A Long Short-term Memory Neural Network for Improved Twins' Voice Differentiation

Stallone Bruno-Ray Sabatier

Follow this and additional works at: https://researchrepository.wvu.edu/etd

Recommended Citation

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.
A Long Short-Term Memory Neural Network for Improved Twins’ Voice Differentiation

Stallone Bruno-Ray Sabatier

Thesis submitted to the
Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Jeremy M. Dawson, Ph.D., Chair
Nasser Nasrabadi, Ph.D.
Thirimachos Bourlai, Ph.D.

Lane Department of Computer Science and Electrical Engineering
Morgantown, WV
2018

Keywords: Speaker Recognition, Neural Networks, i-vector, Identical Twins, Long Short-Term Memory

Copyright © 2018 Stallone Bruno-Ray Sabatier
Abstract

A Long Short-Term Memory Neural Network for Improved Twins’ Voice Differentiation

Stallone Bruno-Ray Sabatier

Although successfully implemented in certain situations, the reliability of speaker recognition (SR) decreases due to speaker and channel variability present between enrollment and evaluation samples, as well as the available length of speech utterances. The issue of speaker variability becomes more pronounced when the number of speakers in the evaluation set increases, as there is a higher probability that two voices may sound more alike. The differentiation of intra-twin pairs’ voices can be beneficial to general SR because inclusion can simulate this effect. This is due to the fact that their shared voice similarities are comparable to the potential inter-speaker similarities one would expect in a large database of enrolled speakers. Furthermore, twin occurrence has steadily increased over the past thirty years. With these considerations in mind, there have been few research efforts analyzing the impact of identical twins on SR, and they have been lacking in terms of the corpus of individuals sampled and/or the technology employed.

In this research effort, a recurrent neural network that specializes in processing time series data, specifically the long short-term memory (LSTM) network, is evaluated on a large corpus of identical twins’ speech collected over two years with multiple speaking modes. The LSTM’s recurrent capability enables the exploitation of higher level speech features which are hypothesized to be more variant between identical twins. The LSTM is configured as a single network, and in a Siamese fashion, to evaluate the performance of varied utterance lengths and speech features. Matching results are analyzed and discussed in comparison to state of the art i-vector methodologies. Results in terms of the equal error rate (EER) indicate that, as the length of the enrollment and test utterances are reduced, the LSTM outperforms the i-vector system by 17% for two seconds of speech data. Of the three speech features investigated in the Siamese configuration, mel-frequency spectral coefficients resulted in the highest rate of twin voice differentiation with an EER of 8.57% for six seconds of data. Lastly, in comparison to other twins’ voice studies, the introduction of more individuals degrades performance with respect to male speakers from nearly 0% to 0.598% EER. However, the results from these experiments are far less than female trials in other studies.
Acknowledgements

I would like to thank my research advisor, Dr. Jeremy Dawson, for giving me the opportunity to be a graduate research assistant and experiencing the full research process from data collection to algorithm implementation. Dr. Dawson has given his patience, encouragement, trust and guidance which allowed me the liberty of discovering exciting and novel machine learning methods. I would also like to thank Dr. Nasser Nasrabadi for sharing his expertise in neural networks and for the beneficial and lively discussions on the topic. I also give my appreciation to Dr. Thirimachos Bourlai for the valuable technical feedback and teaching me the fundamentals of pattern recognition.

I dedicate this work to my family starting with my mother, Lela, my step-father, Chris, my father, Bruno and my grandmother and grandfather, Rosemary and Edward, for their love, patience and continued support in making this research effort a reality.

I thank WVU faculty for helping me through my graduate career as well as my colleagues, Amirsina Torfi, Morgan Trester, Matthew Pachol, Seth Leffel, Sobhan Soleymani, Domenick Poster, Fariborz Taherkhani, Veeru Talreja, Ali Dabouei, Douglas Kerr, Casey Norville, Christo Robison, and all the students from AERB 306 for taking this journey together and having fun along the way. Without all of you, this work would have not been made possible.
Table of Contents

List of Figures ........................................................................................................................................ vi
List of Tables ........................................................................................................................................ vii

Chapter 1: Introduction .......................................................................................................................... 1
  1.1 Motivation and Objectives ................................................................................................................. 2
    1.1.1 Thesis Contribution ..................................................................................................................... 4
  1.2 Overview of Monozygotic Twins ......................................................................................................... 5
    1.2.1 Biometrics and Twins .................................................................................................................. 7
  1.3 The Speech Signal ............................................................................................................................... 8
    1.3.1 Speech Fundamentals ................................................................................................................ 8
    1.3.2 Speech Features .......................................................................................................................... 11
  1.4 Speaker Recognition ........................................................................................................................... 14
    1.4.1 Speaker Verification ................................................................................................................. 15
    1.4.2 Voice Studies on Twins ........................................................................................................... 16
    1.4.3 Other Issues within Speaker Recognition ............................................................................... 22
  1.5 Neural Networks ............................................................................................................................... 23
    1.5.1 Brief History and Fundamentals of Neural Networks ............................................................... 23
    1.5.2 Recurrent Neural Networks .................................................................................................... 26
  1.6 Research Goals .................................................................................................................................. 30

Chapter 2: Background ............................................................................................................................ 32
  2.1 Speech Feature Extraction ................................................................................................................. 33
    2.1.1 Mel-frequency Cepstral Coefficients ....................................................................................... 33
    2.1.2 Other Low-Level Features: Jitter and Shimmer ..................................................................... 36
    2.1.3 High-Level Features ............................................................................................................... 37
  2.2 Speaker Verification ........................................................................................................................... 39
    2.2.1 Protocol ..................................................................................................................................... 40
  2.3 i-vector .............................................................................................................................................. 41
  2.4 Neural Network Information Flow .................................................................................................... 44
    2.4.1 Forward Pass ............................................................................................................................ 44
    2.4.2 Backward Pass .......................................................................................................................... 47
  2.5 Recurrent Neural Network Information Flow .................................................................................... 50
    2.5.1 Long Short-Term Memory Network Information Flow ............................................................... 51
  2.6 Siamese Networks .............................................................................................................................. 55
Chapter 3: Methodology ........................................................................................................57
  3.1 Datasets ..........................................................................................................................58
    3.1.1 Collection ................................................................................................................58
    3.1.2 Post-processing .......................................................................................................59
  3.2 LSTM and i-vector Comparison ....................................................................................60
    3.2.1 Feature Extraction .................................................................................................61
    3.2.2 Baseline i-vector Algorithm ..................................................................................62
    3.2.3 LSTM Algorithm ...................................................................................................62
  3.3 Twin Voice Differentiation with Siamese LSTM .........................................................66
    3.3.1 Feature Extraction .................................................................................................66
    3.3.2 Siamese LSTM Architecture ................................................................................67

Chapter 4: Results and Discussion ..................................................................................69
  4.1 LSTM Architecture Optimization ................................................................................70
    4.1.1 Analysis of Number of Layers and Units ...............................................................70
  4.2 LSTM and Baseline Comparison ................................................................................71
    4.2.1 Effect of Type of Speech Data on Recognition Performance ...............................71
    4.2.2 Do twins pose a problem to speaker verification? ...............................................73
    4.2.3 Effect of Utterance Length ....................................................................................73
  4.3 Twin Voice Differentiation with Siamese LSTM Results ............................................75
    4.3.1 Comparison of Different Types of Speech Features in Twin Voice Separation ....75

Chapter 5: Conclusion ......................................................................................................79
  5.1 Summary of Results ....................................................................................................80
  5.2 Future Directions .........................................................................................................81

References ..........................................................................................................................82
List of Figures

Figure 1.1. Rate of Twin Births Between 1980 and 2009 [4] ................................................................. 6
Figure 1.2. Flow Diagram of Vocal Production, Spectrograms of Phoneme Resonances, Phonemes and Words
Red lines indicate formants. Image by Ish Ishwar, distributed under a CC-BY 2.0 license .............................. 10
Figure 1.3. Summary of Various Speech Features with Benefits and Drawbacks as well as Perceptual Cues for Humans ......................................................................................................................... 14
Figure 1.4. Block Diagram of a Typical Speaker Verification or Identification System, adapted from [34] .......... 16
Figure 1.5. Artificial Neuron and Biological Neuron Anatomical Comparison. Blue Lines Indicate Analogous Components ......................................................................................................................................... 24
Figure 1.6. Multi-Layer Perceptron Topology with I input units, J hidden neurons and K output neurons. Wij and Wjk are the weight matrices for the input-hidden and hidden-output layers, respectively ........................................ 25
Figure 1.7. Unfurling of an RNN with mapping of input vectors xt to hidden states ht ..................................... 27
Figure 1.8. Computational Graph of a Canonical LSTM Network with Hidden State Vectors ht and Input Vectors xt ............................................................................................................................................ 29

Figure 2.1. Mel-frequency spaced filter banks. Filters are linearly spaced below 1000 Hz and logarithmically spaced above 1000 Hz .................................................................................................................. 36
Figure 2.2. Hypothetical Speaker Score Distributions in Verification Scenario. Equal Error Rate (EER) is the point of intersection of Distributions .................................................................................................. 40
Figure 2.3. Example loss function curve of a hypothetical NN in terms of two weights w1 and w2 on the x and y axis, respectively, and the associated loss on the z-axis. The x and y coordinates represent values of the two weights, and the height represents the error of the training set when evaluated at each weight setting [73] ...... 46
Figure 2.4. Simple RNN structure showing weight matrices of input, hidden and output layers. Orange loops indicate recurrent connections ........................................................................................................... 51
Figure 2.5. Internal View of Memory Block of LSTM ................................................................................... 52
Figure 2.6. Siamese Neural Network Architecture with Sub-Nets O(X1) and O(X2), Shared Weights W and Contrastive Cost Function Scoring Module ............................................................................. 55

Figure 3.1. Internal Diagram of WhisperRoom used to collect speech data ................................................... 59
Figure 3.2. Protocol for Verification for NN Architecture .............................................................................. 64
Figure 3.3. Proposed LSTM Architecture. Layer sizes indicate the range of units investigated in each layer. Each yellow, black and white circle represents a Memory Block, Input Neuron, and Standard Neuron, Respectively ................................................................................................................................. 65

Figure 4.1. EERs for Multiple LSTM Layers and Units for Experiment 1 on Condition 1b ............................ 70
Figure 4.2. Box and Whisker Plots for Various Speech Features and Lengths in a Siamese LSTM Network ...... 77
Figure 4.3. Input and Output Distributions for the MFEC features for six seconds of speech using the Siamese LSTM network .......................................................................................................................... 78
List of Tables

Table 1. 1. Summary of Recent Automatic Speaker Recognition Research on Identical Twin Voices ...............22
Table 3.1. Combined Amount of Read and Conversational Speech for All Devices in Datasets 1 and 2 ............60
Table 3.2. Amount of Read and Conversational Speech for Each Device for Dataset 3 ..................................60
Table 3.3. Speaking Modes for Each Verification Trial ....................................................................................61
Table 3.4. Speaker Feature Map Sizes for Experiment 2 (Timesteps x Feature Length) .................................67
Table 4.1. LSTM and i-vector EER Comparisons for Multiple Train-Test Scenarios with Twin Miss Percentage Rate ..................................................................................................................................................72
Table 4.2. EERs for i-vector and LSTM in Matched Short Utterance Scenarios for Condition 2 ..................74
Table 4.3. EERs for Genuine/Imposter Pair Comparison in Siamese Network for Various Lengths and Features ........................................................................................................................................................76
Chapter 1:
Introduction
1.1 Motivation and Objectives

Biometrics is the process of recognizing individuals based on their unique physical and idiosyncratic traits. Its application has permeated throughout our society in a variety of ways, including fingerprints replacing passwords for access to cell phones and facial recognition aiding border control and expediting check-in procedure at the airport. At this very moment, people are authorizing financial transactions with voice commands over the phone using their bank’s automated services. In fact, voice biometric authentication systems are becoming increasingly popular due to low-cost of implementation, a high degree of acceptance by users, and the ability for the user to remotely access privileged services and data. From 2016 to 2021, the global market for voice recognition technologies will have a compound annual growth rate of 12.1% resulting in a projected increase of $80 billion over this 5-year time period [1].

The commercial sector is the most recent application area to experience the benefits of voice technology. Speaker recognition (SRE) techniques have historically been used in forensic applications, being used as a tool to aid in investigations and to provide supporting evidence in a court of law. A notable example of the former was the successful identification of “Jihadi John” in the 2014 slaying of James Foley, enabled by audio capture of the perpetrator’s voice along with other biological traits [2].

State of the art SRE using i-vectors [3] achieves an equal error rate (EER) of 4.64% with noise and 2.37% without noise in recent NIST evaluations composed of 273 female speakers with telephone speech. Although successfully implemented in certain situations, the reliability of SRE decreases due to speaker and channel variability present between enrollment and evaluation samples such as voice aging, environmental noise, recording channel mismatch and the available length of speech utterances. The issue of speaker variability becomes more pronounced when the number of speakers in the evaluation set increases as there is a higher probability that two voices may sound more alike. The differentiation of intra-twin pairs’ voices can be beneficial to general SRE because their inclusion can simulate this effect since their shared voice similarities are comparable to potential inter-speaker similarities one would
expect in a large database of enrolled speakers. Furthermore, twin occurrence, has steadily increased over the past thirty years [4]. As voice biometrics become more prevalent, and more people become enrolled, these systems need to be robust to a wide array of situations and individuals.

Short utterance speaker recognition (SUSR) is becoming an increasingly important sub-field of SRE. This is mainly motivated by its convenience for the end user in commercial applications as well as improving the recognition performance of forensic style speech utterances which are commonly less than a few seconds of duration. Even state of the art algorithms’ performance degrades drastically as the length of utterances are reduced. This drawback has spawned research in several compensation techniques [5-6] yet results have been less than satisfactory.

Bearing these considerations in mind, there have been few research efforts analyzing the impact of identical twins on SRE, but they have been lacking in terms of the corpus and/or the technology employed. Twin voice datasets with more than 100 individuals are virtually non-existent, and the small datasets that are available do not include real world conditions such as noise, channel mismatch and different speaking modes, as well as the addition of fraternal and non-twin siblings [7]. Algorithmically, traditional SRE systems operate on low-level, short-term spectral features that approximate the speech anatomy and associated vocal resonance of an individual. Therefore, spectral components of speech are heavily influenced by genetic factors. However, human speech also possesses learned, behavioral components, so an opportunity exists to investigate the relationships between anatomical and behavioral components using high-level speech features that may have more variation between twins as they grow older and exhibit independence from one another. These features, however, occur over a larger window of time that traditional systems cannot faithfully consider as the temporal dependence between windows is lost. Research efforts employed using traditional methods, such as the Gaussian Mixed Model Universal Background Model (GMM-UBM) [8] and i-vector [3] are not able to consider longer contextual speech features since they only accept short timeframe vector representations. Furthermore,
these methods are developed with a dataset inconsistent of the type of speech data that would be encountered during the enrollment and test phases. Due to the scarcity and limitations of twins’ datasets, proper development of these methods would reduce the number of subjects able to be tested.

Recent advances in pattern recognition and machine learning have been enabled through the resurgence of biologically-inspired artificial neural networks (ANNs). These systems are crudely modeled on the operation of the human brain, and operate by tuning one or more input parameters to achieve a desired output. They exist in a variety of architectures which are often specialized for specific applications. For time-dependent data such as video and speech, recurrent neural networks (RNNs) are well suited. This type of network often provides the best performance since learning is a function of all previous inputs effectively creating a memory. A special type of RNN called the Long Short-Term Memory (LSTM) Network [9] has shown promising results in automatic speech recognition (ASR) [10] and in text-dependent speaker verification [11] but has yet to be adequately explored in a text-independent setting.

1.1.1 Thesis Contribution

In a field that relies on the inherent physical differences of individuals to perform accurate biometric recognition, the main contribution of this research effort is to design and implement an ANN capable of discriminating individuals with the most extreme class of similarity—identical twins. This will be done by implementing an LSTM as well as a Siamese LSTM network specializing in differentiating highly-similar classes and processing longer sequential data. The latter capability enables the direct exploitation of different types of speech features that are advantageous to achieving improved twin differentiation. The network will be trained in a generative fashion from the inherent differences existing between many comparisons of co-twins’ voices. Thus, the criteria for highly similar voice differentiation will be determined heuristically from voice sample precedents and not pre-configured or pre-decreed by the
network designer. To the best of the author’s knowledge, a Siamese LSTM has not been implemented in the context of speaker verification, let alone focusing on identical twins. Results from the proposed method will be compared with a baseline i-vector approach.

These investigations are made possible by valuable datasets of a large corpus of identical and fraternal twins’ speech collected over four years. The present dataset is the largest known dataset of twins’ speech ever collected and allows these issues to be investigated more than any previous extent. Other types of speech features will also be evaluated alongside traditional features to determine their viability in SRE in the context of twin voices in a state of the art RNN.

From a larger scope, the study of identical twins also yields two other valuable contributions:
1.) The elucidation of the nature vs. nurture argument because heritability’s impact on biometric modalities can be ascertained with the performance comparison between subjects of varying degrees of familial relation. For example, monozygotic and dizygotic twins. With the genetic component fixed, environmental influences may illuminate a path in understanding the age old posit of nature vs. nature.
2.) Identical twins can serve as the ultimate stress test to the capabilities of biometric technologies. If such a system can acceptably perform accurate differentiation on these types of individuals then it is presumed that the system should separate non-related individuals with greater ease.

1.2 Overview of Monozygotic Twins

1.2.1 Development and Occurrence

The conception and pre-natal growth process of twins is not unique to humans but is nevertheless uncommon. During the fertilization period, a female may release one or multiple oocytes (eggs) from the ovaries which become fertilized by a male gamete (sperm) resulting in a zygote. In the event a female releases multiple oocytes, there is a possibility that separate gametes could fertilize each oocyte resulting
in dizygotic offspring who are not identical and only share 25% of their genes. The resulting offspring are commonly referred to as fraternal, non-identical, or dizygotic (DZ) twins and may not necessarily be the same gender. In the event that the zygote cleavages into two, monozygotic (MZ) twins are produced with 100% of the same genetic makeup. It is commonly believed that identical siblings develop when the matured zygote, now an embryo, splits four to six days after fertilization. If this milestone is delayed, even by two days, complete separation of the embryos is unlikely causing conjoined or Siamese twins [12].

Twin births are becoming less and less rare. From 1980 to 2009, there was a 76% increase in twin births due to artificial reproductive technology enabling more women over the age of 40 to conceive. This fostered a 200% increase in twin birth rate [4]. Currently, four out of every 1000 births results in identical twins and two-thirds of all twin births are fraternal. The rate of fraternal twinning also depends on ethnicity with Asians having an approximate 0.3% incidence and people of African descent having as much 1-4% [13]. Figure 1.1 displays the increasing trend of twin births in America during the past thirty years.

Figure 1.1. Rate of Twin Births Between 1980 and 2009 [4]
1.2.1 Biometrics and Twins

Identical twins are a phenomenon that have interested researchers in many fields, not only for their rarity and exoticness, but in also understanding the contribution of organic and learned traits in biological organisms. From birth, identical twins exhibit the highest degree of similarity among humans. However, as they get older and their lives diverge from one another, environmental influences may begin to play a role in their development. For many years, identical twins were unable to be differentiated by DNA sequencing. Only until recently the method of “ultra-deep next generation sequencing” has been successful as shown in the resolution of a paternity dispute among two identical twins [14]. This method derives its ability from more discerning random genetic mutations that occur over time.

For a measure of a biological trait to be used as a biometric, it must satisfy these certain criteria among others:

- **Permanence** -- the trait should not change with time
- **Acceptability** -- it should not be overly cumbersome for the subject to provide a sample
- **Distinctiveness** -- the characteristics of the trait should serve to uniquely identify that individual

For this last reason, there has been much investigation into identical twins’ impact on biometric technologies as well as a potential remedy through more robust methods of discrimination. These methods include utilizing nurture components which may offer more opportunity in divergence as they are related to environmental factors. For example, in facial recognition, Le et al [15] used age associated features such as “crow’s feet” around the eyes, nose and mouth with a 96% success rate in twin identification. Moreover, Mahalingam and Ricanek [16] have shown that automatic twin facial differentiation accuracy improves with increasing age. In another facial study, Paone et al [17] evaluated seven face matchers on a dataset of 126 pairs of identical twins and found that performance was consistently lower than non-related individuals under various imaging conditions. The trend continues
with analysis of twins’ fingerprints by Liu and Srihari [18] who discovered decreasing fingerprint ridge pattern similarity of 188 pairs of identical, 39 pairs of fraternal, and other non-twins with equal error rates (EERs) of 5.09%, 4.88% and 3.33%, respectively.

In terms of the vocal characteristics of twins, Ryalls et al [19] measured voice onset time (VOT) between two pairs of twins aged 21 and 70. The seventy-year old twins lived separate lives on the North and Southeast coast of America while the 21-year old twins remained together. They found that the older pair had much larger VOTs, corresponding to their respective regional dialects, and concluded that vocal source features are more genetically influenced than vocal tract filtering. A comprehensive survey of biometric studies on identical twins can be found in [20]. Of all the modalities examined, face has received the most attention as the datasets available are the largest and most standardized. There still exists a need for a multi-modal dataset containing MZ and DZ twins to further advance research in twin biometrics since the amount of data for other modalities is severely lacking compared to face. The largest twin face dataset is composed of 1902 faces scraped from the web. On the other hand, the largest voice dataset contains less than 50 pairs of twins.

1.3 The Speech Signal

A short description of the mechanisms and factors producing speech as well as its perception by humans is provided in this section. The complex, underlying processes are important in understanding the motivation behind subsequent speech signal processing techniques.

1.3.1 Speech Fundamentals

Human speech is rich in information as it conveys many different aspects of semantic and identifying content. From it, one can determine what is being said, how it is being said, and ultimately, who is saying it. These speech cues create the unique voice print of an individual. Humans can use these types of
hierarchical speech features to instantaneously decipher the identity of the speaking individual. However, for computers this is a very difficult task with far-reaching significance as it represents one of the last major hurdles to fluid interaction between human and machine. This emerging field of study is referred to as natural language processing (NLP) and encompasses emotion and semantic understanding or broadly, speech recognition and generation.

Speech is produced by the phonated excitation of the vocal cords followed by a filtering of the vocal tract and other articulators such as the lips, tongue, teeth and nose [21]. When air is pushed by the vocal cords it passes through oscillating glottal folds creating a certain pitch. The faster the folds oscillate the higher the pitch of the modulated sound and vice versa. The natural, unaltered frequency of oscillation is called the fundamental frequency ($f_0$) and depends on the length, tension and mass of the glottal folds [22]. The subsequent filtering of the vocal tract creates a resonance that corresponds to a particular spectral peak, or in linguistic terms, a formant being produced.

Spoken languages consist of a lexicon of words which can be further decomposed into sequences of discrete sounds called phonemes. For example, the word “twin” can be split into the phonemes “tuh-whi-nuh”. A combination of formants, with the lowest frequency formant denoted by F1 and so on, create the unique acoustic signature of a phoneme, which sequentially combined with other phonemes, creates words. The extent to which the vocal tract and other articulators affect the production of phonemes varies for consonants and vowels [21]. Figure 1.2 shows a flow diagram of the vocal production anatomy with the dark spectral peaks in the spectrogram of the vocal tract resonance corresponding to the formants. A combination of formants creates the linguistic building blocks or phonemes which, in turn, create a speaker’s lexicon.
The semantic content is not the only information that can be gleaned from a speech signal. The spoken dialect may be captured spectrally as a systematic shift in formant frequencies [23]. In other words, a phoneme may be characteristically modified revealing the idiosyncrasy of an individual. For instance, the spectrogram of the “o” vowel utterance for a person raised in West Virginia may have a downward sloping spectral trajectory of the F1 frequency compared to its canonical form.

Our speech is a unique, complex signal that is not only dependent on the physical characteristics of our anatomy but also the intricate movement and coloring thereof resulting from environmental factors such as socio-economic status, living region, personality type, education and habits. Since twins live in close proximity to each other throughout most of their childhood and adolescence, the learned component of speech is expected to be highly similar among twins as well as non-twin siblings. This is evidenced by Charlet and Lecha [24] who studied text-dependent ASR with 33 families and discovered that the brothers were often misidentified with each other.
However, there is also some evidence that personal speech behavior is phonetically different, even between identical twin pairs [7]. In an acoustic phonetic analysis of three pairs of identical twins, Nolan and Oh [25] found phonetic differences of different type and size in all three, which led them to the conclusion, “…that identical twins are not necessarily phonetically identical and that they make use of the leeway allowed them by the phonological system.” This hypothesis of idiosyncratic variation in speech production is supported by Johnson and Azara [26]. In a series of listening experiments with identical twins as well as unrelated speakers they found that twins’ voices can be distinguished significantly but are more often confused than unrelated speakers and that, “analysis of the perceptual space for talkers showed that the difference between identical twins was in some cases as large as the difference between unrelated talkers.”

There is also a third element contributing to twins’ voices that is often overlooked in studies involving twins. Epigenetics is the study of how gene expression, other than the underlying DNA sequence, may change throughout time which could lead to further dissimilarities between biologically related individuals. This is corroborated by Fraga et al [27] who concluded, “although twins are epigenetically indistinguishable during the early years of life, older MZ twins exhibited remarkable differences in their overall content and genomic distribution… affecting their gene-expression portrait.”

1.3.2 Speech Features

Humans tend to use several linguistic levels like lexicon, prosody or phonetics to recognize others by their voice. These levels of information are more related to learned habits or style, and they are mainly manifested in the dialect, sociolect or idiolect of the speaker.

In general, there are three sources of inter-speaker variation: the vocal tract, vocal cords and speaking style. Unique audio clues about the speaker manifest themselves in a variety of ways and can be classified on a spectrum of high to low level features as shown in Figure 3. High level features such
as the speaker’s lexicon are learned speaking habits whereas low-level features such as fundamental voice frequency are related to the physical characteristics of the speaker. Typically, low-level short term spectral features are computed over 20-30 ms windows and correlate to the timbre, i.e. the color of the speech. Mid-level features such as prosody include characteristics such as intonation, rhythm and speaking rate as well as dialect. Prosodic parameters are mainly manifested as sound duration, selection and intensity variation in syllables, words and phrases. To accurately describe these features, one must enlarge the window of speech to hundreds of milliseconds or even to the scope of syllables and words which can occupy a second or more. For this reason, they are also known as supra-segmental features and from them, the accent, emotion and even language background can be ascertained. Unfortunately, these features are more susceptible to impersonation and mimicry as they are based on learned behaviors. High-level features focus on conversation level characteristics such as word choice, semantics and sentence formulation and may occupy the scope of a second or several. Common methods of modeling are N-gram modeling that find the probability or occurrence that N number of phonemes or words are detected in succession.

The most important prosodic feature is the $f_0$ and it stems from both organic and learned factors. As an example, the value of $f_0$ correlates to the size of the larynx [28] while the temporal variation in $f_0$ can be tied to the person’s unique behavior of speech. Appending this feature with low level spectral features has shown to provide improvement in SRE. Other types of prosodic features that have been implemented include pause statistics, phone duration and energy modulations [29-31]. However, it was found by [30] that $f_0$ related features gave the highest performance. Some studies [32-33] have analyzed this type of feature within the context of twins and have found that the mean and variability of the $f_0$ and the envelope of the voice spectrum above the first formant are more similar among identical twins than fraternal.
According to [22], an ideal speech feature should satisfy the following list of criteria:

- Low intra-speaker variability
- High inter-speaker variability
- Difficulty for impersonation/mimicry
- Natural and occur often in speech
- Exhibit robustness against noise and distortion
- Not perturbed by aging or health of speaker

Traditionally, low-level features have been the most prominent in the field because they maximize the first three criteria listed above due to the fact they model the unique anatomy of a speaker’s vocal production system and are more difficult to impersonate. Hence, the more similar the vocal tracts, the more similar the recognition scores should be. This is an assumption that holds true for most non-related individuals. However, when dealing with individuals of similar or identical anatomy, a new feature set should be utilized. Each type of feature has its benefits and drawbacks. For instance, high level features are said to be more robust to degradation and channel effects but are more difficult to extract and more training data is required. Conversely, low-level features are language and text-independent but are easily affected by noise and channel mismatch [22]. Figure 1.3 presents a summary of several speech features.
1.4 Speaker Recognition

Although speech and speaker recognition share a strong overlap in types of features used and matching methods, there is a clear difference in application that should be addressed. Speaker recognition (SRE) is the process of recognizing who is speaking while automatic speech recognition (ASR) aims to determine what is being said from a spoken utterance. This may be confusing to some since, in both fields, speech is the underlying evidence for making the decision. The difference lies then in the objective since the two tasks are negatively correlated, e.g. ASR attempts to remove any speaker variance focusing on the linguistic content while SRE attempts to examine as much intra-speaker data as possible regardless of the linguistic content. SRE systems can be split into text-dependent, where the recognition phrases are known beforehand or text-independent where there is no prior knowledge of the phrases spoken. The latter is a much more difficult task in SRE. The term “voice recognition” has been applied to both
concepts but is more commonly used to indicate whether a voice is present in a recording.

Traditional SRE deals with utterances of lengths greater than ten seconds. Dealing with samples shorter than this duration is categorized into a sub-branch of SRE dubbed short utterance speaker recognition (SUSR). Automatic methods are composed of two main parts: the front-end to transform raw speech waveforms into more compact and less statistically redundant features and the back-end which is responsible for modelling speakers and scoring test utterances.

1.4.1 Speaker Verification

SRE is composed of two principle subclasses: speaker identification (SID) and speaker verification (SV). The difference is the matching scenarios between the reference and test utterances. SID compares each input test utterance to a database of N enrolled speakers while SV can be thought of as a simplified version of SID where only two matches are being made, the test utterance with the claimed speaker identity model (the true hypothesis) and then with a background model (alternative hypothesis) [8]. For example, in a traditional GMM-UBM framework, a development speaker set is used to create a background speaker model representing alternative speakers to which test speech features are directly compared to accept or reject the claimed identity of an individual. Enrollment speaker features are also adapted from the background speaker model to create individual speaker models. Figure 1.4 shows the basic procedure for a speaker recognition system. The exact mechanisms and protocols of SV will be further explained in Chapter 3.
1.4.2 Voice Studies on Twins

In the visual realm, the potential issue of twins’ faces, fingerprints, and irises has been illuminated and remedies prescribed but, on the other hand, the issue of voice has not reached a sound conclusion due to limited databases and generalized approaches. In other words, there remains a disparity between investigations of twin SRE and other biometric modalities regarding the number of subjects tested, types of data, and algorithms employed. This presents a major gap in twin speech research. Again, since speech acquisition is dependent on genetic and environmental factors there may be more potential for intra-pair discrimination than other modalities.

Most SRE research has concluded that, whether done by man or machine, some speakers are simply more easily recognized than other individuals and that most of the errors may be attributed to only a few speakers [35]. For this reason, it can be difficult to completely assess the impact that twins have on SRE. Several studies on the voices of MZ and DZ twins have been undertaken to attempt to assess the degree of their similarity as well as comparison to that of non-related individuals. One such study [16] determined that there was a greater perceptual resemblance in identical twin speech than
compared to non-twins using human listening experiments and analysis of the $f_0$. Another study [36] delved deeper and examined multiple voice features such as vocal range, $f_0$, maximum voice intensity and vibrato intensity among others between 31 MZ twin pairs and a comparable group of non-related speakers. The study also found that there was no intra-pair correlation of differences as the twins aged.

In addition to researching twin voices with automatic, acoustic and articulatory approaches, there have been studies that investigate discrimination with perceptual experiments. When asked to choose the correct speaker based on a voice sample comprised of two sentences spoken by twins, human listening experiments resulted in an 82% and 74% success rate for identifying female and male twins, respectively [37]. Other series of listening experiments involving identical twins as well as unrelated speakers [26] found that twins’ voices can be distinguished by humans, but are more often confused than unrelated speakers. Lastly, in another perceptual experiment [24], family members of MZ twins listened to one six-second audio clip and were asked to identify which of the twins was the speaker with a resulting EER of 8.2%.

There has also been an exploration into other voice features that may prove viable in further twin differentiation. Sebastian et al [38] performed human listening experiments as well as acoustic analysis and determined that one such feature, shimmer, deviated more significantly between intra-twin pairs than other features suggesting its potential effectiveness. Other parameters such as $f_0$ which correlate to perceived speaker similarity did not prove to be discriminant as supported by an 80% success rate in listening experiments in identifying the twins’ voices as different speakers. However, the study focused on only ten identical twins between the ages of 11-15.

The viability of shimmer among twins has also been corroborated by Weirich and Lancia [39]. They found that among a few vocal parameters that shimmer and harmonic-to-noise ratio (HNR) were significantly different for two out of four sets of twins examined. One twin in the pair exhibited a low HNR value compared to her twin describing the hoarse quality of her voice. This could be explained by
her profession as a teacher and having to repeatedly yell at students.

The aforementioned studies on different voice features are unanimous in the following respects: mean $f_0$ does not seem to be a useful feature in differentiation of related individuals as much as it is for non-related speakers since it is largely dependent on physical factors such as the size and length of the vocal folds and supra-laryngeal vocal tract. The mean $f_0$ range is similar between MZ and DZ twins, owing to its dependence on learned behavior, and voice quality factors, such as HNR, shimmer and jitter seem to be more discriminative since they can be altered by environmental factors although they are low level features.

1.4.2.1 Automatic Speaker Recognition Studies on Twins

This section will thoroughly describe results of selected SRE studies on twins since they are most relevant to and form the basis of experiments in the present study.

At the same time as investigating SRE on twins’ voices an investigation on SRE’s robustness and discrimination power is being tested based on one of the most extreme cases of similarity. Although low-level features are not completely independent on high-level factors, a system with augmented high-level features as input might prove to be more discriminatory and useful in the recognition of twins as well as speech data recorded in mismatched situations. Unfortunately, datasets containing greater than 60 identical twins are virtually non-existent and do not include forensic style conditions such as audio channel mismatch, noise, and different speaking modes (conversational, read, spontaneous, etc.) [7].

Researchers in both engineering and linguistics disciplines have performed studies on automatic methods of SRE concerning twins. Kunzel [40] employed a commercial forensic voice matcher with 19 MFCCs appended with their deltas resulting in a 39-dimensional feature vector computed every ten ms on a corpus of nine male and 26 female pairs. The speech data consisted of 80 seconds of conversational speech and 35-45 seconds of read speech. Two different tests were conducted: 1). enrollment utterances
were spontaneous and test utterances were read speech and 2.) enrollment and test utterances were spontaneous speech. Both tests were conducted for three possible subject matching scenarios with their respective likelihood ratio (LR) distributions: 1. Inter-speaker (non-target), 2. Intra twin pair, 3. Intra-speaker (target). For the male twins, perfect recognition was achieved except for intra-pair matching of Test 2 with an EER of 11%. This means that, in a strict sense, six of the nine pairs of twins exceeded the unity threshold indicating the decision of the co-twin as the target speaker. In other words, there was a confusion of the twins in a pair. However, in relative terms of the LRs for the inter-speaker distributions, on the order of $10^{-2}$, one could say the system was successful in differentiating all male twins for this case.

The female experiments were not as successful. For Test 1, EERs of 0.5% and 19.2% were obtained for the inter-speaker and intra-twin pair scenarios, respectively. As before though, in relative terms, one may say the system was successful in telling apart the female twins in a pair. The worst performance was found for Test 2 with EERs of 4.4% and 48%. This 48% EER strictly indicates that co-twin distinctions are just as likely as non-distinctions. Thirteen of the intra-twin pair LRs for Test 2 were higher than unity but still smaller than the target LRs for her respective twin. However, a total of 11 twins received equal or greater LRs than her respective twin, meaning in both absolute and relative terms, the twins were misidentified.

Scores were weighed using a reference population that did not match the conditions of the test population (number of speakers, speaking mode, channel characteristics and even language), according to [23], which may have produced unreliable results. If the system had been calibrated correctly, the LRs for non-target trials may have been smaller than unity eliminating the need for relative LR scoring. At any rate, the system was not able to distinguish 11 of the 26 female pairs. The final conclusions of the study are that the genetic similarity of individuals tested does indeed affect the performance of the SRE system employed and furthermore, is inferior to female voices over male voices. Explained by Kunzel,
“as a consequence of the higher $f_0$ of female voices, the spacing of the harmonics is less dense than male voices, which in turn yields less speaker information in the spectrum.” It is also interesting to note that when both enrollment and test utterances were spontaneous speech, the EER was increased due to more intra-speaker variability.

In another automatic SRE study of twins [41], the authors used a dataset of 49 twin pairs that was predominantly female speakers (40 pairs of females, 9 pairs of males). The data consisted of 60 seconds of a poem reading and a five second date of birth reading. In the first test, the enrollment consisted of using the first 30 seconds of the read poem and the remaining portion for testing. For the second test, the date of birth reading was used as the test utterance. Two matching configurations were performed as well: one where each test speaker was matched with every other enrolled speaker (OVERALL) and the other where the test speaker could only be matched with him/herself and co-twin (TWIN). Instead of using MFCCs, the authors used another type of low-level feature, linear predictive cepstral coefficients (LPCCs), along with a GMM-UBM framework and not unlike [40], they used unconstrained cohort score normalization (UCN) in an attempt to mitigate an improperly developed background dataset by using the three highest scoring competitive speakers to normalize the score obtained from each twin when matched with the other’s reference model. However, unlike [40] the UCN was more supervised and tailored. For the OVERALL matching, UCN reduced the EERs from 2.8% to 0.5% and 0.4% to 0% for the date of birth and poem utterances, respectively. The TWIN configuration resulted in more error with UCN driving down the error from 10.4% to 1% and 5.2% to around 0% again for date of birth and poem utterances, respectively. The authors justify the reduction of errors due to UCN, “exploiting the non-genetic characteristics of the twins’ voices to enhance discrimination.” However, mismatching may have occurred since the date of birth utterances for each twin is obviously the same resulting in a text-dependent situation for co-twin matching and a text-independent situation for all others. The gender factor did not seem to be an issue in this study.
The only study to investigate not only MZ twins but also DZ twins and their siblings was conducted by San Segundo and Kunzel [42]. The same set of matching experiments as in [40] were performed with MFCCs input into the Batvox 4.1 system updated with GMM channel factor analysis methods along with 31 cohort speakers for score normalization. A corpus of 24 male MZ twins, 10 DZ twins, eight brothers and 12 unrelated speakers all with 120 seconds of speech was analyzed. Their hypothesis proved correct that the similarity coefficient values between these groups of people would decrease in order of the amount of shared genetic relationships, i.e. the similarity values decreased in the order of MZ, DZ, non-twin siblings and non-related speakers. This is expected since the speech features used largely depend on vocal production anatomy. With a closer look at MZ twin pairs, they found an EER of 9.9% when testing MZ intra-pair comparisons like the EER obtained in [40], although only male speakers and conversational data was used. This means that, of the 12 pairs of MZ twins tested, two were misidentified with his co-twin. The overreaching significance of this study was finding out that MZ twins were harder to differentiate than the lesser related DZ twins meaning that voice is strongly influenced by genetics.

Kunzel [40] encountered very high EERs for matched conditions where both the enrollment and test utterances were composed of spontaneous speech. This could be because the amount of intra-speaker variability of each twin introduced in the spontaneous samples created an overlap of speech characteristics of the twin pair. To drive down this error, higher-level features could be used that would better characterize voice cues unique to each speaker that mere vocal tract modeling through MFCCs could not perceive.

As discovered by Segundo, the degree of similarity was in concordance with the degree of difficulty of speaker differentiation, leading to the conclusion that MFCC features become less discriminant as the degree of shared genetics becomes greater. The substitution, or at least, the complementation of higher level speech features, may prove to alleviate the decreased differentiability of highly related individuals. All studies on twins’ voices whether it be automatic or perceptual conclude...
that the degree of difficulty in differentiating speakers is in decreasing order of MZ twins, DZ twins and non-related speakers.

A summary of the results for the most recent and most comparative studies to the present research effort are presented in Table 1.1.

<table>
<thead>
<tr>
<th>Study</th>
<th>No. of Subjects (Pairs)</th>
<th>Type of Speech Data</th>
<th>Method</th>
<th>Intra-Pair Comparisons Results (EER)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ariyaeinia (2008)</td>
<td>40 Female 9 Male</td>
<td>Read</td>
<td>GMM-UBM w/UCN</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>~0%</td>
<td>Matched: N/A</td>
</tr>
<tr>
<td>Kunzel (2010)</td>
<td>26 Female 9 Male</td>
<td>Read and Spontaneous</td>
<td>Batvox 3.1</td>
<td>48% Female 11% Male 19.2% Female 0% Male</td>
</tr>
<tr>
<td>San Segundo (2015)</td>
<td>24 Male 10 Male</td>
<td>Conversational</td>
<td>Batvox 4.1</td>
<td>9.9%</td>
</tr>
</tbody>
</table>

1.4.3 Other Issues within Speaker Recognition

There are two main areas of weaknesses where research has sought to improve SRE, speaker variability and channel variability. Speaker variability is the intrinsic variability of one person’s speech due to emotion, health, age, disguise, etc., between recordings. Channel variability arises from comparing two recordings done with diverse audio capture devices, environments or varying distances between the source and microphone along with contributing factors such as noise, spectrum coloring and reverberation. Both types of errors are collectively known as session variability and remain to be the central obstacle in SRE. When such recording factors are inconsistent between the enrollment and test recordings, they are said to be under mismatched conditions and, in some cases, have shown to degrade performance to unsatisfactory results. Many efforts have been made to compensate for channel mismatch situations such as modification of speech features [43], adjustment of speaker models [44] or score normalization [45]. A comprehensive analysis of channel compensation techniques was done by [46] and found that the most effective system with a 4.4% EER employed a 50 eigenchannel adaptation scheme.
similar to the pre-cursor of the i-vector framework. Of the two variability issues, research has focused more on the channel variability side due to difficulty in obtaining useful data that exhibits different forms of intra-speaker variability. For instance, to accurately assess the impact of aging on speaker recognition, one would have to collect speech data from participants over a relatively long length of time. In another example, to investigate voice spoofing, one would need a corpus containing participants purposefully disguising their voice.

1.5 Neural Networks

A type of machine learning algorithm, based on biological processes of the brain, has given way to many advances in the field of pattern recognition. This is evidenced by the performance gains in numerous pattern recognition tasks during recent computer vision and computer hearing challenges. This chapter gives an overview of neural networks in the context of SRE. Section 1.5.1 will briefly review the history of neural networks along with their fundamentals. The fundamentals of ANNs will be discussed through the description of multi-layer perceptrons (MLPs) since they are a simple yet powerful architecture on which all ANNs are based. Section 1.5.2 will delve into a special breed of neural networks more suited for sequential pattern labeling and Section 1.5.2.1 will conclude with a special case of architecture and the motivation for its implementation in this thesis.

1.5.1 Brief History and Fundamentals of Neural Networks

Nature exhibits the most elegant and effective engineering solutions. It is for this reason artificial neural networks (ANNs) were first developed as crude mathematical models of the operation of biological brains. Over the past 50 years, extensive research in this field has experienced lulls and leaps ultimately leading to many advancements in pattern recognition and the foundation of artificial intelligence.

ANNs are composed of layers of sub-processing units called neurons bound together with
weighted connections. In comparison to the biological model, the nucleus performs the summation and activation of the data within a neuron. The output of the artificial neuron is analogous to the axon and the weighted connections of the artificial neuron represent the strength of the junction between the axon terminals and the dendrites of the following neuron. These junctions are referred to as synapses. Networks determine these synaptic strengths through a learning process where training data is fed into the network with an accompanying class label. For human infants, this is not unlike the association of the term “dog” while being conditioned with an example of the nature of a dog. Another example of a supervised learning process in terms of speech is one second segments of a speaker utterance being consecutively input to a network with its appropriate speaker label fed at the output. The weights are fine-tuned in a supervised manner until the difference between the actual and desired output is minimized thus achieving correct classification. Figure 1.5 shows a visual comparison of a biological and artificial neuron and Figure 1.6 shows a simple, fully connected neural network architecture.

Figure 1.5. Artificial Neuron (top) and Biological Neuron (bottom) Anatomical Comparison. Blue Lines Indicate Analogous Components
One of the first neural networks ever created was the Mark I Perceptron by Frank Rosenblatt [47] in 1957. This NN was implemented in custom hardware with weights being encoded with potentiometers and photocells for imaging input alphabetic characters to perform subsequent classification. Neurons were based off the threshold logic unit by McCulloch and Pitts [48] in 1943 that used a threshold transfer function similar to a Heaviside step function.

The limitations of the Mark I Perceptron were due to the fact it was a single layer network. These types of networks are only able to separate classes of a small subset of applications that are already linearly separable in their native domain. Even a rudimentary non-linear input-output mapping, such as the XOR function, was unable to be modeled by a perceptron as shown by Minsky and Papert [49] in 1969. To overcome this problem, Widrow and Hoff [50] combined Perceptrons to create the first MLP referred to as MADALINE. This network was hardware implemented as well and consisted of three
layers of ADALINE units with a sinusoidal activation function. Although this three-layer network could separate more complex, non-linear class relationships, the non-differentiable nature of the activation function required a custom training algorithm to be used which was unable to correctly update the weights of the hidden-output connections preventing the network from converging to the desired outputs [51]. Due to overpromises and perceived road blocks, research in NNs began to dwindle and the 1970s saw calm in the field.

NNs at their heart strive to find some decision boundary in a new domain to reliably and effectively classify data. MLPs have been proven to be able to fit any continuous function given a sufficient number of non-linear units, bestowing them the name of universal function approximators [52]. As data is propagated through the network through each successive layer, it is transformed from the input data space into new representations so that classes are more easily separable.

NNs learn by performing two successive algorithms—a forward pass where input data is propagated throughout the network with the current setting of weights, and a backward pass where weights are adjusted according to the amount of error at the output. The neural network section of Chapter 2 describes these processes further.

1.5.2 Recurrent Neural Networks

The types of networks described in the previous section belong to a class of ANNs called feedforward neural networks (FFNNs) with the perceptron being its simplest form. These types of networks are inherently unsuited for dealing with temporal data. Through the fertile field of research over the past 50 years, scientists began to understand the role of cyclic connections in forming memory within these dynamic networks. These feedback loops essentially give the network the ability to form a history of previous inputs to outputs. Recurrent neural networks (RNNs) are a type of ANN that feature these feedback loops as displayed in Figure 7 creating a set of hidden states. This gives RNNs the theoretical
ability to remember patterns deep in time, since their hidden state is a function of all past hidden states. This capability makes them well suited for temporal data and lists as they can operate on sequences of vectors with the temporal data propagated through many time steps. This makes RNNs adept to speech data as the network can consider longer speech fragments with valuable speech information such as sentence formulation and intonation, vocabulary choice, speaking rate, duration of phonemes, rhythm and emphasis. It is true that DNNs can slightly extend their window of context by stacking multiple window segments into one large input vector but the time dependency between each stacked input is lost since there are no recursive connections inside the network.

The advantage of remembering the history of input sequences has made RNNs very popular for time series prediction and classification such as speech synthesis [53], financial market analysis [54], sentiment analysis [55] and even language translation [56]. Another advantage of these types of networks is that their parameters are iteratively adjusted by only three weight matrices, significantly reducing the number of parameters. For this reason, RNNs can be viewed as the unfurling of a chain of repeating modules of a neural network. In this portrayal, RNNs can be roughly interpreted as FFNNs as shown in Figure 1.7. However, re-using the same parameters does have its disadvantages. The network parameters will be described in more detail in the later sections.

![Figure 1.7. Unfurling of an RNN with mapping of input vectors $x_t$ to hidden states $h_t$.](image)
RNNs learn through an iterative process called back-propagation through time (BPTT), a technique similar to regular back propagation where the gradient of a cost function with respect to the weights of a network is descended iteratively. It was found that parameter updating was sufficient for short-term dependencies but was not powerful enough for longer ones (>10 timesteps) due to the vanishing and exploding gradient problem [57], although ad-hoc RNNs can be specially configured for adequate parameter updating for a certain application. The vanishing/exploding gradient problem states that the derivative of the state of a dynamic system at a certain time $t$ with respect to the initial starting time decreases or increases exponentially as time increases making parameter updates minute and insignificant or too large thus destabilizing the system making it fail to track longer-term dependencies. It would seem then that traditional RNNs, although theoretically able to process and remember contextual information, fail at their practical implementation. This problem could be alleviated with an RNN using a more-well designed recurrence formula.

### 1.5.2.1 Long Short-Term Memory Networks

LSTM networks [9] at their core are a special type of RNN using a more complex recurrence function allowing the network to remember longer-term dependencies than a regular RNN. The long-short oxymoron in the nomenclature refers to the “short-term” activation of internal units containing recent history as well as the slower changing parameter updating of a memory cell which forms “long-term” experience based memory. The updating is regulated by a set of gates in the hidden state connection which let the network decide which information is beneficial to guard, add or remove for a given task. These functions are analogous to a CPU’s ability to write, read and reset data. A flow diagram of a typical LSTM is shown in Figure 1.8. An explicit description of the LSTMs recurrence function will be presented in Section 2.4.1.
The most common LSTM is composed of three gates and a memory cell. Not unlike a traditional RNN, LSTMs continuously update a hidden state on each time iteration. LSTM’s more dynamic behavior, however, comes from the use of a memory cell that affects the output of each hidden state. These memory cells give the network the freedom to determine what information to guard, add, or delete for a given task. These gates can be interpreted as filters carefully kneading internal representations of sequences for maximal classification power.

As for the efficacy of LSTM networks, [11] showed that their LSTM architecture outperformed a DNN with a relative improvement in EER by 30% in a text-dependent end-to-end verification scenario. To the best of the author’s knowledge, no other study has attempted to utilize $f_0$ characteristics along with other prosodic/higher level features with an LSTM in the application of speaker recognition. Previous studies, such as [58] have sought to perform emotion recognition, [59] sought to perform language recognition and [60] used prosodic features with a BLSTM for non-native verbal test answer scoring and [61] used these features as well for the application of overlapping speech detection. [62] incorporated $f_0$ features along with cepstral features but only used an RNN. However, they did show that
the inclusion of this feature increased performance but the number of subjects tested was only 16 speakers. [63] used LSTMs with the goal of building text-to-speech synthesis systems to generate more speech data. In conclusion, prosodic features were found to improve performance in all the aforementioned studies regardless of the application.

In the context of speaker verification, the dominant approach has been to feed in utterance or frame level speech data into a DNN and then average the activation vectors in the last layer over the full sequence resulting in one vector that represents the speaker model—the so-called d-vector. This technique draws criticism for two main reasons: the loss function used to optimize the network and the limited context of speech data in deriving the d-vector. A typical loss function used is the softmax loss which scales poorly with increasing speaker classes in addition to requiring much speaker data to accurately estimate the network parameters. This problem can be mitigated with candidate sampling [64]. However, estimation due to short contextual speech windows remains a problem. A potential remedy exists in concatenating multiple frame feature vectors into one large vector for input that provides the network with marginally more speech context.

### 1.6 Research Goals

With a background of neural networks and their operation as well as the characteristics of the human voice, some objectives can be clearly defined. The goals of this research effort are to compare and analyze the performance of various speech features for identical twin voice separation. This thesis will also compare the performance of the LSTM network with other SRE algorithms in various training and testing scenarios with variable length data. Lastly, this thesis will determine which hyper-parameters will provide the best performance. The performance metrics will be based upon the EER and twin misidentification percentage rate.

The coincidence of using Siamese networks for improved intra-pair discrimination of twins is
acknowledged. As many engineering problems have been so elegantly solved by nature already, it seems natural to continue in this direction. The organization of this thesis is as follows: Chapter 1 provided an overview of identical twins in the context of biometrics and more specifically an analysis of their effect on speaker recognition through studies examining different types of speech features. The class of pattern recognizers used in this thesis, neural networks, were discussed from their fundamentals to recent advances in LSTMs. Chapter 2 will discuss these concepts in more detail along with derivation of the various types of speech features and the operation of various neural networks as well as the motivation for using them in this research effort. Chapter 3 will describe the corpus of data and the algorithms used as well as outline the set of experiments that will be taken to assess the performance of certain speech features and network architectures on the recognition of twins’ voices. Chapter 4 will compare the results from the i-vector baseline and LSTM methodologies and provide a discussion of the results from various lengths and types of speech features. Chapter 5 will conclude with a summary of the results, future directions and other applications for which the proposed neural network would be well-suited.
Chapter 2: Theory
2.1 Speech Feature Extraction

2.1.1 Mel-frequency Cepstral Coefficients

Conventional speaker recognition systems typically employ vocal tract related acoustic features for discriminative purposes. The most commonly used feature in SRE are mel-frequency cepstral coefficients (MFCCs) [65] as they represent formants by modeling the resonance of the vocal tract. The information from the vocal tract has been found to be the most valuable for ASR and SRE. These parameters are the most prevalent representations of the speech signal and contain a high degree of speaker specificity and have been shown to achieve a good balance between descriptive power, complexity and dimensionality [66]. For these reasons, MFCCs are the most commonly used features in speech technologies.

Since the speech signal is ever changing, it is windowed into short segments ranging from 25-40 ms in order to be processed with Fourier analysis. For example, these windows may overlap with a stride of 10 ms between each frame. From a signal processing standpoint, speech can be modeled as the convolution between the vocal source glottis impulse and the filtering of the vocal tract. For a discrete speech signal frame,

\[ s(n) = e(n) * v(n) \]  

Eq. 1

where \( e(n) \) is the excitation source and \( v(n) \) is the vocal source filtering of a single point sample \( n \), a discrete Fourier transform (DFT) is applied to \( s(n) \) to reveal richer, more valuable frequency domain information about the signal:

\[ S(k) = \sum_{n=1}^{N} s(n) e^{-\frac{j(2\pi kn)}{N}} \quad 1 \leq k \leq K \]  

Eq. 2
where $K$ is the resolution of the transform, $k$ is the particular frequency bin and $N$ is the total number of samples. The DFT results in a complex-valued function containing both magnitude and phase. It is generally believed that the phase contains no useful information about the signal so the periodogram is calculated rendering the spectral power density,

$$P(k) = \frac{1}{N} |S(k)|^2$$

Eq. 3

At this point the power spectrum is passed through a filter bank with strategically spaced filters to mimic human audio perception. The human ear is not sensitive enough to discern minor variations in frequency and this limitation becomes more pronounced as the frequencies increases. For this reason, the mel-frequency spaced filter bank, as shown in Figure 9, with its lower filters spaced closer together, is applied to the power spectrum. The typical number of filters is between 20 and 40.

The energy in each filter bank is summed and a logarithm is applied for its mathematical property of transforming multiplicative relationships into additive ones as well as that humans do not hear volume changes on a linear scale, e.g. to double the loudness of a sound, the sound must have eight times more energy. Finally, another type of Fourier transform, the discrete cosine transform (DCT), is applied to each frequency band serving to de-correlate the energies (due to the overlapping windows previously applied). The final MFCCs are calculated according to:

$$C_n = \sum_{m=1}^{M} \left[ \log Y(m) \right] \cos \left( \frac{\pi n}{M} \left( m - \frac{1}{2} \right) \right)$$

Eq. 4

where $Y(m), m = 1, \ldots, M$, are the outputs of the M-channel filter bank and $n$ is the index of the cepstral
coefficient. The final acoustic feature vector is obtained by disregarding all coefficients above the 12 to 13 lowest excluding the first coefficient which is the mean value of the speech signal containing very little discriminant information.

Sometimes, first and second order derivatives of the MFCCs are appended to the feature vector to better model the co-articulation of phonemes and formant transitions. These derivatives are sometimes referred to as the acceleration and trajectory, respectively, and account for the spectro-temporal aspects of speech.

$$d_i = \frac{\sum_{m=1}^{M} m(c_{i+m} - c_{i-m})}{2 \sum_{m=1}^{M} m^2}$$

Eq. 5

In principle, since MFCCs are largely based on anatomical parameters such as the size, shape and condition of the vocal tract, identical twins should no doubt pose a problem to SRE systems using these features.

In addition to MFCCs being used to represent the speech data into the system, another related-feature will be investigated that may be more suitable for improved differentiation. This type of feature is called mel-frequency spectral coefficients (MFSCs) and are identical to MFCCs except for the last step where the DCT is performed. When the DCT is applied, the locality of the original input data is lost which could be detrimental for local characteristics of the convolutional operations of a convolutional neural network (CNN). Instead of performing the DCT, the log energies are guarded resulting in a feature vector of the same length as the number of mel-filters as shown in Figure 2,1.
2.1.2 Other Low-Level Features: Jitter and Shimmer

Jitter and shimmer are two low-level short-term spectral features that have shown promise as a discriminative speech feature in twins’ SRE. These two features are measurements of the periodic variations in fundamental frequency and amplitude, respectively, and are perceived as vocal quality characteristics such as buzziness, hoarseness, or breathiness. Mathematically, they can be calculated according to:

\[
Jitter(\text{absolute}) = \frac{1}{N-1} \sum_{i=1}^{N-1} |T_i - T_{i+1}|
\]

Eq. 6

\[
Shimmer(dB) = \frac{1}{N-1} \sum_{i=1}^{N-1} \left| 20 \log \left( \frac{A_{i+1}}{A_i} \right) \right|
\]

Eq. 7
where $T_i$ are the chosen $f_0$ period lengths, $A_i$ are the chosen peak-to-peak amplitude values and $N$ is the number of chosen $f_0$ periods. Jitter can be interpreted as the acoustic measurement of how much a given period differs from the period that immediately follows it and shimmer is the short-term variability of the peak-to-peak amplitude in decibels. Speaker-specific values of these measurements can be affected by environmental factors such as smoking, yelling, aging or any other strain on the voice. It has also been reported that significant differences in shimmer and jitter values can be encountered based on speaking style.

These features are not viable on their own. It is the appending of these features with traditional and/or prosodic features that boosts performance. Farrus et al [67] offer an interesting analysis of prosodic and traditional systems appended with a so-called “JitShim” feature set. Both types of systems improved with the addition of jitter and shimmer features. The lowest EER was achieved with a system incorporating all three types of features yielding an EER of 6.8% on telephone conversation speech.

### 2.1.3 High-Level Features

Human listeners subconsciously consider many factors when determining who spoke a certain utterance such as the timbre of the voice, accent and pronunciation, and the idiolect of the speaker, i.e. the speaker’s characteristic use of vocabulary and grammar. This lexicon, along with the speaker’s semantics, encompasses the higher-level features of speech. The main approach of quantifying these features has been using a sequence of tokens or binary trees detailing the co-occurrence of certain words, phones and prosodic gestures [22]. Motivated by the success of modern language recognizers, these tokens may be used individually or as a combination. To classify these token features, an N-gram modeling approach is usually taken where the joint probability of N consecutive tokens is estimated for a token sequence $\{\beta_1, \beta_2, ..., \beta_T\}$ where, $\beta_t \in V$ and $V$ is a finite vocabulary of an utterance. For example, the phone bigrams of the sequence “you_know” are (y,o), (o,u), (u,_), (_,k), (k,n), (n,o), (o,w) and the word bigrams
are (you, know). These N-gram statistics have been used in the vector space [68] as well as with entropy measurements [69].

The SuperSid project [31] was successful in fusing low level GMM cepstra features with mid- and high-level features using a single layer perceptron classifier. With the combination of nine multi-level features they were able to drive down the EER of the baseline GMM system (using only low-level features) from 0.7% to 0.2% thus improving speaker recognition with complementary high-level features. Subsets of systems were also explored and one of the best was found to fuse GMM cepstra, prosodic and idiolectal features.

High-level features are valuable features that have been investigated for improved speaker differentiability. However, they can be difficult to extract since they are inherently revealed over longer periods of time. For this reason, traditional SRE methods such as GMM-UBM, i-vector and even deep neural network (DNN) approaches do not directly extract these features since they only accept a fixed length speech feature computed over small (~25 ms) windows with no temporal tracking. Larger context speech data at the utterance level may be input to such systems, however, by concatenating multiple window feature vectors into a larger vector to derive speaker representations at a cost of slower computation. One study [11] using DNNs found that speaker representations trained on utterance level data gave better performance compared to frame level from an EER of 3.86% to 2.9% suggesting the benefit of contextual information in speaker verification applications. In the case of twins, genetic factors are the same so the assumption is made that high-level features as well as certain low-level features such as shimmer and jitter could be more discerning due to environmental factors.
2.2 Speaker Verification

Delving deeper into SV, the basis to accept or reject a claimed identity is determined by the log likelihood ratio (LLR),

$$\Lambda(X) = \log p(X|\lambda_{hyp}) - \log p(X|\lambda_{\overline{hyp}})$$

Eq. 8

where $X$ is the speech utterance and $\lambda_{hyp}$ and $\lambda_{\overline{hyp}}$ are mathematical models for the hypothesized and alternative speakers, respectively. If the test speaker’s utterance scores are above a certain threshold, the speaker is verified as the claimed identity. The main goal of SV is to find ways to compute values for these two terms [8]. SV can be viewed as one identification trial where the alternative speaker model is not one individual but a strategized negative case representing all other speakers. For this reason, SV is the more computationally efficient and easier operating mode of the two. Technically, the LR is approximated with probabilistic estimations of the target speaker’s variability and the variability of the ‘rest of the speakers in the world’ i.e. the UBM with normalization and transformation procedures that turn the original similarity score into the final LR. Graphically, the EER is the value where the genuine and imposter distributions intersect as shown for example in Figure 2,2. The area of the overlapping distributions represents a region of uncertainty where acceptance decisions cannot be confidently made and hence serves as the most accepted performance metric in speaker verification applications. Intuitively, this performance metric is adjusted to the point where the rate of false acceptance and false rejection are equal, hence the name “equal error” rate, since one would not typically desire a system biased on passing one type of error over the other. However, there are applications where this threshold may be adjusted, for instance, in high security applications where the repercussions of a false rejection would not be as severe as a false acceptance. Due to the nature of each operating mode, the commercial sector typically uses speech technology as a verification tool such as to control access to a restricted area,
information or service while identification finds more use in forensic applications.

![Target and Imposter Score Distributions](image)

*Figure 2.2. Hypothetical Speaker Score Distributions in Verification Scenario. Equal Error Rate (EER) is the point of intersection of Distributions. For this example, the EER would be approximately 10%*

### 2.2.1 Protocol

There are three main steps in the SV protocol which are described further:

**Development:**

In this phase a general speaker representation is created by a model. This representation depends on the type of model being used (Gaussian subspace model or DNN). In Gaussian models, an i-vector is created from input speech utterances calculated with maximum likelihood estimation (MLE) for the model loss. On the other hand, a DNN uses d-vectors to model this representation that is calculated using a softmax loss. Both models attempt to optimize an internal speaker representation to be speaker discriminative and can use frame or utterance level speech for the input data representation in a supervised fashion.
Enrollment:
After appropriate representations have been created and the frameworks are primed, speakers may be enrolled into the system to create their own unique model. The enrollment speaker models are generated from their respective speaker utterances. The i-vector framework adapts the development i-vectors to each speaker whereas a DNN enrollment model is typically the average of the outputs of the final layer before the softmax layer for each enrollment utterance or frame and is referred to as the d-vector.

Verification:
In the final step, each verification trial is scored to accept or reject the claimed identity of an individual. A test utterance is fed into the network and its representation is generated (an i-vector or d-vector). In the i-vector framework, this test utterance is compared to the i-vectors of the claimed speaker and that of the development model. A fast and linear scoring strategy is used to compute the LLR. If this ratio exceeds a pre-determined threshold, the claim speaker is verified. However, in a DNN framework, there is no single development d-vector of which to compare test utterances so a one-vs-all setup is performed where the test utterance is compared to each enrolled speaker model. The most common similarity metric for comparing the d-vectors is the cosine similarity function.

2.3 i-vector
The de-facto reference method of voice matching for text-independent speech is the GMM-UBM method. This matching approach works by fitting the input features to Gaussian mixture models as shown in Equation 1 and 2,

\[ p(x|\lambda) = \sum_{j=1}^{M} w_j p_j(x) \]

Eq. 1
and

\[ p_j(x) = \frac{1}{(2\pi)^{D/2}|\Sigma_j|^{1/2}} \exp\left\{ -\frac{1}{2} (x - \mu_j)^\prime (\Sigma_j)^{-1} (x - \mu_j) \right\} \]

Eq. 2

where \( \mu_j \) is a \( D \times 1 \) mean vector, \( \Sigma_j \) is a \( D \times D \) covariance matrix, and \( w_j \) represents the mixture weights that sum to one.

The UBM is a GMM that is fit by expectation-maximization (EM) from features of a population and conditions similar to the recognition phase and serves as the alternative hypothesis of speaker identity. During enrollment, speech features are adapted from the UBM via Bayesian adaptation to create a speaker representation. At the recognition phase, test models are weighed against the UBM to determine a log-likelihood ratio, Equation 8. A probabilistic speaker model is chosen to be a weighted linear combination of Gaussian densities for an input \( D \)-dimensional feature vector.

The classical GMM-UBM framework was improved upon by modeling the session variability through factor analysis methods [70]. Speech data from the same speaker could result in different mean vectors due to channel and speaker variability that could, in turn, give poor verification performance. A solution to this is reducing acoustic feature vectors into low-dimensional fixed length representations called i-vectors. Speaker-dependent GMM mean vectors are concatenated into a “supervector”,

\[ M = m + Tw \]

Eq.4

with mean \( m \) derived from the speaker independent UBM supervector and mean offset \( Tw \). Speaker and channel variabilities are bound together in a total variability subspace \( T \) representing the principle directions of speaker and channel variability of a large collection of speech data. Each eigen-dimension of \( T \) is controlled by a standard normally distributed vector of factors, \( w \), which allows the supervectors
to adapt by moving in directions of high variability and is referred to as the i-vector.

Since the extracted i-vectors include information about the channel characteristics, subsequent techniques must be done to negate any effects from the channel. Linear Discriminant Analysis (LDA) is one such technique capable of performing intersession compensation by means of dimensionality reduction which removes non-speaker related directions. Minute non-genetic differences in the twins’ speech resulting from dialect variation or habitual nature (e.g. smoking or yelling) should be emphasized using LDA which aims to maximize inter-speaker variability whilst minimizing intra-speaker variability.

Before verification trials are scored, the LDA channel compensated i-vectors are modelled with Probabilistic LDA (PLDA) [71] to better model the distribution of i-vector features. Let \( \hat{\mathbf{w}} \) be the channel compensated i-vectors, then:

\[
\hat{\mathbf{w}}_r = \bar{\mathbf{w}} + Ux + \epsilon_r
\]

Eq.5

where \( r = 1, \ldots, R \) are recordings for a given speaker; \( U \) is the eigenvoice subspace with \( x \) being the speaker latent factors denoting the position in the subspace; \( \bar{\mathbf{w}} \) is the mean i-vector of the training dataset and \( \epsilon_r \) is a residual noise term capturing any other variability not present in the speaker factors. Given two i-vectors \( \{ \mathbf{w}_1, \mathbf{w}_2 \} \), all verification trials are scored simultaneously with a likelihood ratio calculation of:

\[
ll_r = \ln \frac{P(\mathbf{w}_1, \mathbf{w}_2|H_1)}{P(\mathbf{w}_1|H_0)P(\mathbf{w}_2|H_0)}
\]

Eq.6

where \( H_1 \) and \( H_0 \) are the same and different speaker hypotheses, respectively.
2.4 Neural Network Information Flow

2.4.1 Forward Pass

The forward pass of a neural network is where input data is propagated throughout the network resulting in an output at the last layer. With the weight from element $i$ of input vector $\mathbf{x}$ to unit $j$ as $w_{ij}$,

$$a_j = \sigma \sum_{i=1}^{I} w_{ij} x_i$$

Eq. 9

where $I$ is the total number of input elements. The output activation $a_j$ of the $j^{th}$ neuron in a layer is calculated as the transformed weighted sum of all outputs from the previous layer (or the input data vector $X$ if the first hidden layer). An activation function, $\sigma(u)$, is applied to the summed input effectively squashing the outputs to a specified range. Typically, all neurons in a layer have the same activation function.

Common activation functions are the hyperbolic tangent, logistic sigmoid or ReLU [72]. Non-linear activation functions are an important part in NNs, for without them, they would not be able to transform the input data space into effective internal representations. For instance, a layer employing a hyperbolic tangent activation function transforms the input data according to:

$$\sigma = \tanh(Wx + b)$$

Eq. 10

The input data vector, $x$, is first linearly transformed with the weight matrix $W$ followed by a translation by the bias vector $b$. Finally, the point-wise application of the hyperbolic function stretches and shrinks the already affine transformed data space. The combination of subsequent layers transforms the data from its raw representation to higher level ones allowing the network to more easily separate the data with
hyper plane decision boundaries in the final layer. Networks with deeper and wider layers are capable of further manipulation of the input data space.

The choice of activation functions for the output layer and number of outputs depend on a task-by-task basis. For instance, in the classification of $K$ speakers, we would want to output $K$ numbers corresponding to the score of each speaker label given an input speech utterance. Since the target output is known for each iteration, loss functions are used to quantify the amount of disagreement from the desired and actual output vectors to fine-tune parameter estimation. A typical loss (or cost) function such as the following is referred to as the cross-entropy softmax loss,

$$L_k = -t_k \log(y_k)$$  \hspace{1cm} \text{Eq. 11}$$

and,

$$y_k = \frac{e^{f_{y_k}}}{\sum_c e^{f_c}}$$  \hspace{1cm} \text{Eq. 12}$$

is the softmax function. $f_c$ is the $c^{th}$ element in the vector of class scores $f$ and $f_{y_k}$ is the score of the correct class label $y_k$ given the input speech vector $x_k$ for speaker $k$ parameterized by weight matrix $W$. This type of loss function is referred to as a softmax classifier. The softmax function inside the log performs exponentiation and division of the unnormalized log probabilities in $f$ resulting in the normalized class probabilities that sum to one. A one-hot binary vector coding scheme where all elements are zero except for the element $s$ which corresponds to the correct speaker class, is used to specify the desired output for each input speech vector.

Since many sets of weights could result in the same loss, often a regularization term is added to the loss function in order to better utilize the weights of the network. A regularization penalty such as:
where \( p \) and \( q \) are the indices of the weight matrix \( W \), is commonly used and is known as the L2 norm. The data loss (eq. 11) and regularization loss (eq. 13) can be combined to create the full loss of the network,

\[
L = \frac{1}{N} \sum_s L_s + \lambda R(W)
\]

Eq. 14

where \( N \) is the number of training examples and \( \lambda \) is a weighting coefficient. The minimization of this extended loss function encourages a preference on a certain set of weight values that are of smaller magnitude and hence more dispersed. Intuitively, with more dispersed weight values, the network considers more elements of the input speech vector. An example loss function curve is shown in Figure 2.3.

---

Figure 2.3. Example loss function curve of a hypothetical NN in terms of two weights \( w_1 \) and \( w_2 \) on the x and y axis, respectively, and the associated loss on the z-axis. The x and y coordinates represent values of the two weights, and the height represents the error of the training set when evaluated at each weight setting [73]
Recently popularized “deep” neural networks are extensions of the MLP which contain two or more hidden layers. Given that non-linear activation functions are being used, the addition of more layers increases the power of NNs [51]. However, an MLP with multiple linear hidden layers is only equivalent to one linear hidden layer MLP.

2.4.2 Backward Pass

It wasn’t until 1986 that there was a resurgence in the ANN field thanks to a rediscovery of a gradient descent algorithm by Rumelhart et al [50] called backpropagation. This is a learning process performed for error minimization and curve fitting after training examples have been presented to the network. It works by descending the gradient of an error function. Gradient descent learning attempts to find a point in some parameter space, e.g. neural network weight space that minimizes an error function. The weight space and error function define an error surface. If there were only two weights, an error surface could be visualized as a hilly landscape, like in Fig. 11. However, in practical problems there can be thousands of weights, making the error surface difficult to visualize.

Gradient descent navigates the multi-dimensional loss function to find the global minimum representing the ideal parameterization of weight matrices. Backpropagation is the technique used to effectively calculate the gradient and requires that the activation functions be differentiable. The derivative of the loss function with respect to each of the network’s weights is found using a recursive application of the chain rule to determine each neuron’s contribution to the total error and then adjusting it in the direction of the negative slope. This process is repeated until the desired output is reached as the complex cost function is descended iteratively.

The following presents the derivation of the backpropagation algorithm using a cross-entropy softmax output activation. The derivative of the error function with respect to the weights connecting layer $j$ to $k$ can be expressed as,
\[ \frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial y_k} \frac{\partial f_{y_k}}{\partial f_{y_l}} \]  
Eq. 15

\[ \frac{\partial E}{\partial y_k} = -\frac{t_k}{y_k} \]  
Eq. 16

\[ \frac{\partial y_k}{\partial f_{y_l}} = \left\{ \frac{e^{f_{y_k}}}{\sum_c e^c} - \left( \frac{e^{f_{y_k}}}{\sum_c e^c} \right)^2 \right\} = y_k(k - y_k) \quad k = i \]  
Eq. 17

\[ = \left\{ -\frac{e^{f_{y_k}}}{\sum_c e^c} \right\} = -y_k y_i \quad k \neq i \]  
Eq. 18

\[ \frac{\partial E}{\partial f_{y_k}} = \sum_i \frac{\partial E}{\partial y_l} \frac{\partial y_l}{\partial f_{y_k}} \]  
Eq. 19

\[ = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial f_{y_k}} - \sum_{i \neq k} \frac{\partial E}{\partial y_l} \frac{\partial y_i}{\partial f_{y_k}} \]  
Eq. 20

\[ = -t_k(1 - y_k) + \sum_{i \neq k} t_i y_i \]  
Eq. 21

\[ = -t_k + y_k \sum_i t_i \]  
Eq. 22

\[ = y_k - t_k \]  
Eq. 23

The gradient for the weights in the top layer is,
\[
\frac{\partial E}{\partial w_{jk}} = \Sigma_k \frac{\partial E}{\partial f_{yk}} \frac{\partial f_{yk}}{\partial w_{jk}} = (y_k - t_k)h_j
\]

Eq. 24

\[
h_j = \frac{1}{1 + e^{-f_{yj}^1}}
\]

Eq. 25

\(j\) indexes the hidden units and \(f_{yj}^1\) is the weighted input sum at hidden unit \(j\). For units in the hidden layer,

\[
\frac{\partial E}{\partial f_{yj}^1} = \Sigma_k \frac{\partial E}{\partial f_{yk}} \frac{\partial f_{yk}}{\partial h_j} \frac{\partial h_j}{\partial f_{yj}^1} = \Sigma_k (y_k - t_k)(w_{jk})(h_j(1 - h_j))
\]

Eq. 26

Then a weight \(w_{ij}\) connecting input unit \(i\) to hidden unit \(j\) has the gradient,

\[
\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial f_{yj}^1} \frac{\partial f_{yj}^1}{\partial w_{ij}} = \Sigma_k (y_k - t_k)(w_{jk})(h_j(1 - h_j))(x_i)
\]

Eq. 27

With this final expression, one can compute the gradient for each neuron’s activity with respect to the input data.

The re-discovery of the backpropagation algorithm ushered in a renewed interest in the field of neural networks but did not remedy the lack of convergence. It wasn’t until twenty years later that Hinton and Salakhutdinov [74] attempted to first pre-train networks then fine-tune the parameters with backpropagation. Performance gains were met but this assumption proved to be false as the main problems were improper weight initializations along with sigmoid activation functions. Since neural networks are generative algorithms, they require many instances of diversely labeled training data alongside powerful computing to process this data. These realizations and subsequent remedies bolstered neural networks as the top competitor in recent computer perception challenges.
One such improvement worth noting is batch normalization [75]. Typically, learning is aided by initializing parameter values with zero mean and unit variance since the derivative of the activation functions is highest in this region hence creating larger weight updates. As the updating continues, the mean and variance are altered from this value resulting in slower convergence. This effect is more pronounced as updating permeates into deeper and deeper layers. Batch normalization re-establishes the normalizations for every batch of input data making higher learning rates possible. Moreover, batch normalization also acts as a regularizer, reducing or even eliminating the need for another popular technique called Dropout [76].

2.5 Recurrent Neural Network Information Flow

Given an input sequence \( X = \{x_1, ..., x_T\} \), from \( t = 1, ..., T \), a sequence of hidden states \( H = \{h_1, ..., h_T\} \) and output states \( Y = \{y_1, ..., y_T\} \) are calculated according to,

\[
h_t = W_f[x_t; h_{t-1}] = \sigma(W_{ij}x_t + W_{jj}h_{t-1} + b_j)
\]

Eq. 28

\[
y_t = W_{jk}h_t
\]

Eq. 29

where \( W_{ij} \), \( W_{jj} \) and \( W_{jk} \) are the input-hidden, hidden-hidden and hidden-output matrices, respectively. As usual, a bias vector \( b_j \) is added. \( W_f \) represents the combined weight matrices of \( W_{ij} \) and \( W_{jj} \). Notice in equations (28) and (29) that the network is iteratively adjusted by only three weight matrices significantly reducing the number of parameters needs since weights matrices are re-used. Since the same function is being applied to each timestep, input and output sequences can be processed without regard to length. Therefore, RNNs offer a lot of flexibility in their ability to change the sequence length of inputs.
and outputs whereas FFNNs only accept and output fixed length vectors. For instance, an RNN might be able to accept an image of a fixed length and output a variable sized vector encoding words to caption an image. For sentiment analysis, the network would output a sentiment to describe whether a variable length sequence of words is positive or negative. Furthermore, RNNs can combine these flexibilities with, as an example, language translation, where an arbitrary length sentence in English would result in an arbitrary length French translation. Recent improvements have been made in this application by [55] and marketed by Google. The following figure, Figure 2.4, presents a diagram of a simple RNN.

![Simple RNN structure showing weight matrices of input, hidden and output layers. Orange loops indicate recurrent connections.](image)

**Figure 2.4.** Simple RNN structure showing weight matrices of input, hidden and output layers. Orange loops indicate recurrent connections

### 2.5.1 Long Short-Term Memory Network Information Flow

The memory block of the LSTM is composed of four elements: the cell state $C_t$, an output gate that determines the cell state’s contribution to the next hidden state, and a forget and input gate that omit and add data to the cell state based on the current input and previous hidden state. These three gates govern
the flow of information within the cell state. Figure 2.5 shows an internal view of the memory block.

![Internal View of Memory Block of LSTM](image)

For timestep, $t = \{ 1, ..., T \}$, updates are performed with weight matrices $W_f, W_i, W_o, W_c$ and bias vectors $b_f, b_i, b_o, b_c$. The forget gate activation vector,

$$ f_t = \sigma(W_f [x_t; h_{t-1}] + b_f) \quad \text{Eq. 30} $$

concatenates the previous hidden state and current input vectors, multiplies them with the forget gate weight matrix and passes them through a sigmoid activation layer effectively scaling each element in the cell state between $[0, 1]$. The forget gate enables the cell state’s recurrent connection to preserve or discard itself in the presence of trivial data or when it has become obsolete. This resetting function is key to avoiding exploding gradients. Adding information to the cell state is performed by a decoupled
process. First, the input gate decides what information is useful to add to the cell state based upon,

\[ i_t = \sigma(W_i[x_t; h_{t-1}] + b_i) \]  \hspace{1cm} \text{Eq. 31}

which works in a similar manner as the forget gate by outputting a vector of elements between zero and one. This activation is multiplied with a vector of candidate values,

\[ \tilde{C}_t = \sigma(W_C[x_t; h_{t-1}] + b_c) \]  \hspace{1cm} \text{Eq. 32}

given based off of a separate weight matrix and the current input. The cell state is now updated according to,

\[ C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \]  \hspace{1cm} \text{Eq. 33}

Any unnecessary data in the old state is dampened or cleared and candidate values, scaled by the input gate, are added. Intuitively, this can be interpreted as deciding what to add to the cell state from the candidate values, and how much to add from the input gate. This seemingly redundant process allows for a richer recurrence formula in the LSTM. The final step is to output the updated hidden state. This step is like the previous step but candidate values will be pulled from the updated cell state rather than the previous hidden state along with current input values. The output gate activation and new hidden state are calculated with,

\[ o_t = \sigma(W_o[x_t; h_{t-1}] + b_o) \]  \hspace{1cm} \text{Eq. 34}
The output gate essentially determines what information from the cell state will be leaked into the next hidden state after a hyperbolic transformation of the cell state squeezing all values between [-1, 1].

The ability for LSTMs to extend its effective memory as much as a hundred times longer than RNNs lies in the utilization of the cell state (long term memory) pipeline. Notice in equation 33 that the cell state recurrence function is formed from the additive relationship between the previous cell state and candidate values, both modulated with binary gates. During weight optimization, gradients are injected back into the system and are copied at additive nodes resulting in an uninhibited flow of gradients between time steps. Moreover, since weight optimizations are not being continuously updated, BPTT is able to skip certain timesteps in the past extending the effective memory, i.e. it only updates parameters that are necessary. This phenomenon is referred to as the Constant Error Carousel (CEC) [9]. On the other hand, standard RNNs have a transformative recurrence function that could result in the weight matrix being serially multiplied by a factor. If this factor is greater than one, the gradient explodes. Conversely, if it’s less than one, the gradient vanishes. It is interesting to note that partial derivatives on functions composed of additive relations of variables allows for more efficient backpropagation. This principle is fundamental to the increased capability of LSTM networks which will be described later in this section.

Improvements have been made to the baseline LSTM algorithm such as forget gates [77] and peephole connections [78] which allow the gates to look at previous cell states as described by equations 36-38. This facilitates the gates to precisely time different events.

\[
h_t = o_t \cdot \tanh C_t
\]  
Eq. 35

\[
f_t = \sigma(W_f[C_{t-1}; x_t; h_{t-1}] + b_f)
\]  
Eq. 36
\[ i_t = \sigma(W_i[C_{t-1};x_t;h_{t-1}] + b_i) \]  \hspace{1cm} \text{Eq. 37}

\[ o_t = \sigma(W_o[C_t;x_t;h_{t-1}] + b_o) \]  \hspace{1cm} \text{Eq. 38}

2.6 Siamese Networks

Siamese networks are a type of ANN architecture consisting of two identical sub-networks each sharing the same configuration, parameters and updating. They excel in applications where two comparable input data patterns need to be scored for similarity such as in signature verification [79] or sentence semantic similarity [54] as well as general biometric verification. Since the two sub-nets share weights and biases, the number of parameters is significantly reduced leading to less training data required and a lower chance of overfitting. Each subnet outputs a transformed representation of one side of an input pattern pair feeding to a separate module that is used for comparison and scoring. Figure 2.6 shows a general configuration of a Siamese network.

Figure 2.6. Siamese Neural Network Architecture with Sub-Nets \( O(X_1) \) and \( O(X_2) \), Shared Weights \( W \) and Contrastive Cost Function Scoring Module
The cost function to be optimized in such an architecture takes a slightly different form than traditional ones as its main goal is to minimize the distance between same class pairs whilst maximizing the distance between different class pairs not unlike LDA. An example cost function capable of this goal is the contrastive cost function [80]. This function uses a Euclidean distance formula,

\[ D = \| O(X_1) - O(X_2) \|^2 \]  

Eq. 39

where \( O \) is the output of each subnet in response to input data pair \( X_1 \) and \( X_2 \). The full loss function is,

\[ L = \frac{1-Y}{2} D^2 + \frac{Y}{2} \{ \max(0, m - D) \}^2 \]  

Eq. 40

where \( m > 0 \) is the margin, and \( Y = 0 \) or \( 1 \) for genuine or imposter pairs, respectively. Depending on the genuine and imposter pair label, high dimensionality vectors which are mapped to a low-dimensionality manifold are pushed together or pulled apart by the pre-defined margin that creates a radius around the network outputs. Only when dissimilar pairs are mapped within this radius do they contribute to the loss function. The \( \frac{1}{2} \) constant on each term is again added to simplify the derivative for gradient descent. The cost function does not necessarily have to be tied to the output vectors but can operate on any arbitrary layer outputs such as in [81] which optimizes a part of a hidden layer instead of the final output layer neurons in order to optimize speaker-specific information and disregard the rest.
Chapter 3: Methodology
3.1 Datasets

3.1.1 Collection

The corpus used in these experiments was composed of two datasets collected during two days in each respective year at a twin festival inside of WhisperRoom MDL 4284 double-wall sound isolation booths with three different devices: a Neumann TLM107 condenser microphone, a Logitech Webcam C615 with a built-in microphone, and a Sanken COS-11D BP lapel microphone connected to a Canon 5D Mark II or Mark III DSLR camera. An internal diagram of the booth can be found in Figure 3.1. Audio and video were recorded for each applicable device. The audio from these devices were at sample rates and bit depths 44.1 kHz/24-bit, 48 kHz/16-bit, and 48 kHz/24-bit/stereo, respectively. The operator could communicate with the participant via computer speakers in the booth and a dynamic microphone routed in a talkback system through the audio interface’s mixing software. The session procedures were as follows: participants were asked to read aloud his/her random identification number, then were prompted to read aloud the rainbow passage. Afterwards, an interview portion with participants answering questions about him/herself was conducted. Dataset 3 used the same three devices, sound isolation booth and collection procedure but was collected in the on-campus laboratory from the public. Participants were asked to return no earlier than one month. Two separate sessions were collected each day. Raw lengths of data were three minutes for each recorded session at the festival and five minutes in the laboratory. The scripted speech (rainbow passage) was approximately 30s in duration and the unscripted speech for the festival and on-campus collections were approximately 70s and 180s in duration, respectively.
3.1.2. Post-processing

Post-processing efforts were undertaken to segment the data into scripted, unscripted audio tracks, as well as subject and interviewer portions. This was achieved automatically in MATLAB due to collection procedure modifications in dataset 2 and the latter part of dataset 3. The three device files for each session were simultaneously segmented which meant they had to be re-sampled to a uniform sample rate of 48 kHz. Dataset 1 and the former part of dataset 3 required manual segmentation in Pro Tools. Samples that were imported into Pro Tools had to be re-sampled to 48 kHz as well to match the session sample rate. The final segmented audio files are subject-only read (scripted) and conversational (unscripted) data as well as an operator file recorded through the Neumann microphone during the interview phase. The subject-only unscripted data consists of the reading of the identification number followed by his/her answers to the interview questions. All files were exported at their original sample rate and bit depth. The Canon files were exported in mono format. Audio from the video files was ripped at their original sample
rate and bit depth using Microsoft Visual Studio and MATLAB. Due to variation in the collection process, not all devices were able to be segmented in the same manner resulting in varied amounts of device files for dataset 3. A summary of the number of subjects and sessions for the datasets are shown in Tables 3.1 and 3.2.

<table>
<thead>
<tr>
<th># Sessions</th>
<th>Total</th>
<th>Male</th>
<th>Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>362</td>
<td>83</td>
<td>279</td>
</tr>
<tr>
<td>2</td>
<td>98</td>
<td>44</td>
<td>54</td>
</tr>
<tr>
<td>3</td>
<td>26</td>
<td>10</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>19</td>
<td>6</td>
<td>13</td>
</tr>
<tr>
<td>Total</td>
<td>505</td>
<td>143</td>
<td>362</td>
</tr>
</tbody>
</table>

Table 3.1. Combined Amount of Read and Conversational Speech for All Devices in Datasets 1 and 2

<table>
<thead>
<tr>
<th>Session</th>
<th>Neumann Total</th>
<th>Male</th>
<th>Female</th>
<th>Canon Total</th>
<th>Male</th>
<th>Female</th>
<th>Logitech Total</th>
<th>Male</th>
<th>Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>49</td>
<td>24</td>
<td>25</td>
<td>39</td>
<td>18</td>
<td>21</td>
<td>82</td>
<td>34</td>
<td>48</td>
</tr>
<tr>
<td>2</td>
<td>428</td>
<td>208</td>
<td>220</td>
<td>429</td>
<td>214</td>
<td>215</td>
<td>369</td>
<td>190</td>
<td>179</td>
</tr>
<tr>
<td>3</td>
<td>14</td>
<td>9</td>
<td>5</td>
<td>12</td>
<td>6</td>
<td>6</td>
<td>20</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>125</td>
<td>74</td>
<td>51</td>
<td>86</td>
<td>53</td>
<td>33</td>
<td>83</td>
<td>51</td>
<td>32</td>
</tr>
<tr>
<td>Total</td>
<td>616</td>
<td>315</td>
<td>301</td>
<td>566</td>
<td>291</td>
<td>275</td>
<td>554</td>
<td>285</td>
<td>269</td>
</tr>
</tbody>
</table>

Table 3.2. Amount of Read and Conversational Speech for Each Device for Dataset 3

3.2 LSTM and i-vector Comparison

For the first set of experiments, a series of train-test scenarios were chosen to allow for a comparison to previous twins’ voice studies as well as to meet conditions similar to recent NIST SREs. Different speaking mode data was alternated for train-test conditions to investigate how the inclusion of the speaker’s environmentally related speech habits affected performance in addition to constraining the phonetic content of speech through read passages. Each train-test condition was performed on a corpus of twin and non-twin speakers.

334 twins, 256 of which were females were used from datasets 1 and 2 for the twin speaker group. The comparative non-twin speaker group was created with one twin of each identical (167) and fraternal
twin (31) pair along with 74 non-twins from datasets 1 and 2 and 62 individuals from dataset 3 not present in the UBM. The gender ratio was the same as the twin speaker group. The following verification experiments in Table 3.3 were conducted for each group of speakers.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Conversational</td>
<td>Read</td>
</tr>
<tr>
<td>2</td>
<td>Conversational</td>
<td>Conversational</td>
</tr>
<tr>
<td>3</td>
<td>Read</td>
<td>Conversational</td>
</tr>
</tbody>
</table>

Each condition consisted of 30s of training speech and 10s test speech. Condition 2 used the 10s of speech after the first 30s for testing. In addition, a baseline experiment involving the maximum available data for Condition 1 (60s for train and 30s for test) was performed. Speaker enrollment models were trained using utterances from all three devices whereas test utterances were from the Neumann microphone. Each speaker could be matched with his/her twin as well as every other speaker. Performance metrics are reported in terms of EERs. In addition, the number of genuine and imposter samples for the experiments was balanced making the EER a reliable measure.

### 3.2.1 Feature Extraction

Before feature extraction, audio files were down-sampled to 16 kHz and, with the use of VOICEBOX [82], non-speech containing frames were removed followed by extraction of twelve dimensional MFCCs including the zeroth cepstral coefficient along with their delta and delta-delta coefficients from 25 ms windows at a frame stride of 10 ms using a 24-channel mel-filter bank. Features were further processed with dithering and global cepstral mean variance normalization over each whole utterance.
3.2.2 Baseline i-vector Algorithm

The traditional i-vector system was used to provide a baseline for comparison to the proposed LSTM architecture. The implementation of the i-vector method was accomplished using the MSR Identity ToolBox in MATLAB [83]. A 1024-component gender independent UBM was trained with full-length files for each device from a balanced ratio of male to female genders as shown in Table 1. In other words, all available data (read and conversational speech) in dataset 3 was used to create the UBM. A 400-dimensional total variable subspace and 200-dimensional LDA were specified to compute i-vectors for the development data. The same parameters for the i-vector framework in the development data were used for the enrollment and test utterances.

3.2.3 LSTM Algorithm

The effectiveness of the LSTM was tested against the traditional i-vector system with the same benchmark of experiments. With the same parameters for extracting MFCCs, a 30s sound file would result in a feature matrix of approximately 3000 frames by 39 features. The number of timesteps for an LSTM is generally between 200-400. Timesteps larger than this range result in dramatically reduced convergence speed. For this reason, the speaker feature maps needed to be manipulated in such a way to provide for correct training. For the initial assessment of the LSTM, two types of data formats were presented to the network 1.) stacked vectors and 2.) “statefulness” across batches. In the first representation, every 15 frames of a feature sequence, without overlapping, were concatenated to form a new feature map of dimension 200 frames by 585 features. In this way a full audio file could be fit into one sample to the network. Sample utterances less than 3000 timesteps were padded with frames full of zeros to reach the desired input data length.

The second data representation consisted of splitting input utterances across batches. The high-level API, Keras, enables the network to be set to ‘stateful’ where the hidden state vector is re-used at
index $I$ in each batch, effectively stringing together longer temporal dependencies across batches. The speaker feature maps were then presented as $N$ subsequences of size 200 frames by 39 features. This timestep size corresponds to roughly two seconds of speech data.

The baseline LSTM for determining the data representation consisted of one LSTM layer with 128 memory blocks followed by a fully connected layer, all prior to a softmax layer with cardinality $N = 554$, which is the maximum number of speakers in the development dataset. Since the LSTM outputs a single vector for each sequence, target labels were created for each sequence resulting in a sequence classification mode of operation. Stochastic gradient descent with a learning rate of 0.01 and RMSprop optimization was applied to the training procedure. To promote a faster convergence time, all development speaker feature maps were zero mean normalized with unit variance. Condition 1b with was used for this initial experimentation. After full convergence was achieved, the network parameters were fixed and the enrollment utterances were input to the network. Each enrolled speaker had his/her utterances input consecutively into the network and the d-vectors averaged. The enrollment and test data were also normalized. The data was formatted in the same fashion for each phase of the verification protocol as shown in Figure 3.2.
Between the two formats of data representation in the network, initial experimentation concluded that the stacked vector speech representation had better performance over the stateful network setting with EERs of 17.52% and 21.41%, respectively.

The next step was to determine what network architecture resulted in the lowest EER. Various layer depths and units were examined for Condition 1b. Since it has been shown in studies such as [84] the number of hidden layers and memory blocks was not varied greatly since the scope of this thesis seeks to compare the performance of different types of speech features which will be investigated in the next set of experiments. The range of network architectures examined is displayed in Figure 17. To improve generalization, a dropout layer was used at the input layer and dense layer with a keep probability of 0.5. Each enrolled subject had at most 12 full utterances input into the network. The output activation vectors were averaged from all utterances and were used to model the speaker. One sample containing the test utterance was fed into the network during evaluation outputting an output vector with the cosine similarity metric used to score each trial. Because of the inherent nature of the neural network, each test utterance was compared with every enrolled speaker model in a “one vs all” comparison. This
had the advantage of using multiple binary classifiers that usually are faster to converge especially given the number of output classes and high dimensionality of the input features. The dense layer used a linear activation function as in [11]. EER is reported as the performance metric for experiment one even though the test trials are more similar to a classification set-up. A false acceptance was declared if a competing enrolled speaker scored higher than the target.

Building on the previous findings, an LSTM with the architecture and range of hyper-parameters as shown in Figure 3.3 was investigated. The highest performing network was used in the comparison to the i-vector framework. The maximum number of LSTM layers investigated was four since each added layer resulted in many more computations thus increasing the convergence time considerably. The size of the densely connected layer was fixed at 384 units which is near the dimension of the speaker model i-vectors for more consistent comparison. Besides the number of units and layers, the rest of the hyper-parameters remained the same from the initial data format analysis. Results from this experimentation are shown in Chapter 4.

![Proposed LSTM Architecture](image)

Figure 3.3. Proposed LSTM Architecture. Layer sizes indicate the range of units investigated in each layer. Each yellow, black and white circle represents a Memory Block, Input Neuron, and Standard Neuron, respectively. The softmax layer is removed during the enrollment and evaluation phases.
3.3 Twin Voice Differentiation with Siamese LSTM

The second set of experiments sought to determine if a network could differentiate co-twins in a pair using various utterance lengths and features. The same twin datasets as before were organized for five-fold cross validation experiments with the network being re-trained upon each new iteration of validation. In other words, the network was independently trained with a random 80% split of the speakers and the remaining portion set aside for testing. Training speaker utterances consisted of all three device files when available to provide a sort of multi-style training procedure with audio data exhibiting various channel effects and noise. These added devices files were used in the pursuit of more robust generalization during test time as well as providing more training samples to the system. In addition, recordings from different sessions, some a year apart for repeat participants, were used for training utterances. The train-test scenario was that of Condition 3.

3.3.1 Feature Extraction

Various lengths and types of features were analyzed. Three types of features were extracted and fed to the network: MFCCs, MFECs, and the raw audio data. The first two features are the most prevalent features in speech technology. A frame length of 25 ms and stride of 10 ms was used for both features. Forty filter banks were used for each feature resulting in 40-dimensional feature for the MFECs and a 12-dimensional feature vector for the MFCCs. The appendage of the derivatives and double derivates was performed resulting in final feature vector lengths of 120 and 39, respectively. All features (including the raw data) were normalized to zero mean and unit variance over the training set for more efficient parameter updating as well as voice activity detection and cepstral mean variance normalization before feature extraction where applicable.

Continuing the data format from the previous experiments, features or raw data were stacked to limit the amount of timesteps per utterance sequence to a maximum of 400 timesteps. To this end, input
feature sizes were formatted as displayed in Table 3.4. Longer utterance lengths were not used as this significantly reduced the number of samples used for imposter pair-wise comparison.

Table 3.4. Speaker Feature Map Sizes for Experiment 2 (Timesteps x Feature Length)

<table>
<thead>
<tr>
<th>Seconds</th>
<th>MFCC</th>
<th>MFEC</th>
<th>RAW</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>196 x 39</td>
<td>196 x 120</td>
<td>200 x 160</td>
</tr>
<tr>
<td>4</td>
<td>392 x 39</td>
<td>392 x 120</td>
<td>400 x 160</td>
</tr>
<tr>
<td>6</td>
<td>294 x 78</td>
<td>294 x 240</td>
<td>300 x 320</td>
</tr>
</tbody>
</table>

### 3.3.2 Siamese LSTM Architecture

For this type of task, a specialized network called a Siamese LSTM was used that is adept at separating highly similar classes. The network parameters were again held fixed from the development phase of the previous set of experiments as well as weights being transferred to the Siamese network with the training phase fine-tuning the parameters. The network architecture used was the three-layer LSTM with 128 units in each layer. The size of the d-vector was also set to 384 units. A simplified form of the multi-class cross-entropy loss for two classes was used for these experiments, the binary cross-entropy loss. Each leg of the Siamese network output a 128-dimensional vector. The element-wise absolute difference was calculated and fed into a densely connected layer with a sigmoid activation function with one unit resulting in a score between zero and one which was fed into the binary loss function. The same learning rate and optimizer from the previous experiment was used.

The rainbow files for each twin were split into the above length files for each device at a stride length of one second and from them, random combinations were created for genuine and imposter pairs. Imposter pairs were created by matching samples within each intra-twin pair. The maximum number of genuine and imposter pairs was set to 60 for training and testing resulting in a unity ratio. Testing
utterances consisted of only the Neumann conversational files. The total number of pair-wise samples for the two second data was approximately 33,000 samples for training and 4,000 for testing. A batch size of 500 samples was used for each experiment.
Chapter 4:
Results and Discussion
4.1 LSTM Architecture Optimization

Before comparisons could be made with the i-vector system, the optimum hyper-parameters for the LSTM were determined through a set of experiments in which the number of LSTM memory blocks and layers were varied.

4.1.1 Analysis of Number of Layers and Units

The following figure, Figure 4.1, summarizes the results for each configuration.

![Figure 4.1. EERs for Multiple LSTM Layers and Units for Experiment 1 on Condition 1b](image)

As can be seen from the above figure, the addition of more memory blocks had a more profound effect when the depth of the network is smaller. This effect is more pronounced with decreasing depth size as the one-layer LSTM exhibits a 16% relative improvement in EER from 62 to 128 memory blocks.
However, an increase to 256 units degrades the generalization capabilities of the one-layer network as well as the four-layer network. A possible explanation is that the networks has too many free parameters that overfit the data from the development phase thus resulting in poor model extraction in the latter phases. The largest gain in performance resulted from increasing the depth size as shown in a 51% relative improvement from one to three layers. As the depth of the network increases, the number of hidden units becomes less significant leading to the conclusion that depth is more important than layer size which supports previous findings for deep LSTM networks [10]. The least amount of performance gains from the addition of units resulted from the four-layer LSTM. The massive number of parameters in the network likely overfit the data resulting in worst classification than its three-layer counterpart for the same number of units. The chosen architecture for comparison of the i-vector was the three-layer LSTM with 128 units since the 256-unit counterpart resulted in marginally better results but with the added burden of more parameters to optimize.

4.2 LSTM and Baseline Comparison

A comparison of twin and non-twin corpus of speakers were performed for the LSTM and i-vector systems for multiple enrollment and evaluation scenarios.

4.2.1 Effect of Type of Speech Data on Recognition Performance

Table 4.1 presents the EERs for each experiment for the twin and non-twin trials. LLR scores were compiled for each intra-twin and intra-pair trial to determine the percentage of speakers who were falsely matched with his/her twin. A false match was determined if the LLR score for the factor analysis approach or cosine similarity for the deep learning approach was higher for the respective twin than him/herself.
Table 4.1. LSTM and i-vector EER Comparisons for Multiple Train-Test Scenarios with Twin Miss Percentage Rate

<table>
<thead>
<tr>
<th>Train</th>
<th>Test</th>
<th>Condition</th>
<th>Duration (s)</th>
<th>Twins</th>
<th>Non-twins</th>
<th>Twin Miss %</th>
<th>Twins</th>
<th>Non-twins</th>
<th>Twin Miss %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conversational</td>
<td>Read</td>
<td>1</td>
<td>60-30</td>
<td>0.598</td>
<td>0.297</td>
<td>0.3</td>
<td>5.09</td>
<td>4.5</td>
<td>5.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1a</td>
<td>30-10</td>
<td>2.01</td>
<td>1.666</td>
<td>2.4</td>
<td>5.18</td>
<td>4.73</td>
<td>5.69</td>
</tr>
<tr>
<td>Conversational</td>
<td>Conversational</td>
<td>2</td>
<td>30-10</td>
<td>3.57</td>
<td>2.258</td>
<td>4.49</td>
<td>5.72</td>
<td>5.19</td>
<td>6.59</td>
</tr>
<tr>
<td>Read</td>
<td>Conversational</td>
<td>3</td>
<td>30-10</td>
<td>2.99</td>
<td>2.083</td>
<td>3.59</td>
<td>7.05</td>
<td>6.21</td>
<td>8.08</td>
</tr>
</tbody>
</table>

According to both methodologies investigated, the inclusion of identical twins in an enrolled database does tend to degrade the performance, although according to Table 6, the EERs between the twin and non-twin subjects seem to result in a smaller disparity for the LSTM method suggesting the LSTM’s use of contextual information exploiting the higher-level speech features that were hypothesized to be more variant between the voices of identical twins.

The type of speech data also played a role in verification performance as shown in the EERs of the equal duration cases. For Condition 1 and 3, training or testing utterances for each twin were read speech with the same speaking mode and wording, thus the acoustic variations in the twins lie solely in their voices. The lower EER for Condition 1a suggests that this constraint is more important for test utterances. For the i-vector system, twin differentiation was poorest when enrollment and evaluation utterances were of conversational data lending to higher intra-speaker variability as a result of the allowance of speech traits not so dependent on the speaker’s anatomy. However, in this condition, the LSTM exhibited the smallest difference in EERs possibly due to the LSTM’s increased exposure to speech patterns in the extended utterances and hence unique prosodic and other higher-level speech content of the speaker able to be exploited. The non-twin subjects also followed this trend.

Counter-intuitively, the longer range contextual speech information in this condition did not improve the performance of the LSTM proportionally to the improvement of the i-vector as the inclusion of more than double the length of speech data produced only marginally better results than the matched
conditions. An explanation could be the format of the input data as sequences may have not been correctly interpreted by the LSTM or, the BPTT mechanism, was unable to update weights as the time sequences were too long.

4.2.2 Do twins pose a problem to speaker verification?

The results from these experiments show that identical twins consistently pose a greater problem to the distinguishing capabilities of voice matching algorithms than non-related individuals as corroborated by Segundo et al [42]. Due to the nature of the present data, it was not possible to replicate all previous works’ experiments. However, one comparison can be made with [40] in the extended Condition 1. With the introduction of more individuals, performance degraded with respect to male speakers. On the other hand, the resulting EER was far less than female trials in [40]. Of the 36 total twins that were misidentified in all conditions, six were male. This result is expected since it is known in the speaker recognition community that female speakers tend to be more difficult to distinguish than their male counterparts. Although a higher EER was encountered for the LSTM comparison to the i-vector methodology, the LSTM did not seem to have a problem with one specific gender like the i-vector as the twin pair-wise confusion gender ratio was more balanced. The misidentified twin rates varied from 0.3% to 4.49% for the best cases. However, the low percentage of misidentified twins is not trivial. Scaled by the United States’ twin population of nearly 2 million, a 2% miss rate equates to roughly 40,000 twins with the potential of being misidentified.

4.2.3 Effect of Utterance Length

The next set of experiments determined the performance of each system in a SUSR setting with equal enrollment and test utterance lengths in a text-independent setting from Condition 2 (matched conversational speech). Table 4.2 presents the results for these experiments.
Table 4.2. EERs for i-vector and LSTM in Matched Short Utterance Scenarios for Condition 2

<table>
<thead>
<tr>
<th>Duration (s)</th>
<th>i-vector</th>
<th>LSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enrollment Test Twins</td>
<td>Enrollment Test Twins</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>26.77</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>23.19</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>17.03</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>9.53</td>
</tr>
</tbody>
</table>

From Table 7, the one second utterances resulted in the highest EERs as expected. As the length of utterances increased, the i-vector exhibited a larger relative improvement than the LSTM method through an exponential decrease. The largest gains resulted from the four to eight second durations as the i-vector can acquire sufficient statistics to adequately describe a speaker model in the Gaussian subspace. In the context of this research effort, the i-vector was superior to the LSTM for speech utterances longer than eight seconds. However, the LSTM performed better using utterances of one to four seconds. Improvements of 17% and 12% can be seen for the two and one second conditions, respectively. This trend is in line with [85] who performed language identification with an LSTM and i-vector and found the LSTM outperformed the other by 26% with three seconds worth of speech data and accuracy over 50% with 0.5 seconds of data.

A possible explanation for the LSTM’s effectiveness in short utterances is that it can take advantage of two complementary components of the speaker feature map. In the i-vector framework, independent frame-level feature vectors are summarized in a Gaussian subspace resulting in a latent speaker i-vector refined with linear discriminant analysis as the primary speaker model. On the other hand, in a LSTM framework, feature vectors are considered in a sequence resulting in one output activation vector. The temporal nature of the speech feature sequences should have added a second level of speaker discriminant information in addition to the actual frame level spectrogram estimates. This is beneficial when input data is scarce.

The findings from this study do not necessarily agree with the hypothesis that the LSTM should
have higher accuracy with longer speech utterances. A few reasons may be issues in the verification protocol as it pertains to a neural network structure which include a more suitable loss function during the development phase. Since development utterances for each speaker were of a varied amount, (6-24) this may not have provided enough data for certain speakers for estimation of the softmax cross-entropy loss. This applies as well in the enrollment phase where speakers had a variable number of utterances. For this reason, a weighting score function should be utilized to account for variation in the number of available utterances per speaker. More refinements to the architecture and hyper parameters of the LSTM might also be necessary to improve or surpass the performance of factor analysis methods. Lastly, the speaker feature maps may not have been formatted correctly to take advantage of the ‘stateful’ setting of the network. Again, since there were variable amounts of lengths and utterances per speaker, the data formatting would have been too time-consuming and prone to error.

When input data is scarce and formatting of the feature maps for the LSTM are straightforward, these temporal correlations between frames can then provide additional information, as shown in the case of one to four seconds worth of speech data where the LSTM outperformed the i-vector methodology.

### 4.3 Twin Voice Differentiation with Siamese LSTM Results

The next experiment involved investigating the performance of different speech features for improved twin voice differentiation. The table below presents the results from using the three most common speech features for SRE in a five-fold cross validation set-up.

#### 4.3.1 Comparison of Different Types of Speech Features in Twin Voice Separation

For most of the experiments, the network converged at approximately 50 epochs. The longer utterance data required approximately 25 epochs more. Table 4.3 displays the EER comparison for multiple
utterance lengths and features using a three-layer LSTM with 128 units in each layer on Condition 3 (read and conversational speech for training and testing, respectively).

<table>
<thead>
<tr>
<th>Seconds</th>
<th>MFCC</th>
<th>MFEC</th>
<th>RAW</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>21.07 ± 2.11</td>
<td>18.3 ± 1.81</td>
<td>28.66 ± 2.59</td>
</tr>
<tr>
<td>4</td>
<td>14.85 ± 1.40</td>
<td>14.21 ± 1.42</td>
<td>23.07 ± 1.77</td>
</tr>
<tr>
<td>6</td>
<td>10.22 ± 1.06</td>
<td>8.57 ± 0.91</td>
<td>17.5 ± 1.49</td>
</tr>
</tbody>
</table>

Table 4.3. EERs for Genuine/Imposter Pair Comparison in Siamese Network for Various Lengths and Features

As expected with the feature comparison of experiment two, the MFECs performed the greatest with an EER of 8.57% with the longest utterance value of 6 seconds, even though the amount of training samples was slightly reduced. This is reasonable since the feature vector is of greater dimension (roughly 3x long as the MFCCs). The MFCCs encompass a final computation of the DCT which separates the supposed glottal source excitation from the vocal tract filtering to improve inter-speaker discrimination. However, in the emerging deep learning approach to speaker recognition, MFECs are the pre-dominantly used feature [11] for its effectiveness and this research effort corroborates this trend. Even though the glottal source and vocal tract are components of the vocal production anatomy (and hence more similar between identical twins) the findings in this thesis conclude that both components are valuable for improved twin voice differentiation. The worst performing feature was the raw audio data. In fact, no past work has shown improvements with raw-waveform over a log-mel trained neural network [86] and, again, the findings of this research effort suggest this to be valid. Figure 4.2 depicts the results of the experiment with a box and whisker plot of EERs.
From the box and whisker plots above, the input raw data features consistently and decisively perform worse than the other two spectral features. For data lengths of two and four seconds, the MFECs tend to, on average, result in better recognition performance but, due to the standard deviations of the results, a concrete conclusion cannot be made of a feature’s effectiveness over another. The six second comparisons, however, tend to show the superiority of MFECs over MFCCs. As the length of data becomes larger, the quartile range of the EERs seem to narrow. An explanation may be that the pair-wise sample segments become longer leading to less random variation between samples in addition to the
LSTM taking advantage of more contextual data. The effectiveness of the MFECs can be further seen in Figure 4.3 which shows the distribution of genuine and imposter pairs before and after being processed by the network. The output of the network was measured from the output activation of the densely connected layer. Scores consist of the Euclidean distance between frames of MFEC features and output activations of each respective network before the scoring module. While the input representations of the speaker classes are initially completely overlapping due to high similarity between intra-speaker and intra-twin feature distribution, the network learns to differentiate these difficult classes with reasonable accuracy.

As with all generative training methods, the proportions of multi-class data are very important in reliable and accurate training. For instance, during training of the network, a major concern was the presentation of imbalanced class ratios due to the variable nature of the available data per speaker which could have caused a bias for a certain speaker. This was not necessarily a problem with experiment two since the ratio of positive and negative samples were near unity.
Chapter 5:
Conclusion
5.1 Summary of Results

The main goal of this research effort was to determine if identical twins posed a problem to SR and to design and evaluate a system that could take advantage of certain speech features as well as evaluate those features believed to be more discriminatory between twins. Such a system, the LSTM, proved to be more effective than baseline methods when speech utterances were four seconds or less. This is attributed to the leveraging of additional sequential information the i-vector is unable to consider. Furthermore, the i-vector cannot gather sufficient data when speech utterances are sparse resulting in poor performance. However, when speech utterances were greater than four seconds, the baseline i-vector algorithm outperformed the LSTM network, perhaps due to improper input data formatting and tracking of lengthy utterances. Even with large variations in data lengths (from 30s to 60s), results from the LSTM did not drastically improve. This does not mean the deep recurrent method can never be a superior alternative to Gaussian subspace methods in this setting. Refinements to the LSTM architecture and hyper-parameters may serve to drive down the EER in future investigations. Both systems, however, show that the inclusion of identical twins in a speaker database degrades the recognition performance in comparison to a non-twin database. In comparison to other twins’ voice studies, the inclusion of more twins from 35 to 334 resulted in a higher EER than [20] from roughly 0% to 0.569%.

According to experiment two, investigation into the types of features and sequence lengths suggest that more contextual data results in improved separation of genuine and imposter classes. This is evidenced by the performance gains of 18.3% to 8.57% EERs for the MFEC features using two and six seconds worth of data, respectively. Although the inclusion of more speech data produced marginally better results in class separation, the increase was not proportional to that of the i-vector in the previous experiments. As the length of the utterances become longer, however, this correlation may be poorly tracked due to the implementation of the LSTM in this research effort or a short-coming of the BPTT mechanism. The i-vector drastically improves when input data becomes more abundant.
5.2 Future Directions

A twin voice corpus that collects metadata such as place of birth and place of living could be useful for further investigation into dialects. It would also be interesting to collect voice data from individuals younger than 18 years of age as the voice changes more drastically during these years. Due to standards on the collection of human subjects, we were not able to acquire this type of data.

Recurrent Siamese networks would excel in other applications of separating highly similar temporal sequences such as music copyright infringement cases where disputes are made based on the similarity of one piece of music to another. With the amount of current and past lawsuits concerning music copyright infringement, a robust and effective algorithm for resolving these lawsuits could be a lucrative commercial endeavor. Another interesting direction would be to combine frequency domain convolutional neural networks.
References


[73] Ng, Andrew. “Gradient Descent.” 01 and 02: Introduction, Regression Analysis, and Gradient Descent. URL: http://www.holehouse.org/mlclass/01_02_Introduction_regression_analysis_and_gr.html


