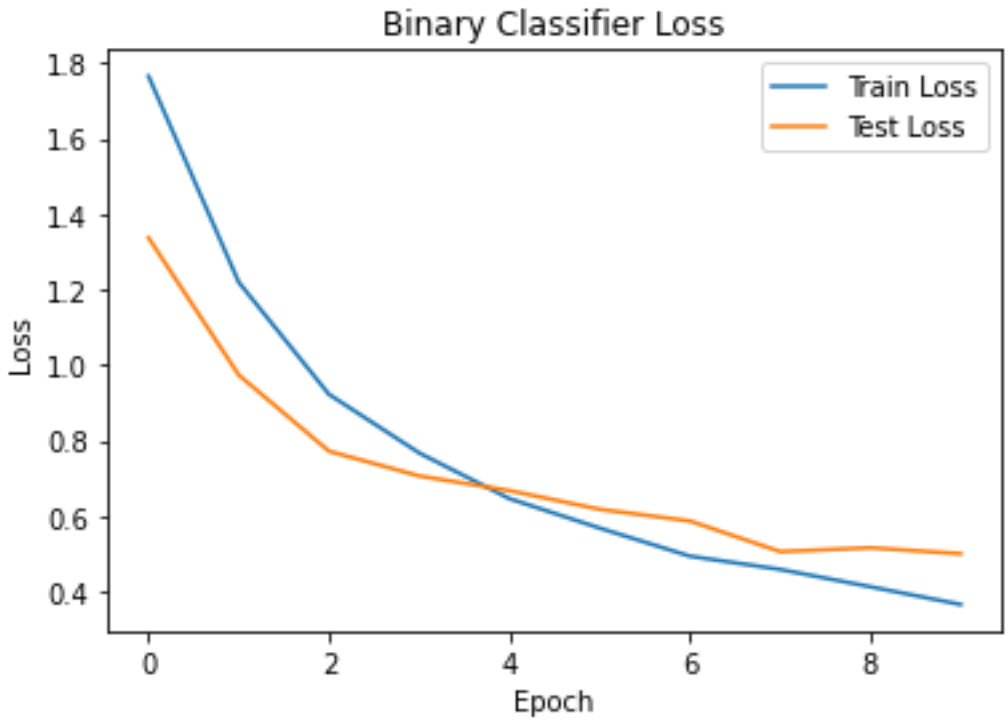
Graphs of performance metrics for the Binary Classifier are contained within this section. Two performance metrics were selected: validation loss and validation accuracy. These two metrics were plotted in a graph versus training epoch. The training epoch is a measure of time – a higher number epoch means that the model has been training for longer. An inverse relationship between loss and accuracy was expected. That is, the loss versus epoch graph would trend downward and that the accuracy versus epoch graph would trend upward, as the model trained.

4.2.1 Validation Loss

The following plot is a loss versus time graph of the Binary Classifier's loss values as it trains. The loss of a model is a metric which acts as a proxy to the model's performance. It is a measure of how far the model's prediction is from the actual label. Larger loss values indicate that the model is performing poorly – that is the model prediction is far from the true value. Because the loss is a performance metric, it is expected that as the model trains for more epochs, the loss values will decrease. Figure 10 below shows a loss versus epoch graph of the Binary Classifier.

Figure 10

Binary Classifier Loss Values Versus Epoch



As expected of successful model training, both the train and test loss curves trend downward as the epoch values increase. The train loss has a descends quicker and terminates at a lower value than the test loss. The train loss terminated at a value of 0.38 where the test loss terminated at a value of 0.57. Should model training have continued for more epochs, it is expected that the loss curves would asymptotically approach a loss value which is likely going to be the lowest it can get for this dataset and model architecture. Because this research was to prove the overarching framework, extensive time was not spent in hyperparameter optimization. Given more time to conduct more experimentation, it is possible that the model can achieve lower loss values. These

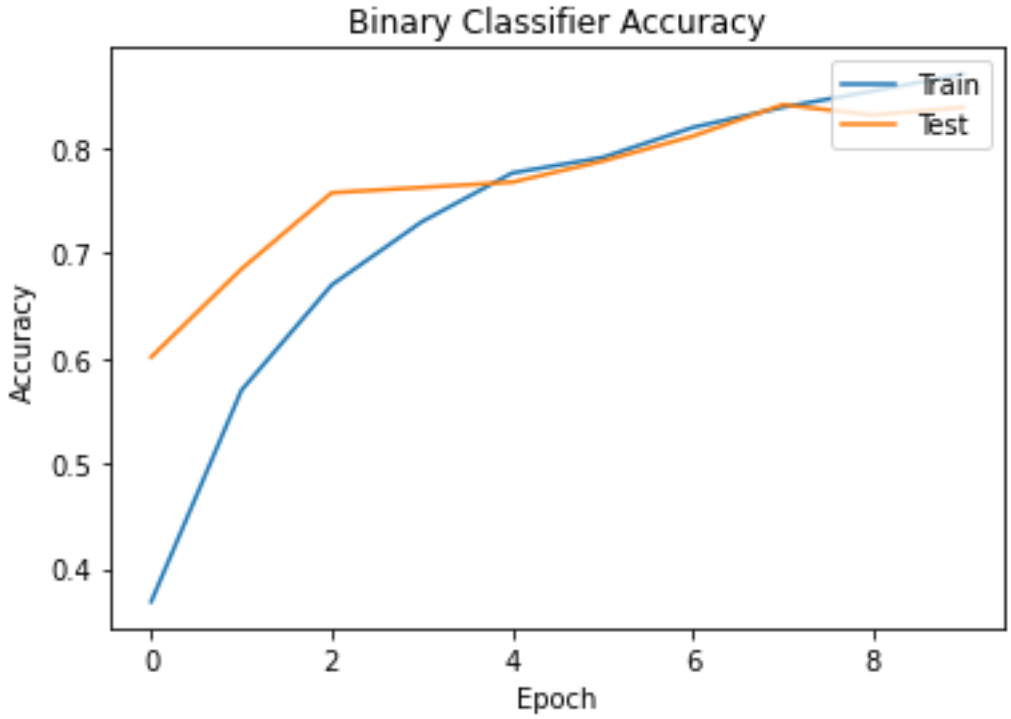
optimizations yield diminishing returns, however, as the training and tuning of hyperparameters takes time and does not produce corresponding results.

4.2.2 Validation Accuracy

The following plot is an accuracy versus time graph of the Binary Classifier's accuracy values as it trains. It is expected that as the model trains for longer, it will learn the features in the dataset. As a result, the model will increase in its accuracy. This should hold true for both the train and test datasets. Once the model sees and learns from more data in the train set, its weights, biases, and parameters will continue to update. These updates will yield better generalization on the test set – data that it has not seen before. Successful model training yields accuracy plots that trend upward. Figure 11 below shows an accuracy versus epoch plot for the Binary Classifier.

Figure 11

Binary Classifier Accuracy Values Versus Epoch



The Binary Classifier trained successfully and achieved acceptable accuracy values. As expected, the accuracy curves trended upwards as the model trained for more epochs. The Binary Classifier achieved a final train accuracy of 97% and a final test accuracy of 88%. Much like the loss values, there are diminishing returns for extensive hyperparameter optimization. Training for more epochs, modifying model architecture, testing different loss functions will all have an impact on model accuracy. This impact, however, is not guaranteed to be positive. Hyperparameter optimization very well may improve model accuracy. However, it comes with the tradeoff of investing time which will yield insignificant gains given the model's adequate performance.

The Binary Classifier proved a successful technique to classify words into two classes: ATC or ENGLISH. Each audio file was first converted into spectrogram images. These images were then fed into a CNN model suited for image classification. The Binary Classifier portion of the framework proved successful – as it classified word by word audio files with 88% accuracy.

4.3 Transcription Models

There were four experiments conducted to test the transcription models. The following sections contain the results for testing on all four scenarios for the transcription step of the framework. Each experiment resulted in a calculated CER. It was expected that models tested on their corresponding dataset would perform better than models tested on the opposing dataset. Table 6 below shows the average CERs on both the train and test sets for the four combinations of models and datasets.

Table 6

Character Error Rates for the Four Combinations of Models and Datasets

Model – Dataset Combination	Train CER	Test CER
ATC – ATC	79%	75%
ATC – ENGLISH	-	83%
ENGLISH – ENGLISH	76%	74%
ENGLISH – ATC	-	86%

The four conducted experiments yielded results that were expected of the transcription step. Models tested on their corresponding datasets had lower CERs than when they were tested on the opposing datasets. The ATC model achieved a CER of 75%

on ATC words and 83% on ENGLISH words. The ENGLISH model had a CER of 74% on ENGLISH words and 86% on ATC words. These CERs are certainly not adequate performance for a live, mission critical system. In the best case, one out of every four letters, or every fourth letter, would be correct in the word level transcription. It can be very difficult to make sense of the word given these errors. It becomes next to impossible to make sense of an entire sentence if all of the words have significant errors in them. This is in the best case, where the Binary Classifier predicts the class of the word correctly. In the cases where the word is given to the wrong transcription model, the accuracies suffer even more. In the worst case, the ENGLISH model on ATC words, only 14% of the characters would be transcribed correctly. Given these results, it is difficult to say that the transcription step had adequate performance when state of the art models regularly achieve single digit character and word error rates.

4.4 Overall Framework Performance

The following section contains the performance of the overall framework as well as a comparison to two baseline Wav2Vec 2.0 models. Table 7 below compares average WERs and corresponding 95% confidence intervals for three models: the framework, a pretrained Wav2Vec 2.0 model and a Wav2Vec 2.0 trained, from scratch, on this dataset.

Table 7

Word Error Rates for the Framework and Generalized Models

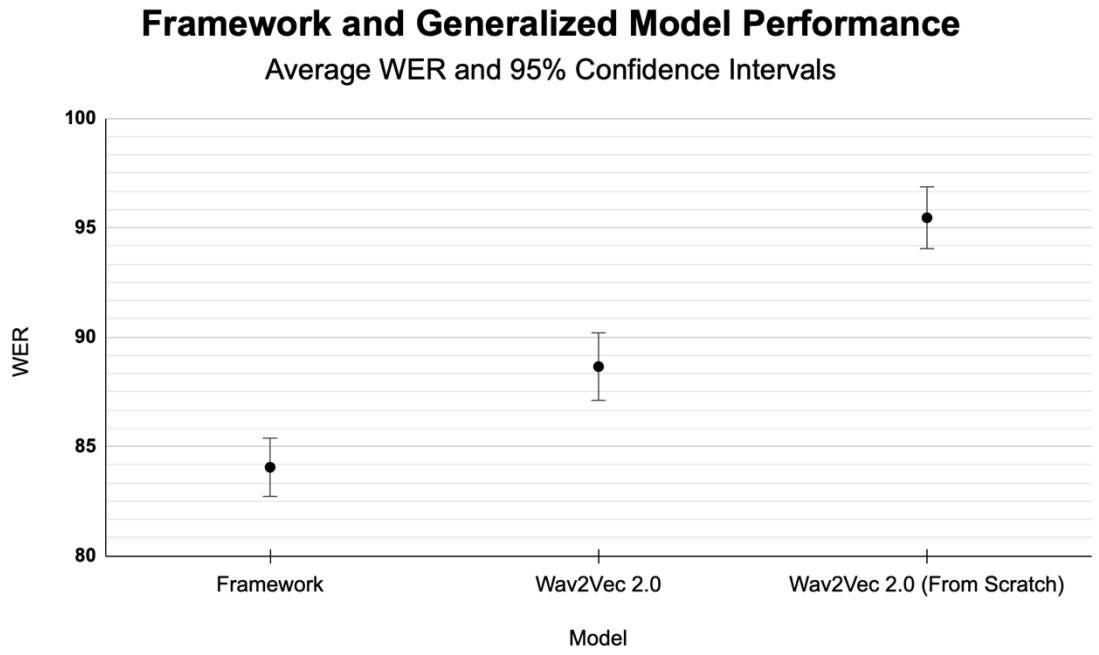
	Framework	**Wav2Vec2.0 (pretrained)**	**Wav2Vec2.0 (trained from scratch)**
WER	84.05%	88.65%	95.45%
95% CI	[82.72, 85.38]	[89.87, 89.87]	[94.04, 96.86]

Percent Improvement	11.40%	6.80%	0.00%

The proposed framework achieved an 84.05% average WER on all the audio files in this dataset. A pretrained Wav2Vec 2.0 model had an 88.65% WER. Training Wav2Vec 2.0 from scratch yielded a 95.45% WER. To analyze and properly compare these results, the averages and 95% confidence intervals were plotted on the same graph to determine whether the results were statistically significant. Figure 12 below is a graph depicting the performance of the three models.

Figure 12

Framework Performance Compared to Generalized Models



The framework performed the best out of the three tested models, with statistical significance. All three of the models had very high WERs that are not fit for mission

critical systems. In the case of the Wav2Vec model developed from scratch, training on minutes of data yielded the highest WER, because there was simply not enough data to learn from. For the pretrained Wav2Vec 2.0, trained on thousands of hours of data, yielded better performance, but still far from adequate. The wide range of features and larger dataset helped improve the performance, but it was still far from state of the art. The unique, word-by-word framework presented in this thesis proved to improve performance compared to the other two models, with statistical significance across 20 test datasets. This is a promising step for the future of this research. This type of dataset and word-by-word transcription may prove to be a feasible method for ASR in ATC.

In summary, this chapter presented detailed results and relevant discussion on how the proposed framework performed compared to available standard speech transcription models. The following chapter provides a conclusion, limitations of the research and future recommendations.

Chapter 5

Conclusion

Chapter 5 contains a summary of the chapters and accomplishments of this thesis. The research objectives and accomplishment as presented in Chapter 1 are reviewed. In addition, recommendations for future research and improvements are given.

5.1 Thesis Review

Chapter 1 presented the problem solved by this research as well as an overview of the goals and objectives of this thesis. Chapter 2 was a literature review of the current state of affairs for ASR in ATC. Chapter 3 is a detailed examination of the methodology and framework used in this research to accomplish the goals laid out in Chapter 1. Chapter 4 presented the results of the experimentation explained in Chapter 3. Finally, this chapter, Chapter 5, is a summary of the work and accomplishments of this thesis.

5.2 Summary of Accomplishments

The goal of this thesis was to evaluate feasibility of a novel distributed machine learning framework for ASR in ATC designed to combat the common problems of the domain. The results demonstrated that a word-by-word, binary classification and transcription architecture is a feasible approach to accomplish ASR on ATC data. Compared to baseline state-of-the-art ASR models, the framework presented in this research achieved better performance. The objectives laid out in Chapter 1 are reviewed and a summary of related accomplishments is given:

1. *To determine the best speech enhancement method for ATC audio data.*

- Seven spectral subtractive speech enhancement methods were tested for their performance on ATC data. PESQ and STOI scores were used to measure

performance. All audio samples in the dataset were enhanced using the seven enhancement methods and the PESQ and STOI scores were calculated. Given this data, the average and 95% confidence intervals were plotted to determine which enhancement method performed best, with statistical significance.

2. *To develop a word-by-word ATC audio corpus.*

- A word-by-word ATC dataset was developed using a combination of FAA provided data and an open-source ATC audio corpus. This dataset was tailor-made for the framework. Every audio file had a corresponding spectrogram and two labels. One label was a binary class indicating which of the two classes the word belonged to, either ATC or ENGLISH. The other label was a transcription of the word spoken in the audio file.

3. *To train a Binary Classifier to distinguish between ATC and ENGLISH words.*

- A Binary Classifier was trained to classify words into two classes. It took advantage of data preprocessing by converting audio files into spectrogram images. This preprocessing then allowed the Binary Classifier architecture to use convolutional layers suited for image detection. The Binary Classifier achieved an accuracy of 88%.

4. *To train ASR models to specifically transcribe ATC and ENGLISH words.*

- Two ASR transcription models were trained to transcribe words belonging to the two classes. That is, a transcription model was trained specifically on ATC words with the sole purpose of transcribing only words belonging to the ATC class. The same was done for a model to transcribe ENGLISH words. The models used a pretrained Wav2Vec 2.0 and fine-tuned its parameters to achieve greater accuracy

for their transcription tasks. The ATC model achieved a 75% CER and the ENGLISH model achieved a 74% CER.

5. *To transcribe ATC audio on a word level basis.*

- ATC audio files containing entire sentences were transcribed on a word-by-by basis using the proposed framework. First, every word was fed into a Binary Classifier which would predict which class the word belonged to – either ATC or ENGLISH. Then, the word was given to a purpose-built transcription model trained to transcribe words belonging to one of the two classes. ATC words were transcribed by an ATC transcription model and ENGLISH words were transcribed by an ENGLISH transcription model. With every word transcribed and the sequence of words known from the original audio file, a transcription of the entire spoken sentence was pieced together.

6. *To analyze performance of the developed framework in comparison to the current state of the art ASR models.*

- The performance of the proposed framework was determined using a WER accuracy metric which compared the transcription to the label. The framework achieved an 84% WER on this dataset. The framework was then compared to two baseline models: an off the shelf, pretrained Wav2Vec 2.0 and a Wav2Vec 2.0 trained from scratch on the developed dataset. The pretrained model had a WER of 88% and the model trained from scratch had a WER of 95%. The framework performed better than the two baselines, with statistical significance. However, this performance is still far from adequate for a mission critical system such as ATC. These results are promising for the future of this framework and application. However, further testing

needs to be conducted to improve the framework's accuracy and achieve sufficient performance.

5.3 Limitations of the Research and Recommendations for Future Work

The biggest challenge for this research was the severe lack of data. In totality, there were only about 2 minutes of audio in the entire dataset. Many state of the art ASR systems are trained on thousands of hours of audio. However, as presented in Chapter 4 and concluded in Chapter 5, the proposed distributed machine learning framework performed better than the generalized models. This suggests that testing of this framework on a larger dataset may provide significant performance gains. More research efforts and funds should be allocated towards development of a word-by-word dataset in the mold laid out in this thesis. This would further validate the framework to be a viable solution to the unique problem of ASR in ATC.

A modification of this framework should be tested in the Binary Classification step. In this work, a CNN based model was used. This required the audio data to be converted into spectrogram images. However, it is worth investigating other traditional machine learning methods to achieve Binary Classification. Some of these methods include Gaussian Mixture Models and Support Vector Machines. These methods would use the time series audio data and would not require the audio files to be converted into spectrograms. If these methods provide similar or better performance than the CNN based Binary Classification approach used in this work, it will eliminate a data preprocessing step and simplify the framework without diminishing performance.

Another point of improvement in future research is optimizing the transcription step of the framework. In this work, the two transcription models operated independent of

each other. To improve this, game theory can be implemented such that the two models act as players who will act as a unit to optimize their output. This may prove to be a fruitful area of research as it may increase the framework's robustness to any errors the Binary Classifier may make. This increased robustness may likely result in greater framework accuracy and performance.

In terms of practicality, this model requires a very specific type of dataset for training and testing. Word-by-word audio corpora are not common. In addition, in a live, deployed scenario, this model would not function properly on a full audio file containing a sentence. Any conversation between a pilot and ATCO would need to be split word by word for the model to transcribe it. This is far too time consuming for a human to manually do for every sentence that each party speaks. However, a future area of research is to train a segmentation algorithm which would split up the audio files automatically into word-by-word segments. The output of this segmentation algorithm would then be fed into the transcription framework and the ATC audio could be transcribed.

References

[1] N. J. Nilsson, *The Quest for Artificial Intelligence*. 2009. doi: 10.1017/cbo9780511819346.

[2] T. Hwang, "Computational Power and the Social Impact of Artificial Intelligence," *SSRN Electronic Journal*, 2018, doi: 10.2139/ssrn.3147971.

[3] K. Sekaran, P. Chandana, J. R. V. Jeny, M. N. Meqdad, and S. Kadry, "Design of optimal search engine using text summarization through artificial intelligence techniques," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 18, no. 3, 2020, doi: 10.12928/TELKOMNIKA.v18i3.14028.

[4] J. W. Kim, B. H. Lee, M. J. Shaw, H. L. Chang, and M. Nelson, "Application of decision-tree induction techniques to personalized advertisements on internet storefronts," *International Journal of Electronic Commerce*, vol. 5, no. 3, 2001, doi: 10.1080/10864415.2001.11044215.

[5] J. Zhao, B. Liang, and Q. Chen, "The key technology toward the self-driving car," *International Journal of Intelligent Unmanned Systems*, vol. 6, no. 1. 2018. Doi: 10.1108/IJIUS-08-2017-0008.

[6] Khaled M. Alhawiti, "Advances in Artificial Intelligence Using Speech Recognition," *Fundamentals of Speaker Recognition*, vol. 9, no. 6, 2015.

[7] R. Verganti, L. Vendraminelli, and M. Iansiti, "Innovation and Design in the Age of Artificial Intelligence," *Journal of Product Innovation Management*, vol. 37, no. 3, 2020, doi: 10.1111/jpim.12523.

[8] T. Kumar and M. Trakru, "the Colossal Impact of Artificial Intelligence in E - Commerce : Statistics and Facts," *International Research Journal of Engineering and Technology*, vol. 570, no. May, 2019.

[9] G. Terzopoulos and M. Satratzemi, "Voice assistants and smart speakers in everyday life and in education," *Informatics in Education*, vol. 19, no. 3, 2020, doi: 10.15388/infedu.2020.21.

[10] R. Kheir and T. Way, "Inclusion of deaf students in computer science classes using real-time speech transcription," *ACM SIGCSE Bulletin*, vol. 39, no. 3, 2007, doi: 10.1145/1269900.1268860.

[11] M. B. Hoy, "Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants," *Med Ref Serv Q*, vol. 37, no. 1, 2018, doi: 10.1080/02763869.2018.1404391.

[12] S. Badrinath and H. Balakrishnan, "Automatic Speech Recognition for Air Traffic Control Communications," *Transportation Research Record: Journal of the*

Transportation Research Board, vol. 2676, no. 1, 2022, doi: 10.1177/03611981211036359.

[13] G. Azam and M. T. Islam, "Design and Fabrication of a Voice Controlled Wheelchair for Physically Disabled People," *International Conference on Physics Sustainable Development & Technology (ICPSDT-2015)*, no. October, 2015.

[14] B. H. Juang and L. R. Rabiner, "Automatic Speech Recognition – A Brief History of the Technology Development," *Elsevier Encyclopedia of Language and Linguistics*, vol. 50, no. 2, 2004.

[15] J. A. Updegrove and S. Jafer, "Optimization of air traffic control training at the federal aviation administration academy," *Aerospace*, vol. 4, no. 4, 2017, doi: 10.3390/aerospace4040050.

[16] G. Li, S. P. Baker, J. G. Grabowski, and G. W. Rebok, "Factors associated with pilot error in aviation crashes," *Aviat Space Environ Med*, vol. 72, no. 1, 2001.

[17] G. H. Millman, "Atmospheric Effects on VHF and UHF Propagation," *Proceedings of the IRE*, vol. 46, no. 8, 1958, doi: 10.1109/JRPROC.1958.286970.

[18] J. Berg, C. Bustad, L. Jonsson, L. Mossberg, and D. Nyberg, "Perceived audio quality of realistic FM and DAB+ radio broadcasting systems," *AES: Journal of the Audio Engineering Society*, vol. 61, no. 10, 2013.

[19] H. A. Whale, *Effects of Ionospheric Scattering on Very-Long-Distance Radio Communication*. 1969. doi: 10.1007/978-1-4899-6545-5.

[20] "Noise reduction with multiple microphones: A unified treatment," in *Springer Topics in Signal Processing*, vol. 1, 2008. doi: 10.1007/978-3-540-78612-2_5.

[21] S. Alharbi *et al.*, "Automatic Speech Recognition: Systematic Literature Review," *IEEE Access*, vol. 9. 2021. doi: 10.1109/ACCESS.2021.3112535.

[22] Y. Lin, L. Deng, Z. Chen, X. Wu, J. Zhang, and B. Yang, "A Real-Time ATC Safety Monitoring Framework Using a Deep Learning Approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, 2020, doi: 10.1109/TITS.2019.2940992.

[23] V. N. Nguyen and H. Holone, "Possibilities, Challenges and the State of the Art of Automatic Speech Recognition in Air Traffic Control," *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 9, no. 8, pp. 1916–1925, 2015.

[24] B. Yang *et al.*, "ATCSpeech: A multilingual pilot-controller speech corpus from real air traffic control environment," in *Proceedings of the Annual Conference of*

the International Speech Communication Association, INTERSPEECH, 2020, vol. 2020-October. doi: 10.21437/Interspeech.2020-1020.

[25] Y. Lin *et al.*, "ATCSpeechNet: A multilingual end-to-end speech recognition framework for air traffic control systems," *Appl Soft Comput*, vol. 112, 2021, doi: 10.1016/j.asoc.2021.107847.

[26] K. Zhou, Q. Yang, X. S. Sun, S. H. Liu, and J. J. Lu, "Improved CTC-Attention Based End-to-End Speech Recognition on Air Traffic Control," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11936 LNCS. doi: 10.1007/978-3-030-36204-1_15.

[27] A. V. Haridas, R. Marimuthu, and V. G. Sivakumar, "A critical review and analysis on techniques of speech recognition: The road ahead," *International Journal of Knowledge-Based and Intelligent Engineering Systems*, vol. 22, no. 1, 2018, doi: 10.3233/KES-180374.

[28] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing – Proceedings*, 2015, vol. 2015-August. Doi: 10.1109/ICASSP.2015.7178964.

[29] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Estève, "TED-LIUM 3: Twice as Much Data and Corpus Repartition for Experiments on Speaker Adaptation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11096 LNAI. doi: 10.1007/978-3-319-99579-3_21.

[30] Y. Lin, Q. Li, B. Yang, Z. Yan, H. Tan, and Z. Chen, "Improving speech recognition models with small samples for air traffic control systems," *Neurocomputing*, vol. 445, 2021, doi: 10.1016/j.neucom.2020.08.092.

[31] Y. Lin, D. Guo, J. Zhang, Z. Chen, and B. Yang, "A Unified Framework for Multilingual Speech Recognition in Air Traffic Control Systems," *IEEE Trans Neural Netw Learn Syst*, vol. 32, no. 8, 2021, doi: 10.1109/TNNLS.2020.3015830.

[32] S. Ali, S. Tanweer, S. Khalid, and N. Rao, "Mel Frequency Cepstral Coefficient: A Review," 2021. doi: 10.4108/eai.27-2-2020.2303173.

[33] T. Fu and X. Wu, "Multi-scale feature based convolutional neural networks for large vocabulary speech recognition," 2017. doi: 10.1109/ICME.2017.8019385.

[34] J. C. Alderson, "Air safety, language assessment policy, and policy implementation: The case of aviation english," *Annual Review of Applied Linguistics*, vol. 29. 2009. doi: 10.1017/S0267190509090138.

[35] P. Fan, D. Guo, Y. Lin, B. Yang, and J. Zhang, “Speech recognition for air traffic control via feature learning and end-to-end training,” *IEEE ICASSP 2022*, Nov. 2021.

[36] N. Hou, X. Tian, E. S. Chng, B. Ma, and H. Li, “Improving air traffic control speech intelligibility by reducing speaking rate effectively,” in *Proceedings of the 2017 International Conference on Asian Language Processing, IALP 2017*, 2018, vol. 2018-January. doi: 10.1109/IALP.2017.8300578.

[37] ICAO 9835, “Manual on the Implementation of ICAO Language Proficiency Requirements,” *Strategies*, 2010.

[38] I. Cohen and B. Berdugo, “Noise estimation by minima controlled recursive averaging for robust speech enhancement,” *IEEE Signal Process Lett*, vol. 9, no. 1, 2002, doi: 10.1109/97.988717.

[39] I. Cohen, “Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 5, 2003, doi: 10.1109/TSA.2003.811544.

[40] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, “Perceptual evaluation of speech quality (PESQ) - A new method for speech quality assessment of telephone networks and codecs,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2001, vol. 2. doi: 10.1109/icassp.2001.941023.

[41] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, “A short-time objective intelligibility measure for time-frequency weighted noisy speech,” 2010. doi: 10.1109/ICASSP.2010.5495701.

[42] J. J. Godfrey, “Air Traffic Control Complete,” *1-58563-024-1*, 1994.

[43] R. Martin, “Speech enhancement based on minimum mean-square error estimation and supergaussian priors,” *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, 2005, doi: 10.1109/TSA.2005.851927.

[44] G. Doblinger, “Computationally efficient speech enhancement by spectral minima tracking in subbands,” *Proceedings of EUROSPEECH*, vol. 2, 1995.

[45] K. V. Sørensen and S. V. Andersen, “Speech enhancement with natural sounding residual noise based on connected time-frequency speech presence regions,” *EURASIP J Appl Signal Processing*, vol. 2005, no. 18, 2005, doi: 10.1155/ASP.2005.2954.

[46] C. H. YOU and B. MA, “Spectral-domain speech enhancement for speech recognition,” *Speech Commun*, vol. 94, pp. 30–41, Nov. 2017, doi: 10.1016/j.specom.2017.08.007.

[47] P. C. Loizou, “Speech Databases and {MATLAB} codec,” in *Speech Enhancement Theory and Practice*, 2007.

[48] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, “An algorithm for intelligibility prediction of time-frequency weighted noisy speech,” *IEEE Trans Audio Speech Lang Process*, vol. 19, no. 7, 2011, doi: 10.1109/TASL.2011.2114881.

[49] B. Zhang, J. Leitner, and S. Thornton, “Audio Recognition using Mel Spectrograms and Convolution Neural Networks,” 2019.

[50] Y. Zeng, H. Mao, D. Peng, and Z. Yi, “Spectrogram based multi-task audio classification,” *Multimed Tools Appl*, vol. 78, no. 3, 2019, doi: 10.1007/s11042-017-5539-3.

[51] F. Sultana, A. Sufian, and P. Dutta, “Advancements in image classification using convolutional neural network,” 2018. doi: 10.1109/ICRCICN.2018.8718718.

[52] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *Advances in Neural Information Processing Systems*, 2020, vol. 2020-December.

Appendix A

Glossary of Terms

1. ATC: Air Traffic Control
2. ATCO: Air Traffic Controller
3. ICAO: International Civil Aviation Organization
4. MCRA: Minima Controlled Recursive Averaging [38]
5. MCRA2: Minima Controlled Recursive Averaging 2 [39]
6. PESQ: Perceptual Evaluation of Speech Quality [40]
7. STOI: Short Term Objective Intelligibility [41]