DEVELOPING ROBUST TEXT INDEPENDENT SPEAKER RECOGNITION USING DEEP LEARNING MODELS

WONDIMU LAMBAMO ANITO

A DISSERTATION SUBMITTED TO
THE DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING

Presented in Fulfillment of the Requirement for the Degree of Doctor of Philosophy in Computer Science and Engineering

OFFICE OF GRADUATE STUDIES
ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY

June, 2024

Adama, Ethiopia

DEVELOPING ROBUST TEXT INDEPENDENT SPEAKER RECOGNITION USING DEEP LEARNING MODELS

WONDIMU LAMBAMO ANITO

SUPERVISORS:

1. RAMASAMY SRINIVASAGAN (Professor): MAIN SUPERVISOR
2. WORKU JIFARA (Associate Professor): CO-SUPERVISOR

A DISSERTATION SUBMITTED TO
THE DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING

Presented in Fulfillment of the Requirement for the Degree of Doctor of Philosophy in Computer Science and Engineering

OFFICE OF GRADUATE STUDIES
ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY

June, 2024

Adama, Ethiopia

Declaration

I hereby declare that this Dissertation entitled ***"Developing Robust Text Independent Speaker Recognition Using Deep Learning Models"*** is my original work. That is, it has not been submitted for the award of any academic degree, diploma or certificate in any other university. All sources of materials used for this thesis have been duly acknowledged through appropriate citations.

Wondimu Lambamo	_____	June 25, 2024
Student Name	Signature	Date

Recommendation

I/we, the supervisor(s) of this dissertation, hereby certify that I/we have read and revised the dissertation entitled ***"Developing Robust Text Independent Speaker Recognition Using Deep Learning Models"*** prepared under my/our guidance by ***Wondimu Lambamo*** submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science and Engineering. Therefore, I/we recommend the submission of the dissertation to the department for further review and defense.

Ramasamy Srinivasagan (Professor)	_____	June 26, 2024
Major Supervisor	Signature	Date
Worku Jifara (Associate *Professor*)	_____	June 26, 2024
Co-supervisor	Signature	Date

Approval Page of Ph.D. Dissertation

I/we hereby certify that the recommendations and suggestions made by the board of examiners are appropriately incorporated into the final version of the dissertation entitled ***"Developing Robust Text Independent Speaker Recognition Using Deep Learning Models"*** by ***Wondimu Lambamo***.

Ramasamy Srinivasagan (Professor)	_____	June 26, 2024
Major Supervisor	Signature	Date
Worku Jifara (Associate *Professor*)	_____	June 26, 2024
Co-supervisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the dissertation open defense by ***Wondimu Lambamo*** have read and evaluated the dissertation entitled ***"Developing Robust Text Independent Speaker Recognition Using Deep Learning Models"*** and examined the candidate during open defense. This is, therefore, to certify that the dissertation is accepted for partial fulfillment of the requirement of the degree of Doctor of Philosophy in Computer Science and Engineering.

Mesfin Abebe (Associate Professor)	_____	_____
Chairperson	Signature	Date
Teklu Urgessa (Associate Professor)	_____	_____
Internal Examiner 1	Signature	Date
DP Sharma (Professor)	_____	28/06/2024
External Examiner 1	Signature	Date
Asrat Mulatu (Associate Professor)	_____	_____
External Examiner 2	Signature	Date

Finally, approval and acceptance of the dissertation is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the candidate's Department Graduate Council (DGC) and School Graduate Committee (SGC).

_____	_____	_____
Department Head	Signature	Date
_____	_____	_____
School Dean	Signature	Date
_____	_____	_____
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGEMENT

Jesus the son of God, I just thanks for the unspeakable gifts received from you. When I think your mercy, love, and forgiveness always get excited and glorify you. It is so much more than the tangible blessings that I have also received beyond all measure. That this work has ended up well was out of range of what I could do on my own. I would like to glorify you in my entire life for all your blessing.

This dissertation have been completed with the support, cooperation and guidance of many people. Foremost, I would like to take this opportunity to express my deepest gratitude to my supervisors, Prof. Ramasamy Srinivasagan and Assoc. Prof. Worku Jifara for accepting to supervise and providing me the opportunity to research on the topics of my interest. I owe special thanks to both for their guidance, support, patience and cooperation at each level of this dissertation progress. Their trust and encouragement in me motivated me to finalize the dissertation effectively.

I would also like to express my gratitude to my parents, to my father Lambamo Anito and to my mother Shogame Abide. Dear my dad, I love you and always remember you in my entire life for the love you have shown to me. Although you couldn't participate in this happiness all this success is dedicated to you because you were my legend. My father have been motivating and showing me how much education is powerful to change human life. Dear mom, I am very glad for getting this chance to thank for all your sacrifices you paid to get this achievement and your strength. You gave me unconditional love at every challenges, downs and ups we faced at the early age. The challenge we face at the early age and your strength pushed me to this achievement. Moreover, I would like to thank my dear brothers and sisters for their love, support and respect to each other which made me happy and helped to pass all the challenges.

My heartfelt thanks goes to Adama Science and Technology University for all the cooperation, support and guidance during the stay in the university. Especially, I would like to thank Computer Science and Engineering department heads and staffs for sharing knowledge, skill and experience in all circumstances. It would not have been possible getting the opportunity of PhD at early age without the effort of the department heads and staffs in launching PhD program

in computer science and engineering. I got a lot of academic and personal experience from them during the stay at the university. Their cooperation, support, and guidance in all aspect encouraged me to complete the dissertation.

Also, I would like to thank Wachemo University for providing sponsorship for my PhD study and appropriate time for the study leave. The completion of this work would not have been possible without the support of the Wachemo University.

I also, extend my gratitude to my former colleagues (i.e., Wollega University Computer Science staffs) and current colleagues (i.e., Wachemo University Computer Science staffs). We have shared a lot of happiness and challenges in the academic and social conditions. Without sharing the knowledge, idea, experiences it couldn't be possible to reach this stage. The science I used in this work and in my life was obtained from these colleagues.

With great pleasure I would like to thank my home church, Anidelicho Full Gospel Believers church for the support by praying day and night for my success. Without the support of the church by praying and motivating I could not defeat this world's challenge. Their praying strengthen me at every challenges, ups and downs. Jesus Christ the son of the living God provided me wisdom and strength by their praying.

Lastly, but not least I would like to express my special gratitude to Mr. Abera Osamo the former teacher of Kecha Elementary School. Reaching this stage couldn't have been possible without his early support to return back to my grade one class. Because of class room limitation the director of the school returned me and two female students who have smaller ages to KG. After two months Mr. Abera Osamo also returned us to the grade one class because of free space in the class. His support motivated me at every stage of my study and I am very glad having such a father teachers in my country.

Table of Contents

List of Tables

List of Figures and Illustrations

List of Acronyms and Abbreviations

2DCNN	Two Dimensional Convolutional Neural Network
Adam	Adaptive Momentum
ANN	Artificial Neural Network
BGD	Batch Gradient Descent
BiGRU	Bidirectional Gated Recurrent Unit
BiLSTM	Bidirectional Long Short Term Memory
CNN	Convolutional Neural Network
DCT	Discrete Cosine Transform
DNA	Deoxyribonucleic Acid
ECAPA	Emphasized Channel Attention, Propagation and Aggregation
DWT	Dynamic Window Time
ECG	Echocardiography
EEG	Electroencephalogram
EER	Equal Error Rate
ERB	Equal Rectangular Bandwidth
FC	Fully Connected
FFT	Fast Fourier Transform
GFCC	Gammatone Frequency Cepstral Coefficient
GMM	Gaussian Mixture Model
GMM-UBM	Gaussian Mixture Model/Universal Background Model
GRU	Gated Recurrent Unit
GTFB	Gammatone Filter Bank
HMM	Hidden Markov Model
LPC	Linear Predictive Coding
LPCC	Linear Predictive Cepstral Coefficient
LSTM	Long Short Term Memory
MFCC	Mel Frequency Cepstral Coefficient
PLDA	Probabilistic Linear Discriminant Analysis
PLP	Perceptual Linear Prediction
ReLu	Rectifier Linear Unit

ResNet	Residual Network
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SNR	Signal to Noise Ratio
SOP	Self-Organizing Map
SVM	Support Vector Machine
TDNN	Time Delay Neural Network
TIMIT	Texas Instruments Massachusetts Institute of Technology
VGG	Visual Geometry Group
VoIP	Voice Over Internet Protocol
VQ	Vector Quantization

Abstract

Speaker recognition is the process of classifying/identifying a person from others based on speech characteristics. It has crucial applications in security, surveillance, forensics and financial transactions. The performance of the speaker recognition systems was good in the clean speech and without mismatch. However, the performance of the speaker recognition systems gets degraded under noisy and mismatched conditions. Several studies have been conducted in speaker recognition using machine learning methods to enhance performance in noisy environments. Recently, deep learning models outperformed machine learning methods in speaker recognition. Moreover, hybrid models of convolutional neural networks (CNN) and enhanced variants of recurrent neural networks (RNN) have shown better performance in image classification and natural language processing. However, only limited attempts have been conducted using hybrid CNN and RNN variants to enhance speaker recognition performance under noisy conditions. The features which have good performance in speaker recognition using machine learning methods were not as effective as spectrogram and cochleogram in deep learning-based speaker recognition. However, the noise robustness of the cochleogram and spectrogram was not analyzed in speaker recognition using deep learning models to employ the more robust feature in noisy conditions. In this study, a text-independent speaker recognition using deep learning models have been developed for noisy conditions. First, the noise robustness analysis of cochleogram and spectrogram in speaker recognition using deep learning were conducted to select the more robust feature. Then, the speaker recognition model using hybrid CNN and enhanced RNN variants have been developed to enhance the performance under noisy conditions. The enhanced RNN variants employed in this study include long short-term memory (LSTM), bidirectional LSTM (BiLSTM), gated recurrent unit (GRU) and bidirectional GRU (BiGRU). Cochleogram have shown better noise robustness at each signal-to-noise ratio (SNR) level and are used as an input in each of the speaker recognition models developed in this study. The experiments have been conducted on the VoxCeleb1 audio dataset with real-world and white Gaussian noises at the SNR level of -5dB to 20dB and without additive noises. The speaker recognition using hybrid CNN and BiGRU on the cochleogram input was proposed for noisy conditions in this study because of its higher performance. The proposed model has achieved speaker identification accuracy of 93.15% to 98.60% on the dataset with real-world noise at SNR of -5dB to 20dB, respectively and 98.85% on the dataset

without additive noise. The equal error rate (EER) of the proposed model on the dataset with real-world noise at SNR of -5dB to 20dB ranges from 10.55% to 0.47%, respectively and 0.37% on the dataset without additive noise. The comparison with the existing works also confirmed that the proposed model has higher performance than existing works.

Keywords: *Speaker recognition, cochleogram, spectrogram, convolutional neural network, bidirectional gated recurrent unit, real world noise, white gaussian noise*

CHAPTER ONE

1. INTRODUCTION

1.1. Speech Processing

Speech is the most natural and comfortable form of communication for human beings (Delna, 2019). Nowadays the advancement in technologies has increased the need for human-to-machine communication. Speech is also very important for human-to-machine communication because human beings mostly prefer speech-based communication unless they are impaired in speech communication (Schafer, 1994). For human-machine communications speech signals should be transformed into appropriate forms for both human and machine. Speech processing is the mechanism to transform natural speech signals into appropriate formats for the analysis of speech components, understanding and generating human speech (Jacob Benesty, 2008). It is very important for efficient use of storage, computational and transmission resources during communication or interaction. Various applications of speech processing includes speech analysis or synthesis, recognition, and coding (Ben, 2008). Speech analysis is the process of extracting or obtaining important information from the speech signal for specific applications (Toledano, Ramos, Gonzalez-Dominguez, & González-Rodríguez, 2009). It mainly focuses on the speech production mechanism to simulate the effect of each speech production system. Speech Coding is also the mechanism of converting speech signals into more compressed forms to save storage, minimize computation costs and increase transmission speed (B, Anees, & Yadava, 2023).

In recognition using the speech characteristics, speech processing has crucial role in obtaining important information from the speech. Some of the basic recognitions in which speech processing could be employed include speech recognition, speaker recognition, language recognition and gender. To facilitate human-machine communication, speech recognition converts speech signals into text form (Abdel-Hamid, et al., 2014). Speech is an acceptable method of communication for human beings and text is the appropriate format for representing information on the computer. Speech is also processed to extract language information to understand the message (Li, Ma, & Aik, 2013). Speaker recognition performs classification, detection, segmenting and clustering the speaker by speech processing and using algorithms to train and classify the speaker. In this study,

the researchers have focused on speaker recognition rather than other speech processing applications. The detailed classification of speech processing applications which was adopted from the study (Campbell, 1997) is presented in Figure 1.

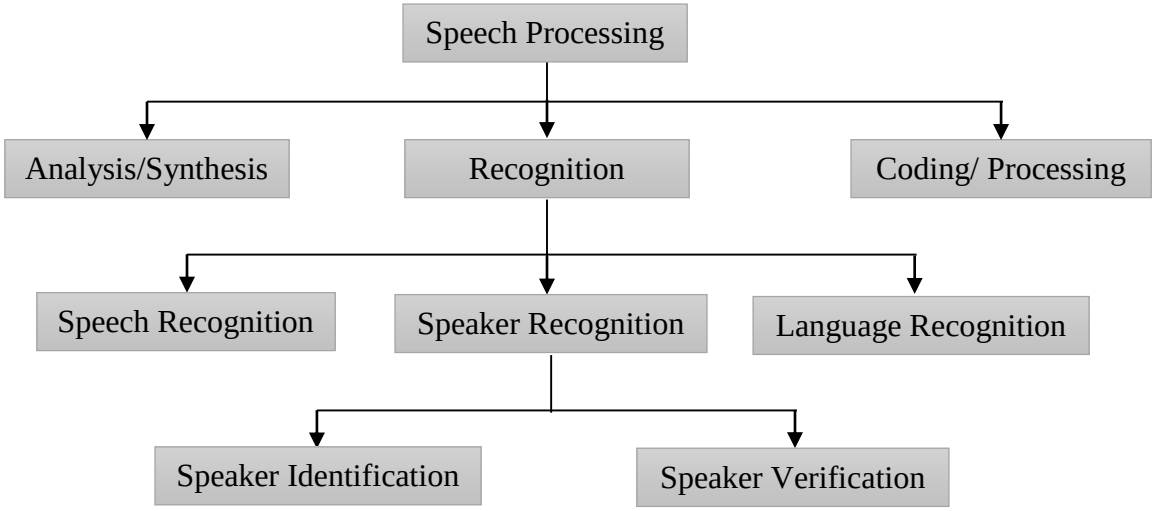

Figure 1: Speech Processing Applications

1.2. Speaker Recognition

Biometrics is a mechanism by which computers identify people by measuring some aspect of an individual's anatomy/physiology and other behavioral characteristic (Institute, 2024). It has crucial applications in classifying, identifying, and detecting an individual based on physiological and behavioral characteristics. Some of the basic biometrics which has been employed in real-world applications include face recognition, deoxyribonucleic acid (DNA) recognition, fingerprint recognition, gait recognition, iris recognition and speaker recognition (Mróz-Gorgoń, Wodo, Andrych, Caban-Piaskowska, & Kozyra, 2022).

Speaker recognition (Kinnunen & Li, 2010) is a biometrics that uses speech characteristics for classifying or identifying a person from other speakers. Speech contains various information about the speaker, language, gender, and others which is important for speech-processing applications (Safavi, Russell, & Jančovič, 2018). The speech of two individual cannot be the same in any circumstances because of the unique structure, size and shape of the speech production system components such as the vocal tract, larynx and others (Kinkiri & Keates, 2020). In addition, every person has a unique way of speaking style, timing between two words, choice of word, and so on

(Gustavo, 2007). The speaker recognition systems extract important physiological and behavioral characteristics of the speaker from the speech for classifying and identifying persons.

Speech-based systems are easily accessible and acceptable by the users and implementation requires resources of low cost. These advantages increased the demand for systems that use speech in various application areas. Therefore, speaker recognition attracted the interest of researchers, organizations or companies, and users of various areas.

In the speaker recognition system, there are two basic components such as feature extraction and classification algorithm (Paulose, Mathew, & Thomas, 2017). Both components have a major role in the performance of the speaker recognition systems. The better the feature extraction technique and classification algorithm performs better speaker recognition. Feature extraction is crucial for obtaining important attributes for training classification algorithms or models (Labied & Belangour, 2021). The classification algorithms learn the pattern or correlation between the features of the speech for classifying the speaker. There are two types of classification algorithms widely used in speaker recognition which are machine learning algorithms (Ayadi, Hassan, Abdelnaby, & Elgendy, 2017) and deep neural networks (India, Safari, & Hernando, 2019).

Machine learning algorithms use statistical methods to learn from the feature patterns in the data and generalize unseen data, then perform the tasks without explicit programs. Some of the machine learning algorithms employed in speaker recognition are Gaussian mixture model (GMM) (Kumar & Bhukya, 2022), support vector machine (SVM) (Wang J. , et al., 2017), and vector quantization (VQ).

The deep neural network also referred to as deep learning model (Mobiny & Najarian, 2018) is a complex interconnection of neural networks designed to solve complex problems based on the structure and function of the human brain and nervous system. Deep learning models can automatically learn complex pattern interdependency between the features during training for classifying or predicting (Shon, Tang, & Glass, 2019). A common type of deep learning models in speaker recognition include convolutional neural network (CNN) (Torfi, Dawson, & Nasrabadi, 2018), recurrent neural network (RNN) (Ye & Yang, 2021) and their variants. Recent studies (Ashar, Shahid, & Mushtaq, 2020), have shown that deep learning models have better performance than machine learning models in image classification, natural language processing, computer

vision, robotics, and speaker recognition. In this study, the researchers have focused on the deep learning models to develop a text independent speaker recognition model for noisy conditions.

1.2.1. Types of Speaker Recognition

Speaker recognition can be classified into speaker identification and speaker verification (Chauhan & Chandra, 2017). Speaker identification automatically classifies an individual who gives an utterance from a registered or trained set of speakers in the model. In speaker identification, the speaker is always considered as from the trained or registered speaker. The system determines the speaker as the one who has the maximum similarity ratio. In speaker identification, the likelihood ratio of the test sample is computed from each class trained in the model. The person who claimed with the utterance can be decided as the speaker class of the maximum likelihood ratio. Identification is also known as closed set recognition, in which each claimed utterance is expected to be from known speakers. Both verification and identification follow a sequence of steps such as speech pre-processing, feature extraction, training and classification (Chowdhury, Zunair, & Mohammed, 2020). Features extraction techniques and classification models are important components that have critical importance in the performance of the systems (Paulose, Mathew, & Thomas, 2017).

Speaker verification systems determine whether the person speaking to the system exists or not in the trained speaker database (Kanervisto, Vestman, Sahidullah, Hautamäki, & Kinnunen, 2017). In speaker verification, there are only two states of decision that either accept the speaker or reject. The likelihood ratio of test utterances is computed from each speaker class. Then the speaker with the maximum likelihood ratio is selected and compared with the threshold values to make a decision. Speaker verification is an open-set classification where samples are assumed to come from known speakers or unknown speakers. In this dissertation, the researchers focused on enhancing the performance of both speaker identification and verification under noisy conditions.

Moreover, the speaker recognition could be conducted either in text-dependent (Bhattacharya, Alam, Stafylakis, & Kenny, 2016) or text-independent speaker recognition (Ayadi, Hassan, Abdelnaby, & Elgendy, 2017). In text-dependent speaker recognition, the text of the training and test utterance should always be the same. This type of speaker recognition is very static, not attractive for the users and it is not a natural form of recognition for human beings. In text-

independent speaker recognition, the text of the training and recognition utterance should not be the same or not fixed. Human beings can recognize speakers irrespective of the content of the utterance and text. The text-independent method restricts speakers from uttering the text of their interest which reduces the interest of users and does not consider the real-world recognition scenario. The text-independent method is more flexible, follows the real-world scenario of recognition and attracts the interest of the users in various areas. Therefore, this dissertation focused on text-independent speaker recognition because its working principle was attractive and based on the real-world recognition scenario of human beings.

1.2.2. Applications of Speaker Recognition

Speaker recognition systems has a vital role in identifying, classifying, clustering, and detecting a speaker based on the speech characteristics. Therefore, speaker recognition has important applications in the real-world. Some of the basic speaker recognition applications in real-world include Access control (Karthik, Aju, & Anish, 2014) and (Selvan, Joseph, & Babu, 2013), forensic investigation (Han, et al., 2011), surveillance (Alegre, Soldi, Evans, Fauve, & Liu, 2014), and financial transactions (Singh, R.A.Khan, & Shree, 2012).

Surveillance: The advancement in technology increased the need to perform tasks remotely (Eva & Jozef, 2015). Some real-world scenarios need monitoring or recognizing individuals remotely by using speech characteristics without physical contact. For example, to detect terrorists from telephone communication speaker recognition systems are very important (Ramasubramanian, 2012). Speaker recognition systems play a great role in monitoring the activities, behavior, and information of the speaker. The information during the surveillance helps security agencies and other respective organizations to make decisions on the speaker. These activities are common in criminals' identification by using electronic eavesdropping.

Forensics: Recently, human-human communication through information communication technology and social media has been increasing from day to day (Hansen & Hasan, 2015). Some people transmit hate speech, harassment and insults during unhealthy communications. In such conditions, speaker comparison is very important for classifying the speaker with the claimed speech to make decisions. Speaker recognition systems help forensic investigators by automatically classifying the speaker based on speech characteristics.

Access Control: Some resources should be allowed for only privileged people. The resources could be physical devices (i.e., storage, houses, and cars), data, applications and networks (Wang, et al., 2015). Access control is a method of restricting access to sensitive information to only authorized users. This can be achieved through the use of passwords, multi-factor authentication, and role-based access control. These methods ensure that only those with the proper authorization can access sensitive data, reducing the risk of data breaches and unauthorized access. Speaker recognition systems are very important biometrics to implement access control in various real-world application areas (Alaliyat, Waaler, Dyvik, Oucheikh, & Hameed, 2021).

Front end of speech recognition: Speech and speaker recognition are dual research areas in the sense that speaker variability is one of the major problems in speech recognition, whereas in speaker recognition it is an advantage. Speaker recognition technology could be used to reduce speaker variability in speech recognition systems by speaker adaptation. For example, a speech recognition system could have a speaker gating unit that recognizes who is speaking. Then, the system could adapt its speech recognizer parameters to suit better for the current speaker, or to select a speaker-dependent speech recognizer from its database.

1.3. Motivation of the Study

Nowadays human-to-machine communication is increasing from day to day for various reasons. For example, most people request services from organizations such as banks (Nhat, 2024), business applications (Mikel, 2024), and education (Marcela Hernandez-de-Menendez, Escobar, & Arinez, 2021) through online platforms that require personal identification. Most of the human-machine interaction requires recognizing the person based on what they know (e.g., patterns, passwords, and secret questions), what they have (e.g., keys, cards, tokens) and what they are (Guennouni, Mansouri, & Ahaitouf, 2019). What they have also known as biometrics, which uses the physical and behavioral characteristics of the individuals to recognize. Both what a person knows and has can be forgotten and stolen which may restrict a person from getting the required access/service (Rhynearson, 2024), whereas biometrics cannot be forgotten, stolen and user continent. These importance of biometrics motivated the researchers to conduct in speaker recognition because speaker recognition is one of the biometric techniques.

During the interaction between the human beings using speech, the auditory system naturally identifies a person based on the speech characteristics. Speech is one of the natural way of communication which could be preferred by majority of the human beings. For human-to-machine interaction, speaker recognition systems simulate human auditory system to identifying the person based on the speech characteristics. Speech is accessible from the remote or without standing in front of the machine which is very crucial for various types of applications such as surveillance, video conference and other remote communication and service provision. These advantages motivated the researchers to conduct in the area of speaker recognition.

Recently, the advancement in technologies has increased the demand for identification by using speaker recognition. For instance organizations, companies, business areas, financial institutions, hospitals and others have high demand in speaker recognition for authentication, security, surveillance and other purposes (Han, et al., 2011). In addition, deep learning model applications in various areas including speaker recognition have attracted our interest (McLaren, Lei, & Ferrer, 2015). The availability of public speech datasets to experiment with the speaker recognition models has also motivated the researcher to conduct studies in the area. The efforts conducted by other researchers in speaker recognition and the progress in the area have attracted the researcher's interest. The speaker recognition area has several gaps that need engagement of many researchers to achieve better performance in real-world applications. Only limited studies have been conducted in speaker recognition areas by using a deep learning approach to enhance performance under noisy conditions. These points motivated the researchers to conduct the study in speaker recognition to enhance the performance under noisy conditions using a deep learning model.

1.4. Statement of the Problem

Speech is highly variable and it could be affected by environmental conditions and the speaker's physiological or behavioral changes. The performance of the speaker recognition systems was good in the clean speech or without noise. However, the performance of the speaker recognition systems gets degraded under noisy and mismatched conditions.

Several studies have been conducted in speaker recognition to enhance the performance of the systems under noisy and mismatched conditions. Most of the previous studies conducted in speaker recognition for noisy and mismatched conditions employed machine learning methods and

hand-crafted feature. For instance, in the study (Kaur, Bhushan, & Singh, 2016), the speaker recognition system have been developed using GMM and GFCC features for noisy conditions. The study (Liu & K., 2018), conducted the noise robustness analysis of the MFCC and GFCC in speaker recognition using machine learning methods, the GFCC feature has shown better performance than the MFCC feature. In the research works (Li & Huang, 2011) and (Valero & Alias, 2012), GFCC features surpassed the accuracy of MFCC features in speaker recognition under environmental noises.

Recently, deep learning models outperformed machine learning models in speaker recognition (Costantini, Cesarini, & Brenna, 2023), image classification (Dong, et al., 2022) and natural language processing. Specifically, hybrid models of CNN and enhanced RNN variants have shown better performance than single models of each in the above stated areas. For instance, in the research work conducted by (Banjara, Mishra, Rathi, Karki, & Shakya, 2021), the hybrid model of CNN and standard RNN have achieved better performance than CNN and RNN in speech recognition. Another study (Islam, Islam, Hashim, Rashid, & Bari, 2022), demonstrated that the hybrid model of CNN with BiLSTM or BiGRU model has outperformed the existing works in arrhythmia detection using the ECG signals. The detailed report of the literature in chapter two of this study confirms that hybrid models of the CNN and enhanced variants of RNN have better performance in various areas. However, only limited attempts have been conducted in speaker recognition using hybrid CNN and enhanced RNN variants.

Moreover, the feature which were common in machine learning-based speaker recognition were not as effective as in deep learning-based speaker recognition. For example, in the study model (Jung, Heo, Kim, Shim, & Yu, 2019) and (Saritha, Azharuddin, Hussain, & Choudhury, 2022), the raw waveform of the speech have shown better performance than MFCC and GFCC features in speaker recognition using the deep learning. In another study (Bunrit, Inkian, Kerdprasop, & Kerdprasop, 2019), spectrogram outperformed raw waveform and MFCC features in speaker recognition using deep learning model. In the study (Sharan & Moir, 2019), cochleogram feature outperformed the spectrogram in acoustic event recognition by using the CNN model. The detailed report in the literature review (chapter two) also confirmed that the spectrogram and cochleogram are more effective in speaker recognition using a deep learning approach. However, the noise robustness of the cochleogram and spectrogram was not analyzed in speaker recognition using the

deep learning approaches at different level of SNR to select the better features for the noisy conditions. In this study, a text-independent speaker recognition model using deep learning models have been developed for noisy conditions. First, the noise robustness analysis of cochleogram and spectrogram in speaker recognition were conducted using deep learning models to select the more robust feature for the speaker recognition model development. Then, the speaker recognition models using hybrid CNN and enhanced RNN variants (i.e., LSTM, BiLSTM, GRU, and BiGRU) were developed using the more noise robust input for noisy conditions. The model with the highest performance was proposed in this study for speaker recognition under noisy conditions. The effectiveness of the proposed model was presented by comparing with the previous works.

1.5. Research Questions

1. Which type of speech representation or feature can achieve better performance in speaker recognition using a deep learning model under noisy conditions?
2. How speaker recognition model with better performance can be developed by using a deep learning methods for noisy conditions?
3. Which hybrid model of CNN and enhanced RNN variants could achieve the better performance in speaker recognition under noisy conditions?

1.6. Objectives of the study

1.6.1. General Objective

Generally, this study was aimed to develop text independent speaker recognition model using deep learning models to enhance the performance under noisy conditions.

1.6.2. Specific Objectives

The following list of specific objectives were achieved during the development of text-independent speaker recognition model using deep learning models.

- To prepare a datasets from the appropriate public speech datasets
- To generate Cochleogram and spectrograms from each utterance of the datasets.
- To analyze the noise robustness of the cochleogram and spectrogram in speaker recognition using deep learning.

- To develop speaker recognition models for noisy conditions using hybrid CNN and enhanced RNN variants.
- To evaluate the speaker recognition models at different levels of SNR.
- To compare the proposed model's performance with the state of the arts in the area

1.7. The Scope of the study

The speaker recognition systems can be developed using the methods text-dependent and text-independent. To limit the scope this dissertation was focused on the text-independent method because of its flexibility, user attraction and support of real-world recognition scenarios. Since the objective of this dissertation was to enhance the performance of the speaker recognition models under noisy conditions, the features and models were evaluated on the real-world noise and/or white Gaussian noises. There are several real-world noise types which could be added to the original datasets and evaluate the speaker recognition models, but the most common types of real-world noises such babble, restaurant and street noises were focused to evaluate the performance of the models and features. There are unlimited levels of signal-to-noise ratio in which the noise-added datasets can be generated from the original dataset. In this study, the dataset with real-world noise and white Gaussian noise were generated from the original dataset at the SNR level ranging from -5dB to 20dB in the interval of 5dB. In speaker recognition, there are different types of features (input of the models). In our work, the spectrogram features that were most frequently used input in the deep learning-based speaker recognition and cochleogram which was a more noise-robust feature for speaker recognition under noisy conditions using deep learning were selected. Moreover, there are several deep learning model architectures which could be applicable for speaker recognition, to limit the scope of the study the most common types of deep learning model architectures such as convolutional neural network architectures and recurrent neural network architecture were focused. In this study, the CNN architectures were employed to analyze the noise robustness of the cochleogram and spectrogram in speaker recognition. Moreover, the speaker recognition model using hybrid CNN and enhanced variants of RNN models have been developed for noisy conditions. Several public speech datasets are available to experiment the speaker recognition performance of the models. Some of them have only a few utterances in each speaker class which were primarily prepared for machine learning methods and others have a large number of samples for each speaker in the dataset which were prepared targeting deep learning

models. Some of them were collected from a clean environment, unequal gender distribution, and lack of accent variations. In this study, the VoxCeleb1 audio dataset was selected because it was open source dataset which have large number of speaker classes with significant number of utterances in each classes and it was prepared for speaker recognition models experiments.

1.8. Limitations of the study

The dataset selected for this study and other datasets available for speaker recognition didn't not considered speaker emotions during preparation. Although the selected dataset have different types of emotions of the speaker it is not explicitly categorized based on these emotions. Therefore, this study has limitation of investigation in the effect of the speaker emotion in the features and speaker recognition models. The robustness of the cochleogram and spectrogram for speaker emotion changes and its effect in the speaker recognition were not considered because of the limitation in discussed above in the dataset. Moreover, most of the datasets available in public did not prepared considering the aging varieties in the samples. This study have limitation of performance evaluation of the models and features in the effect of aging in speaker recognition.

1.9. List of Publications

List of articles published from the results of this dissertation are listed as follows:

I. Wondimu Lambamo, Ramasamy Srinivasagan and Worku Jifara "*Analyzing Noise Robustness of Cochleogram and Mel Spectrogram Features in Deep Learning Based Speaker Recognition*" Applied Sciences, 2022, Vol. 13, Issue 569, DOI: https://doi.org/10.3390/app13010569
II. Wondimu Lambamo, Ramasamy Srinivasagan, Worku Jifara, and Ali Alzahrani "*Speaker identification under noisy conditions using hybrid convolutional neural network and gated recurrent unit*" IAES International Journal of Artificial Intelligence, 2024, Vol. 13, Number 1, DOI: http://doi.org/10.11591/ijai.v13.i1.pp1050-1062
III. Wondimu Lambamo, Ramasamy Srinivasagan, and Worku Jifara "*Speaker Identification under Noisy Conditions using Hybrid Deep Learning Model*", Communications in Computer and Information Science (CCIS), 2024, DOI: https://doi.org/10.1007/978-3-031-57624-9_9

1.10. Contributions

The main focus of this dissertation was developing robust text-independent speaker recognition model using a deep learning approaches. The publications selected for this dissertation were original research papers in the speaker recognition areas. For all publications, the authors took care of the implementation, result analysis, and major paper writing. The supervisors (i.e., Main supervisor and co-supervisor) helped with reviewing, editing, suggesting and guiding during manuscript writing and publication. The researchers' contribution to each paper is detailed below:

In publication **I**, the noise robustness of the cochleogram and spectrogram features were analyzed in speaker recognition by using a deep learning approach. This helps scientific society in the area to easily select appropriate input/features during the development of speaker identification and verification models for a noisy conditions using deep learning approaches. Both features were analyzed in speaker identification and verification. The CNN architectures such as Basic 2DCNN, VGG-16, ResNet50, ECAP-TDNN and TitaNet models have been employed for the analysis. The evaluation result confirmed that cochleogram is more robust than the spectrogram in speaker recognition under noisy conditions using deep learning.

In publication **II**, the speaker identification model was developed by using a hybrid CNN and GRU model with the cochleogram input for the noisy conditions. The model was evaluated on the dataset with real-world noise and white Gaussian noises at different levels of SNR. In this publication, the proposed model have achieved better performance than existing works in speaker recognition. Moreover, the researchers have shown the direction how hybrid models of CNN and enhanced RNN variants have better performance in speaker recognition.

In Publication **III**, Bidirectional gated recurrent units have been employed in combination with the CNN model in speaker recognition with the cochleogram input. This publication was aimed to show the impact of forward and backward direction features correlation effect on speaker recognition. The result confirmed that the model proposed in this publication have better performance than other models developed in this study and the existing works.

1.11. Dissertation Outline

The remaining sections of the dissertation were organized as follows:

- Chapter two presents a brief literature review of various concepts and related works conducted in speaker recognition. In section 2.1, the concept of deep learning model were discussed. In section 2.2, the detailed review on convolutional neural network were presented. Some of the standard CNN architectures like visual geometry group (VGG-16 and VGG-19) were presented in detail. Section 2.3, presents the recurrent neural network and its varieties such as LSTM, BiLSTM, GRU and BiGRU architectures. The detailed concept of speech representation methods for speaker recognition and various features such as MFCC, GFCC, spectrogram and cochleogram were presented in section 2.4. In section 2.5, the speaker recognition systems developed by using machine learning methods were presented in depth. Section 2.6, presents the speaker recognition models conducted by using deep learning models. The speaker recognition systems developed using hybrid approaches have been reviewed and presented in section 2.7. In section 2.8, public datasets appropriate for speaker recognition system evaluation were presented. The related works conducted in the speaker recognition area were presented in section 2.9.
- Chapter Three presents the methodology followed to develop the text independent speaker recognition model using deep learning models for noisy conditions. In section 3.1, introduced the methodology followed in this study. Section 3.2, presents the dataset preparation methods to evaluate the models and features. In section 3.3, the cochleogram and spectrogram generation process for speaker recognition was presented. The methods followed for noise robustness analysis of the cochleogram and spectrogram in speaker recognition using deep learning was presented in section 3.4. Another section 3.5, presents the speaker recognition model development method and the proposed speaker recognition model in this study for noisy conditions. Section 3.7, presents implementation details of the each experiment.
- Chapter Four discusses the Results and Discussion. In section 4.1, the results of noise robustness analysis of cochleogram and spectrogram were presented. Section 4.2, presents the results of speaker recognition using CNN-LSTM, CNN-BiLSTM, CNN-GRU and CNN-BiGRU.

- The conclusion and recommendations were also presented after the last chapter
- Sample codes and other important information were presented in the appendix section of this dissertation.

CHAPTER TWO

2. LITERATURE REVIEW

2.1. Deep Learning Models

Deep learning models are the subset of machine learning methods that have the interconnection of a large number of neural networks to solve complex problems (Soori, Arezoo, & Dastres, 2023). Most of the deep learning models automatically learn the pattern between the features from the input data during training. In deep learning, there is no explicit program to perform each of the tasks but the model itself learns from the patterns in the data (Costantini, Cesarini, & Brenna, 2023). The neural networks in deep learning help the model to learn patterns in the data which works based on the structure and functions of the human brain and nervous systems. The relationship between artificial intelligence, machine learning and deep learning (Solutions, 2023) is presented in Figure 2.

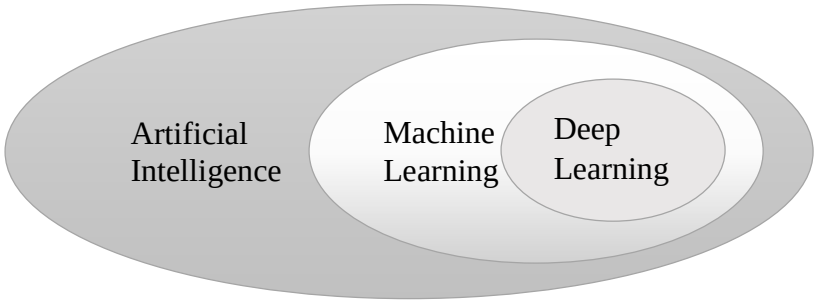

Figure 2: Relationship between deep learning, machine learning and artificial intelligence

Recently, deep learning models have been preferred in various application areas because of their performance (Group, 2022). Deep learning models automatically extract important features from the input data and learn complex representations of the data. Unlike machine learning, deep learning models do not require human intervention in feature extraction. Deep learning models have vital applications in areas such as computer vision, natural language processing, robotics, speech recognition and other applications (Solutions, 2023). In computer vision, deep learning models could be employed in object detection and recognition, image classification and segmentation. In natural language processing deep learning models are used for automatic text generation, language translation, sentiment analysis, and speech and speaker recognition tasks. In

addition, deep learning could have important applications in game playing, robotics and control systems. In these areas, the models could be employed for recognition, classification, clustering, prediction and segmenting. Some of the basic deep learning model architectures are convolutional neural network (CNN) and recurrent neural network (RNN) (Shekhar & Roy,2021) which are discussed in the following subsections.

2.2. Convolutional Neural Network

Convolutional Neural Network (CNN) is a type of deep learning model which have a feed-forward interconnection of the neural network (Abdel-Hamid, et al., 2014). CNN models are also referred to as *convnets*. It is primarily developed to operate on image input or two-dimensional data. The models developed from CNN were also successful in time series data analysis (Banjara, Mishra, Rathi, Karki, & Shakya, 2021). Various architectures of the CNN model were widely applied in areas like medical image classification, computer vision, biometric recognition, speech recognition, and so on (Ashar, Shahid, & Mushtaq, 2020). The models have been employed to identify, recognize, detect, segment and classify objects in the images. Unlike traditional machine learning models, the CNN models automatically extract features from the input data without human intervention (Gurbuz, J.Gowdy, & Tufekci, 2002). The CNN models can extract a few trainable parameters by using weight sharing which has an advantage of saving computational resources (O'Shea & Nash, 2015). These also enhance the generalization ability of the model and avoid overfitting. The feature extraction layers and classification layers learn concurrently during the training of the CNN models. Depending on the specific applications, the CNN models with different numbers of layers (i.e.; simple to complex models) can be developed. The basic components of the CNN models are the convolutional layer, pooling layer and fully connected layer (Wang, et al., 2018). Moreover, CNN architectures consist of various functions and hyperparameters at different levels of the network or layers. Important functions in CNN architectures include activation function, loss function, optimizers and regularization functions (Salehghaffari, 2018). Various types of standard architectures of CNN were employed in speaker recognition, some of them are visual geometry group (VGG), residual network (ResNet), GoogLeNet, and AlexNet with various number of layers. Both VGG and ResNet are employed for evaluating the noise robustness of cochleogram and spectrogram features in speaker recognition

and their detail is presented below in this section. Sample CNN architecture block diagram which was adopted from (O'Shea & Nash, 2015) is presented in Figure 3.

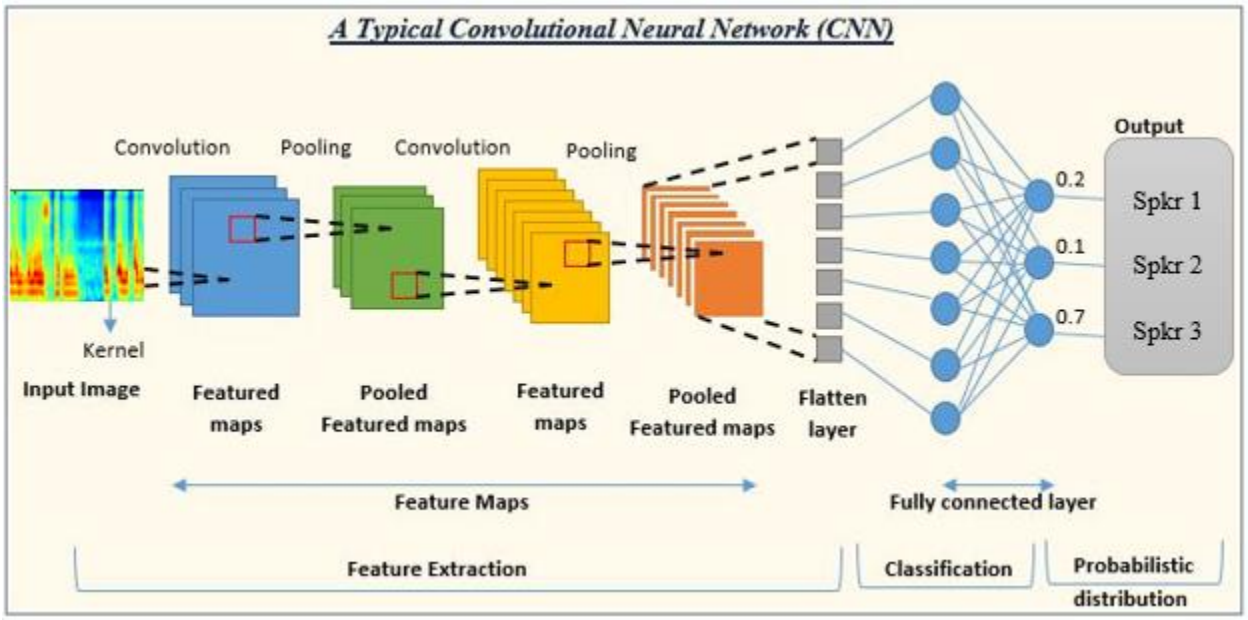

Figure 3: Convolutional Neural Network

Convolutional layers: The convolutional layer is the most important component in CNN models (Abdel-Hamid, et al., 2014). It consists of kernels or filters that contain the parameters learnable by the model during the training (Soleymani, Dabouei, Mehdi, Kazemi, & Dawson, 2019). The filters convolve with the input of the layer to extract the feature maps. In these layers, kernels with matrices of equal length and width are used to learn feature extraction. One or more convolution layers are interconnected in the model to extract the unique features from the input. The convolution layers at the beginning of the model extract the basic features such as textures, lines and edges, while the deeper layers extract more abstract features based on the information from the previous layers (McLaren, Lei, & Ferrer, 2015). In addition, padding is important to preserve the spatial dimensions of the input image after convolution operations on a feature map (Prachi, Nahiyan, Habibullah, & Khan, 2022). The padding adds extra pixels around the border of the input feature map before the convolution.

Pooling Layers: The feature maps obtained from the convolution layer are larger which overwhelms computational resources and causes model overfitting (Bhattacharya, Alam, Stafylakis, & Kenny, 2016). Pooling layers are crucial after each of the convolution layers to

minimize the dimension of feature maps. The pooling layer reduces the number of connections in the network by performing down-sampling and dimensionality reduction on the input data (Costantini, Cesarini, & Brenna, 2023). Concurrently, it maintains the majority of the dominant information (or features) in every step of the pooling stage. Its primary purpose is to alleviate the computational burden and address overfitting issues. The pooling operation produces output feature maps that are more robust against distortion and errors in individual neurons. MaxPooling and AveragePooling have been widely employed in most of the CNN architecture applications (Chowdhury, Zunair, & Mohammed, 2020).

Fully Connected Layers: Every CNN model uses a fully connected layer at the end of the model (Kim & Park, 2021). In a fully connected layer, all the neurons are interconnected to all the neurons in the preceding layer which is similar to the conventional multi-layer perceptron neural network (Soleymani, Dabouei, Mehdi, Kazemi, & Dawson, 2019). The FC layer receives input from the last pooling/convolutional layer, which is a vector created by flattening the feature maps. The FC layer serves as the classifier in the CNN, enabling the network to make predictions.

Activation Function: This function maps the input of the model into the output of the model (Bunrit, Inkian, Kerdprasop, & Kerdprasop, 2019). This means that the activation function decides whether or not to fire a neuron concerning a particular input by creating the corresponding output. The activation function must also have the ability to differentiate, which is an extremely significant feature, as it allows error back-propagation to be used to train the network. The basic activation function types are Sigmoid, Tanh, and ReLu (Ding, Chen, Gong, Zha, & Wang, 2020). In sigmoid, the input is real numbers and the output is between zero and one. In Tanh, the input is real numbers, the output is between -1 and 1. ReLu is mostly commonly used in the CNN context, its input is whole numbers and the output is positive numbers.

Loss Functions: The last fully connected layer in the CNN architecture performs the classification. Loss functions are commonly applied in the classification layer to compute the predicted error/loss that occurs during the model training with the samples (Chowdhury, Zunair, & Mohammed, 2020). The loss function provides the difference between the expected/actual output and the obtained value. The CNN architecture minimizes the loss by learning from the past patterns in the sample. Some of the commonly employed loss functions in deep learning models are Softmax/Cross-Entropy, Euclidean and Hinge loss functions. Softmax loss function has been used for multiclass

classification applications. It represents the output as the probability $p \in \{0, 1\}$. Euclidean loss is commonly employed for regression problems. Hinge loss is employed in problems related to binary classification.

Regularization: Both overfitting and under-fitting affect the performance of the machine learning and deep learning models in various applications. If the model is over-fitted it performs well on the training data and performance is degraded on the test or unseen data, whereas if the model is under-fitted it does not learn sufficient information from the training data and performs well in the test data (Singh, et al., 2021). The model is referred to as just-fitted if it executes well on both training and testing data. Deep learning models use the regularization function to handle over-fitting and under-fitting effects during training the model and predicting the samples (Bhattacharya, Alam, Stafylakis, & Kenny, 2016). Dropout and Drop weight are the two commonly used regularization functions in the deep learning model. Dropout randomly drops the neurons at each training epoch which distributes the feature selection power across the whole group of neurons and forces the model to learn different independent features. Drop weight randomly drops the weights or connection between the neurons at each epoch during the training.

Batch Normalization: This method ensures the performance of the output activations. This performance follows a unit Gaussian distribution (Hourri & Kharroubi, 2019). Subtracting the mean and dividing by the standard deviation will normalize the output at each layer. While it is possible to consider this as a pre-processing task at each layer in the network, it is also possible to differentiate and integrate it with other networks. In addition, it is employed to reduce the "internal covariance shift" of the activation layers. In each layer, the variation in the activation distribution defines the internal covariance shift. This shift becomes very high due to the continuous weight updating through training, which may occur if the samples of the training data are gathered from numerous dissimilar sources (for example, day and night images). Thus, the model will consume extra time for convergence, and in turn, the time required for training will also increase. To resolve this issue, a layer representing the operation of batch normalization is applied in the CNN architecture.

Optimizers: In deep learning models, optimizers are crucial for learning algorithm selection (Muayad, Sahib, & Adnan, 2020). The network parameters should always update through all training epochs, while the network should also look for the locally optimized answer in all training

epochs to minimize the error. Basically, deep learning models employ different types of optimizers based on their applications which include batch gradient descent (BGD), stochastic gradient descent (SGD), mini-batch gradient descent, momentum and adaptive Moment Estimation (Adam) (McLaren, Lei, & Ferrer, 2015).

2.2.1. Visual Geometry Group (VGG)

Visual Geometry Group (VGG), is one of the standard CNN model architecture. It is a multilayer model featured to simulate the relations of the network representational capacity in depth (Jakubec, Lieskovska, & Jarina, 2021). VGG models have better performance in object recognition. The size of the kernels in the VGG model is smaller which has the same influence as the large filters or kernels (Ali & Kumar, 2021). VGG architecture typically uses 3x3 filters to extract learnable parameters from the input data. Filters with small sizes could enhance the CNN performance. Using small-size filters reduces the dimension of feature maps which also reduces computation resources for training models. In addition, a 1x1 convolution is used in the middle of the convolution layers to filter the linear transformation of the input which also manages the network complexity. A 1x1 convolution in VGG architecture learns a linear grouping of the subsequent feature maps. Maxpooling is used after the convolution layers to reduce the dimension of the feature maps which reduces computational resource requirement. There are two standard VGG architectures such as VGG-16 and VGG-19.

VGG-16 architecture: This architecture is one of the varieties of standard VGG networks. It consists of 16 layers from which 13 layers are convolution layers and 3 layers are fully connected (R & Patilkulkarni, 2021). The convolution layers are grouped into five blocks, each of the first two blocks has two convolution layers followed by a maxpooling and the remaining three blocks have three convolution layers followed by a maxpooling. The VGG-16 architectures have three fully connected layers at the end. VGG-16 architecture has achieved the best performance in image recognition compared with the state of the arts. The VGG-16 architecture is illustrated graphically in Figure 4.

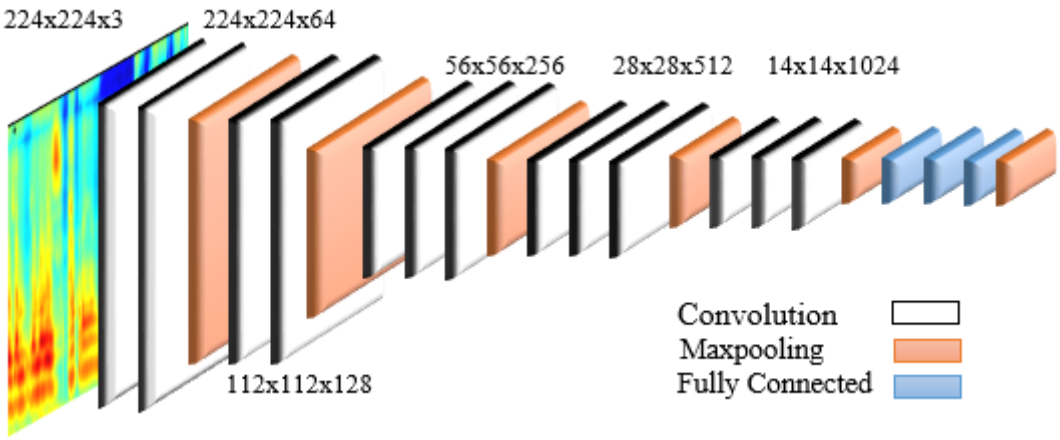

Figure 4: VGG-16 Architecture

VGG-19 architecture: This architecture of VGG has more number of convolution layers than VGG-16. VGG-19 models have nineteen layers from which sixteen layers are convolution and the remaining three layers are fully connected. This type of architecture is important for extracting patterns in complex datasets. Like VGG-16, each of the first two blocks of the VGG-19 contains two convolution layers followed by a maxpooling layer. Unlike VGG-16, each of the deeper three blocks of the VGG-16 contains four convolution layers followed by maxpooling. Similar to VGG-16, this architecture also contains three fully connected layers. The details of the interconnection of layers in the VGG-19 architecture are shown in Figure 5.

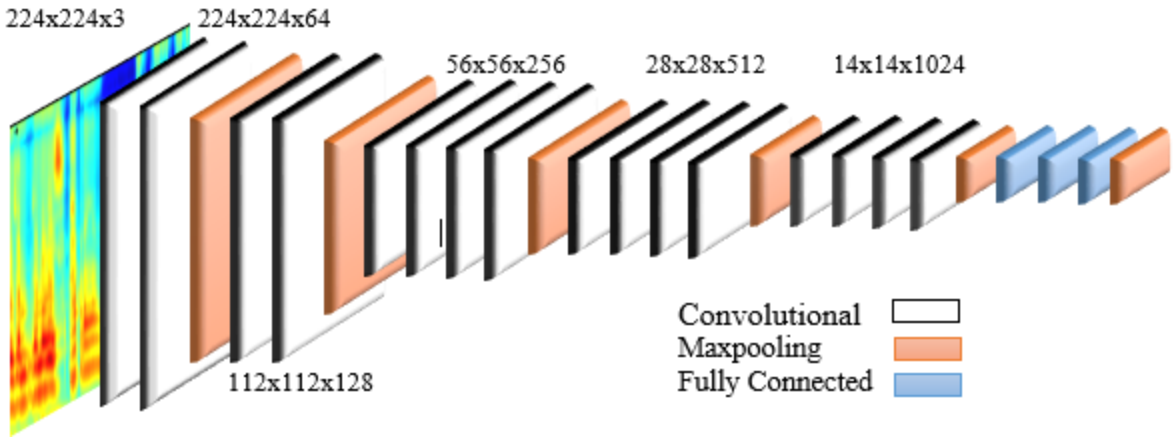

Figure 5: VGG-19 Architecture

2.2.2. Residual Network (ResNet)

Residual Network (ResNet) is one of the standard variants of CNN architectures designed to support large numbers (i.e., hundreds to thousands) of convolutional layers. Most of the CNN models employed in various applications have a limited number of convolution layers and they are not flexible enough to increase the number of layers, which limits the performance (Jakubec, Lieskovska, & Jarina, 2021). Moreover, using a large number of convolution layers in the previous models causes a gradient vanishing problem. ResNet models have a larger number of convolution layers and reduce the gradient vanishing problem by using the concept of bypass to address the problem of training deeper networks (Jung, Heo, Kim, Shim, & Yu, 2019). Various types of ResNet architectures can be designed based on the number of layers. ResNet34 and ResNet50 were commonly employed in various image classification and computer vision applications. In our study, ResNet50 architecture was employed for analyzing the noise robustness of cochleogram and Mel spectrogram features under different ratios of noise. The number of layers in ResNet50 models is 50 layers, 49 of them are convolutional layers and one is a fully connected layer. The first convolutional layer of the ResNet50 models has a filter size of 7x7 with strides of 2. The maxpooling layer with the filter size 3x3 and stride of 2 is inserted after the first convolutional layer. From the remaining convolutional layers 32 have kernel size of 1x1 and 16 convolutional layers have filters of size 3x3. The various ResNet architectures that were adopted from the study (Ruiz, 2018) are presented in Table 1.

Table 1: Different types of ResNet Architectures

Layer Name	Output size	ResNet-18	ResNet-34	ResNet-50	ResNet-101	RessNet-152
Conv1	112x112	7x7, 64, stride 2				
Conv2.x	56x56	3x3 maxpooling, stride 2				
		$\begin{bmatrix}3x3,64\\3x3,64\end{bmatrix}x2$	$\begin{bmatrix}3x3,64\\3x3,64\end{bmatrix}x3$	$\begin{bmatrix}1x1,64\\3x3,64\\1x1,256\end{bmatrix}x3$	$\begin{bmatrix}1x1,64\\3x3,64\\1x1,256\end{bmatrix}x3$	$\begin{bmatrix}1x1,64\\3x3,64\\1x1,256\end{bmatrix}x3$
Conv3.x	28x28	$\begin{bmatrix}3x3,128\\3x3,128\end{bmatrix}x2$	$\begin{bmatrix}3x3,128\\3x3,128\end{bmatrix}x4$	$\begin{bmatrix}1x1,128\\3x3,128\\1x1,512\end{bmatrix}x4$	$\begin{bmatrix}1x1,128\\3x3,128\\1x1,512\end{bmatrix}x4$	$\begin{bmatrix}1x1,128\\3x3,128\\1x1,512\end{bmatrix}x4$
Conv4.x	14x14	$\begin{bmatrix}3x3,256\\3x3,256\end{bmatrix}x2$	$\begin{bmatrix}3x3,128\\3x3,128\end{bmatrix}x6$	$\begin{bmatrix}1x1,256\\3x3,256\\1x1,1024\end{bmatrix}x6$	$\begin{bmatrix}1x1,256\\3x3,256\\1x1,1024\end{bmatrix}x23$	$\begin{bmatrix}1x1,256\\3x3,256\\1x1,1024\end{bmatrix}x36$

Conv5.x	7x7	$\begin{bmatrix}3x3,512\\3x3,512\end{bmatrix}x2$	$\begin{bmatrix}3x3,128\\3x3,128\end{bmatrix}x3$	$\begin{bmatrix}1x1,512\\3x3,512\\1x1,2048\end{bmatrix}x3$	$\begin{bmatrix}1x1,512\\3x3,512\\1x1,2048\end{bmatrix}x3$	$\begin{bmatrix}1x1,512\\3x3,512\\1x1,2048\end{bmatrix}x3$
	1x1	Average pool, FC, Softmax				

2.3. Recurrent Neural Network

Recurrent Neural Network (RNN) is the category of deep learning model which have been widely employed in time-series data analysis. RNN models have cycles within the network in that the current state of the network is directly or indirectly dependent on the previous information either for a short time or a long time (Abd, et al., 2020). This type of model is developed to simulate human memory for information storage (Mobiny & Najarian, 2018). It has memory which stores information about the previous sequences. RNN extracts the correlation between the features sequentially in a time series.

The standard RNN networks have short-term memory only that restricts the network from storing long-term correlation of features (Prachi, Nahiyan, Habibullah, & Khan, 2022). Therefore, standard RNN models can be affected by gradient exploding and vanishing in the features that have a long sequence of dependency. Standard RNN may fail to capture the long-term temporal correlation between the features because of its gradient vanishing and exploding problem. Enhanced variants of RNN have been employed to solve the problem of gradient vanishing and exploding of standard RNN (Hu, Si, Luo, Tang, & Jian, 2021). Enhanced variants of RNN such as Long Short Term Memory (LSTM), Bidirectional Long Short Term Memory (BiLSTM), Gated Recurrent Unit (GRU) and Bidirectional Gated Recurrent Unit (BiGRU) are proposed by various researchers to handle the gradient exploding and vanishing problems.

2.3.1. Long Short Term Memory (LSTM)

The main problem of standard RNN is gradient vanishing and exploding in the tasks that require storing long-time dependency information of the features (Abd, et al., 2020). Gradient vanishing refers to the loss of information in a neural network as connections recur over a longer period. LSTM is an advanced version of RNN models that have the ability to store long-term feature correlation information (Bader, Shahin, Ahmed, & Werghi, 2022). The LSTM models are developed by using the interconnection of gates to manage and control the feature dependency information either in the long term or in the short term. The earlier information on the features

correlation can be used together with the current information to give the output of the current state. LSTM networks have three types of gates such as input, forget, and output gates. Each gate performs specific tasks in managing and controlling the information in the network. The input gate decides the information to be stored in long-term memory. The current input and short-term memory information from the previous step is used in the input gate. Forget gate decides the information should be kept in and discarded from the long-term memory. The output gate produces new short-term memory output from the current input, previous short-term memory, and newly generated long-term memory.

The LSTM networks converge faster during the training of the model due to its improved gradient handling. LSTM models are widely employed in language models, machine translation, handwriting recognition, speech synthesis, image processing, speech recognition and other applications (Nirvana, Mahmud, Habibullah, & Khan, 2022). The detailed structure of the LSTM network cell structure is presented in Figure 6.

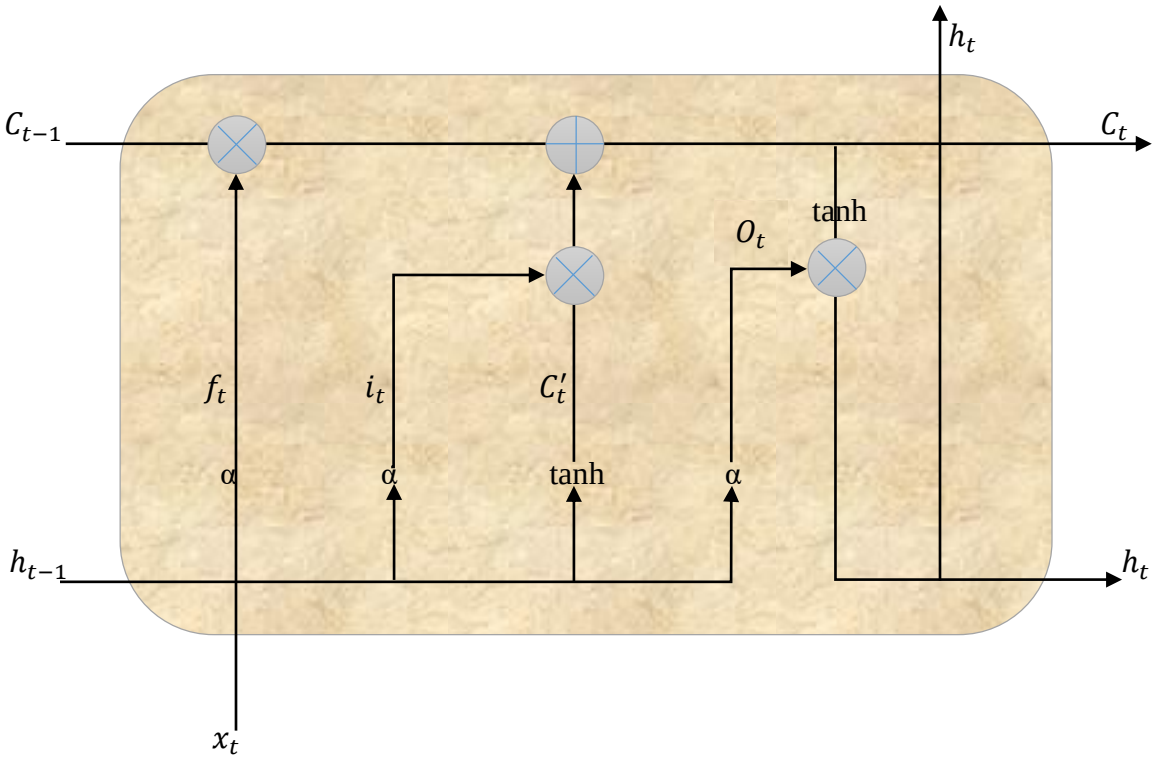

Figure 6: Cell Structure of LSTM

2.3.2. Bidirectional Long Short Term Memory (BiLSTM)

LSTM networks process only in one direction or extract and train the model by the relationship between the features only in one direction. It misses the other direction interdependency information of the features that are required for model training (Nammous, Saeed, & Kobojek, 2022). BiLSTM network is an extension of the LSTM architecture that addresses the limitation of standard LSTM architectures by considering the forward and backward correlation between the features (Emre, Soufleris, Duan, & Heinzelman, 2018). The BiLSTM network consists of two parallel LSTM layers from which one extracts the correlation between the features in the forward direction, while the other extracts the correlation between the features in the backward direction. Bi-LSTM has been widely applied in various sequence modeling tasks such as natural language processing, speech recognition, and sentiment analysis. It has shown promising results in capturing complex patterns and dependencies in sequential data, making it a popular choice for tasks that require an understanding of both past and future context. The detail of the BiLSTM network architecture is presented in Figure 7.

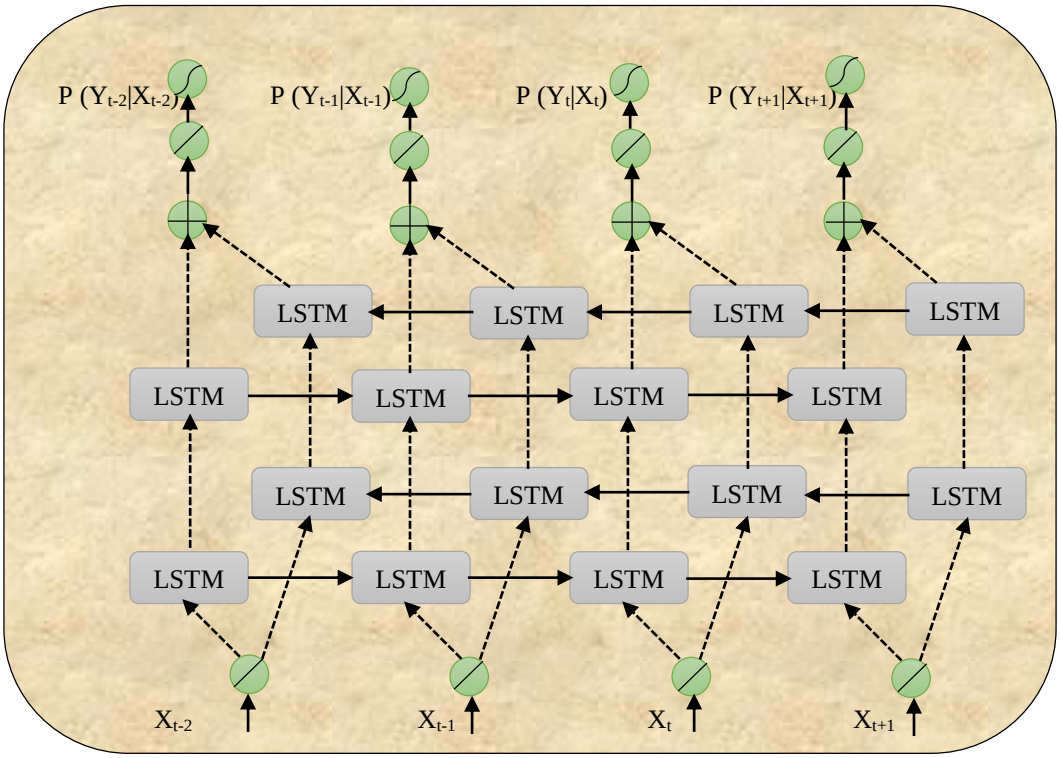

Figure 7: BiLSTM Network Architecture

2.3.3. Gated Recurrent Unit (GRU)

Both LSTM and BiLSTM network architectures have a large number of gates and cost high computational resources because of a large number of parameters (César, Mendes, Torres, & Assis, 2021). An overfitting problem and computational cost are the disadvantages of the LSTM and BiLSTM network models. The GRU architecture has been proposed to handle the limitations of LSTM and BiLSTM network models. GRU is a variant of the RNN which uses gating mechanisms to control and manage the flow of information between cells in the neural network (Ye & Yang, 2021). The gates are responsible for regulating the information to be kept or discarded at each time step. It uses only two gates (i.e., Update and reset gates) to handle the limitation of LSTM architectures. The input gate and forget gate of the LSTM are combined to give the update gate in the GRU network. The update gate determines the amount of information that should pass from the previous cell to the next state. The reset gate decides how much of the past information is needed to neglect. Like LSTM, GRU handles gradient vanishing and exploding by storing the information of long sequence dependency. The GRU's structure enables it to capture dependencies from large sequences of data in an adaptive manner, without discarding information from earlier parts of the sequence. The details of the GRU network models' cell structure are presented in Figure 8 which is adopted from (César, Mendes, Torres, & Assis, 2021).

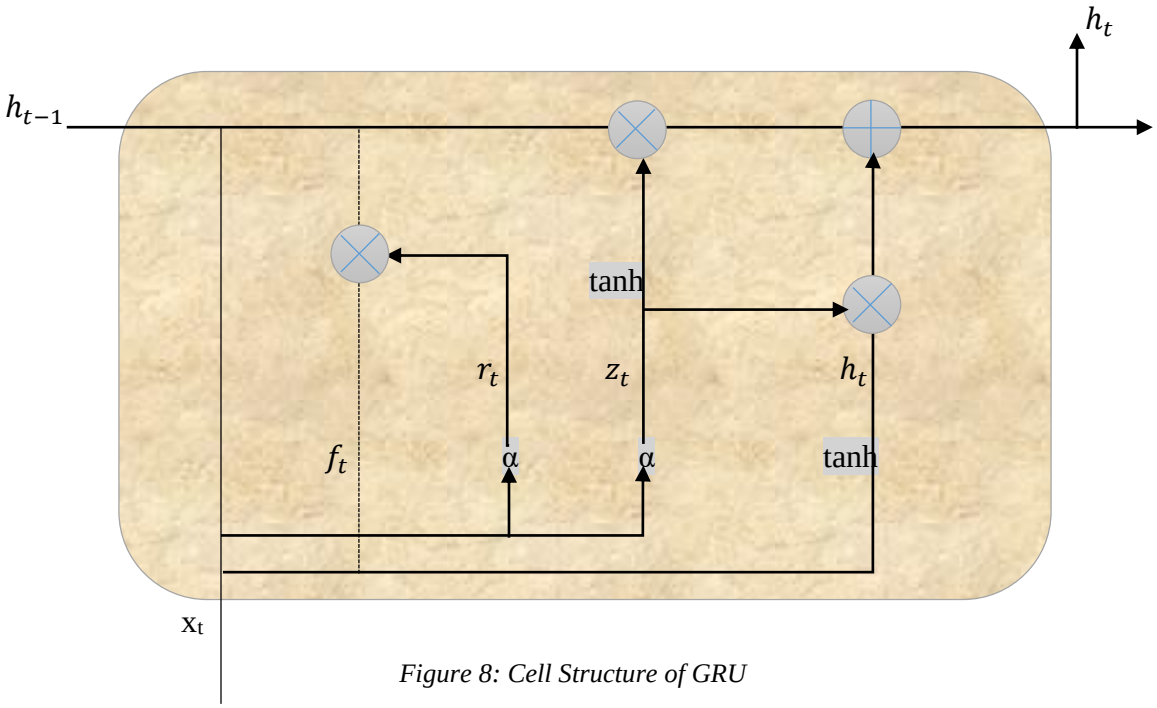

Figure 8: Cell Structure of GRU

2.3.4. Bidirectional Gated Recurrent Unit (BiGRU)

Since GRU networks extract the correlation of features sequentially in one direction only, it miss the dependency information in the other direction (Wu, et al., 2023). Bidirectional gated recurrent units (BiGRU) have been recommended for extracting the backward and forward long-term dependency information of the features sequentially (Guo, Qiao, Sukkarieh, Chai, & He, 2021). The BiGRU layer is the interconnection of GRU layers in forward and backward mechanisms to extract both directions long-term correlation information of the features. Moreover, BiGRU networks converge after a few iterations during the model evaluation. The BiGRU model architecture is presented in Figure 9.

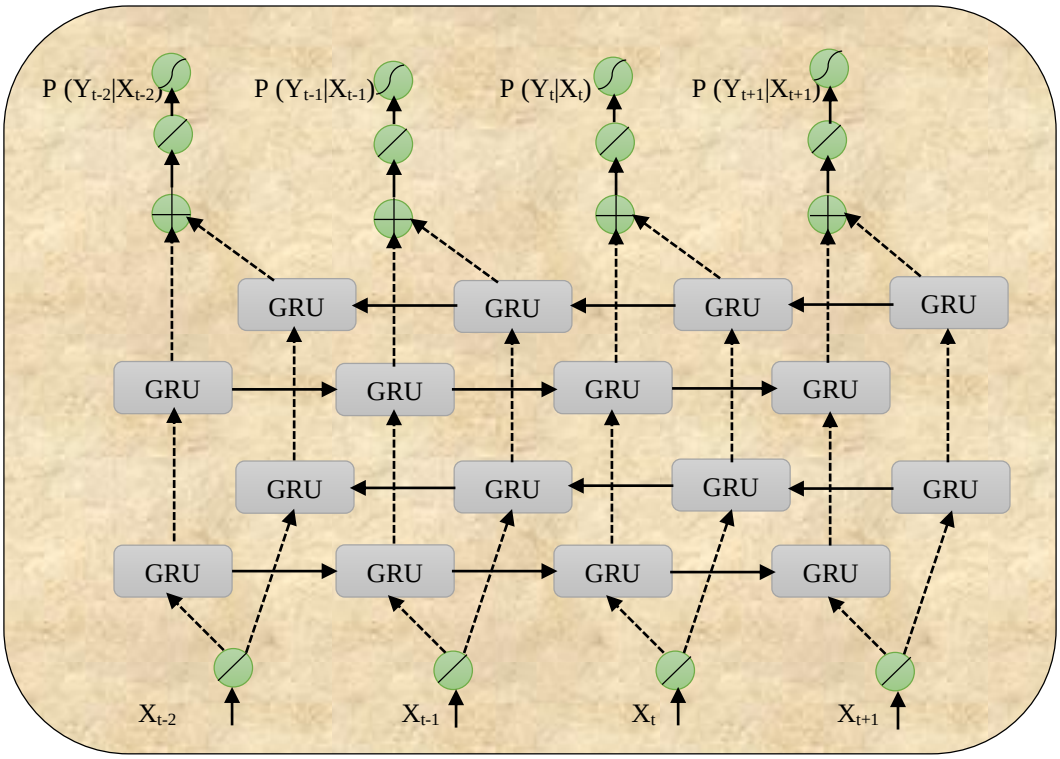

Figure 9: BiGRU Network Architecture

2.4. Speech Representation for Speaker Recognition

Most of the areas such as natural language processing, signal processing and image processing have large datasets for training machine learning or deep learning models. The data in the dataset contains numerous attributes from which some attributes are not important and duplications that may result a poor performance. Speech representation in the appropriate form for the models should be conducted before model training to reduce the size of trainable attributes and minimize

irrelevant and redundant information. Training machine learning or deep learning directly with raw signals often yields poor results because of the high data rate and information redundancy (MathWorks, 2023). Feature extraction is the mechanism to transform the raw data into feature vectors by preserving important information in the raw data (Shrawankar & Thakare, 2013). It identifies the most discriminating characteristics in signals, which a machine learning or a deep learning algorithm can more easily consume. Feature extraction also used as the front-end component and the machine learning approach used as the backend component of the systems. Training machine learning or deep learning models using extracted features yields better performance and saves computational resources than the raw data. Machine learning models use either hand-crafted features or extracted by another algorithm which is not a component of machine learning. In deep learning models feature extraction will be carried out automatically during the model training at some of the layers at the beginning. Feature extraction is very crucial for reducing computational cost by reducing the dimensionality of the data, improving performance by removing noises and irrelevant information, preventing overfitting by reducing the number of training parameters, and better understanding of data (Khalid, Khalil, & Nasreen, 2014).

There are several of feature extraction or speech representation methods for speech processing applications including speech and speaker recognition systems. Some of the common methods are Mel frequency cepstral coefficient (MFCC) (Hanifa, Isa, & Mohamad, 2020), gammatone frequency cepstral coefficient (GFCC) (Sekkate, Khalil, & Adib, 2017), linear predictive cepstral coefficients (LPCC) (Mansour & Lachiri, 2017), and perceptual linear prediction (PLP) and linear prediction coding (LPC) (Kabir, Mridha, Shin, Jahan, & Ohi, 2021). In addition, raw waveform, spectrogram and cochleogram of the speech have recently shown better performance than the above feature types in speech and speaker recognition. The detailed discussion of MFCC, GFCC, Spectrogram and Cochleogram features are presented in subsections 2.4.1 to 2.4.4 respectively.

2.4.1. Mel Frequency Cepstral Coefficient

The Human ear is the most reliable recognizer of the sounds that come from various sources (Dobie RA, 2004). Simulating the working principle of the human auditory system using statistical or algorithmic methods is very crucial for developing reliable recognizers using sounds for various

types of computer technologies. The human ear has critical bandwidth filters to identify or classify sounds, speakers, gender and language (Sahidullah & Saha, 2012).

Mel Frequency Cepstral Coefficient (MFCC) simulates the human auditory system by using the filters spaced linearly for low frequencies and logarithmically for high frequencies (ZRAR & ABDULBASIT, 2022). The mel-frequency scale has linear frequency spacing below 1000 Hz and logarithmic spacing above 1000 Hz. The filters extract important information contained in the speech signal to classify or identify speech, speaker, language and others. According to (Isam, John, David, & Victor, 2019), the MFCC features extraction process follows the sequence of steps such as pre-emphasis, framing, windowing, Fast Fourier Transform (FFT), power spectrum, Mel filter bank and discrete cosine transform (DCT) as shown in figure 10.

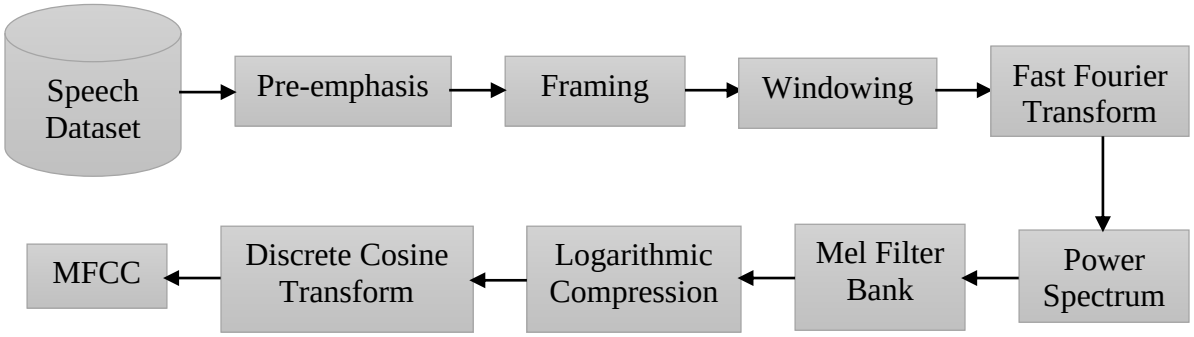

Figure 10: MFCC feature extraction process

MFCC features are very important for medical analysis (e.g., Disease detection, ECG and EEG analysis), industry analysis (e.g., monitoring Gear health, turbine health and pump health), and acoustic analysis. In the research work (Boussaa, Atouf, Atibi, & Bennis, 2016), ECG signal classification for medical purposes was proposed using the MFCC feature and artificial neural network. In another study (Rajesh, 2016), an analysis of MFCC features was proposed for EEG signals classification for cardiovascular disease diagnosis. The gear fault detection using sound under varying conditions for diagnosis was proposed using the MFCC features and ANN (Mian, Choudhary, & Fatima, 2022). In acoustic analysis, MFCC features have been widely employed in speech recognition, speaker recognition, language, and gender recognition. In the study conducted by (Ittichaichareon, Suksri, & Yingthawornsuk, 2012), the speech recognition system have been developed using the MFCC feature and support vector machine approach. The research work conducted by (Nakagawa, Wang, & Ohtsuka, 2012) proposed a speaker verification system using the MFCC feature and machine learning methods. In the reference (Ahmad, Thosar, Nirmal, &

Pande, 2015), a text-independent speaker recognition system was conducted using the MFCC feature and machine learning method. The study in (Leu & Lin, 2017), proposed a speaker identification system using the MFCC feature. Although MFCC features have better performance in clean speech their accuracy gets degraded under background and additive noises (Nasersharif & Akbari, 2007).

2.4.2. Gammatone Frequency Cepstral Coefficient

As discussed above, the performance of the speaker recognition systems that use MFCC features gets degraded under the additive noises and mismatch with training and test data. In the real-world scenario, the speech that could be used for speaker recognition may contain various types of noises. The features that is robust for noise and other changes are important for reliable speaker recognition. Gammatone frequency cepstral coefficients (GFCC) are relatively better than MFCC features for the noisy conditions (Qi, Wang, Xu, & Tejedor, 2013). Unlike MFCC features, the GFCC features could be extracted from the gammatone filters of the speech signals (Zhao & Wang, 2013). In GFCC features, an equal rectangular bandwidth (ERB) scale is useful to represent the resolution of the frequencies of speech samples. In the ERB scale, the low-frequency samples have finer resolution compared with the Mel scale. The GFCC feature extraction follows the sequence of operations as shown in Figure 11 which is adopted from (Huapeng Wang, 2020).

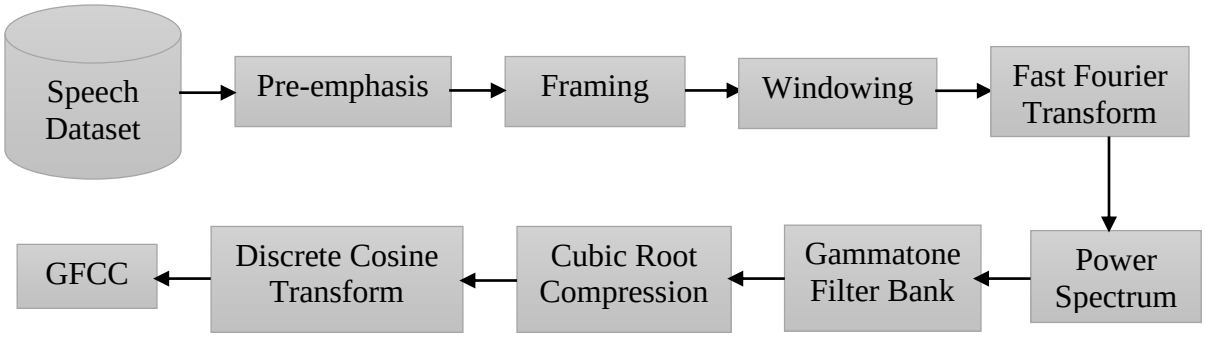

Figure 11: GFCC feature extraction process

Similar to MFCC features GFCC features are very important for various applications including medical analysis, industry analysis and acoustic analysis, especially under noisy conditions. The research work conducted by (Dua, Kumar, & Biswas, 2018) employed GFCC features for speech recognition using Hindi speech by using a machine learning approach. The research work conducted by (Menaka, Karthik, & Kabilan, 2024), proposed autism classification by analyzing

MFCC features in the EEG signal and deep learning model known as the AlexNet model. (Patni, Jagtap, Bhoyar, & Gupta, 2021), conducted to develop speech emotion recognition using GFCC features and convolutional neural networks. In the study (Tazi, Benabbou, & Harti, 2012), the text-independent speaker identification system was proposed using the GFCC feature and machine learning method for noisy and mismatched conditions. The study (A. & Al-Karawi, 2020), employed the GFCC feature and machine learning model for speaker verification to handle the reverberation effect on the performance. GFCC feature is also used to represent the speech based on the inner ear functionality scenario in a very compressed form which misses the detail attributes important for deep neural networks.

2.4.3. Spectrogram

Speech is highly dynamic and directly using speech audio in speech processing applications is not effective. For achieving better performance in various speech processing applications including speaker recognition, it is necessary to convert the audio of the speech into the appropriate format. As discussed in sections 2.3.1 and 2.3.2, both MFCC and GFCC features represent speech in a very compressed to achieve better performance in various speech processing applications and they reduce computational costs, save memory and other resources. Therefore, both features contain limited acoustic and other information that is important during model training and classification phases (Abdul, Setianingsih, & Nasrun, 2021). Deep learning models automatically extract features from the input at the beginning layers during training (Bhattacharya, Alam, Stafylakis, & Kenny, 2016). It also needs large data and attributes of the data to train the network. The spectrogram of the speech is very important for speech processing applications using a deep learning approach (Torfi, Dawson, & Nasrabadi, 2018). The spectrogram represents speech characteristics in a two-dimensional time versus frequency mechanism, which represents the contextual variation of speech. The spectrogram is rich with the acoustic information of the speakers such as energy, pitch, fundamental frequency, formants and timing (Ye & Yang, 2021). It gets generated from the power spectrum of the Mel filter bank. The spectrogram could be generated from the speech signal by applying a sequence of steps such as pre-emphasis, framing, windowing, fast Fourier transform and mel filter bank as shown in Figure 12 which is adopted from (Ye & Yang, 2021).

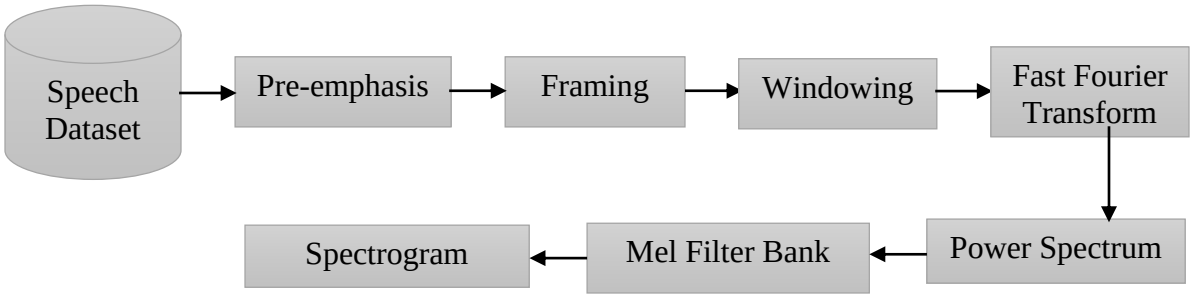

Figure 12: Spectrogram generation process

In the speech samples lower frequency component always dominates the higher frequency component, which may reduce the effectiveness of the systems in noisy environments (Desai & Joshi, 2014). Pre-emphasis is useful to compensate for the energy in high-frequency speech samples concerning energy in the low-frequency samples (Weng, Li, & Guo, 2010). For sample s[n] and emphasis factor f, pre-emphasized sample Y[n] can be calculated according to Equation 1.

$$Y[n] = s[n] - f * s[n-1]; 1 > f \geq 0.91 \quad (1)$$

The speech is very dynamic and obtaining stable information from a long speech is difficult. Framing segments each input speech into short sample points which reduces time variation and helps to get stable acoustic characteristics from the speech (Nakagawa, Wang, & Ohtsuka, 2012). The recommended frame durations in various speech analysis tasks range from 20ms to 50ms with an overlap of 30% to 50% between the frames. Framing causes discontinuity and mismatch between frame segment and original speech. Windowing is applied to reduce unexpected changes between frames and smooth edges of frames (Leu & Lin, 2017). Generally, the hamming window is commonly used in speech analysis for different types of applications. Each frame F[n] can be windowed as shown in Equation 2, and the window function P[n] is calculated as shown in Equation 3.

$$H[n] = F[n] * P[n] \quad (2)$$

$$P[n] = 0.5 - 0.46\, cos\left(\frac{2n\pi}{M-1}\right), \mathrm{M} - 1 \geq \mathrm{n} \geq 0 \quad (3)$$

Obtaining important features from the time domain of the speech signal is difficult. During speech processing applications speech signals should be converted from the time domain into the frequency domain to obtain the characteristics of the acoustics. Fast Fourier Transform (FFT)

computes the frequency domain of the samples from the time domain, the result is known as a spectrum or periodogram of the sample (Toledano, Ramos, Gonzalez-Dominguez, & González-Rodríguez, 2009). FFT of frame signal X(n) with N number of sample points is computed as in Equation 4:

$$X(k) = \sum_{n=0}^{N-1} X(n) e^{-\frac{2j\pi nk}{N}};\ 0 \leq k \leq N-1 \quad (4)$$

The magnitude of the FFT is computed by taking the absolute value of X(k), which is used as an input to compute the power spectrum of each FFT. Mel Scale is obtained by applying triangular bandpass filters at each of the power spectrum values. Mel-scale simulates non-linear human auditory perception, by being more discriminative at lower frequencies and less discriminative at higher frequencies. For the given frequency f in Hz, the Mel scale is computed as in Equation 5:

$$m = 2595\, log_{10}\left(1 + \frac{f}{700}\right) \quad (5)$$

Finally, Spectrogram is generated by stacking all the Mel filter bank values of each frame of the given speech signal.

The spectrograms also can be employed for medical analysis, acoustic analysis and other applications. Most deep learning-based speech processing applications employ spectrogram as an input because of its rich acoustic features. The research work conducted by (Yenigalla, et al., 2018), employed a spectrogram of the speech and convolutional neural network for emotion recognition. Another study (Singh, et al., 2021), also employed a spectrogram and CNN for language recognition. The speech recognition model proposed in (Shah & Chandra, 2020), has been conducted by using a machine learning approach and spectrogram of the speech signal. As demonstrated by (Yağmur & ÖZKURT, 2019), the heartbeat classification from the ECG signals has been proposed using spectrogram analysis and the CNN model. In the study (Demile & Mulatu, 2020), combination of spectrogram, MFCC and spectral features were employed for detecting environmental sound warning using the multilayer perceptron neural network.

2.4.4. Cochleogram

In the above sections, how MFCC, GFCC and spectrogram could be obtained from the speech signal were discussed in detail. Like MFCC features, spectrograms could be obtained from the power spectrum of the Mel filter bank. Although spectrograms are rich in acoustic features they

could be affected by additive noises and other environmental and speaker-related effects. Cochleogram is the representation of speech signal in a two-dimensional (2D) time-frequency domain from the power spectrum of the gammatone filters to employ in speech processing applications (Ayoub, Jamal, & Arsalane, 2016). The time and frequency of the speech were represented in the x-axis and y-axis of the cochleogram. The color in the cochleogram image represents the amplitude of the sample. Cochleogram can be obtained from the gammatone filters in the equal rectangular bandwidth (ERB) scale. ERB measures the psychoacoustics of the speech to determine the estimated bandwidths of human hearing filters (Huapeng Wang, 2020). Gammatone filters simulate the human auditory system which helps to model the speaker, language, speech and other information from the speech (Zhao & Wang, 2013). It has advantages in representing speech of low frequency in finer resolution.

Like GFCC features, cochleogram could be obtained from the power spectrum of the speech signal's gammatone filters which makes them preferable for speech analysis under noisy conditions. Unlike GFCC features, cochleogram is rich in acoustic information to train data and attribute-intensive models. Cochleogram could be obtained from the speech signal by applying a sequence of tasks such as pre-emphasis, framing, applying a window function computing fast Fourier transform, gammatone filter bank, and computing power spectrum from gammatone filters as shown in figure 13. The detailed description and importance of pre-emphasis, framing, windowing and fast Fourier transform are presented in spectrogram section or in 2.4.3.

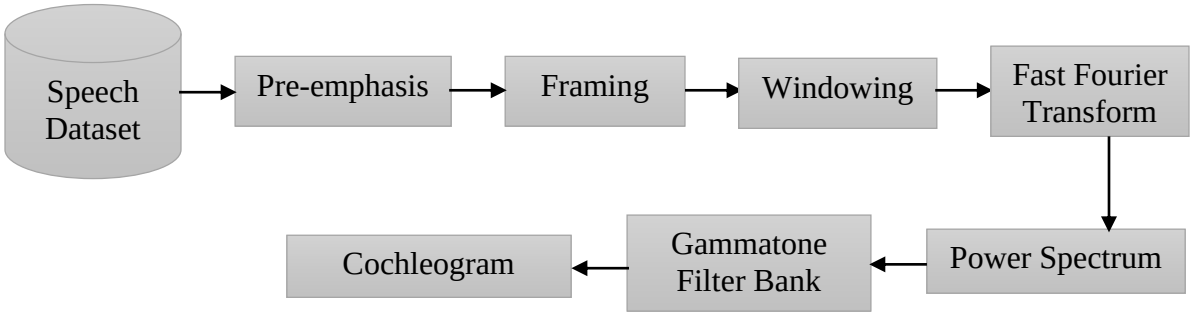

Figure 13: Cochleogram Generation Process

For the FFT of the speech frames with amplitude *a* and time t, gammatone filter of filter order n and phase shift φ can be computed according to Equation 6.

$$g(t) = at^{n-1}e^{-2\pi b_m t}\, cos(\pi f_c t + \varphi) \quad (6)$$

Central frequency f_c for the mth gammatone filter is computed from the low-frequency f_L and high-frequency f_H according to equation 7.

$$f_c = \left(-\frac{1000}{4.37}\right) + \left(f_H + \frac{1000}{4.37}\right) exp\left(\frac{m}{M}\left(-ln\left(f_H + \frac{1000}{4.37}\right) + ln\left(f_L + \frac{1000}{4.37}\right)\right)\right) \quad (7)$$

The Equal Rectangular Bandwidth (ERB) scale of the central frequency fc can be calculated as in Equation 8.

$$\mathrm{ERB}(\mathrm{f_c}) = 24.7\left(4.37 * \left(\frac{\mathrm{f_c}}{1000}\right) + 1\right) \quad (8)$$

The bandwidth b_m of the gammatone filters can be calculated as shown in Equation 9.

$$b_m = 1.019 * ERB(f_c) \quad (9)$$

Each gammatone filter can be used as a feature to model the speaker, speech, language, and emotion that exists in a speech. The power spectrum of the gammatone filters is stacked together to form cochleogram of the given speech. Finally, the proposed model takes the cochleogram as input for speaker identification.

Since cochleogram is rich with acoustic features some speech processing applications employ cochleogram. Sharan and Moir (Sharan & Moir, 2019), employed cochleogram of the speech signal for speech event recognition using a convolutional neural network. The study conducted (Ahmed, Mamun, & Hossain, 2021), has proposed the speaker identification model using noise-adapted CNN and cochleogram input.

2.5. Speaker Recognition using Machine Learning Methods

Several research works have been conducted in the speaker recognition area by using conventional machine learning methods. Some of the conventional machine learning approaches commonly employed in speaker recognition are Gaussian mixture model (GMM) (Kumar & Bhukya, 2022), the gaussian mixture model with universal background (GMM-UBM) (Yan, Men, Yang, & Jiang, 2016), support vector machine (SVM) (Wang J. , et al., 2017), Hidden Markov Model (Gurbuz, J.Gowdy, & Tufekci, 2002) and Vector Quantization or Euclidian distance (Wang, Tang, & Zheng, 2012). These approaches have been employing handcrafted features such as MFCC, GFCC, LPC, LPCC and PLP for speaker recognition. For example, in the studies (Alam, Kinnunen, Kenny,

Ouellet, & O'Shaughnessy, 2013) and (Hossan, Memon, & Gregory, 2010), speaker verification systems were conducted using i-vector and GMM classifiers on the MFCC feature respectively. The speaker recognition proposed in the study (Weng, Li, & Guo, 2010), has used a GMM machine learning approach and MFCC features of the speaker recognition using the short utterances. Speaker recognition conducted with the GMM approach on the MFCC feature was proposed as the biometrics for home device control (Abdul, Setianingsih, & Nasrun, 2021) and remote identification over VoIP (Ajgou, Sbaa, Ghendir, Chamsa, & Taleb-Ahmed, 2014). In the reference (Sharma & Ali, 2015), the GMM method together with the MFCC and inverse MFCC feature extraction techniques were employed in speaker recognition. Moreover, artificial neural network have been employed in the speaker recognition using MFCC features on the dataset prepared using the Amharic language (Belayneh, Urgessa, & T.GopiKrishna, 2019).

The Gaussian Mixture Model (GMM) has been widely employed in speaker recognition, language identification, emotion recognition and gender recognition. The GMM method together with the MFCC feature has been employed to develop speaker recognition using a clean dataset or without noise. In the study (KUMAR, RAJU, Rao, & Satheesh, 2010), the speaker recognition system was proposed using the GMM method and MFCC feature for the dataset collected from the clean environment. Moreover, most of the text-independent speaker recognition systems have been developed using the GMM method. The research work (Nayana, Dominic, & Abraham, 2017), was conducted to develop a text-independent speaker recognition system using the GMM method and PNCC feature. In another study conducted by (Chen, Hsieh, & Lai, 2004), the speaker identification system proposed for a text-independent mode was developed by using the GMM method and MFCC features. The speaker verification system was developed by using the GMM method and MFCC feature in the text-independent method in the study (Hasan, L, & Hansen, 2019). Together with the GMM approach, GFCC features have been widely employed to enhance the performance of speaker recognition systems under noisy and mismatched conditions. For instance, in the study (Ayoub, Jamal, & Arsalane, 2016) GMM method and GFCC features were employed for speaker identification under noisy acoustic datasets. In the reference (Choudhary, Sadhya, & Vinal Patel, 2021), a GMM method and GFCC feature were employed for speaker verification using datasets with real-world noises. In the reference (Moinuddin, 2014), the speaker verification system proposed for the noisy condition was conducted by using the GMM method

and GFCC features. The research work (Kaur, Bhushan, & Singh, 2016), proposed speaker recognition using the GMM method and GFCC feature for the application in biometric security.

Gaussian Mixture Model with Universal Background Model (GMM-UBM) has been also employed in speaker recognition for various applications. In the study (Omid & Hansen, 2014), the speaker verification for the reverberant and mismatch conditions was proposed using the GMM-UBM and MFCC features. In the reference (Islam, Jassim, Cheok, Shamsul, & Zilany, 2017), the speaker identification was developed using the GMM-UBM method and MFCC feature. The speaker verification system developed in the study also employed the GMM-UBM method with the MFCC and DWT features. Another study (Wang & Zhang, 2020), proposed a speaker identification system for forensic applications using the GMM with universal background model (GMM-UBM) and GFCC features in noisy environments.

Support vector machine was also commonly employed in speaker recognition to enhance the performance of the systems in specific applications. In the reference (Wang J. , et al., 2017), Support Vector Machine (SVM) and MFCC feature was proposed for speaker verification systems in abnormal human behavior. Some speaker recognition systems were conducted using vector quantization and handcrafted feature extraction techniques. In the study (Maurya, Kumar, & Agarwal, 2018), a speaker recognition system was proposed using the vector quantization method and MFCC features for the Hindi speech dataset.

2.6. Speaker Recognition using Deep Learning Models

Recently, deep learning models have shown better performance in the areas of computer vision, image classification, natural language processing and other application areas. Some of the basic deep learning models employed in the recent research include convolutional neural network (CNN), recurrent neural network (RNN), Self-Organizing Maps (SOP), and Autoencoders. The CNN and variants of RNN architectures have been widely applied in speech and speaker recognition.

The research work proposed by (Hourri & Kharroubi, 2019), employed deep neural networks and MFCC feature input for speaker recognition. The model has been experimented on the THUYG-20 SRE corpus and performed better than the i-vector method and PLDA on both clean and noisy

datasets. In the study conducted by (McLaren, Lei, & Ferrer, 2015), a speaker identification model was developed using a deep neural network. The model was experimented on the NIST 2012 dataset and it has shown better performance than the Gaussian Mixture Model with Universal Background Model and i-vector approaches in the same dataset.

In the reference (Joseph & Billson, 2020), the deep neural network has been employed to develop a speaker recognition system for emotional speakers. The experiment was conducted on the Ryerson Audio-Visual Database of Emotional Speech and Songs (RAVDESS). To show the effectiveness of the deep neural network in speaker recognition, additional experiments were conducted using machine learning methods on the same dataset. The deep neural networks have achieved better performance than machine learning models. The speaker identification model proposed in (Muayad, Sahib, & Adnan, 2020) was conducted by using CNN architecture and experimented on the TIMIT dataset. The model accuracy has shown superior performance than state of the art in speaker identifications.

In speaker recognition systems conducted using deep neural networks, various types of features or input have been used. Deep neural networks have been employing raw waveform, MFCC features, GFCC features, spectrograms, cochleogram and other forms of speech for speech and speaker recognition. In the research work conducted (Nammous, Saeed, & Kobojek, 2022), the MFCC feature has been used as an input for speaker identification using a bidirectional long short-term memory method to enhance the performance. The experiment was conducted on the Fisher speech corpus. Another study conducted by (Ashar, Shahid, & Mushtaq, 2020), developed a speaker recognition using the CNN architecture and MFCC features for noisy environments. The experiments were conducted on 60 speakers each speaker have four utterances. The speaker verification model developed by (Soleymani, Dabouei, Mehdi, Kazemi, & Dawson, 2019) has employed the MFCC feature image as an input for the Siamese network of CNN architecture. The result has shown improvement over the existing machine learning approach using MFCC features.

Deep learning models employing the MFCC and GFCC features for speaker recognition have achieved better performance than classical classifiers. However, MFCC and GFCC features were not as effective as other inputs like raw waveforms, spectrograms and cochleogram for speaker recognition using deep neural networks. This is because both MFCC and GFCC features have

limited acoustic information about the speaker and it represent acoustic features in a very compressed mechanism in each of the speech features. The raw waveform of the speech was employed in the study (Salvati, Drioli, & Foresti, 2019), for speaker identification using CNN architecture. Here the raw waveform outperformed MFCC features in speaker identification under noisy conditions and slight enhancement in the clean data. In another research conducted by (Salvati, Drioli, & Luca, 2023), the speaker recognition model was developed by using the GFCC features and raw waveform as an input for the CNN architecture to enhance the performance under the noisy environment. Here raw waveform of the speech has better accuracy than MFCC and GFCC features.

Most of the speaker recognition systems conducted using deep learning models have mainly used spectrograms as an input. In the research work conducted by (Farsiani, Izadkhah, & Lotfi, 2022), the speaker identification models have been proposed by using the visual geometry group (VGG-13) architecture of the CNN and spectrogram as the input for the model. The experiment was conducted on the VoxCeleb1 audio dataset and its accuracy was compared with the state-of-the-art in speaker recognition. This model has shown better accuracy than the state of the art in speaker recognition. In the study (Costantini, Cesarini, & Brenna, 2023), spectrograms have achieved superior performance than MFCC in speaker recognition using the deep neural network model CNN. Spectrograms have shown better performance than MFCC features in speaker recognition. In the study (Abd, et al., 2020), spectrogram input has achieved better accuracy than MFCC features in speaker recognition using LSTM architecture of the recurrent neural network. In the reference (Bunrit, Inkian, Kerdprasop, & Kerdprasop, 2019), spectrogram features have shown better accuracy than raw speech signal and MFCC in speaker recognition using deep machine learning. This is because spectrogram features are rich in acoustic features to characterize the speaker which helps the deep learning networks easily learn the correlation between the features.

The speaker recognition systems have been conducted using several deep learning architectures to enhance the performance. The CNN architectures such as visual geometry group (VGG), residual network (ResNet) and other custom CNN architectures with different number of layers have been widely used in speaker recognition. In the reference (Bunrit, Inkian, Kerdprasop, & Kerdprasop, 2019), customized CNN architecture with three convolutional and one fully connected layer has been employed for speaker identification. The experiment was conducted on the Thai language

dataset with five speakers and the accuracy of the customized CNN model with spectrogram input was compared with the accuracy of the model on other features such as MFCC and waveform. The proposed custom CNN with spectrogram input has shown better accuracy. In the research conducted by (Yadav & Rai, 2018), the speaker verification model was developed by using two CNN architectures (i.e., VGG-11 and VGG-13) and spectrogram input. The model was experimented on the VoxCeleb1 dataset and compared the result with other machine learning and deep learning models. This model has the highest performance than other baseline models selected for comparison. In another study (Jakubec, Lieskovska, & Jarina, 2021), the speaker recognition model was proposed by using the ResNet architecture of the CNN which employed the spectrograms as an input. This model also experimented with the VoxCeleb1 to evaluate its effectiveness in realistic conditions. The model has surpassed the VGG model and other existing works in speaker recognition.

Moreover, variants of recurrent neural networks such as long short-term memory (LSTM), bidirectional LSTM, and gated recurrent unit (GRU) have been used in deep learning model-based speaker recognition. In the research (Wang, Xue, Wang, & Liu, 2020), the BiLSTM architecture has been employed with the spectrogram input for speaker recognition. This model was evaluated by using VoxCeleb2 and Aishell-1 datasets and the model has shown good performance.

Enhanced variants of the RNN model include gated recurrent unit (GRU) (Ye & Yang, 2021), long short-term memory (LSTM), and bidirectional LSTM (BiLSTM) (Emre, Soufleris, Duan, & Heinzelman, 2018). The GRU model is an enhanced version of the LSTM and it has the advantages of extracting and learning long-term correlation between features sequentially in only one direction (Liu, Yin, Sun, & Ersoy, 2022). GRU models have fewer parameters than LSTM to minimize the computation cost. Moreover, GRU models handle gradient vanishing problems and converge within a few iterations during model training.

2.7. Speaker Recognition and other Applications using Hybrid Models

Several researchers have followed different kinds of mechanisms to enhance the performance of speaker recognition systems. Most of the research works conducted in the speaker recognition areas have employed a single method/model with either a single feature extraction/input representation type or the fusion of features/inputs. Other research works conducted in the speaker

recognition area employed hybrid methods/models of popular algorithms together with either single feature extraction/input representation type or the fusion of them. This section presents the research works conducted in the speaker recognition and other areas by using hybrid approaches.

Hybrid conventional machine learning approaches have been widely used to develop speaker recognition systems. For instance, the research work conducted by (Mashao, 2005), has shown the effectiveness of hybrid GMM and SVM in speaker recognition. The system was experimented on the TIMIT dataset and the results have shown better performance than earlier works. In another study (Rodríguez, Ruíz, García, & García, 2005), hybrid statistical machine learning models GMM and HMM have been used to develop speaker recognition system. The results confirmed that the hybrid model has better accuracy than each of the separate approaches results. In the research conducted by (Liu, Xie, Yao, & Dai, 2006), the speaker verification system using hybrid GMM and SVM was proposed. The experiments were conducted on the NIST04 dataset and the evaluation results indicate that the proposed system has better performance than the state of the art in speaker recognition using GMM or SVM. In another study (Desai & Joshi, 2014), a hybrid VQ and GMM model was proposed for speaker recognition with the MFCC features. It was evaluated for text-dependent and text-independent modes and has shown promising improvement over the earlier works. In another study conducted by (Chakroun, Beltaïfa, Frikha, & Ben, 2016), the speaker identification system was proposed using the hybrid approaches of GMM and SVM by extracting the super vectors from the speech. The experiments have been conducted on the TIMIT dataset. The results of the proposed system in this study have been compared with the accuracy of SVM, GMM and existing works. It has achieved superior performance than other models.

Some studies conducted in speaker recognition have employed hybrid approaches across conventional machine learning approaches and deep learning models. In the research work conducted by (Liu, Wu, Li, Li, & Shen, 2018), the speaker recognition system was developed by using the hybrid CNN which was from deep learning architecture and the GMM method was from the conventional machine learning model. The spectrogram of the speech was employed as an input to the network by using CNN's first layer. The model has shown promising improvements over each of the individual models such as CNN and GMM. In another study (Nainan & Kulkarni, 2021), a hybrid model of SVM and one-dimensional CNN (1DCNN) was proposed in speaker recognition. The model has shown superior performance over the GMM model.

Recently, combinations of the CNN models with the enhanced variants of the RNN model have been achieving better performance in speech recognition, image classification, time series data analysis and other natural language processing. The study (Alam, Fathan, & Hyun, 2021), proposed a hybrid model of three deep learning architectures such as CNN, LSTM and TDNN for speaker verification. The experiments were conducted on three datasets such as NIST SRE 20016, VoxCeleb1 and short-duration speaker verification (SDSV) challenge 2021. The results confirmed that the proposed model in this work outperformed other architectures such as TDNN, TDNN-LSTM and other existing works. In the studies (Zhao, et al., 2019; Bader, Shahin, Ahmed, & Werghi, 2022), a hybrid network of CNN and LSTM models exhibited performance improvement in speaker verification and identification. Another study (Shekhar & Roy, 2021), employed a hybrid network of the CNN and BiLSTM for language identification using spectrograms of speech and the results have shown improvement in existing works. The study (Liu, Liu, Fan, Zhong, & Du, 2017), also indicated the effectiveness of a hybrid networks of CNN and BiLSTM models in audio-visual recognition for biometric applications. In the study (Ye & Yang, 2021), a hybrid CNN and GRU network were employed for speaker identification with the spectrogram of Aishell-1 datasets.

Moreover, hybrid models of deep learning models have been employed in various areas and shown better performances. For example, in the study (Wilkinson & Niesler, 2021), hybrid CNN and BiLSTM model have been employed for voice activity detector for resource limited areas. The model were constructed from two CNN and two BiLSTM models which minimizes the complexity of the architecture. The experiment was conducted on the AVA speech dataset. The proposed CNN-BiLSTM model outperformed ResNet and other works in the area. In the reference (Shekhar & Roy, 2021), hybrid models of the CNN and BiLSTM with the two CNN and two BiLSTM layers followed by other layers were employed for language identification. The model were experimented on the four Indian languages and spectrogram of the utterances were used as an input for the model. This model have shown a promising improvement over the existing works.

2.8. Public Datasets for Speaker Recognition

In this section, some of the public speech datasets applicable to speaker recognition have been presented. Several public datasets are available for the training, development, and evaluation of speech and speaker recognition systems. The public dataset saves the energy of researchers

because it helps to experiment on the available datasets and the researchers focus on scientific contributions in the areas. Sometimes the dataset prepared for speech recognition is interchangeably applicable in speaker recognition applications. Some of the well-known and most used public datasets in speaker recognition include TIMIT, LibriSpeech, Aishell-1, and VoxCeleb1.

TIMIT: - Also referred to as the DARPA TIMIT Acoustic-Phonemic Continuous Speech Corpus. This dataset is a continuous acoustic-phonemic speech corpus prepared by Texas Instruments, Massachusetts Institute of Technology and Stanford Research Institute SRI international. The TIMIT speech corpus where collected from the 630 people who speak the English language, the speech was collected from the eight major dialect regions of the United States. The TIMIT speech dataset consists of a total of 6300 utterances, which means 10 utterances from each speaker. Each of the utterances in the dataset has a sampling frequency of 16 kHz which is stored in the waveform files. The dataset was designed to provide speech data for acoustic-phonetic studies and the development and evaluation of automatic speech and speaker recognition systems.

Several studies in speaker recognition have been experimented on the TIMIT dataset. In the study (Al-Kaltakchi, Woo, Dlay, & Chambers, 2017), the TIMIT speech dataset has been used to experiment speaker identification system proposed using i-vector and MFCC features. In the reference (Ali & Kumar, 2021), the speaker recognition model developed using hybrid TDNN and LSTM has been experimented on the TIMIT public dataset. In another study conducted by (LI, ZHANG, XU, MA, & GAO, 2021), the speaker recognition system was experimented with using the TIMIT dataset.

Aishell-1: This dataset was developed by Beijing Shell Technology Co., Ltd. It contains approximately 520 hours of Chinese Mandarin speech from 400 speakers recorded simultaneously on three different devices with associated transcripts. The goal of the collection was to support speech recognition system development in 11 domains, including smart homes, autonomous driving, entertainment, finance and science and technology. Participants read 500 sentences covering the domains; sentences were chosen for their speech and phonetic characteristics. Speakers were recruited from different accent areas across China, including North, South and Yue-Gui-Min regions. There were 214 female speakers and 186 male speakers, constituting 53% and

47% of the database, respectively. Additional demographic information about the participants is included in this release. Speech was recorded in a quiet indoor environment on a high-fidelity microphone and two mobile phones (Android and iOS). All speech is presented as 16-bit FLAC compressed wav files; the microphone speech sample rate is 44.1 kHz and the phone speech sample rate is 16 kHz. Each speech file ranges from approximately 1 second to 14 seconds in length. Transcripts are stored as UTF-8 encoded plain text files and are not time-aligned.

Several research works conducted in speaker recognition also experimented on the Aishell-1 dataset. In the research (Hu, Si, Luo, Tang, & Jian, 2021), the speaker recognition system developed by using hybrid deep learning models of CNN and LSTM has been experimented on the Aishell-1 speech dataset. In the reference (Wang, Xue, Wang, & Liu, 2020), the speaker identification model developed by using hybrid CNN and BiLSTM architecture was evaluated on the Aishell-1 dataset. Another study conducted by (Ye & Yang, 2021), employed the Aishell-1 dataset to evaluate the speaker identification model developed by using GRU architecture.

LibriSpeech: This dataset is an audiobook dataset containing both text and speech, a corpus of approximately 1000 hours of 16 kHz read English speeches written by Vassil Panayotov. The dataset contains 251 speakers. Data is derived from reading audiobooks from the LibriVox project and is carefully segmented and consistent. Cut and organized into text-annotated audio files of about 10 seconds each, ideal for getting started.

Some studies conducted in speaker recognition areas have been evaluated by using the LibriSpeech dataset. As demonstrated in the research (Naoyuki Kanda, 2020), the LibriSpeech dataset was employed to evaluate the speaker recognition performance of the model was developed using the attention-based encoder-decoder method. The speaker recognition model developed by using the CNN architecture and the raw waveform input (Ravanelli & Bengio, 2019) experimented on the LibriSpeech dataset. In the reference (Wang, et al., 2018), the speaker recognition model proposed for a smart home solution with the small dataset using the CNN model was experimented on the LibriSpeech dataset. Another research work (Prachi, Nahiyan, Habibullah, & Khan, 2022), employed TIMIT and LibriSpeech dataset to evaluate the speaker recognition performance developed using hybrid CNN and LSTM.

VoxCeleb1: This dataset was primarily collected for the development and evaluation of speaker recognition systems. The dataset contains an appropriate ratio of samples recorded in various environmental conditions, genders, accents and age groups. The VoxCeleb1 dataset consists of a vast amount of speech data collected from celebrities in different languages. It provides a diverse set of speakers and variations in recording conditions, making it suitable for training and evaluating speaker verification models. In recent studies, the VoxCeleb1 dataset has been commonly employed to evaluate speaker recognition in the real-world scenario. In the research (Rao, Li, Lavrukhin, & Ginsburg, 2020), the speaker identification and verification conducted based on the text-independent mode using CNN have been evaluated using the VoxCeleb1 dataset. Another study (Li, et al., 2022), evaluated the speaker recognition model developed by using the CNN on the VoxCeleb1 dataset. In the reference (Xie, Nagrani, Son, & Zisserman, 2019), the VoxCeleb1 dataset was also used to evaluate the speaker recognition performance which was proposed for wild using the CNN approach. In the research work (Nagrani, Son, Xie, & Zisserman, 2020), the speaker verification system proposed for the wild and developed using the CNN architecture has employed the VoxCeleb1 dataset for evaluation. Generally, most of the recent studies conducted in speaker recognition have employed the VoxCeleb1 dataset. The reason was that the VoxCeleb1 dataset consists of a large number of speakers with an appropriate number of samples from each speaker which was collected from various environmental conditions and speaker distribution to perform speaker recognition using the deep learning models. In this study, the VoxCeleb1 audio dataset were selected to evaluate the features and models in speaker recognition.

2.9. Related works

Several research works have been conducted in speaker recognition areas to enhance the performance of the systems and achieve robustness to various environmental and mismatch conditions. In this section, the researchers have presented some of the studies which are more related to this dissertation.

In speaker recognition, research works have been conducted in text-independent mode for noisy conditions by using the machine learning method. In the research conducted (Aldhaheri & AlSaadi, 2004), the speaker recognition system was developed using SVM and cepstral coefficient proposed for the text-independent mode under the noisy environment. The model was evaluated at the SNR

of 5dB to 20dB in the interval of 5dB. In the reference (Wang, Tang, & Zheng, 2012), a speaker identification system was developed using vector quantization and MFCC feature for text-independent application under noisy environment. The evaluation has been performed on the YOHO dataset at different types of noises such as restaurant, street and pop-song noises at the SNR of 10dB and 15dB for each noise type. The research work conducted by (Choudhary, Sadhya, & Vinal Patel, 2021), proposed a speaker recognition system for clean and noisy conditions based on the text-independent mode using the GMM method and GFCC feature. In another study (Wang & Zhang, 2020), the speaker identification system based on the text-independent mode was developed using the GMM method and fusion of the GFCC and PNCC features was proposed for the noisy conditions. The evaluation has been conducted using different types of noises such as babble, buccaner1, destroyengine, factory, leopard and other noise types at the SNR of 0dB to 15dB in intervals of 5dB. In the study (Alabbasi, Jalil, & Hasan, 2020), speaker identification using the GMM-UBM method and GFCC feature has been developed based on the text-independent mode and proposed for noisy conditions. The model was evaluated on the noise types such as babble, factory 1, pink and white noises at the SNR of 0dB to 15dB in intervals of 5dB. In the study conducted by (Benhafid, Yasmine, & Amrouche, 2021), the speaker identification system proposed for noisy environments have been conducted using the GMM method and GFCC features. The evaluation of the system have been conducted by using the noise types babble, factory and subway at the SNR of -6dB to 18dB in interval of the 6dB.

Some studies have been conducted in speaker recognition using a deep learning model to enhance performance under noisy conditions. In the study (Abd, et al., 2020), the speaker recognition model for the text-independent mode has been developed using the LSTM and radon transform of the spectrogram to enhance the performance under noisy conditions. In the study (Taherian, Wang, Chang, & Wang, 2020), the deep neural network model was employed to develop speaker recognition for noisy conditions by using the GFCC features for various types of channels.

Moreover, a number of studies have been conducted in speaker recognition using hybrid deep learning model of CNN and RNN variants. In the study (Nirvana, Mahmud, Habibullah, & Khan, 2022), hybrid model of CNN and LSTM have been employed for speaker recognition. The model consists of two CNN and two LSTM layers. The experiment was conducted on the TIMIT dataset. Spectrogram were used as an input for the model and the model have shown promising

improvement over the individual models. In the study (Ye & Yang, 2021), the speaker identification model have been developed using hybrid CNN and GRU model on the spectrogram of the Aishell-1 dataset. The model in this study employed two CNN layers and two GRU layers followed by the fully connected layer. The model have shown better direction for the further investigation using hybrid models of deep learning models.

As discussed above, most of the speaker recognition systems developed for noisy and mismatched conditions commonly employed machine learning methods. These methods widely used the GFCC feature for recognition application under noisy conditions. The studies have confirmed that deep learning models have better performance than machine learning methods in image classification, speech recognition and speaker recognition. Moreover, hybrid models of CNN and enhanced RNN variants have shown improved performance in image classification, natural language processing, computer vision and other biometrics. However, only limited studies have employed hybrid CNN and enhanced RNN variants in speaker recognition areas. In addition, there is no attempt conducted using hybrid models of CNN and RNN variants for speaker recognition under noisy conditions. The features such as MFCC and GFCC which were good in speaker and speech recognition using machine learning method were not as effective as other features such as raw waveform, spectrogram and cochleogram in speaker recognition using deep learning. In most of the speaker, speech, emotion and language recognition, spectrogram of the speech were widely employed. In some speech processing applications using deep learning, cochleogram have shown better performance than spectrogram. However, none of the study conducted to analyze the noise robustness of cochleogram and spectrogram in speaker recognition using deep learning model to select the better feature for noisy conditions. In this study, text-independent speaker recognition using deep learning models was developed for noisy conditions. The robustness focused in this study was mainly to real world noise and white gaussian noise at different level of noise ratio. First, the noise robustness analysis of spectrogram and cochleogram features in speaker recognition were conducted using deep learning model. Based on the analysis the better features for noisy condition were selected for further model performance improvement under noisy conditions. Second, the speaker recognition models using a hybrid models of CNN and enhanced variants of RNN were developed on the selected feature for the noisy conditions. Then, the model with the better performance were proposed for the speaker recognition application under noisy

conditions. Finally, the proposed model performance were compared with the existing works to indicate the effectiveness of the proposed model.

CHAPTER THREE

3. METHODOLOGY

3.1. Introduction

This section discusses the methods followed to develop text-independent speaker recognition model for noisy conditions using deep learning models. In section 3.2, the research design method is presented to illustrate the detailed method followed in this study. In section 3.3, the datasets were prepared to evaluate each of the models and features in speaker recognition. Section 3.4, describes the spectrogram and cochleogram generation methods for evaluating both features noise robustness using deep learning and for evaluating the speaker recognition models developed for the noisy conditions in this study. In section 3.5, the description of the model selection method were presented in detail. In section 3.6, the methods followed to analyze the noise robustness of cochleogram and spectrogram features in the speaker recognition were presented in detail. In subsections of 3.6, the details of how CNN architectures such as basic 2DCNN, VGG-16, ResNet50, ECAPA-TDNN and TitaNet models have been employed for noise robustness analysis of cochleogram and spectrogram features in speaker recognition were presented in detail. Section 3.7, presents the methods followed to develop a speaker recognition models for the noisy conditions using deep learning approaches. In subsections 3.7.1, the methods followed to develop speaker recognition model using a hybrid CNN and LSTM model for noisy conditions were discussed. In section 3.7.2, the details of the speaker recognition model development process using a hybrid CNN and BiLSTM model for noisy conditions was presented. In section 3.7.3, the speaker recognition model using a hybrid CNN and GRU for noisy conditions were presented in detail. In section 3.7.4, the proposed model architecture was presented in detail. Section 3.8, presents the implementation details of this dissertation.

3.2. Research Design Method

Employing an appropriate research design plays crucial role in conducting the research. The design science research method were mainly focused in this study because its advantage in the development of models, systems, and frameworks. Design science research method provides an organized method for creating, refining and evaluating the models, systems and frameworks of the research. The design science research method follows a sequence of steps during conducting the

research such as problem identification and motivation, literature review, defining the goal, methods, tools, development of the model, evaluation and communication (F. & L., 2018).

Problem identification and motivation are necessary to show the gap in the existing speaker recognition models under the noisy conditions. In this stage, the researchers have identified the gaps in conventional machine learning method and the impact of features in the performance of the models under the noisy condition. The motivation for speaker recognition was driven by the possibility of its advantages in security, surveillance, forensic investigation and financial transactions. The identification and motivation of this topic are usually guided by a survey of the literature.

For the developing of speaker recognition model for noisy conditions using deep learning models, the assessment of previous studies, publications, and research pertaining in the area of speaker recognition were conducted. Finding and compiling pertinent literature that discusses issues, approaches, algorithms, and strategies utilized for the analysis of cochleogram and spectrogram features in speaker recognition using deep learning models at different levels of SNR and for the development of the speaker recognition model for noisy conditions using deep learning approach.

In the context of developing a speaker recognition model for noisy conditions using deep learning approach, the objective is to precisely define the goals which should be achieved at each level and at the end of the research work.

Design and development of the speaker recognition model for noisy conditions using deep learning approach refers to the conceptual design, organizing and carrying out experiments to verify its effectiveness, figuring out its architecture and functionality. This stage is essential for converting the specified needs and goals into a solution.

Communication is the final stage in the design science research and in our study which discusses the research findings, conclusions, and insights to a variety of perspective. This procedure includes sharing knowledge both orally and in writing through documents, research papers, conference presentations, technical reports, and documentation. The purpose of communication is to inform and educate stakeholders on the capabilities, limitations, and potential applications of the speaker recognition model

3.3. Dataset and Preparation Method

The speech dataset used in this work was obtained from the public VoxCeleb1 audio files dataset which was found in the source (Group, 2022). The main reasons for selecting VoxCeleb1 audio dataset were because it was collected in the real world conditions, contains significant number speakers in the dataset, representative amount of utterances in each speaker classes and appropriate ratio of gender, age and accents . Moreover, the VoxCeleb1 dataset has large number of data which is important to train deep learning models because deep learning models are data intensive. The VoxCeleb1 speech dataset was primarily prepared for speech-processing applications including speaker recognition. It was a free and publicly available dataset that helped the researcher to conduct various speech-processing experiments. The VoxCeleb1 dataset contains 153,516 utterances which were collected from 1251 speakers. The utterances were extracted from different kinds of celebrities uploaded to YouTube. The total number of male speakers and female speakers in the VoxCeleb1 speech dataset is approximately 55% and 45% respectively. The speakers span a wide range of different ethnicities, accents, professions and ages. The utterances with different types of English language accents were included in the dataset. Since each utterance was obtained from celebrity videos recorded in the real-world, it contains various types of speaking styles, age groups, language backgrounds, and professions. The sample frequency in the utterances of the dataset is 16 KHz. In each speaker class of the dataset, the total number of utterances and their length varies. Primarily, the original split (i.e., dev and test split) of the dataset was first merged into one to split for our experiment. In our experiment, the dataset was split into 80% for training, 10% for validation and 10% for testing. Although the VoxCeleb1 dataset contains noises in various ratios, it was assumed as a clean dataset.

The real-world noise and white Gaussian noise added to VoxCeleb1 datasets at the signal-to-noise ratio (SNR) of -5dB to 20dB in intervals of 5dB were generated from the original VoxCeleb1 dataset. The real-world noises such as babble, restaurant and street noises were used randomly to obtain real-world noise-added VoxCeleb1 datasets at the specified SNR levels. The noises used in this study were obtained from the source (Ellis, 2002). The main reason for selecting these real-world noise types were because it was very common in the real world commination and recognition. The white Gaussian noises which were generated by using Python code were employed to generate the VoxCeleb1 datasets with WGN at the specified SNR levels.

3.4. Spectrogram and Cochleogram Generation Process

Generation of cochleogram and spectrogram were conducted to convert utterances into the formats which represent speaker characteristics for recognition and deep learning models input tensor. Spectrogram gets generated directly from the Mel filter bank of the utterances which is rich in acoustic features of the speaker, whereas MFCC feature gets generated from the Mel filter bank of the utterances by employing logarithmic compression and discrete cosine transform which minimizes the acoustic feature of the speaker. Cochleogram gets generated directly from the gammatone filter banks of the utterances which is also rich in acoustic information of the speaker, whereas the GFCC feature gets generated from the gammatone filter banks of the utterances by employing cubic root compression and discrete cosine transform which reduces the acoustic information of the speaker. Acoustic information in the cochleogram and spectrogram is very crucial to train the deep learning model for speech processing applications including speaker recognition. Therefore, the cochleogram and spectrogram generation from the original and generated datasets were conducted for evaluating both features noise robustness in speaker recognition and to select better robust feature to use as an input for the speaker recognition models developed in this study. In this section the process followed to generate cochleogram and spectrogram were presented.

3.4.1. Spectrogram Generation Process

In this subsection, the details of the spectrogram generation process for the analysis of noise robustness in speaker recognition were discussed. Spectrogram were mainly focused because it was commonly employed in speaker recognition using the deep learning model. In this study, the spectrogram generation from each utterance follows pre-emphasis, framing, hamming window, fast Fourier transform, mel filter bank and power spectrum steps as presented in figure 14.

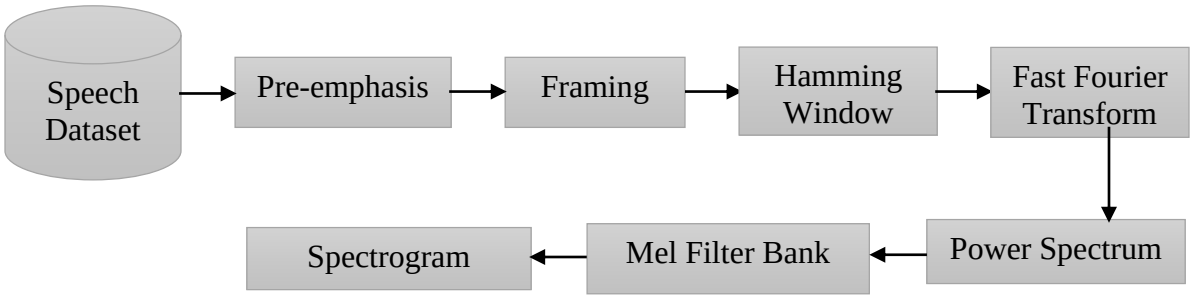

Figure 14: Spectrogram Generation Process

Input speech was pre-emphasized with the emphasis factor of 0.97 to minimize background noise and dynamicity of the speech. Then, the pre-emphasized speech was framed into segments of 30ms to get stable acoustic characteristics of the speakers. Hamming window was applied with the overlapping of 10ms (i.e. 40% of the frame) between adjacent frames to reduce discontinuity between the frames. Finding the acoustic characteristics of the speaker from the time domain of the speech frame is not efficient when compared with the frequency domain. Fast Fourier Transform (FFT) was employed to convert the frames from the time domain into the frequency domain. FFT of 128 filters and 2048 points were extracted from each of the frames in this study. Mel Filter Bank was applied to convert the frequency scale into Mel Scale representation simulate human auditory system. The power spectrum was computed from each of the Mel Scale to generate spectrogram features. Finally, the spectrogram features of the frames are stacked together to generate a spectrogram of an utterance. Sample raw waveform of the clean speech signal and its spectrogram are shown in Figures 15 and 16 respectively. Sample raw waveform of the speech with babble noise at SNR of 5dB and its spectrogram is shown in Figure 17 and 18 respectively.

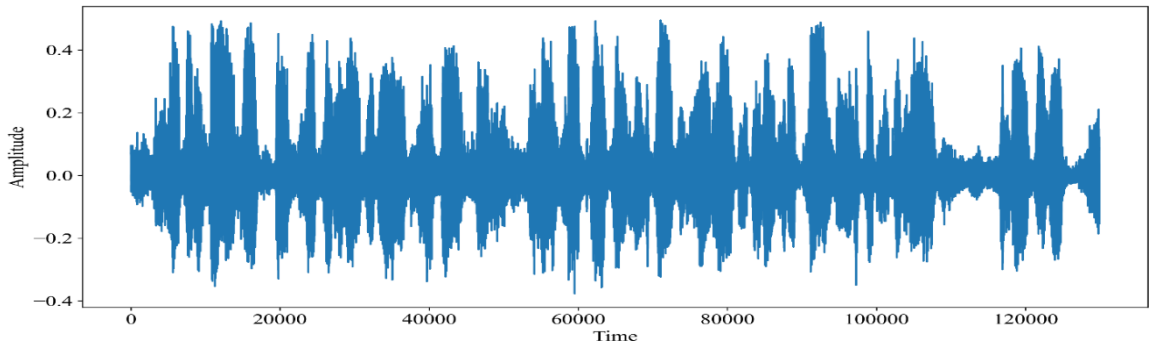

Figure 15: Sample raw waveform of clean speech

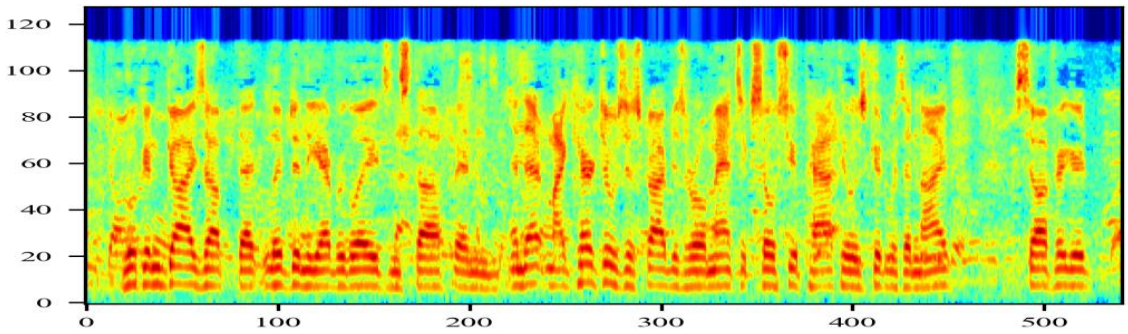

Figure 16: Spectrogram of the sample clean speech in fig.13

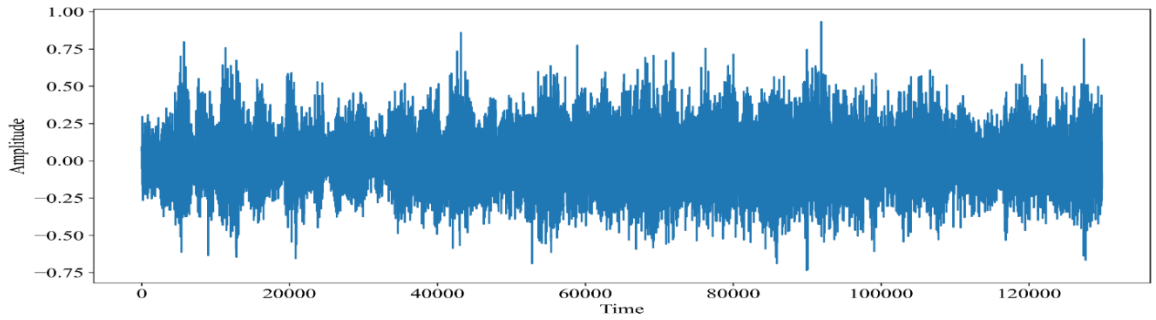

Figure 17: Sample raw waveform of the speech with babble noise at SNR of 5dB

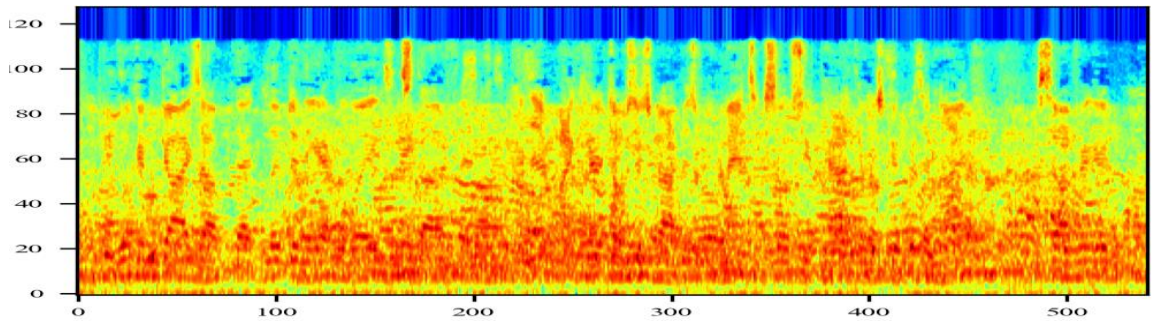

Figure 18: Spectrogram of the sample speech with babble noise at SNR of 5dB in figure 15

3.4.2. Cochleogram Generation Process

The Cochleogram is another two-dimensional time-frequency representation of speech to simulate the human auditory system based on the functionality of the inner ear or cochlea. Similar to spectrogram features, the cochleogram was rich in acoustic information about the speaker which is important for training the deep learning networks. In this study, Cochleogram generation has been conducted based on the sequence of tasks in Figure 19.

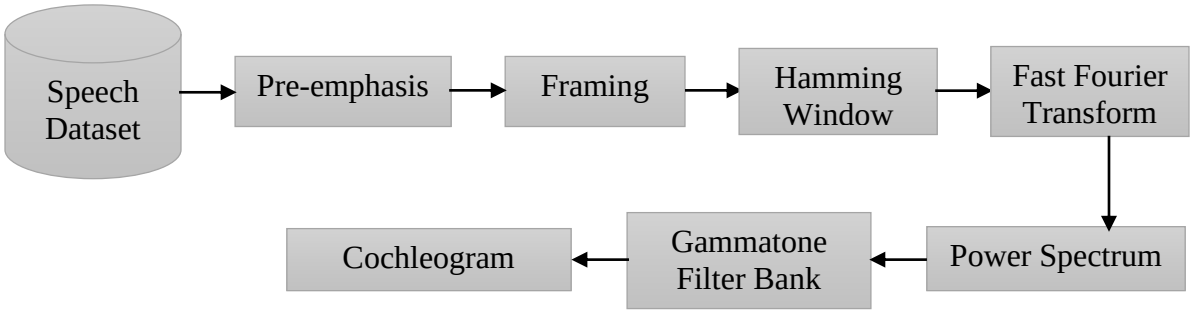

Figure 19: Cochleogram Generation Process

First, speech was pre-emphasized with an emphasis factor of 0.97, then it was framed into segments of 30ms with 10ms overlap and the hamming window was applied. From each frame the fast Fourier transform with the filters of 128 and filter points of 2048 were computed. To compute

gammatone filters from each FFT, the lower frequency was set to 0Hz and the higher frequency of 16 KHz was used. The central frequency of the gammatone filter was computed from the lower and higher frequencies. Next, ERB scale of the central frequencies was computed. Then, bandwidth of the gammatone filters was computed from the ERB scale of central frequency. Gammatone filters were computed from each of the FFT based on central frequency, bandwidth, time, amplitude, and filter orders. Each of the gammatone filter's power spectrum was stacked together to generate cochleogram of the utterance. The cochleogram of size 224x224x3 was generated which an important shape for the deep was learning input tensor of the models used in this study. The cochleogram of the sample speech waveform in Figure 15 and 17 are shown in the Figure 20 and 21 respectively.

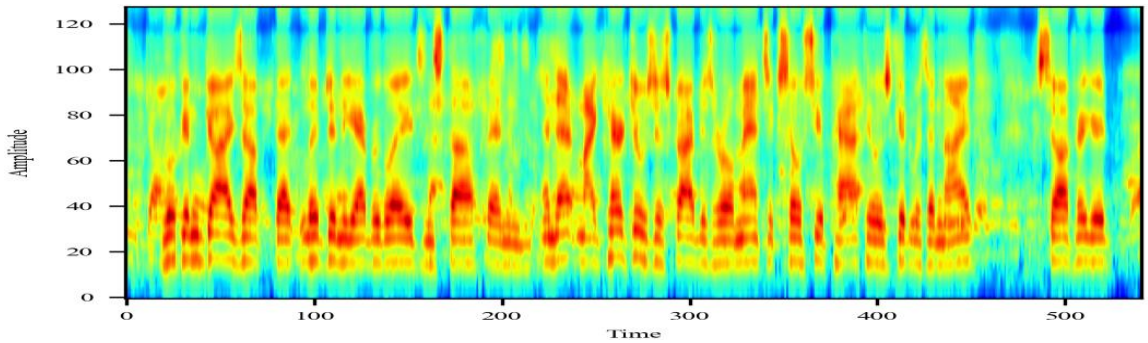

Figure 20: Sample cochleogram of the clean speech

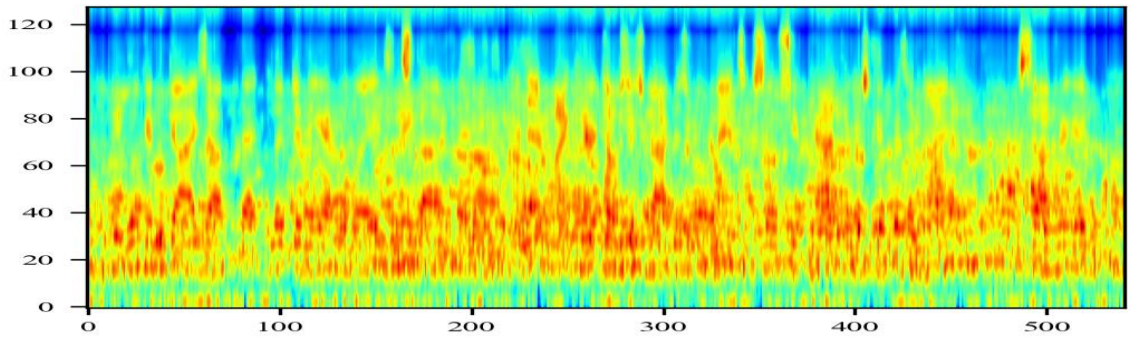

Figure 21: Sample Cochleogram of the speech with babble noise at SNR of 5dB

3.5. Model Selection Method

Model selection is one of the important component in machine learning and deep learning research works. Different type of machine learning and deep learning models are efficient based on the dataset types and size (labeled and unlabeled), areas of application and so on. Similarly, selection

of the features is also very crucial to achieve better performance in various areas. In case of speaker recognition, deep learning models have outperformed deep learning models because of the large number of neural networks. Because of the performance of the deep learning model we have employed the deep learning approach for our study. From the deep learning architectures, both convolutional neural network and recurrent neural networks were commonly employed in various studies in the speaker recognition areas. They also achieved better performance than other architectures. Therefore, both convolutional neural network (CNN) and enhanced recurrent neural network (RNN) variants were selected for this study. From the CNN architecture, customized 2DCNN, VGG16, ResNet50, ECAPA-TDNN and TitaNet models were selected for the noise robustness analysis of the cochleogram and spectrogram features in speaker recognition. These models were selected because they have better performance in the existing speaker recognition and image classification applications. For the speaker recognition model development, the CNN and RNN variants such as LSTM, BiLSTM, GRU and BiGRU were selected because of their good performance in various areas. The combination of CNN and RNN variants were employed together with the more robust feature to develop the speaker recognition models.

3.6. Noise Robustness Analysis of Cochleogram and Spectrogram

This section presents the methods followed to analyze the noise robustness of cochleogram and spectrogram features in speaker recognition. In the speech processing applications using machine learning methods, MFCC and GFCC features have better performance in clean and noisy conditions respectively. However, MFCC and GFCC features were not as effective as other inputs like raw waveform, spectrogram, and cochleogram in speech processing applications using deep learning model. However, the noise robustness of the cochleogram and spectrogram have not been conducted to select the better feature for speaker recognition under noisy condition. In this study, the noise robustness analysis of cochleogram and spectrogram in speaker recognition using deep learning model were conducted. The analysis was conducted by using basic 2DCNN, VGG-16, ResNet50, ECAPA-TDNN and TitaNet models. These architectures were selected because of their better performance in the area of speaker recognition, image classification and other computer vision applications. The evaluation was conducted on the VoxCeleb1 dataset with real-world noise at the signal-to-noise ratio ranging from -5dB to 20dB in the interval of 5dB and without additive

noises. Analysis of both features were conducted on speaker identification and verification at the SNR stated above.

3.6.1. Analysis of Cochleogram and Spectrogram using Basic 2DCNN

The basic 2DCNN model employed in this study was developed by customized basic layers of CNN architectures. The CNN models have an advantage in automatically extracting features from the input and adaptive learning from the pattern of the features. It has better performance in various applications when compared with the machine learning models. Since CNN architectures also have good performance in speaker recognition, this study employed it to analyze the noise robustness of cochleogram and spectrogram in speaker recognition. The basic 2DCNN model employed in this study consists of six layers (four convolutional and two fully connected). Each of the convolutional layers was followed by maxpooling and batch normalization layers. The basic 2DCNN model architecture is presented in Figure 22.

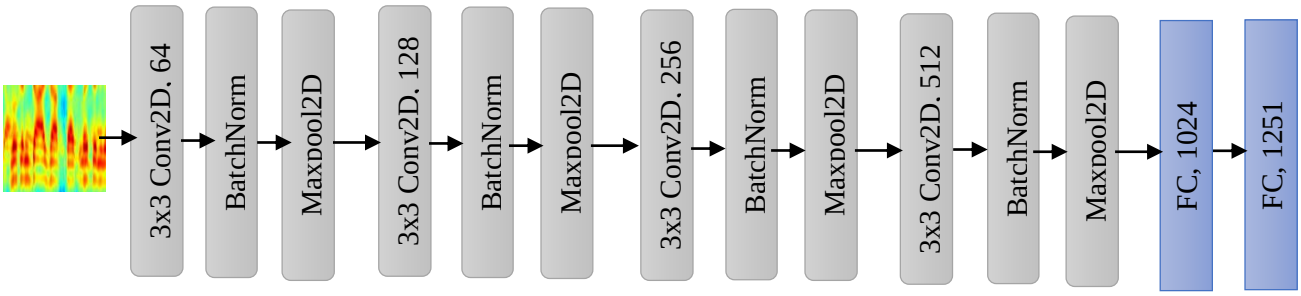

Figure 22: Basic 2DCNN Architecture

This model have employed an input (i.e., cochleogram or spectrogram) of the shape 224x224x3 which was appropriate for the input tensor. The initial batch size for the basic 2DCNN model was 32. The filter sizes of the first, second, third and fourth convolutional layers are 64, 128, 256 and 512 respectively. The kernel size of 3x3, the same padding and ReLu activation have been used in each of the convolutional layers. Each maxpooling layers in the model have a pool size of 2x2 and stride of 2x2. The flatten layer was connected immediately after the last maxpooling to convert the feature maps into the input shape of the fully connected layer. The dropout of 0.5 (50%) was employed after the flatten layer. To compute the loss at each epoch, categorical_crossentropy has been employed together with the RMSprop optimizer. To evaluate the noise robustness of the cochleogram and spectrogram in speaker identification, the accuracy metrics have been employed. The noise robustness of cochleogram and spectrogram in speaker verification have been evaluated using an equal error rate (EER) metrics. The softmax activation function with 1251 classes has

been used to map the input into the output. The model was trained for 20 epochs, at each epoch loss and accuracy were computed. The detailed implementation summary of the basic 2DCNN architecture used in this study is presented in Table 2.

Table 2: Basic 2DCNN Model Summary

Layer No	Layer	Description	Output Shape	Param#
1	Input (224x224x3)	Cochleogram or Spectrogram	-	-
2	Conv2D	f = 64, k = (3,3), p=same, a=ReLu	(None, 112,112,64)	1792
3	BatchNorm	-	(None, 112,112,64)	256
4	Maxpooling2D	Pool size = 2 × 2, s = 2 × 2	(None, 56,56,128)	0
5	Conv2D	f = 128, k = (3,3), p=same, a=ReLu	(None, 56,56,128)	73856
6	BatchNorm	-	(None, 56,56,128)	512
7	Maxpooling2D	Pool size = 2 × 2, s = 2 × 2	(None, 28,28,256)	0
8	Conv2D	f = 256, k = (3,3), p=same, a=ReLu	(None, 28,28,256)	295168
9	BatchNorm	-	(None, 28,28,256)	1024
10	Maxpooling2D	Pool size = 2 × 2, s = 2 × 2	(None, 14,14,512)	0
11	Conv2D	f = 512, k = (3,3), p=same, a=ReLu	(None, 14,14,512)	1180160
12	BatchNorm	-	(None, 14,14,512)	2048
13	Maxpooling2D	Pool size = 2 × 2, s = 2 × 2	(None, 7,7,1024)	0
14	Flatten	-	(None, 50176)	0
15	Fully Connected	1024	(None, 1024)	25691136
16	BatchNormalization	-	(None, 1024)	4096
17	Dropout	-	(None, 1024)	0
18	Fully Connected	1251	(None, 1251)	1282275
Total params: 28,532,323 Trainable params: 28,528,355 Non-trainable params: 3,968				

Where, f = filters, k = kernel size, s = stride, p = padding, a = activation.

3.6.2. Analysis of Cochleogram and Spectrogram using VGG-16

In this section, the methods followed to conduct noise robustness analysis of the cochleogram and spectrogram in speaker recognition by using VGG-16 architecture is presented. The VGG-16 architecture was one of the standard CNN architectures which have good performance in image classification and ILSVRC image classification computation of 2014. The VGG-16 has sixteen

layers (thirteen convolutional and three fully connected). The convolutional layers of this model were grouped into five blocks, each block followed by the maxpooling layer. The first and the second blocks have two convolution layers and the remaining each blocks have three convolutional layers. The VGG-16 architecture employed in this study is shown in Figure 23.

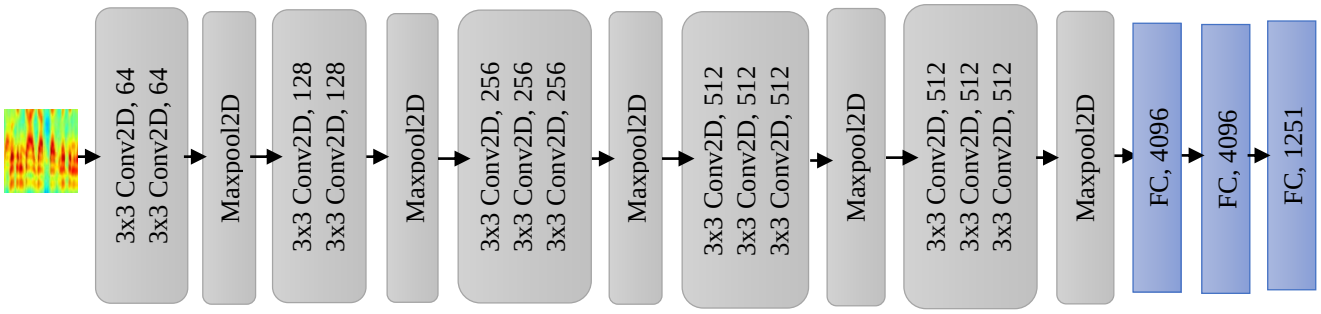

Figure 23: The Architecture of the VGG-16

The filter sizes of the convolution layers in the block one to five 64, 128, 256, 512 and 512 respectively. The kernel size of 3x3, the same padding and ReLu activation have been employed in each of the convolution layers of the model. The maxpooling layer with the pool size of 2x2 and stride of 2x2 has been inserted after each block to reduce the dimension of the training feature maps. The flatten layer was connected after the fifth block and the last maxpooling layer to transform the feature map dimension into the input shape of the fully connected layer. The three fully connected layers were inserted at the end of the model. The first and second fully connected layers in this model used ReLu activation together with 4096 units. The number of units in the last fully connected layer was equal to the number of classes or speakers in the dataset (i.e., 1251) and it employed the softmax activation function. The loss of the model was computed using the categorical_crossentropy function and for model optimization, the RMSprop function was used for optimization. The noise robustness of the cochleogram and spectrogram in speaker identification have been evaluated using accuracy metrics. The EER metrics were used to evaluate the noise robustness of the cochleogram and spectrogram in speaker verification. The detailed summary of the VGG-16 architecture employed in this study is presented in Table 3.

Table 3: VGG-16 Model Summary

Layer No.	Layer	Description	Output Size	Param #
1	Input (224x224x3)	Cochleogram or Spectrogram	-	-
2	Conv2D	f = 64, k = 3 × 3, *p* = same, a = ReLu	(None, 224, 224, 64)	1792
3	Conv2D	f = 64, k = 3 × 3, *p* = same, a = ReLu	(None, 224, 224, 64)	36928
4	Maxpool	pool = 2 × 2, s = 2 × 2	(None, 112, 112, 64)	0
5	Conv2D	f = 128, k = 3 × 3, *p* = same, a = ReLu	(None, 112, 112, 128)	73856
6	Conv2D	f = 128, k = 3 × 3, *p* = same, a = ReLu	(None, 112, 112, 128)	147584
7	Maxpool	pool =2 × 2, s = 2 × 2	(None, 56, 56, 128)	0
8	Conv2D	f = 256, k = 3 × 3, *p* = same, a = ReLu	(None, 56, 56, 256)	295168
9	Conv2D	f = 256, k = 3 × 3, *p* = same, a = ReLu	(None, 56, 56, 256)	590080
10	Conv2D	f = 256, k = 3 × 3, *p* = same, a = ReLu	(None, 56, 56, 256)	590080
11	Maxpool	pool = 2 × 2, s = 2 × 2	(None, 28, 28, 256)	0
12	Conv2D	f = 512, k = 3 × 3, *p* = same, a = ReLu	(None, 28, 28, 512)	1180160
13	Conv2D	f = 512, k = 3 × 3, *p* = same, a = ReLu	(None, 28, 28, 512)	2359808
14	Conv2D	f = 512, k = 3 × 3, *p* = same, a = ReLu	(None, 28, 28, 512)	2359808
15	Maxpool	pool = 2 × 2, s = 2 × 2	(None, 14, 14, 512)	0
16	Conv2D	f = 512, k = 3 × 3, *p* = same, a = ReLu	(None, 14, 14, 512)	2359808
17	Conv2D	f = 512, k = 3 × 3, *p* = same, a = ReLu	(None, 14, 14, 512)	2359808
18	Conv2D	f = 512, k = 3 × 3, *p* = same, a = ReLu	(None, 14, 14, 512)	2359808
19	Maxpool	pool = 2 × 2, s = 2 × 2	(None, 7, 7, 512)	0
20	Flatten		(None, 25088)	0
21	Fully Connected		(None, 4096)	102764544
22	Fully Connected		(None, 4096)	16781312
23	Fully Connected		(None, 1251)	5125347
Total params: 139,385,891 Trainable params: 139,385,891 Non-trainable params: 0				

3.6.3. Analysis of Cochleogram and Spectrogram using ResNet50

This section presents the process of noise robustness analysis of cochleogram and spectrogram features in speaker recognition using ResNet50 architecture. ResNet50 was one of the standard CNN architecture with a large number of convolutional layers each having different sizes of kernel and filters. The ResNet50 architectures have fifty (50) layers (49 convolutional and one fully connected layer). The convolutional layers in the ResNet50 architectures were grouped into five

stages or blocks, each block having different number of iterations. The first block has a single convolutional layer with a kernel size of 7x7 and a filter size of 64 followed by batch normalization and maxpooling of 2x2 pool size and 2x2 strides. The first block iterates only once during the model training. Each of the remaining blocks has three convolutional layers from which two of the convolutional layers (i.e., the first and third) have 1x1 kernel size and one convolutional layer (i.e., the second) has a 3x3 kernel size in each block. In the second block the first two convolution layers have 64 filters and the third convolutional layer has 256 filters. In the third block, the first two convolutional layers have 128 filters and the third convolutional layer has 512 filters. In the fourth block, the first and second convolutional layers have 256 filters and the third convolutional layer has 1024 filters. In the fifth block, the first and second convolutional layers have 512 filters and the third convolutional layer has 2048 filters. The number of iterations of the block one to five were one, three, four, six and three times respectively. Each convolutional layers have ReLu activation and the same padding. AveragePooling was inserted before the fully connected layer to reduce the dimension of the feature maps. The flatten layer was also included to transform the output shape of the AveragePooling into the input shape of the fully connected layers. A fully connected layer with the units equal to the number of classes or speakers in the dataset (i.e., 1251) and softmax activation function were connected at the end of the model to convert the input shape into the output shape of the model. The block diagram for the ResNet50 architecture employed in this study was presented as shown in Figure 24.

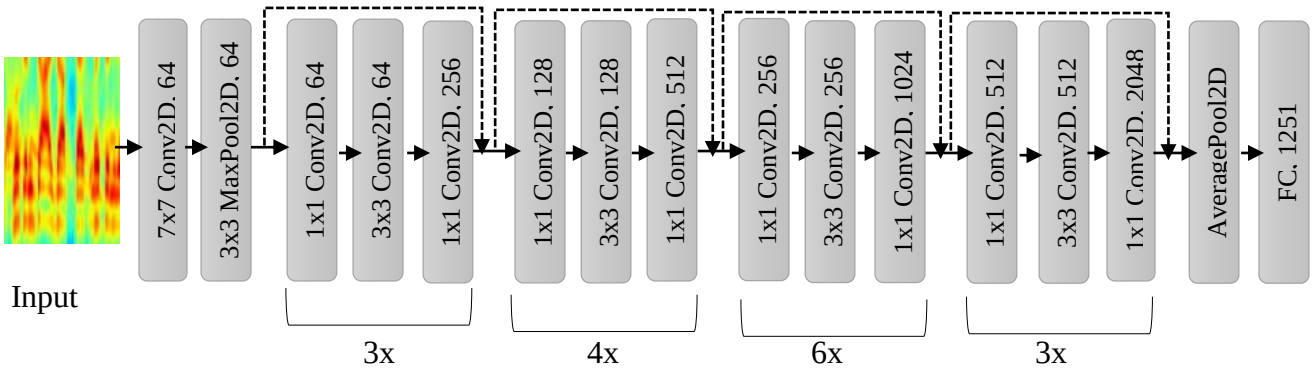

Figure 24: The architecture of the ResNet50 model

To compute the loss of the model at each of the epochs, categorical_crossentropy has been employed together with the RMSprop optimizer. To evaluate the noise robustness of the cochleogram and spectrogram in speaker identification the accuracy metrics were used. The noise robustness of cochleogram and spectrogram in speaker verification were evaluated using the EER

metrics. The detailed summary of the ResNet50 architecture employed in this study is presented in Table 4.

Table 4: ResNet50 Model Summary

Block	Layer	Description	Output	Param#	Iteration
	Input	Cochleogram or Spectrogram	224x224x3	0	
Conv1.x	ZeroPad	Size = 3 × 3	(None, 230, 230, 3)	0	1×
	Conv2D	f = 64, k = 7 × 7, strides = 2 × 2, a = ReLu	(None, 112, 112, 64)	9472	
	MaxPool2D	Pool size = 3 × 3, strides = 2 × 2	(None, 55, 55, 64)	0	
Conv2.x	Conv2D	f = 64, k = 1 × 1, s = 1 × 1, *p* = same, a=ReLu	(None, 55, 55, 64)	4160	3×
	Conv2D	f = 64, k = 3 × 3, s = 1 × 1, *p* = same, a=ReLu	(None, 55, 55, 64)	36928	
	Conv2D	f=256, k = 1 × 1, s = 1 × 1, *p* = same, a=ReLu	(None, 55, 55, 256)	16640	
Conv3.x	Conv2D	f =128, k =1 × 1, s =2 × 2, *p* = same, a =ReLu	(None, 28, 28, 128)	65664	4×
	Conv2D	f =128, k =3 × 3, s =1 × 1, *p* = same, a = ReLu	(None, 28, 28, 128)	147584	
	Conv2D	f =512, k =1 × 1, s =1 × 1, *p* = same, a = ReLu	(None, 28, 28, 512)	131584	
Conv4.x	Conv2D	f =256, k =1 × 1, s =2 × 2, *p* = same, a = ReLu	(None, 14, 14, 256)	262400	6×
	Conv2D	f =256, k =3 × 3, s =1 × 1, *p* =same, a =ReLu	(None, 14, 14, 256)	590080	
	Conv2D	f =1024, k =1 × 1, s =1 × 1, *p* = same, a=ReLu	(None, 14, 14, 1024)	263168	
Conv5.x	Conv2D	f =512, k =1 × 1, s =2 × 2, *p* = same, a = ReLu	(None, 7, 7, 512)	1049088	3×
	Conv2D	f =512, k =3 × 3, s =1 × 1, *p* =same, a =ReLu	(None, 7, 7, 512)	2359808	
	Conv2D	f =2048, k =1 × 1, s =1 × 1, *p* = same, a=ReLu	(None, 7, 7, 2048)	1050624	
-	AveragPool	Pool=2x2, p=same	(None, 4, 4, 2048)	0	1
-	Flatten	-	(None, 32768)	0	1
-	FC	Units=1251, a=softmax	(None, 1251)	40994019	1
Total params: 62,628,451 Trainable params: 62,628,451 Non-trainable params: 0					

3.6.4. Analysis of Cochleogram and Spectrogram using ECAPA-TDNN

In this section, the methods followed to analyze the noise robustness of the cochleogram and spectrogram in speaker recognition using ECAPA-TDNN is presented. The ECAPA-TDNN (Emphasized Channel Attention, Propagation and Aggregation in Time Delay Neural Network), which is a customized CNN architecture and has shown better performance in speaker verification (Desplanques, Thienpondt, & Demuynck, 2020) has been adopted in this study to evaluate noise robustness of cochleogram and spectrogram in speaker recognition. The model has a total of seven

layers (two convolutional, three SE-Res2Block and two fully connected layers). The three SE-ReS2Blocks were consecutively inserted at the second, third and fourth layers that were between the two convolutional layers of the ECAPA-TDNN. Each SE-Res2Block layer has four layers (two convolutional, one dilated convolutional and one SE-Block layer). The output of each SE-Res2Block have been used as an input for the second convolutional layer of the ECAPA-TDNN. The details of ECAPA-TDNN architecture employed in this study is presented in Figure 25.

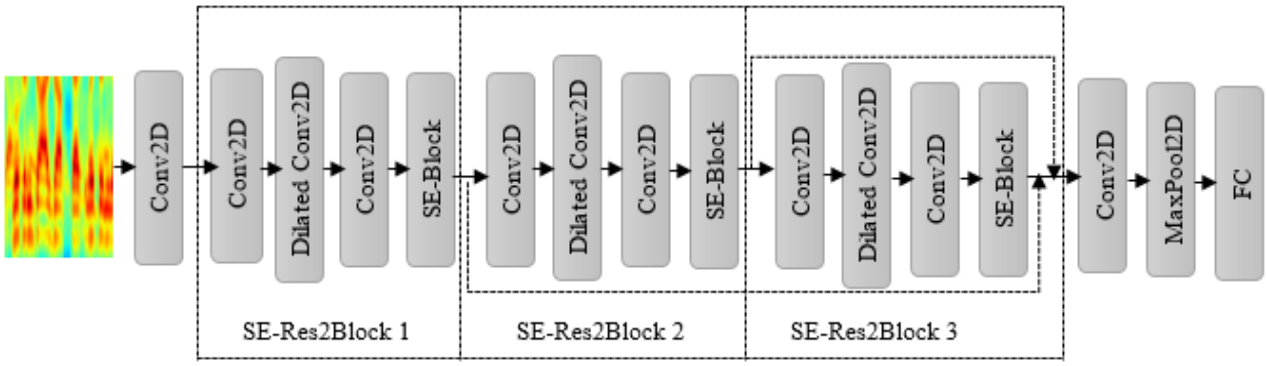

Figure 25: ECAPA-TDNN architecture

Each of the convolutional and dilated convolutional layers have ReLu activation, the same padding, batch normalization and kernel of 3x3. The filter sizes of the first and second convolutional layers outside the SE-Res2Block were 64 and 128, respectively. The filter sizes of the convolution layers in the SE-Res2Block block were 64 and 128 respectively. The dilated convolutional layer in the SE-Res2Block have filter sizes of 64. Maxpooling layer with the pool size of 2x2 and stride of 2x2 has been used between convolutional layer and fully connected layer to reduce the feature dimensions for the fully connected layer. Fully connected layer was connected at the end of the model for classification. In fully connected layer, softmax function and the number of unit equal to 1251. To compute the loss of the model at each of the epochs, categorical_crossentropy has been employed together with the RMSprop optimizer. The noise robustness of cochleogram and spectrogram in speaker identification have been evaluated using accuracy metrics. Whereas, the EER metrics were used to evaluate the noise robustness of cochleogram and spectrogram in speaker verification. The detailed summary of the ECAPA-TDNN architecture employed in this study is presented in Table 5.

Table 5: Implementation details of the ECAPA-TDNN Architecture

Layer No.	Layer	Description	Output Size	Param #
1	Input	Cochleogram or Spectrogram	(None, 224,224,3)	0
2	Conv2D	f = 64, k = 3 × 3, *p* = same, a=ReLu	(None, 224, 224, 64)	1792
3	Conv2D	f = 64, k = 3 × 3, *p* = same, a=ReLu	(None, 224, 224, 64)	36928
4	Conv2D	f = 128, k = 3 × 3, *p* = same, a=ReLu	(None, 224, 224, 128)	73856
5	Conv2D	f = 256, k = 3 × 3, *p* = same, a=ReLu	(None, 224, 224, 256)	295168
6	GlobalAveragePooling2D	-	(None, 256)	0
7	Dense	Units=16, a=ReLu	(None, 16)	4112
8	Dense	Units=32, a=ReLu	(None, 256)	4352
9	Reshape	-	(None, 1, 1, 256)	0
10	Multiply	-	(None, 224, 224, 256)	0
11	MaxPooling2D	Pool=2x2, stride=2x2	(None, 112, 112, 256)	0
12	Dropout	-	(None, 112, 112, 256)	0
13	Conv2D	f = 64, k = 3 × 3, *p* = same, a=ReLu	(None, 112, 112, 64)	147520
14	Conv2D	f = 128, k = 3 × 3, *p* = same, a=ReLu	(None, 112, 112, 128)	73856
15	Conv2D	f = 256, k = 3 × 3, *p* = same, a=ReLu	(None, 112, 112, 256)	295168
16	GlobalAveragePooling2D	-	(None, 256)	0
17	Dense	Units=16, a=ReLu	(None, 16)	4112
18	Dense	Units=32, a=ReLu	(None, 256)	4352
19	Reshape	-	None, 1, 1, 256)	0
20	Multiply	-	None, 112, 112, 256)	0
21	MaxPooling2D	Pool=2x2, stride=2x2	(None, 56, 56, 256)	0
22	Dropout	-	None, 56, 56, 256)	0
23	Conv2D	f = 128, k = 3 × 3, *p* = same, a=ReLu	(None, 56, 56, 128)	295040
24	Conv2D	f = 256, k = 3 × 3, *p* = same, a=ReLu	(None, 56, 56, 256)	295168
25	Conv2D	f = 512, k = 3 × 3, *p* = same, a=ReLu	(None, 56, 56, 512)	1180160
26	GlobalAveragePooling2D	-	(None, 512)	0
27	Dense	Units=32, a=ReLu	(None, 32)	16416
28	Dense	Units=64, a=ReLu	(None, 512)	16896
29	Reshape	-	(None, 1, 1, 512)	0
30	Multiply	-	(None, 56, 56, 512)	0
31	MaxPooling2D	Pool=2x2, stride=2x2	(None, 28, 28, 512)	0
32	Dropout	-	(None, 28, 28, 512)	0
33	Conv2D	f = 256, k = 3 × 3, *p* = same, a=ReLu	(None, 28, 28, 256)	1179904
34	GlobalAveragePooling2D	-	(None, 256)	0

35	Dense	Units=64, a=ReLu	(None, 512)	131584
36	Dropout		(None, 512)	0
37	Dense	Units=1251, a=softmax	(None, 1251)	641763
Total params: 4,698,147 Trainable params: 4,698,147 Non-trainable params: 0				

3.6.5. Analysis of Cochleogram and Spectrogram using TitaNet

This section presents the process followed to analyze the noise robustness of cochleogram and spectrogram in speaker recognition using TitaNet model of CNN architecture. The TitaNet model employed in this study was adopted from the study (Koluguri, Park, & Ginsburg, 2021), because it has better performance in speaker verification. The model contains fourteen (14) basic layers (three convolutional, five depth-wise convolutional, four point-wise convolutional, one SE-Block and one fully connected layer). First, two of the convolutional layers were connected at the beginning of the model consecutively. Then, depth-wise and point-wise convolutional layers which were connected after the two convolutional layers iterates three times and batch normalization were applied at each iteration. The depth-wise convolution has an advantage in extracting fewer number of parameters which reduces cost of computation and overfitting problem. Next, two depth-wise convolutional layers were employed. SE-Block was employed after the last depth-wise convolutional layer. Another point-wise convolutional layer was connected directly after the batch normalization of the second convolutional layer and its output together with the output of the SE-block and second convolutional layer was employed as an input to the third convolutional layer. At the end fully connected layer was employed for classification. The architecture of the TitaNet model employed for robustness analysis of cochleogram and spectrogram features in speaker recognition is shown in Figure 26.

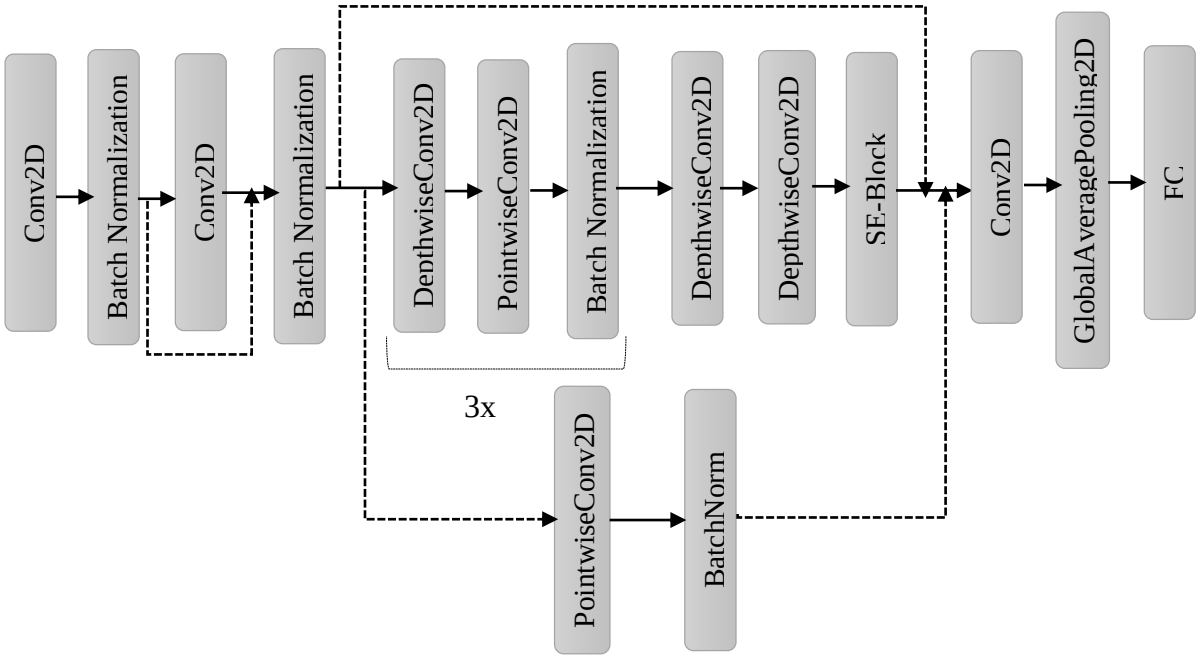

Figure 26: TitaNet Model Architecture

Each convolutional, depth-wise and point-wise (separable) convolutional layers have the kernel size of 3x3, the same padding and ReLu activation. The filter sizes of the first, second, and third convolutional layers were 64, 128, and 256 respectively. The filter sizes of the first, second, third and fourth point-wise convolutional layers were 64, 128, 256 and 512 respectively. The fully connected layer with the softmax activation function and 1251 number of units which was equal to number of speaker class in VoxCeleb1 dataset were employed at the end of the model. To compute the loss of the model at each of the epochs, categorical_crossentropy has been employed together with the RMSprop optimizer. The noise robustness of the cochleogram and spectrogram in speaker identification were evaluated using accuracy metrics, whereas the noise robustness in speaker verification were evaluated using the EER metrics. The detailed implementation summary of the TitaNet model employed in this study is presented in Table 6.

Table 6: TitaNet Architecture Model Summary

Layer No.	Layer	Description	Output Size	Param #
1	Input	Cochleogram or Spectrogram	(None, 224,224,3)	0
2	Conv2D	f = 64, k = 3 × 3, *p* = same, a=ReLu	(None, 224, 224, 64)	1792
3	BatchNormalization	-	(None, 224, 224, 64)	256
4	Conv2D	f = 128, k = 3 × 3, *p* = same, a=ReLu	(None, 224, 224, 128)	73856
5	BatchNormalization	-	(None, 224, 224, 128)	512
6	Concatenate	Concatenates output of layer 2 and 3	(None, 224, 224, 192)	0
7	DepthwiseConv2D	k = 3 × 3, *p* = same, a=ReLu	(None, 224, 224, 192)	1920
8	SeparableConv2D	f = 64, k = 3 × 3, *p* = same, a=ReLu	(None, 224, 224, 64)	14000
9	BatchNormalization	-	(None, 224, 224, 64)	256
10	Dropout	Dropout rate=0.5	(None, 224, 224, 64)	0
11	DepthwiseConv2D	k = 3 × 3, *p* = same, a=ReLu	(None, 224, 224, 64)	640
12	SeparableConv2D	f = 128, k = 3 × 3, *p* = same, a=ReLu	(None, 224, 224, 128)	8896
13	BatchNormalization	-	(None, 224, 224, 128)	512
14	Dropout	-	(None, 224, 224, 128)	0
15	DepthwiseConv2D	k = 3 × 3, *p* = same, a=ReLu	(None, 224, 224, 128)	1280
16	SeparableConv2D	f = 256, k = 3 × 3, *p* = same, a=ReLu	(None, 224, 224, 256)	34176
17	BatchNormalization	-	(None, 224, 224, 256)	1024
18	Dropout	Dropout rate=0.5	(None, 224, 224, 256)	0
19	DepthwiseConv2D	k = 3 × 3, *p* = same, a=ReLu	(None, 224, 224, 256)	2560
20	DepthwiseConv2D	k = 3 × 3, *p* = same, a=ReLu	(None, 224, 224, 256)	2560
21	GlobalAveragePooling2D	-	(None, 256)	0
22	Dense	-	(None, 16)	4112
23	Dense	-	(None, 256)	4352
24	SeparableConv2D	f = 256, k = 3 × 3, *p* = same, a=ReLu	(None, 224, 224, 512)	67200
25	Reshape	-	(None, 1, 1, 256)	0
26	BatchNormalization	-	(None, 224, 224, 512)	2048
27	Concatenate	Concatenates output of layer 3, 23 and 24	(None, 224, 224, 896)	0
28	Conv2D	f = 512, k = 3 × 3, *p* = same, a=ReLu	(None, 224, 224, 512)	4129280
29	GlobalAveragePooling2D	-	(None, 256)	0
30	Dense	-	(None, 1251)	321507
Total params: 4,351,312 Trainable params: 4,349,008 Non-trainable params: 2,304				

3.7. Speaker Recognition Model for Noisy Condition using Deep Learning Models

In section 3.4, the process of noise robustness analysis of cochleogram and spectrogram features in speaker recognition using basic 2DCNN, VGG-16, ResNet50, ECAPA-TDNN and TitaNet models of CNN architectures were presented in detail. The analysis results in Tables 11 and 12 confirmed that cochleogram have better performance than spectrogram in speaker recognition under noisy conditions using deep learning models. Therefore, cochleogram have been selected as an input during the development of speaker recognition models for noisy conditions using deep learning models.

As discussed above in the literature, the hybrid models of CNN and enhanced RNN variants have been showing better performance in various areas including speaker recognition. Enhanced variants of RNN include LSTM, BiLSTM, GRU and BiGRU. However, only limited studies have been conducted using hybrid CNN and enhanced variants of RNN in speaker recognition.

This section presents the methods followed to develop speaker recognition models using hybrid CNN and enhanced RNN variants for noisy conditions. Section 3.5.1, presents the process followed to develop a speaker recognition model using hybrid CNN and LSTM model for the noisy conditions. In section 3.5.2, the methods followed to develop speaker recognition model using hybrid CNN and BiLSTM model for noisy conditions was presented. In section 3.5.3, the method followed to develop a speaker recognition model using a hybrid CNN and GRU model for noisy conditions was presented. Section 3.5.4, presents the methods followed to develop proposed speaker recognition model for noisy condition using hybrid CNN and BiGRU model. Cochleogram was used as an input in each speaker recognition model developed this study.

3.7.1. Speaker Recognition Model using Hybrid CNN and LSTM

The hybrid CNN and LSTM or CNN-LSTM which was employed in speaker recognition mainly evaluated using spectrogram of the clean speech. As confirmed during the analysis of cochleogram and spectrogram features in this study, spectrogram have lower performance than cochleogram in speaker recognition under noisy conditions. However, the CNN-LSTM model with the cochleogram input were not evaluated in speaker recognition at different types of noises and noise to signal ratio. This section presents the methods followed to develop speaker recognition model

using the hybrid CNN and LSTM model for the noisy conditions. The model employed the cochleogram of the size 224x224x3 as an input which was appropriate input tensor shape.

The CNN-LSTM model developed for speaker recognition in this study consists of five layers (two convolutional, two LSTM and one fully connected). The convolutional layers were employed for the short term temporal features extraction and LSTM layers were employed for the long term feature map generation. Cochleogram of the speech was employed as an input because of its rich acoustic feature and better robustness. Both convolutional layers in this model were connected at the beginning of the model. The reshape layer were connected after the second convolutional layer to convert the output shape of CNN layers into the input shape of the LSTM layers. The LSTM layers were connected immediately after the reshape layer. Each LSTM layer used the output of reshape layer as an input and operate in parallel. Concatenate layer were connected after the LSTM layers to convert the output of the LSTM layers into input shape of the fully connected layer. Finally fully connected layer were connected at the end to perform classification. The architecture of the CNN-LSTM model was presented in the figure 27.

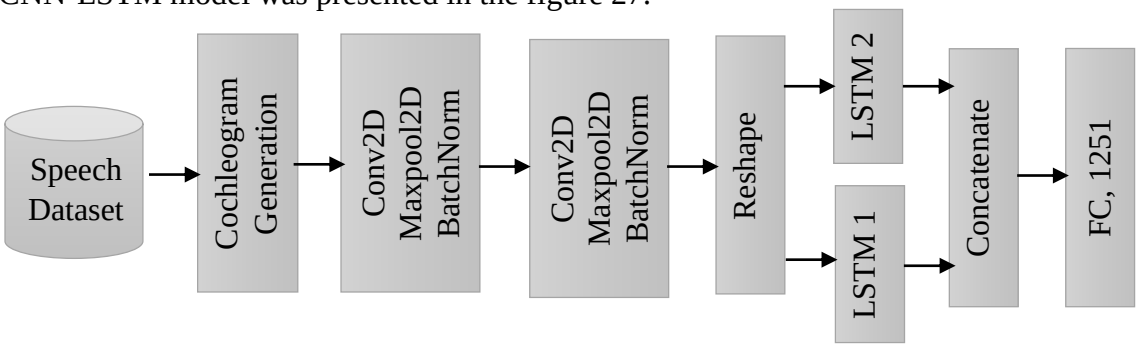

Figure 27: The architecture of CNN-LSTM Model

The filter sizes in the first and second convolutional layers are 16 and 32 respectively. The kernel size of 3x3, the same padding and ReLu activation were used in each convolutional layers. Maxpooling with the pool size of 2x2 and stride of 2x2 have been employed together with batch normalization after each convolutional layer. Each LSTM layers have 256 units to extract long-term correlation between the features. In the fully connected layer the number of units was equal to the number of class sizes or number of speakers in the dataset which was 1251 and softmax activation function has been used. To compute the loss of the model at each of the epochs, categorical-cross-entropy has been employed together with the RMSprop optimizer. The speaker identification and verification performance of CNN-LSTM model were evaluated using the accuracy and EER metrics respectively. The model was trained for 50 epochs, at each epoch loss

and accuracy were computed. The detailed implementation summary of the CNN-LSTM model for speaker recognition under noisy condition is presented in Table 7.

Table 7: CNN-LSTM Model Summary

Layer No.	Layer Name	Description	Output Shape	Param #
1	Input	Cochleogram	(None, 224, 224, 3)	0
2	Conv2D	Filters=16, kernel=(3, 3), padding=same	(None, 224, 224, 64)	1792
3	MaxPooling2D	Pool=(2, 2)	(None, 112, 112, 64)	0
4	BatchNorm	-	(None, 112, 112, 64)	0
5	Conv2D	Filters=32, kernel=(3,3) , padding=same	(None, 112, 112, 128)	73856
6	MaxPooling2D	Pool=(2, 2)	(None, 56, 56, 128)	0
7	BatchNorm	-	(None, 56, 56, 128)	0
8	Reshape	-	(None, 56, 7168)	0
9	LSTM	unit=256, kernel_initializer=he_normal	(None, 256)	7603200
10	LSTM	unit=256, kernel_initializer=he_normal	(None, 256)	7603200
11	Concatenate	-	(None, 512)	0
12	Dense	Filters=512	(None, 1251)	641763
13	Softmax	Classes = 1251	(None, 1251)	0
Total params: 15,923,811 Trainable params: 15,923,811 Non-trainable params: 0				

3.7.2. Speaker Recognition Model using Hybrid CNN and BiLSTM

This section presents the steps followed to develop a speaker recognition model for noisy conditions using a hybrid CNN and BiLSTM or CNN-BiLSTM method. Here the CNN layers were employed to extract short-term dependency between the features, whereas the BiLSTM layers were used to extract long-term correlation between the features during the model training. The cochleogram of the speech was used as an input for this model. The CNN-LSTM model developed for speaker recognition in this study has a total of 5 layers (two convolutional, two bidirectional LSTM and one fully connected). The two convolutional layers were connected at the beginning of the model consecutively. The BiLSTM layers were inserted after the second convolutional layer. The fully connected layer was connected at the end of the layer for classification purposes. The reshape layer was connected between the convolutional and BiLSTM layers to convert the output shape of the convolutional layers into the input shape of the BiLSTM

layers. Both BiLSTM layers (i.e., BiLSTM 1 and BiLSTM 2) are connected and operated in parallel by using the output of the reshape layer as an input. The concatenate layer was inserted after the BiLSTM layers to concatenate the output of both BiLSTM layers to shape the feature maps into the input shape of the fully connected layer. The architecture of the CNN-BiLSTM model employed in this study for speaker recognition under noisy conditions is presented in Figure 28.

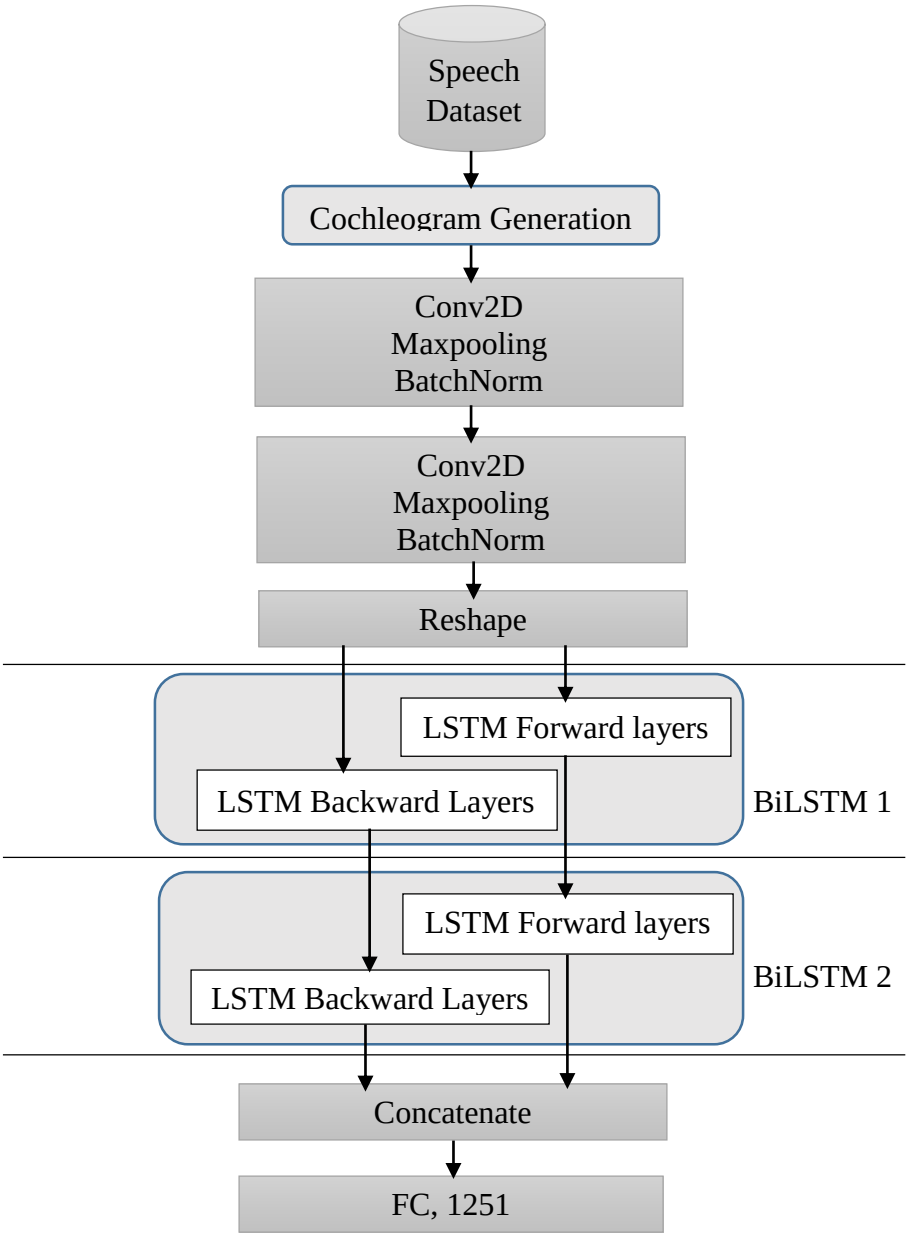

Figure 28: The Architecture of the CNN-BiLSTM Model

Each convolutional layer has the kernel size of 3x3, ReLu activation and the same padding. The filter sizes of the first and second convolutional layers were 64 and 128 respectively. Maxpooling layer with the pool size of 2x2 and stride of 2x2 was connected after each convolutional layer. The BiLSTM layers with the 256 cell units have been employed to extract and adaptively learn the long-term speaker related feature correlation from the cochleogram input. The fully connected layer has 1251 units which is equal to the number of classes or speakers in the dataset of VoxCeleb1 dataset. The softmax activation function was also applied to convert the input of the fully connected layers into the output shape. To compute the loss of the model at each epochs, categorical-crossentropy has been employed together with the RMSprop optimizer. The speaker identification and verification performance of the CNN-BiLSTM model were evaluated using accuracy and EER metrics respectively. The detailed implementation summary of the CNN-BiLSTM model is presented in Table 8.

Table 8: CNN-BiLSTM Model Summary

Layer No.	**Layer Name**	**Description**	**Output Shape**	**Param #**
1	Input	Cochleogram shape 224x224x3	(None, 224, 224, 3)	0
2	Conv2D	Filters=16, kernel=(3, 3), padding=same	(None, 224, 224, 64)	1792
3	MaxPooling2D	Pool=(2, 2)	(None, 112, 112, 64)	0
4	BatchNorm	-	(None, 112, 112, 64)	0
5	Conv2D	Filters=32, kernel=(3,3) , padding=same	(None, 112, 112, 128)	73856
6	MaxPooling2D	Pool=(2, 2)	(None, 56, 56, 128)	0
7	BatchNorm	-	(None, 56, 56, 128)	0
8	Reshape	-	(None, 56, 7168)	0
9	BiLSTM	unit=256, kernel_initializer=he_normal	(None, 512)	15206400
10	BiLSTM	unit=256, kernel_initializer=he_normal	(None, 512)	15206400
11	Concatenate	-	(None, 1024)	0
12	Dense	Filters=1024	(None, 1251)	1282275
13	Softmax	Classes = 1251	(None, 1251)	0
Total params: 31,770,723 Trainable params: 31,770,723 Non-trainable params: 0				

3.7.3. Speaker Recognition Model using Hybrid CNN and GRU

This section presents the steps or methods followed to develop a speaker recognition model for noisy conditions using a hybrid CNN and GRU or CNN-RGU model. Similar to other models, this model employed convolutional layers for extracting short-term correlation between the features, whereas GRU was employed for extracting long-term dependency between the features. There are a total of five layers in this model (two convolutional, two GRU and one fully connected layer). Two of the convolution layers were employed consecutively at the beginning of the model to extract short-term spatial feature correlation. The GRU layers were connected consecutively after the second convolutional layer. Each convolutional layers were followed by the Maxpooling and batch normalization layers. The output shape of the maxpooling layer is not similar to the input shape of the GRU layers. The reshape layer was inserted after the second maxpooling layer to convert the output shape of the convolutional layers into the input shape of the GRU layers. Each GRU layer uses the output of the reshape layer as an input for extracting long-term correlation between the features. A concatenate layer was connected after the GRU layers to convert the output shape of each GRU model into the input shape of the fully connected layers by concatenating both GRU outputs. A fully connected layer was employed at the end of the model for prediction or classification purposes. In Figure 29, the architecture of the CNN-GRU model developed for this study was presented.

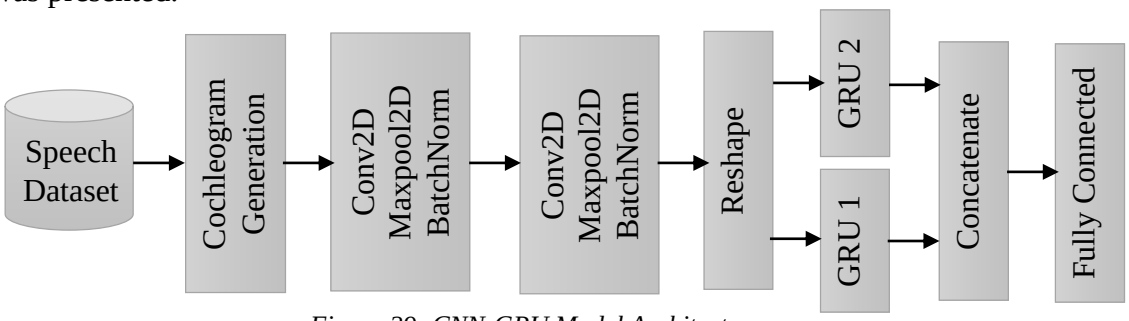

Figure 29: CNN-GRU Model Architecture

Each convolutional layers in the model have a kernel size of 3x3, the same padding, ReLu activation and normal kernel initializer. The size of filters in the first and second convolution layers were 64 and 128, respectively. To reduce the dimension of features, the maxpooling layer of pool size 2x2 and stride of 2x2 has been employed after each of the convolution layers. Batch normalization was also employed after each maximum pooling layer to normalize the features. The number of cell units in each GRU layer is 256 units. A normal kernel initializer was used in

each GRU layer. The fully connected layer had 1251 units which was equal to the number of classes or speakers in the VoxCeleb1 dataset together with the softmax activation function for conversion of fully connected input into the predicted speaker class. The categorical-crossentropy has been employed for computing the loss of the model. The model was also evaluated by using the accuracy metrics. The RMSprop optimizer has been used for model optimization purposes. The model was trained for the 50 epochs. The speaker identification and verification performance of the model were measured by using the accuracy and EER metrics respectively. The implementation details of the CNN-GRU model for speaker recognition under noisy conditions is presented in Table 9.

Table 9: CNN-GRU Model Summary

Layer No.	**Layer name**	**Output shape**	**Param #**
1	Input (Cochleogram)	(None, 224, 224, 3)	0
2	Conv2D	(None, 224, 224, 64)	1792
3	MaxPooling2D	(None, 112, 112, 64)	0
4	Conv2D	(None, 112, 112, 128)	73856
5	MaxPooling2D	(None, 56, 56, 128)	0
6	Reshape	(None, 56, 1280)	0
7	GRU	(None, 256)	5703168
8	GRU	(None, 256)	5703168
9	Concatenate	(None, 512)	0
10	Dense	(None, 1251)	641763
11	Softmax	(None, 1251)	0
Total params: 12,123,747 Trainable params: 12,123,747 Non-trainable params: 0			

3.7.4. Proposed Speaker Recognition Model

In the section 3.5.1, 3.5.2 and 3.5.3, the speaker recognition models using hybrid CNN and LSTM, hybrid CNN and BiLSTM and hybrid CNN and GRU architectures on the cochleogram input were developed to show the effectiveness of the proposed model.

This section presents the methods followed to develop the proposed speaker recognition model. The proposed model have the CNN model and BiGRU model components. The CNN component have an advantage in extracting a short-term correlation between the feature and adaptive learning from the features. The BiGRU component have the advantage in handling gradient vanishing and exploding with the limited number of gates. BiGRU also have an advantage in extracting the forward and backward long-term correlation information of features during speaker recognition. The CNN-BiGRU model or proposed model in this study comprises of five layers (two convolution layers, two BiGRU and one fully connected layer). The cochleogram with the shape of 224x224x3 which was the appropriate input shape for the input tensor was used as an input. Two of the convolution layers were employed consecutively at the beginning of the model to extract short-term spatial feature correlation and learn adaptively from the parameters.

The maxpooling output shape is not similar to the BiGRU layer's input shape. The reshape layer was inserted after the second Maxpool layer to convert the output shape of the convolutional or maxpooling layer into the input shape of the BiGRU layers. The BiGRU layers were connected after the reshape layer to extract long-term feature dependency in the forward and backward directions. Each BiGRU layers consume the output of the reshape layers. The outputs of both BiGRU layers were concatenated to feed the fully connected layer. The architecture of the proposed speaker recognition model is shown in Figure 30.

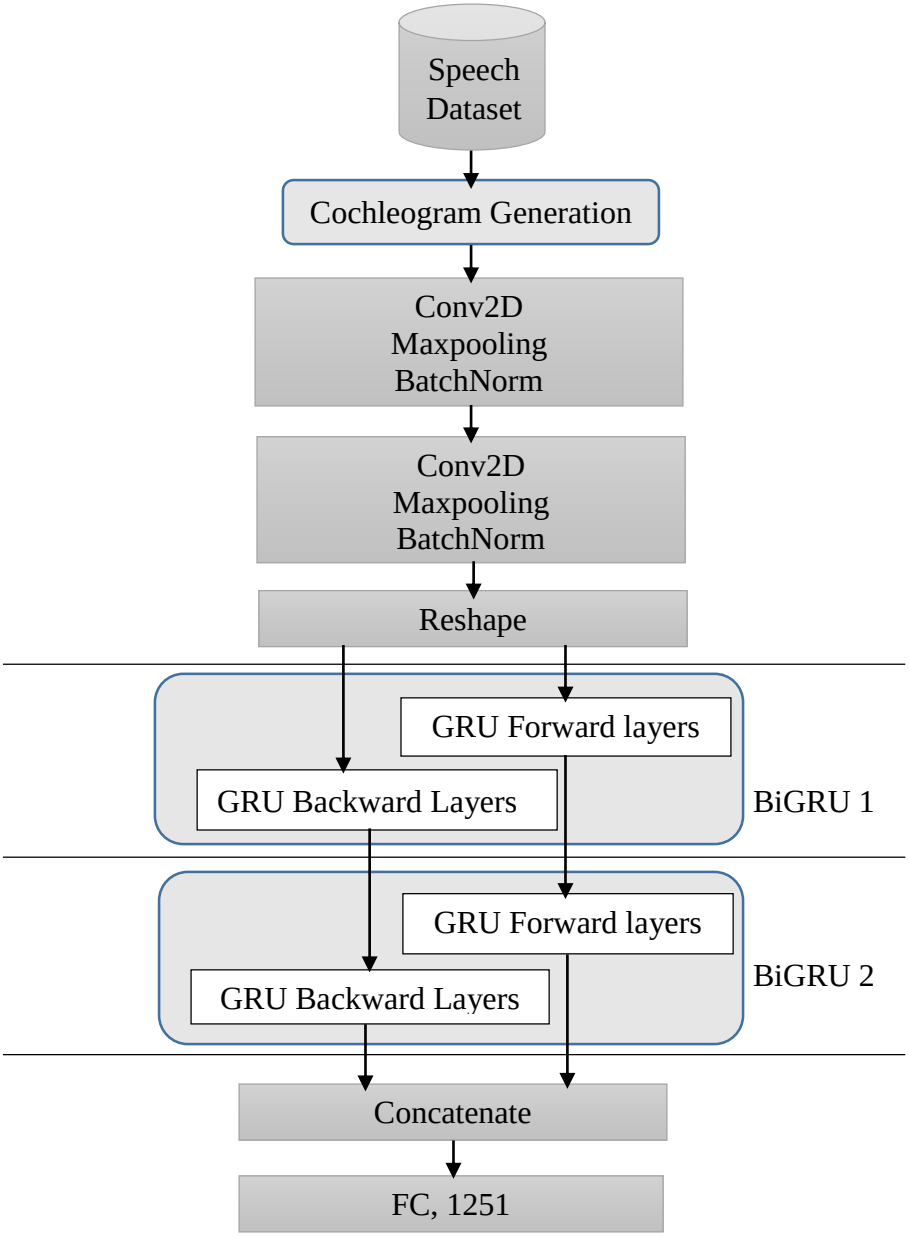

Figure 30: Proposed Speaker Recognition Model Architecture

Each convolutional layers have ReLu activation, the same padding, and kernel size of 3x3. The maxpooling layer with the pool size of 2x2 and stride of 2x2 was inserted after each of the convolutional layers to reduce the dimension of the feature maps of each convolutional layer's output. Batch normalization was also employed after each maxpooling layer to normalize the features. The first convolution layer has the filter size of 64 filters and the second convolution layer has the filter size of 128 filters. In each BiGRU layer, the 256 cell units were employed. The number of units in the fully connected layer employed for classification was equal to the number

of classes or speakers in the VoxCeleb1 dataset which was 1251. The softmax activation function was used in a fully connected layer for mapping into the output. The loss of the model was computed using categorical-crossentropy and RMSprop optimizer was also used for model optimization. The speaker identification and verification performance of the model were measured by using the metrics accuracy and EER respectively. Implementation summary of the proposed model or CNN-BiGRU model is presented in Table 10.

Table 10: CNN-BiGRU Model Summary

Layer No.	Name of Layer	Output Shape	Param #
01	Input (Cochleogram)	(None, 224, 224, 3)	0
02	Convolution	(None, 224, 224, 64)	1792
03	MaxPooling2D	(None, 112, 112, 64)	0
04	BatchNormalization	(None, 112, 112, 64)	0
05	Convolution	(None, 112, 112, 128)	73856
06	Maxpooling2D	(None, 56, 56, 128)	0
07	BatchNormalization	(None, 56, 56, 128)	0
08	Reshape	(None, 56, 7168)	0
09	Bidirectional GRU	(None, 512)	11406336
10	Bidirectional GRU	(None, 512)	11406336
11	Concatenate	(None, 1024)	0
12	Fully Connected	(None, 1251)	1282275
13	Activation	(None, 1251)	0
Total params: 24,171,363 Trainable params: 24,170,979 Non-trainable params: 384			

3.8. Implementations Detail

The implementation of this study was conducted on the NVIDIA GeForce RTX which has a GPU processor, memory size of 12GB and clock speed of 2.5 GHz. Computers with GPU processors operate faster and more effectively in processing images and large data than CPU-based computers. The management of important packages, deployment of a deep learning model and other speech analysis tasks were employed in the anaconda navigator. The JupyterLab 3.2.1 was used for writing, editing and running the Python code during implementation or prototype development. Since the experiments were conducted using deep neural networks, the TensorFlow library was used for working on different types of deep learning models. TensorFlow library was a Python language-based library recommended for deep learning model development purposes. For speech processing purposes (i.e., for cochleogram and spectrogram generation) of this study, the Spafe package (simplified Python audio features extraction) were customized. During cochleogram and spectrogram generation, important packages such as random, librosa, math, matplotlib, and Spafe were employed. Before using the Spafe package for speech processing, it was installed in the anaconda navigator through the CMD.exe module of the anaconda. Additional packages like numpy, pandas, os, pathlib, wave, scipy, pylab, keras and cv2 were employed during model training and classification. The noise robustness analysis of the cochleogram and spectrogram was conducted on the speaker identification and verification. The speaker recognition models developed for noisy conditions have employed cochleogram as an input.

CHAPTER FOUR

4. RESSULTS AND DISCUSSION

This chapter presents the results of the experiment conducted to evaluate the performance of each feature and model in speaker recognition and discusses the results in detail for further understanding. Section 4.1, presents the results of an experiment conducted for noise robustness analysis of cochleogram and spectrogram features in speaker recognition using deep learning on the VoxCeleb1 dataset at SNR of -5dB to 20dB in the interval of 5dB. Moreover, both features' noise robustness analysis results in speaker identification and verification were presented in this section. At each SNR level, the noise robustness of the cochleogram and spectrogram were evaluated in speaker identification and verification by using basic 2DCNN, VGG-16, ResNet50, ECAPA-TDNN and TitaNet models. In section 4.1.1, noise robustness analysis results of cochleogram and spectrogram features in speaker identification were presented. Section 4.1.2, presents the noise robustness analysis results of cochleogram and spectrogram in speaker verification. In section 4.2, the results of the speaker recognition models developed for the noisy conditions were presented. In this section, the results of the models such as CNN-LSTM, CNN-BiLSTM, CNN-GRU and CNN-BiGRU in speaker recognition on the real-world noise and white Gaussian noise added datasets at the SNR of -5dB to 20dB were presented. Section 4.3, presents the comparison of the proposed model performance with the existing works.

4.1. Noise Robustness Analysis Results of Cochleogram and Spectrogram

4.1.1. Analysis Results of Cochleogram and Spectrogram in Speaker Identification

This section presents the evaluation/analysis results of the cochleogram and spectrogram features in speaker identification by using basic 2DCNN, VGG-16, ResNet50, ECAPA-TDNN and TitaNet models on the VoxCeleb1 dataset with real world-noise at the SNR of -5dB to 20dB. The results were presented using the figures, tables and text discussion. For illustrating the analysis results of both cochleogram and spectrogram features in speaker identification at each epoch during training graphically, the researchers have selected a single model (i.e., VGG-16) and its validation accuracy for the specified SNR levels. Each feature's accuracy and loss were presented graphically to illustrate the model's performance and loss at different levels of SNR.

Figure 31, shows the accuracy of the cochleogram and spectrogram features in speaker identification at a SNR of -5dB (i.e., at very high noise) using the VGG-16 model. In this figure, cochleogram has shown superior accuracy than the spectrogram at each epoch. The maximum accuracy of the spectrogram at this SNR level was 51.96%. Cochleogram has achieved a maximum accuracy of 75.77% at the SNR=-5dB. At SNR =-5dB (i.e., at very high noise) cochleogram has shown an improvement of 23.81% over the spectrogram in speaker recognition using VGG-16. Cochleogram features also converge faster than spectrogram during model training. This indicates that cochleogram has better performance than spectrogram features at a very high noise ratio in the speech.

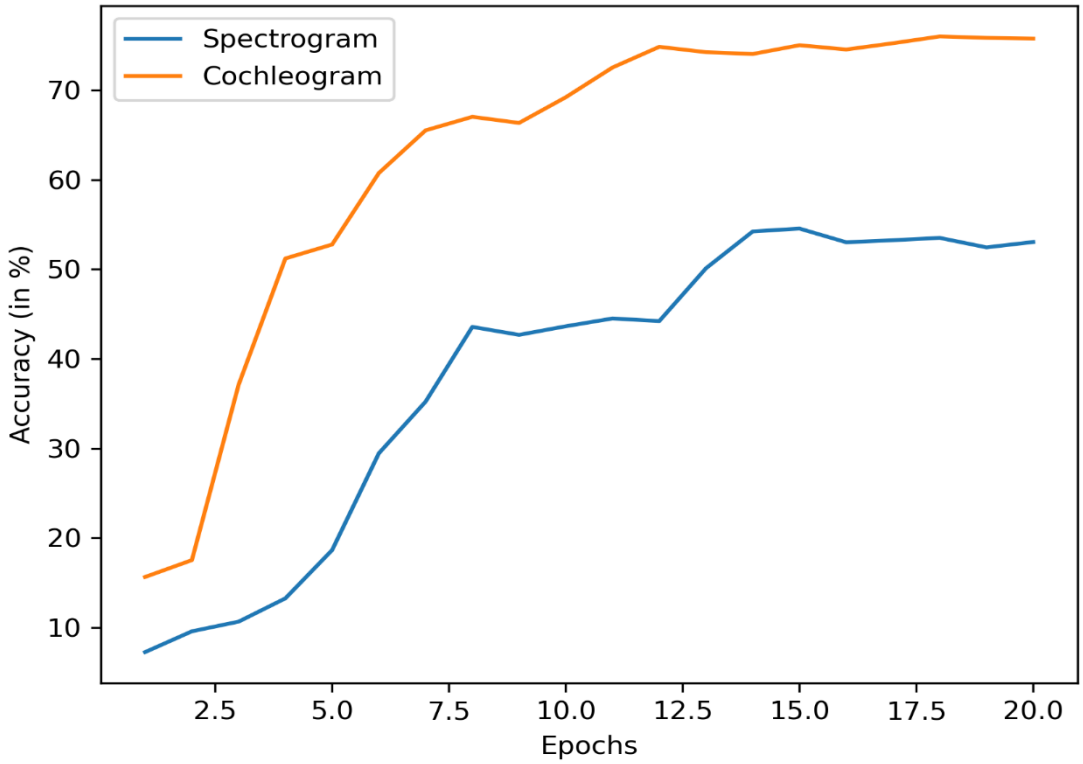

Figure 31: Accuracy of Cochleogram and Spectrogram in speaker identification using VGG-16 at SNR=-5dB

Figure 32, presents the loss of cochleogram and spectrogram features in speaker identification at very high noise (i.e., at SNR=-5dB) using VGG-16. Cochleogram has lower loss than spectrogram at each of the training epochs. The minimum loss of the cochleogram at this SNR level was 6.15 which was smaller than the loss of the spectrogram which have a loss of 7.27. This indicates that cochleogram performs better than spectrogram features at SNR of -5dB.

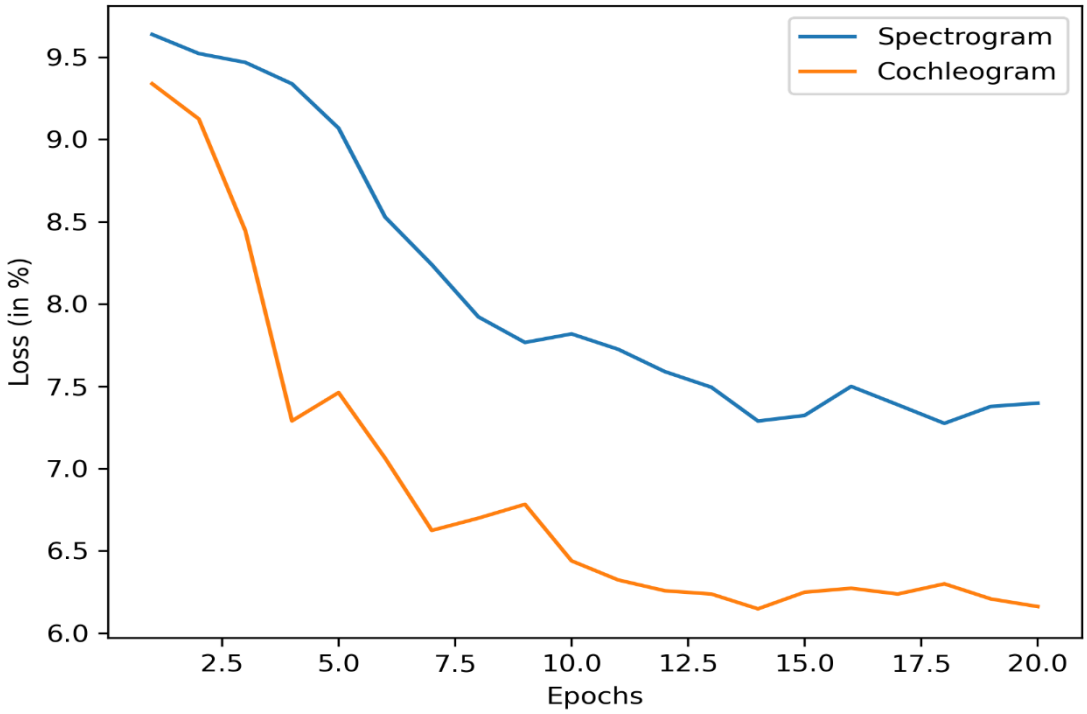

Figure 32: Loss of Cochleogram and Spectrogram in speaker identification at SNR=-5dB using VGG-16

Figure 33, presents the accuracy of the cochleogram and spectrogram in speaker identification during analysis at SNR=0dB level by using VGG-16. In this figure, cochleogram has shown better accuracy than the spectrogram at each epoch of the training. The maximum accuracy of the spectrogram at SNR=0dB was 70.82%, whereas the cochleogram achieved a maximum accuracy of 89.38%. In this SNR level, cochleogram has shown an improvement of 18.56% on the spectrogram features. The results also confirmed that cochleogram was better than the spectrogram at the equal noise and speech ratio.

In Figure 34, the loss of cochleogram and spectrogram feature in a speaker identification at high noise (i.e., at SNR=0dB) using VGG-16 was shown. Cochleogram have achieved the minimum loss compared with the spectrogram at each epoch of this SNR level during training. The least loss of the cochleogram at this SNR level was 5.5, whereas the least loss of the spectrogram was 6.3. This confirmed that cochleogram has a lower loss and higher performance than spectrogram features in speaker identification at the SNR of 0dB level.

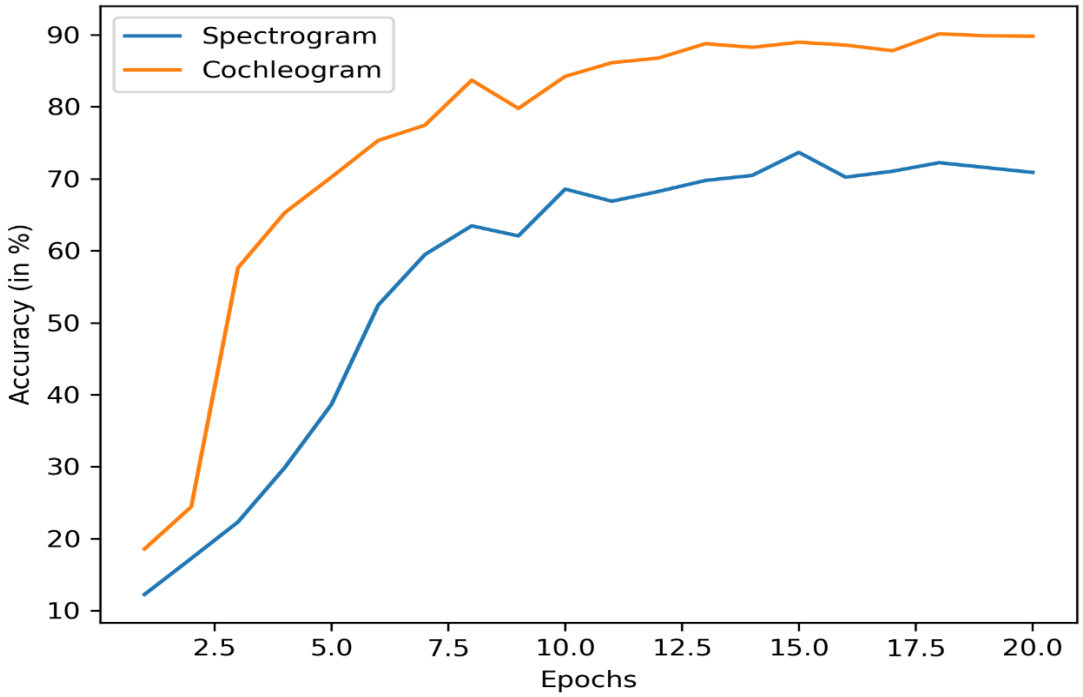

Figure 33: Accuracy of Cochleogram and Spectrogram in speaker identification using VGG-16 at SNR=0dB

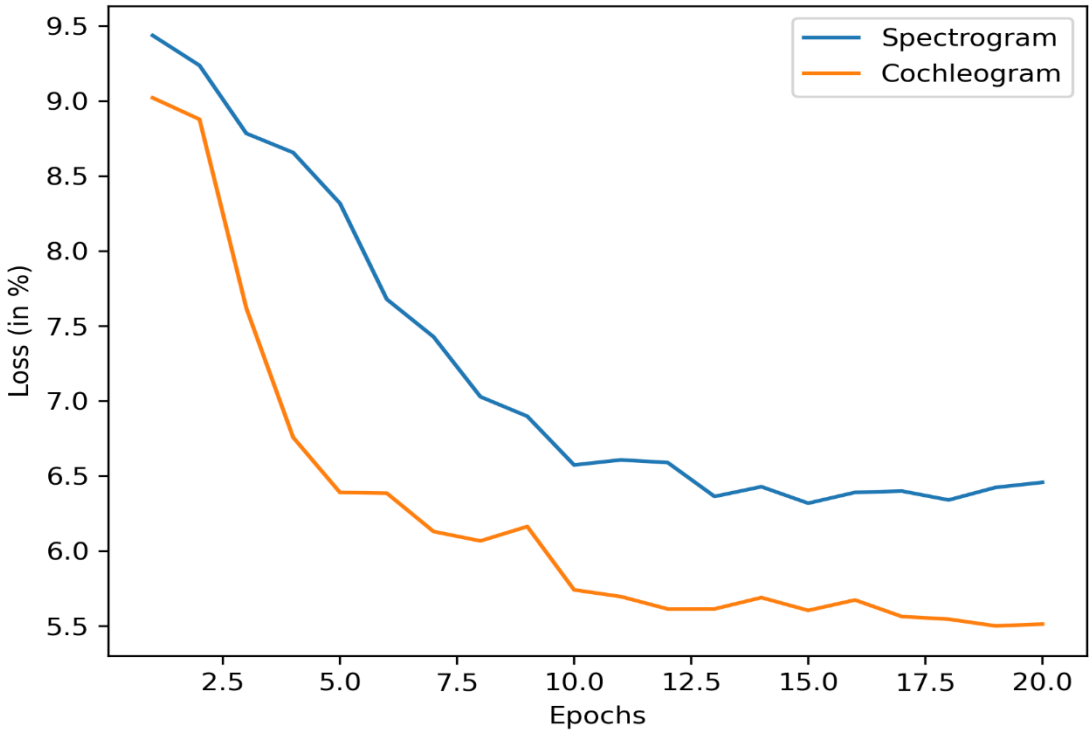

Figure 34: Loss of Cochleogram and Spectrogram in speaker identification at SNR=0dB using VGG-16

In Figure 35, the accuracy of the cochleogram and spectrogram features in speaker identification at a medium loss (i.e., at SNR=5dB) using VGG-16 was presented. The results in this figure

indicate that cochleogram features outperform spectrogram in speaker identification at SNR=5dB. At SNR level of 5dB the maximum accuracy of the spectrogram was 85.30% which was smaller than the maximum accuracy of the cochleogram features which was 93.94%. At SNR level 5dB cochleogram features have shown an improvement of 8.64% on the spectrogram. The results at each epoch of the figure also confirmed that cochleogram have higher performance than spectrogram features at a medium noise ratio in the speech. The model with the cochleogram feature input converged faster than spectrogram features during the training.

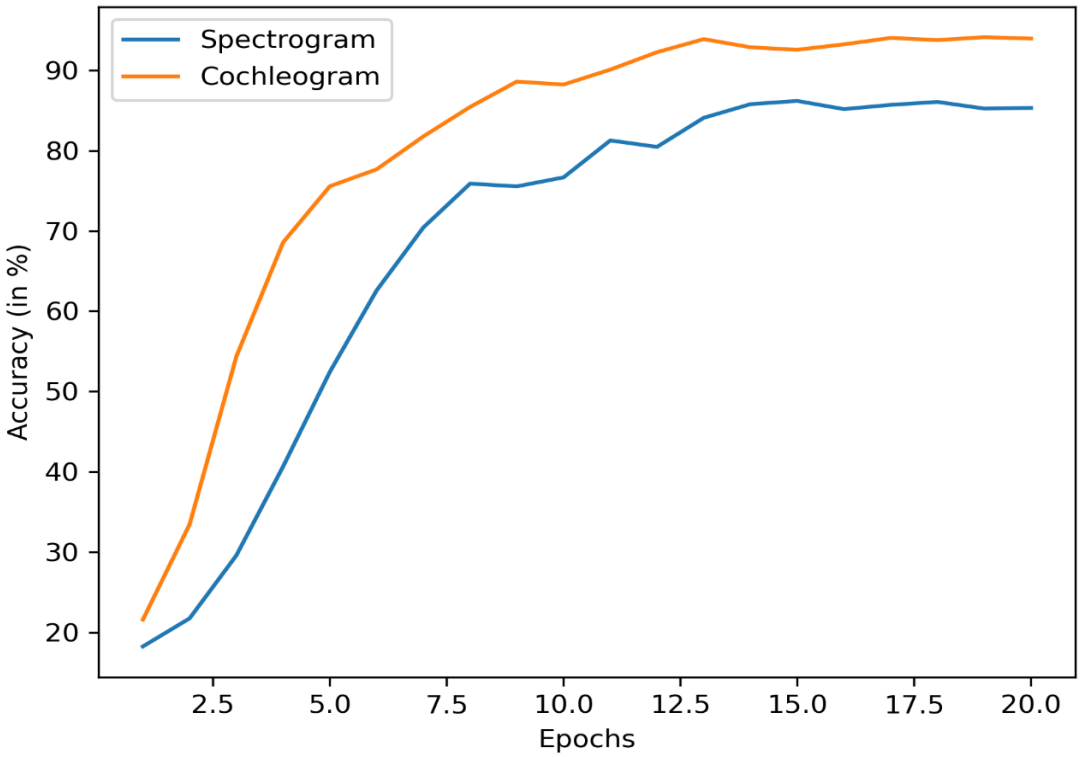

Figure 35: Accuracy of Cochleogram and Spectrogram in speaker identification using VGG-16 at SNR=5dB

Figure 36, shows the loss of the cochleogram and spectrogram features in speaker identification at medium noise ratio (i.e., at SNR=5dB) using VGG-16. Cochleogram features have minimum loss than spectrogram features at SNR of 5dB which shows that cochleogram have better performance than spectrogram. The minimum loss of the cochleogram at this SNR level was 5.288 which was smaller than the loss of the spectrogram that was 5.7. The model confirmed that cochleogram feature has better performance than the spectrogram at the medium noise in the speech.

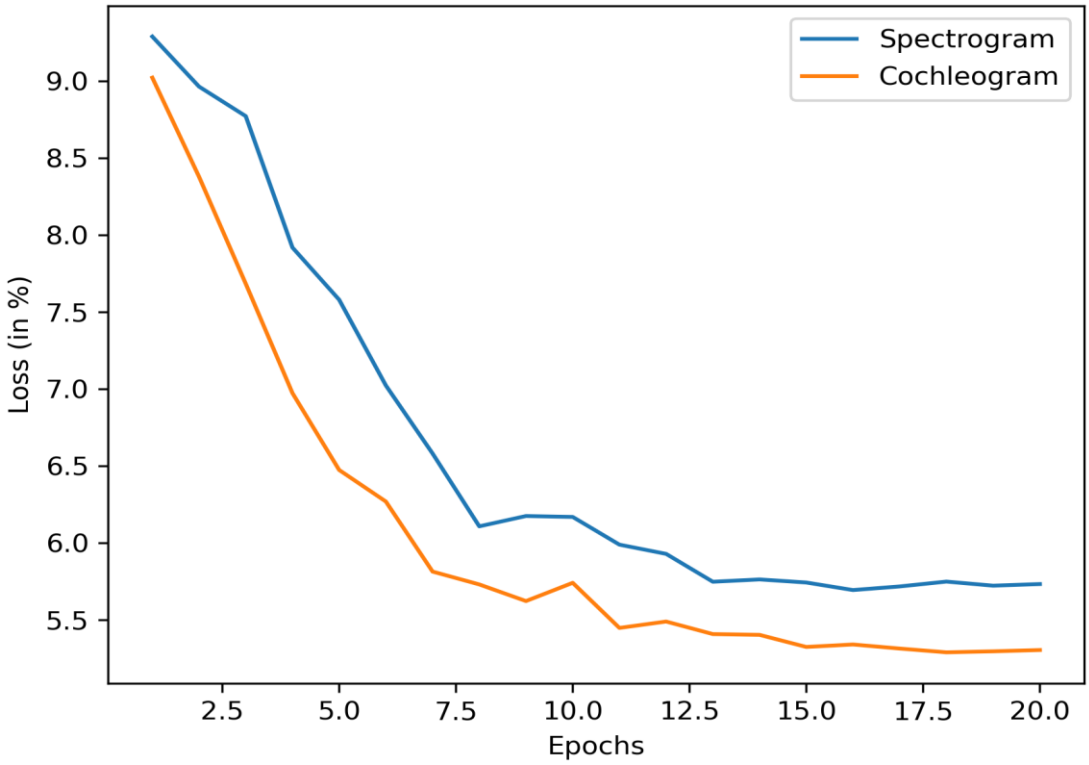

Figure 36: Loss of Cochleogram and Spectrogram in speaker identification at SNR=5dB using VGG-16

The noise robustness analysis results of the cochleogram and spectrogram features in speaker identification at SNR of 10dB using VGG-16 is presented in Figure 37. Cochleogram outperformed spectrogram features at SNR =10dB and showed faster convergence during training of the model. The maximum accuracy of the spectrogram feature at this SNR level was 91.64% and the maximum accuracy of the cochleogram was 95.96%. At this SNR level, cochleogram has shown an improvement of 4.32% which was smaller than the previous improvements. At a low noise ratio level, the speech cochleogram shows smaller improvements over the spectrogram features. Still, cochleogram features have good accuracy at the SNR level 10dB.

Figure 38, presents the loss of cochleogram and spectrogram features in speaker identification at medium noise (i.e., at SNR=10dB) using VGG-16. The results in the figure show that cochleogram have better performance or lower loss at each epoch during training. The minimum loss of the cochleogram at SNR=10dB was 5.18, whereas the minimum loss of the spectrogram was 5.4. The loss at each epoch indicates that cochleogram was better than the spectrogram for speaker identification at medium noises (i.e., at SNR=10dB).

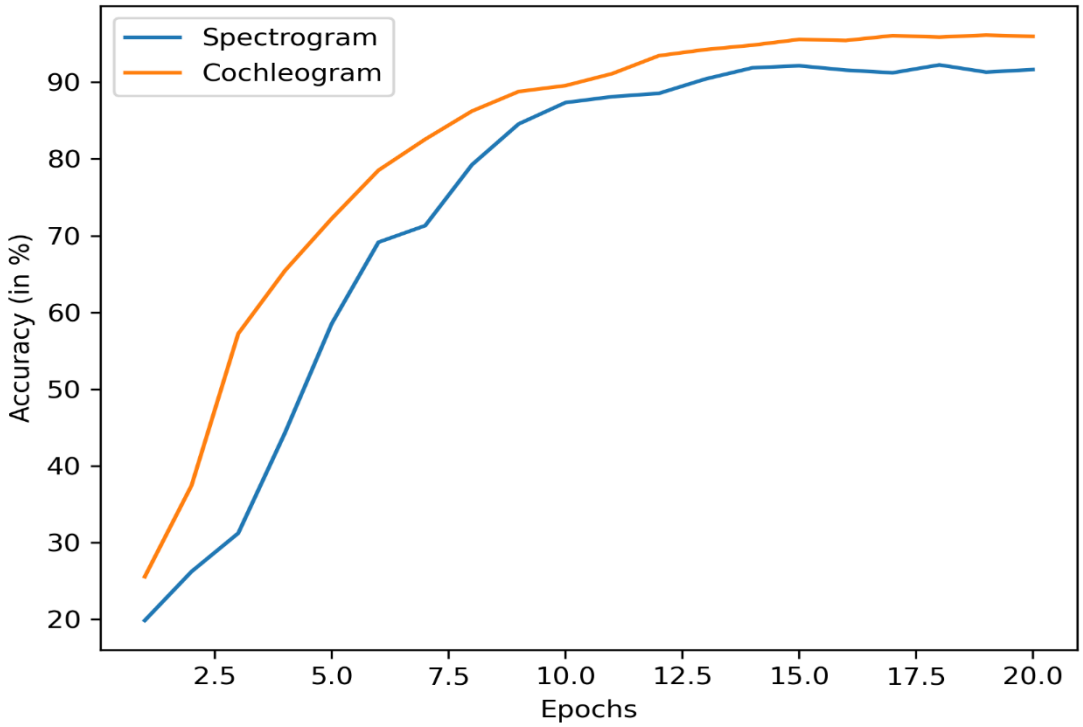

Figure 37: Accuracy of Cochleogram and Spectrogram in speaker identification using VGG-16 at SNR=10dB

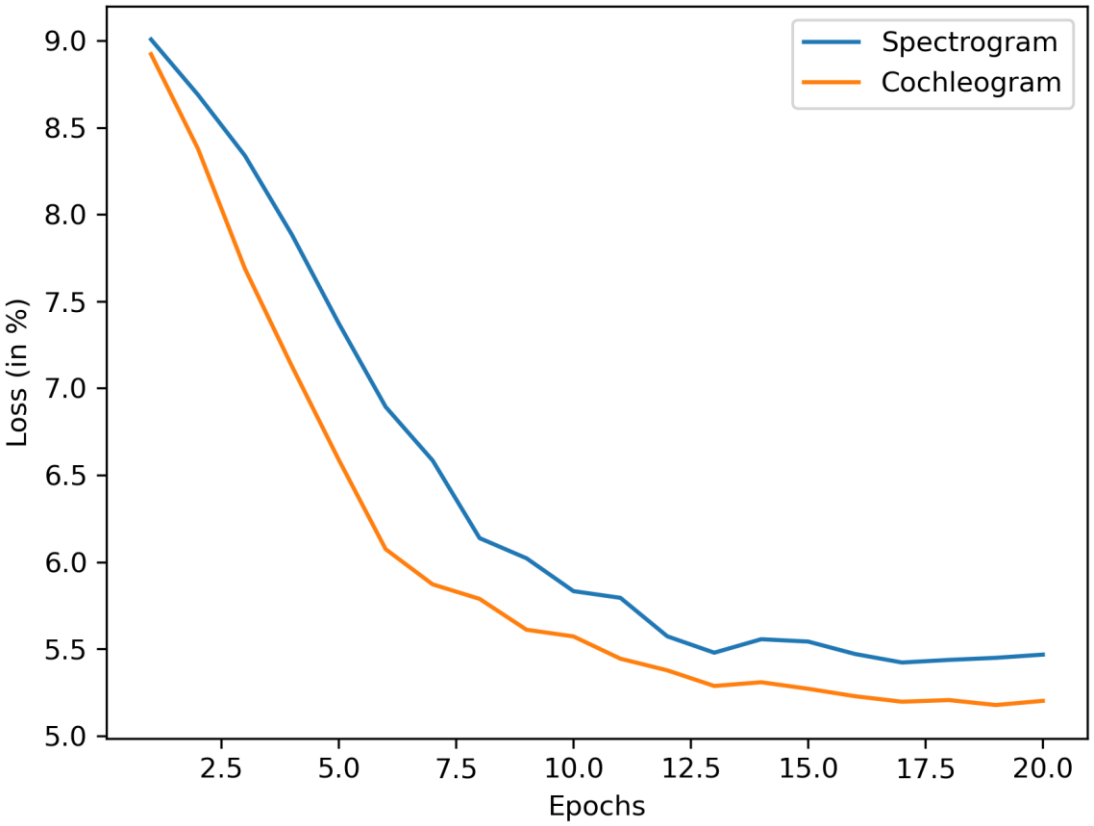

Figure 38: Loss of Cochleogram and Spectrogram in speaker identification at SNR=10dB using VGG-16

Figure 39, presents the noise robustness analysis results of cochleogram and spectrogram features in speaker identification at SNR of 15dB using VGG-16. This figure shows that cochleogram have better accuracy than spectrogram features at the SNR equal to 15dB. The maximum accuracy of the spectrogram was 92.81% which was lower than the maximum accuracy of the cochleogram which was 96.69%. The cochleogram has shown an improvement of 3.95% over the spectrogram features. We can observe that cochleogram features also converge slightly faster than spectrograms.

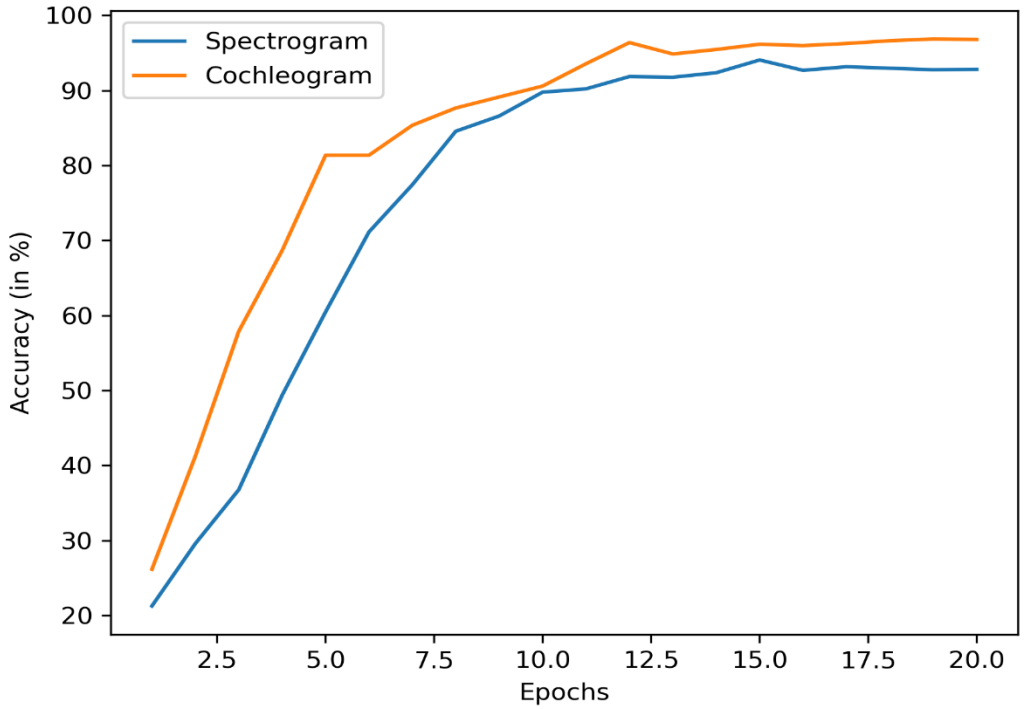

Figure 39: Accuracy of Cochleogram and Spectrogram in speaker identification using VGG-16 at SNR=15dB

In Figure 40, the loss of the cochleogram and spectrogram features in speaker identification at low noise (i.e., at SNR=15dB) using VGG-16 was presented. The results confirmed that cochleogram have a minimum loss or better performance than spectrogram features at this SNR level. The minimum loss of the cochleogram at this SNR level was 5.14, whereas the minimum loss of the spectrogram features was 5.4. Also, both cochleogram and spectrogram achieved minimum loss at this SNR level than previous SNR levels or higher noises.

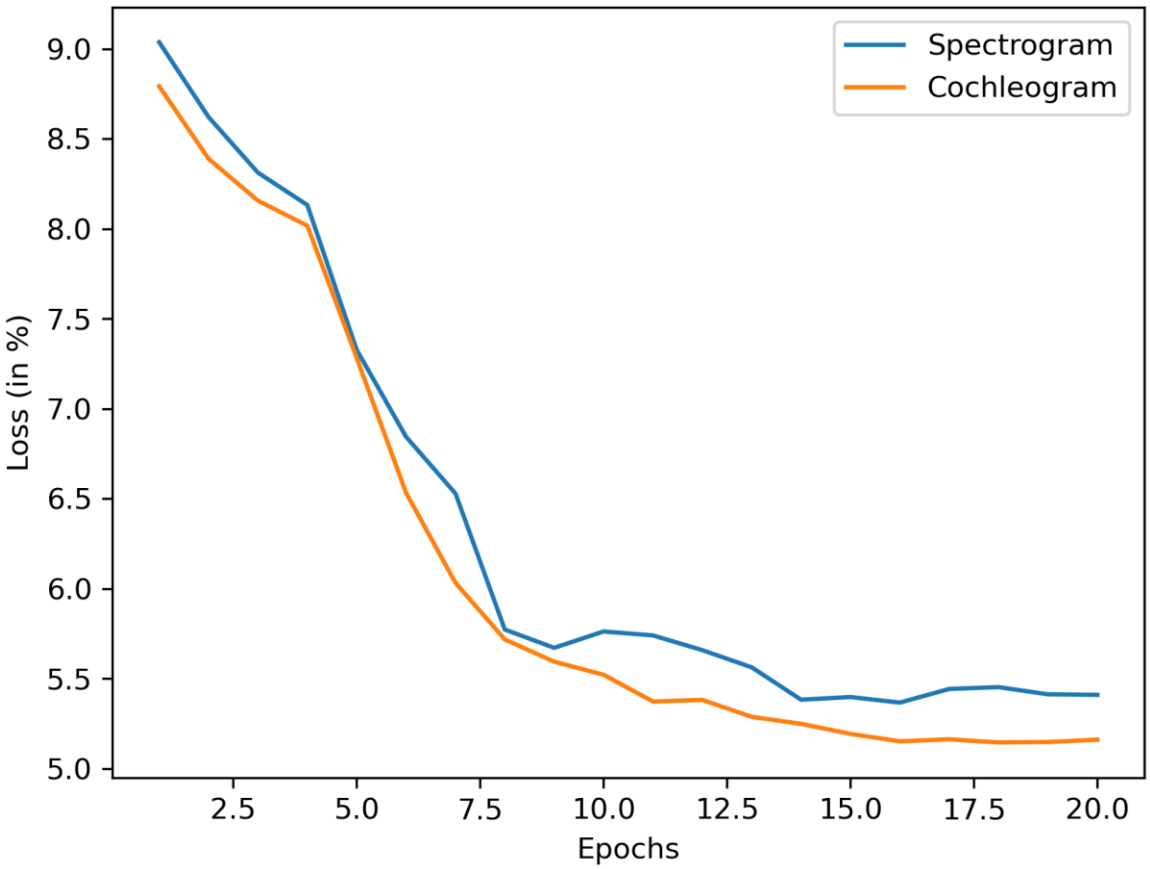

Figure 40: Loss of Cochleogram and Spectrogram in speaker identification at SNR=15dB using VGG-16

Figure 41, presents the evaluation results of the cochleogram and spectrogram features for noise robustness in speaker identification at the SNR of 20dB using VGG-16. In this figure, the maximum accuracy of the spectrogram was 95.77% which was better than its accuracies at the higher noise ratios discussed before. The maximum accuracy of the cochleogram in this figure was 97.32% which was better than its accuracies at the higher noise ratios. Cochleogram outperformed spectrogram at SNR =20dB. In the Figure the maximum accuracy of Cochleogram has shown slight improvement which was equal to 1.55% over the maximum accuracy of the spectrogram.

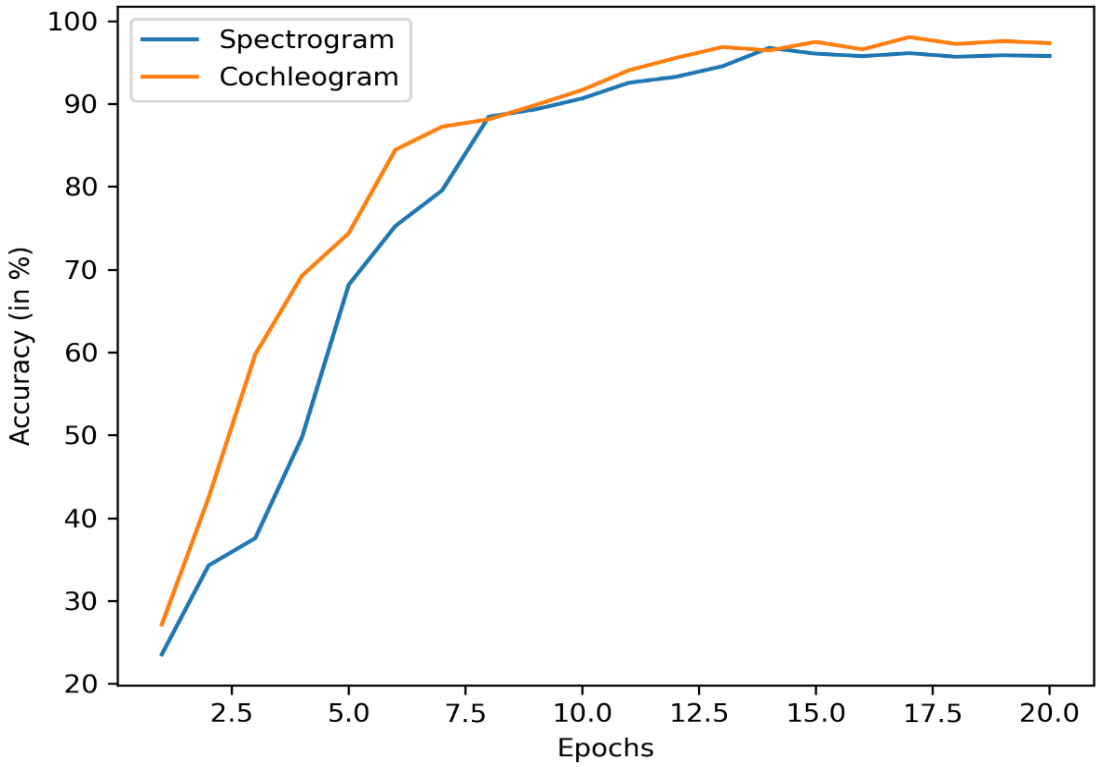

Figure 41: Accuracy of Cochleogram and Spectrogram in speaker identification using VGG-16 at SNR=20dB

In Figure 42, the loss of the cochleogram and spectrogram features in speaker identification at a lower loss (i.e., at SNR=20dB) using VGG-16 was presented. At this SNR level, the loss of the model with cochleogram feature was lower than the loss of the model with the spectrogram features. The minimum loss of cochleogram at this SNR level was 5.12 which was lower than the loss of spectrogram which was 5.166.

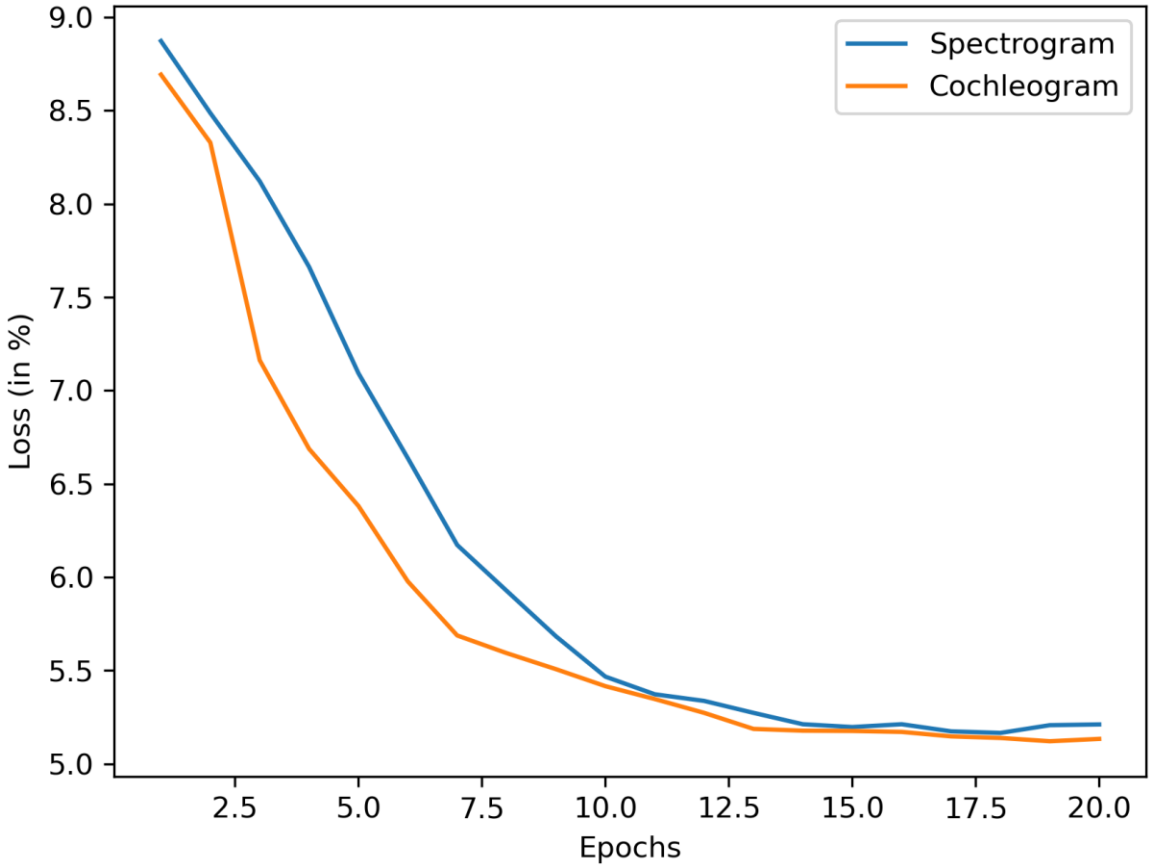

Figure 42: Loss of Cochleogram and Spectrogram in speaker identification at SNR=20dB using VGG-16

Figure 43, presents the accuracy of the cochleogram and spectrogram features in speaker recognition on the dataset without additive noise using VGG-16. Although the dataset does not have additive noise it contains various types of real-world noises at various noise ratios. Cochleogram has shown relatively better accuracy than spectrogram features because the dataset has various types of noises during collection. On the dataset without the additive noise, the maximum accuracy of the spectrogram was 96.93%. The maximum accuracy of the cochleogram on the dataset without additive noise was 98.04% which was better than the spectrogram. At this SNR level, cochleogram has shown a maximum improvement of 1.11% over the spectrogram. In this figure, the accuracy difference of both features was minimal because of a small noise ratio or no additive noise in the speech.

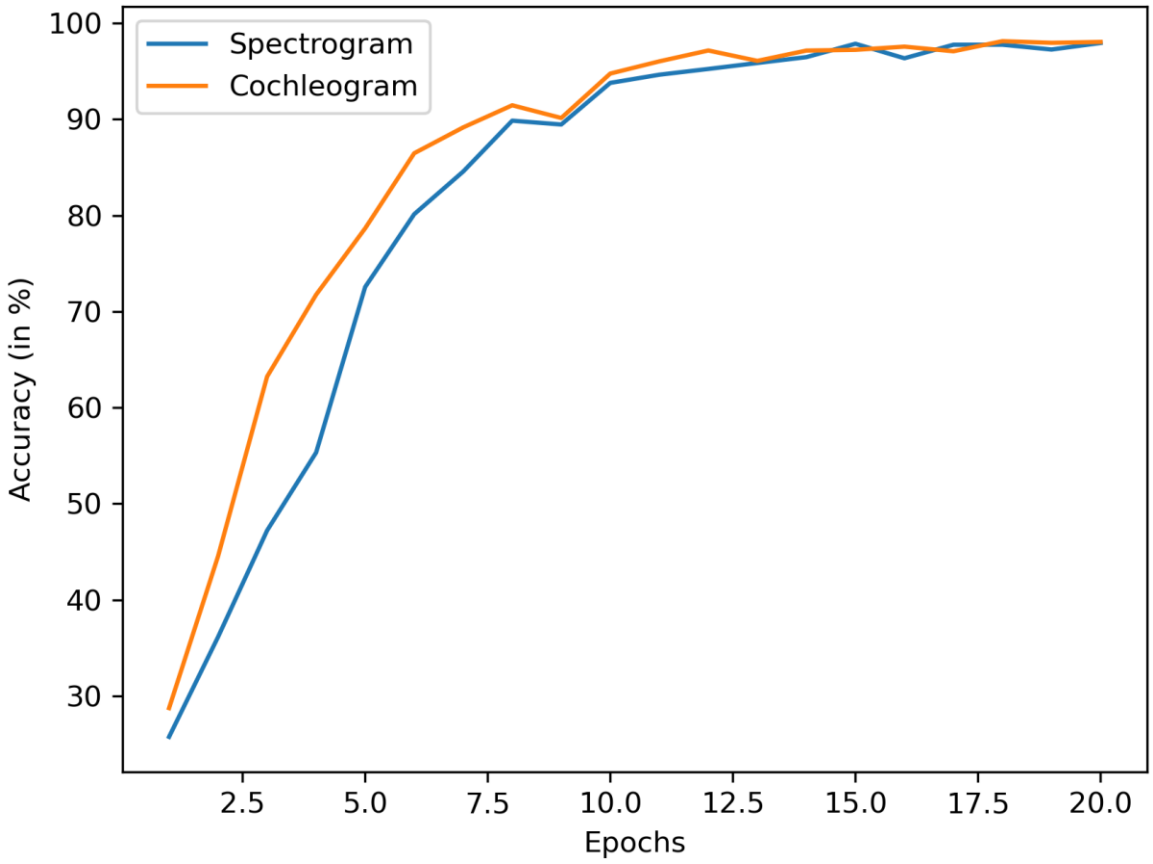

Figure 43: Accuracy of Cochleogram and Spectrogram in Speaker Identification without additive noise using VGG-16

In Figure 44, the loss of the cochleogram and spectrogram features in speaker identification on the dataset without additive noise using the VGG-16 model was presented. The model with the cochleogram input has a smaller loss than the spectrogram input. The minimum loss of the cochleogram was 5.12, whereas the minimum loss of the spectrogram was 5.133. The loss results at each epoch indicate that cochleogram has better performance or minimum loss than the spectrogram on the VoxCeleb1 dataset without additive noise.

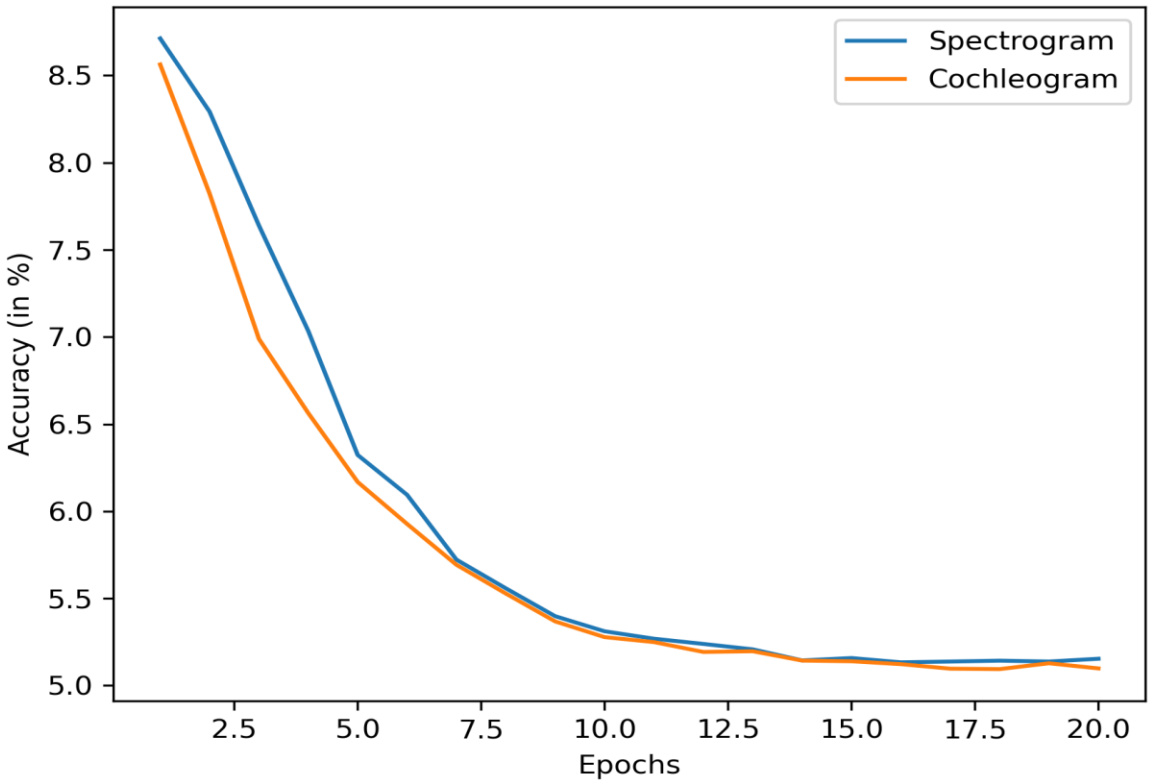

Figure 44: Loss of Cochleogram and Spectrogram in speaker identification without additive noise using VGG-16

Generally, the analysis results of cochleogram and spectrogram features in speaker recognition using VGG-16 confirmed that cochleogram has better performance than spectrogram. At each of the SNR levels and without additive noise cochleogram has shown higher accuracy and lower loss than the spectrogram. In speaker identification at very high noise in the speech, cochleogram showed higher improvement over the spectrogram. The accuracy and loss difference/gap between the cochleogram and spectrogram features were reduced when the noise ratio in the speech was reduced. The results of accuracy and loss discussed above in each figure confirmed that cochleogram input was better than spectrogram in speaker identification under noisy conditions.

Table 11, presents the accuracy of the cochleogram and spectrogram features in speaker identification based on the SNR levels and types of models employed for the evaluation. The results were presented at SNR levels between -5dB to 20dB in intervals of 5dB and based on the models such as basic 2DCNN, VGG-16, ResNet50, ECAPA-TDNN and TitaNet. At SNR=-5dB (i.e., at very high noise in the speech), the minimum and maximum accuracy of the spectrogram in speaker identification was 46.78% and 55.25% respectively. The minimum and maximum accuracies of the cochleogram at the SNR=-5dB were 73.56% and 78.37% respectively. The

results of all models at SNR=-5dB indicated that cochleogram input has better performance than spectrogram in speaker identification. The basic 2DCNN model achieved the lowest accuracy, whereas TitaNet have shown the maximum accuracy at the SNR=-5dB. Cochleogram has shown an improvement of 22.44% up to 26.78% over the spectrogram at SNR=-5dB. At SNR = 0dB the minimum and maximum accuracy of the spectrogram were 67.78% and 73.03% respectively, whereas the minimum and maximum accuracy of the cochleogram were 87.16% and 89.97. The basic 2DCNN model achieved the least accuracy and the TitaNet model achieved the highest accuracy at SNR =0dB on both spectrogram and cochleogram features. Cochleogram input outperformed spectrogram features in speaker identification at the SNR=0dB on all the models used for the evaluation. Cochleogram has achieved an improvement of 16.94% up to 19.38% over the spectrogram in speaker identification at SNR=0dB. At medium additive noise in the speech (i.e., SNR=5dB) the minimum and maximum accuracy of the spectrogram were 81.07% and 87.61% respectively, whereas the minimum and maximum accuracy of the cochleogram were 91.98% and 94.51% respectively. At this SNR level, cochleogram has also shown superior performance than the spectrogram at all the models employed for the evaluation. Cochleogram has shown an improvement of 6.90% up to 10.97% over the spectrogram at SNR=5dB. Similarly, the basic 2DCNN model has the lowest accuracy and the TitaNet model has the highest accuracy at this SNR level on both cochleogram and the spectrogram features. At SNR=10dB the minimum and maximum speaker identification accuracy of the spectrogram was 87.74% and 93.15% respectively, whereas cochleogram achieved the minimum and maximum accuracy of 93.66% and 96.55% respectively. The evaluation results of cochleogram and spectrogram using the specified models confirmed that cochleogram input has a higher performance than the spectrogram in speaker identification at SNR=10dB. On the dataset with a small additive noise ratio on the speech (i.e., at SNR=15dB) the speaker identification accuracy of the cochleogram was better than the spectrogram features. Cochleogram achieved the minimum and maximum accuracy of 94.65% and 97.34% respectively, whereas the spectrogram achieved 89.44% and 94.19% respectively at SNR=15dB. Basic 2DCNN has the smallest accuracy and TitaNet has the highest accuracy at this SNR level. Using cochleogram input an improvement of 3.15% up to 5.21% was achieved over the spectrogram at SNR=15dB. At SNR=20dB and on the dataset without additive noise cochleogram has shown higher accuracy than spectrogram input in all the models used for the evaluation. The minimum and maximum accuracy of the spectrogram at SNR=20dB was 92.16%

and 97.17% respectively, whereas the minimum and maximum accuracy of the cochleogram were 95.47% and 97.81% respectively. Basic 2DCNN have the least accuracy and TitaNet have the highest accuracy on both spectrogram and cochleogram at SNR=20dB. At this SNR level, cochleogram has shown an improvement of 0.64% up to 3.31%. On the dataset without additive noise, cochleogram has also shown relatively better performance than spectrogram features. The minimum and maximum accuracies of the cochleogram were 95.62% and 98.04% respectively, whereas the minimum and maximum accuracy of the spectrogram were 93.61% and 97.55% respectively. Cochleogram has achieved an improvement of 0.49% up to 2.01% over the spectrogram on the dataset without additive noise. In general, cochleogram has shown higher performance than the spectrogram in speaker recognition at all the specified SNR levels and without additive noise on the VoxCeleb1 dataset. All the models result at each of the SNR levels also confirmed that cochleogram has better performance than the spectrogram in speaker identification. At very high noise ratios in the speech (i.e., at SNR=-5dB, 0dB and 5dB) cochleogram has shown higher improvement over the spectrogram. The accuracy or performance difference between the cochleogram and spectrogram was reduced as the noise ratios in the speech get reduced.

Table 11: Analysis results of cochleogram and spectrogram in speaker identification with and without additive noises

Model Type	**Feature Type**	**Accuracy (%) With Additive Noises at SNR**						**Without Added Noise**
		−5dB	**0dB**	**5dB**	**10dB**	**15dB**	**20dB**	
Basic 2DCNN	Spectrogram	46.78	67.78	81.07	87.74	89.44	92.16	93.61
	Cochleogram	73.56	87.16	91.98	93.66	94.65	95.47	95.62
ResNet-50	Spectrogram	48.89	69.91	83.22	89.91	91.63	94.37	96.57
	Cochleogram	74.14	87.88	92.87	94.63	95.63	96.22	97.85
VGG-16	Spectrogram	51.96	70.82	85.30	91.64	92.81	95.77	96.83
	Cochleogram	75.77	89.38	93.94	95.96	96.79	97.32	98.02
ECAPA-TDNN	Spectrogram	53.98	72.25	86.94	92.54	93.67	96.59	96.72
	Cochleogram	76.42	89.75	94.25	96.39	97.15	97.61	97.89
TitaNet	Spectrogram	55.25	73.03	87.61	93.15	94.19	97.17	97.55
	Cochleogram	78.37	89.97	94.51	96.55	97.34	97.81	98.04

4.1.2. Analysis Results of Cochleogram and Spectrogram in Speaker Verification

This section presents the analysis results of cochleogram and spectrogram features in speaker verification on the VoxCeleb1 dataset with real world noise at SNR level between -5dB and 20dB in the interval of 5dB and without additive noises. The evaluations were conducted using the models such as basic 2DCNN, VGG-16, ResNet50, ECAPA-TDNN and TitaNet models. The performance was measured using the equal error rate (EER), in which the minimum EER value model or feature was considered as the better model.

Table 12, presents the EER of the cochleogram and spectrogram features in speaker verification based on the SNR level and model type. At very high noise in the speech (i.e., at SNR=-5dB), the cochleogram features have superior performance (i.e., lowest EER) than spectrogram. The minimum and maximum EER of the spectrogram at SNR=-5dB was 22.97% and 11.36% respectively. Cochleogram have a minimum and maximum EER of 17.83% and 10.82% at SNR =-5dB respectively. In all the models employed for evaluation cochleogram achieved the least EER than spectrogram. Basic 2DCNN model have the highest EER which was 22.97%, whereas TitaNet model have the lowest EER which was 10.82% from all the models. At SNR=-5dB, cochleogram have achieved an improvement of EER from 0.54% to 5.14% over the spectrogram.

At SNR =0dB, cochleogram feature have the lowest EER (highest performance) than spectrogram. The highest and lowest EER of the spectrogram features at SNR=0dB were 18.18% and 8.72% respectively. The maximum and minimum EER of the cochleogram features at SNR=0dB were 14.59% and 7.64% respectively. Both cochleogram and spectrogram features at SNR=0dB have shown highest EER (least performance) using basic 2DCNN, whereas the lowest EER (highest accuracy) were achieved using TitaNet model. At this SNR level, cochleogram have achieved an improvement of 1.04% to 3.59% over the spectrogram features.

The cochleogram feature have the lowest EER (highest performance) than spectrogram at medium noise ratio in the speech (i.e., SNR=5dB). The highest and lowest EER of the spectrogram at this SNR level were 13.37% and 4.92% respectively. Cochleogram feature have shown the maximum and minimum EER of 10.82% and 3.83% respectively at SNR=5dB. Using basic 2DCNN model both features have the lowest performance (highest EER) than in the rest models. Both features

using TitaNet model have least EER (highest performance) than in the models. At this SNR level, cochleogram have achieved an improvement of the 1.14% to 2.82% over the spectrogram.

At the medium noise ratio in the speech (at SNR=10dB), cochleogram have also achieved the maximum performance (least EER) than spectrogram. The highest and lowest EER of the cochleogram at this SNR level was 8.72% and 2.53% respectively which was smaller than the highest and lowest EER of the spectrogram which were 10.45% and 2.70% respectively. Basic 2DCNN model have shown the highest EER or lowest performance on both cochleogram and spectrogram at SNR=10dB. The TitaNet model have achieved the highest performance or least EER at SNR=10dB for both cochleogram and spectrogram. Cochleogram features outperformed spectrogram using all the evaluation models at this SNR level. It has shown an improvement of 0.17% to 1.83 over the spectrogram at this SNR level.

At SNR=15dB, cochleogram have achieved highest performance or the least EER than spectrogram. At this SNR level the highest and lowest EER of the cochleogram were 7.19% and 1.24% respectively. The maximum and minimum EER of the spectrogram features at this SNR level was 9.12 % and 1.30%. Basic 2DCNN have shown the least performance and TitaNet model have achieved the highest performance at this SNR level for both cochleogram and spectrogram features from the rest of the models. In this SNR level cochleogram have shown an improvement of 0.10% to 1.93 respectively.

At the small noise ratio in the speech (at SNR=20dB), cochleogram also achieved better performance (lower EER) than spectrogram. At this SNR level the maximum and minimum EER of the cochleogram were 6.54% and 0.62% respectively. The highest and lowest EER of the spectrogram at SNR=20dB were 8.46 % and 0.84% respectively. The EER results in all the model indicates that cochleogram have better performance than spectrogram at this SNR level. Cochleogram have shown an improvement of 0.21% to 1.92% over the spectrogram at SNR=20dB

On the dataset without additive noise, cochleogram have also achieved better performance than spectrogram because the dataset by itself contains various types of environmental noises at various SNR levels. The lowest and highest EER achieved on the dataset without additive noise using cochleogram were 0.54% and 6.47%. The minimum and maximum EER of the spectrogram in this type of dataset were 0.74% and 8.11% respectively. In this dataset the basic 2DCNN model have

shown the least performance (highest EER) on both cochleogram and spectrogram inputs. The TitaNet model have shown better performance (least EER) than other models on both cochleogram and spectrogram input.

Table 12: Analysis results of cochleogram and spectrogram in speaker verification on dataset with and without additive noises

Model Type	Feature Type	EER (%) at SNR						Without Added Noise
		−5dB	0dB	5dB	10dB	15dB	20dB	
Basic 2DCNN	Spectrogram	22.97	18.18	13.37	10.45	9.12	8.46	8.11
	Cochleogram	17.83	14.59	10.82	8.72	7.19	6.54	6.47
ResNet-50	Spectrogram	19.12	16.05	12.23	10.04	8.70	6.15	5.64
	Cochleogram	16.06	13.23	9.41	8.21	7.92	5.41	4.28
VGG-16	Spectrogram	18.83	15.71	11.92	9.74	8.37	5.86	5.28
	Cochleogram	15.42	12.86	9.22	7.95	6.61	4.55	4.16
ECAPA-TDNN	Spectrogram	13.83	10.72	6.93	3.75	2.06	1.16	0.91
	Cochleogram	11.15	9.68	5.84	2.61	1.30	0.64	0.61
TitaNet	Spectrogram	11.36	8.72	4.92	2.70	1.34	0.83	0.75
	Cochleogram	10.82	7.64	3.78	2.53	1.24	0.62	0.54

Generally, the results in the Tables 11 and 12 show that cochleogram features achieved superior performance than spectrogram in both speaker identification and verification at each of the SNR levels. In the improved deep learning architectures (i.e., in TitaNet model), cochleogram have shown highest performance in comparison with other models. Therefore, cochleogram features are recommended for speaker recognition under noisy conditions using deep learning models.

4.2. Speaker Recognition Performance of the Models under Noisy Conditions

In section 4.1, we have shown that cochleogram features were more robust than spectrogram features in speaker recognition at various level of SNR (i.e., at SNR from -5dB to 20dB) using the models such as: basic 2DCNN, VGG-16, ResNet50, ECAPA-TDNN and TitaNet architectures. Cochleogram have shown superior performance over the spectrogram in both speaker identification and verification in each of the model at the SNR level discussed above. Therefore, the researchers employed cochleogram as an input in the models developed for speaker recognition under noisy condition in this dissertation.

In this section, the evaluation results of each model developed for speaker recognition under noisy condition were presented in detail. The models developed in this study for speaker recognition under noisy conditions includes CNN-LSTM, CNN-BiLSTM, CNN-GRU and CNN-BiGRU. Each models' performance were evaluated on the VoxCeleb1 dataset with real-world and white gaussian noise at the SNR of -5dB to 20dB and without additive noise and the results were discussed clearly in the following subsections. The speaker identification and verification performance of the each model on the dataset with the above noise types and SNR level were discussed. Cochleogram were employed as an input in each model and SNR level. The speaker identification accuracy of each model were presented in section 4.2.1 and the speaker verification performance of each model were presented in section 4.2.2.

4.2.1. Speaker Identification Performance of the Models

This section presents the speaker identification performance of the models such as CNN-LSTM, CNN-BiLSTM, CNN-GRU and CNN-BiGRU which were developed in this study. Each model were evaluated for speaker identification on the VoxCeleb1 dataset without additive noise, with real-world noise and white gaussian noises at the SNR of -5dB to 20dB in the interval of 5dB. Each of the experiment conducted using the above listed models were repeated for 10 rounds. The average of the results at each noise ratios were considered as the overall performance of the model at that specific noise ratio level. To present the model's performance graphically, one training progress of each model on the dataset with the real world noises at SNR of 0dB, 10dB, and 20dB were selected randomly.

Figure 45, illustrates the speaker identification accuracy of the models such as CNN-LSTM, CNN-BiLSTM, CNN-GRU and CNN-BiGRU on the VoxCeleb1 dataset with the real-world noise at SNR=0dB. The figure shows that CNN-LSTM have lowest accuracy, whereas CNN-BiGRU have highest accuracy than other models. The CNN-BiGRU model converges much faster and CNN-LSTM converges very slowly than other models during training. The CNN-BiLSTM and CNN-GRU have an average accuracy at this SNR level.

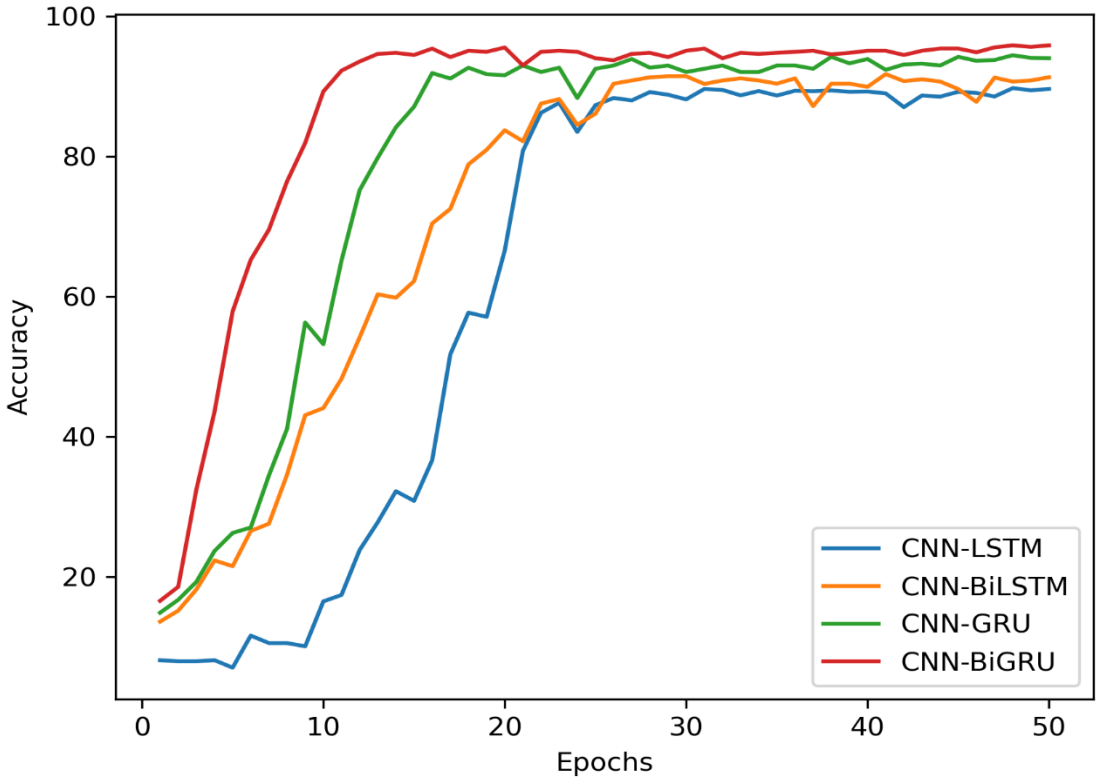

Figure 45: Speaker Identification Accuracy of the Models on the dataset with real world noise at SNR=0dB

In the figure 46, the speaker identification loss of the models such as CNN-LSTM, CNN-BiLSTM, CNN-GRU and CNN-BiGRU models on the VoxCeleb1 dataset with real world noise at the SNR of 0dB were presented. From the figure we can observe that CNN-LSTM have the highest loss in most of the epochs during training. The CNN-BiLSTM model have the lowest loss at each of the epoch during training. This confirms that CNN-BiGRU model have better speaker identification accuracy than other models employed in this study at the SNR=0dB.

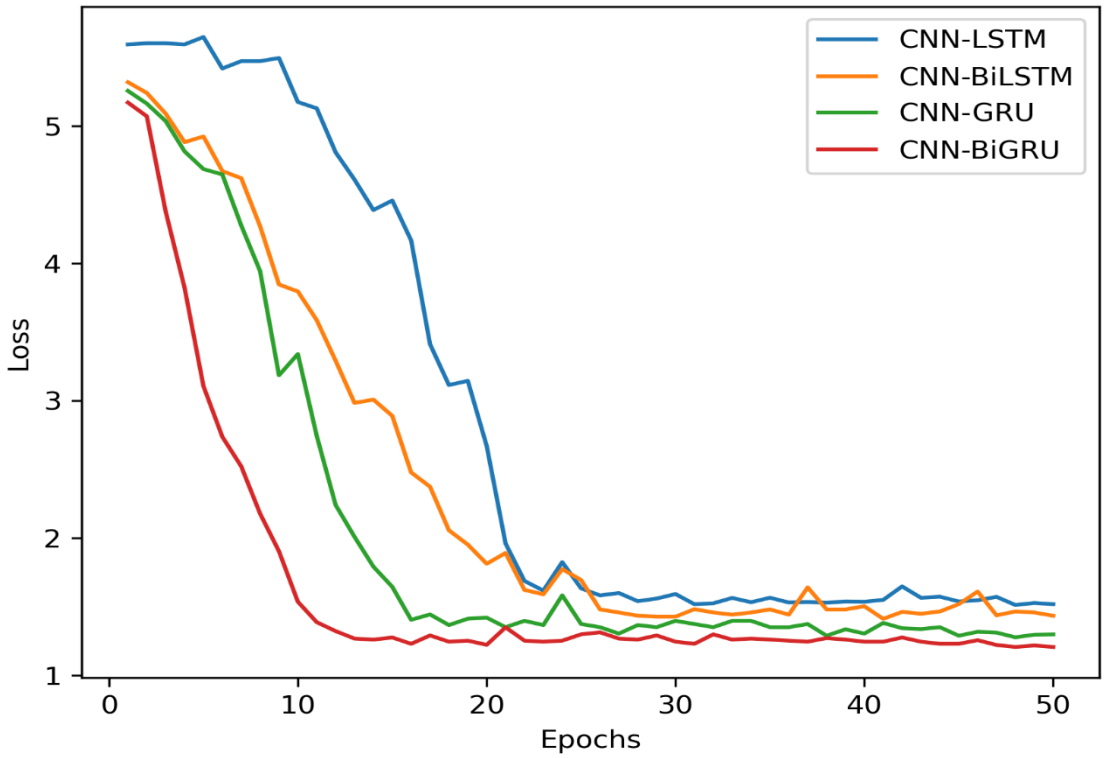

Figure 46: Speaker Identification Loss of Models on the dataset with real world noise at SNR=0dB

Figure 47, presents the speaker identification accuracy of the models such as CNN-LSTM, CNN-BiLSTM, CNN-GRU and CNN-BiGRU on the dataset with the real-world noise at the SNR of 10dB. The lowest and the highest speaker identification accuracy at this SNR level were achieved by CNN-LSTM and CNN-BiGRU models. Other models (i.e., CNN-BiLSTM and CNN-GRU) have shown relatively average accuracy in most of the epochs during the training. From the figure we can observe that CNN-BiGRU model converges much faster than other models. The results in the figure indicates that CNN-BiGRU model have better performance than other models in speaker identification on the dataset with real-world noise at SNR=10dB. Moreover, the CNN-BiGRU model at this SNR level have shown better accuracy than at previous SNR level (i.e., at SNR=0dB)

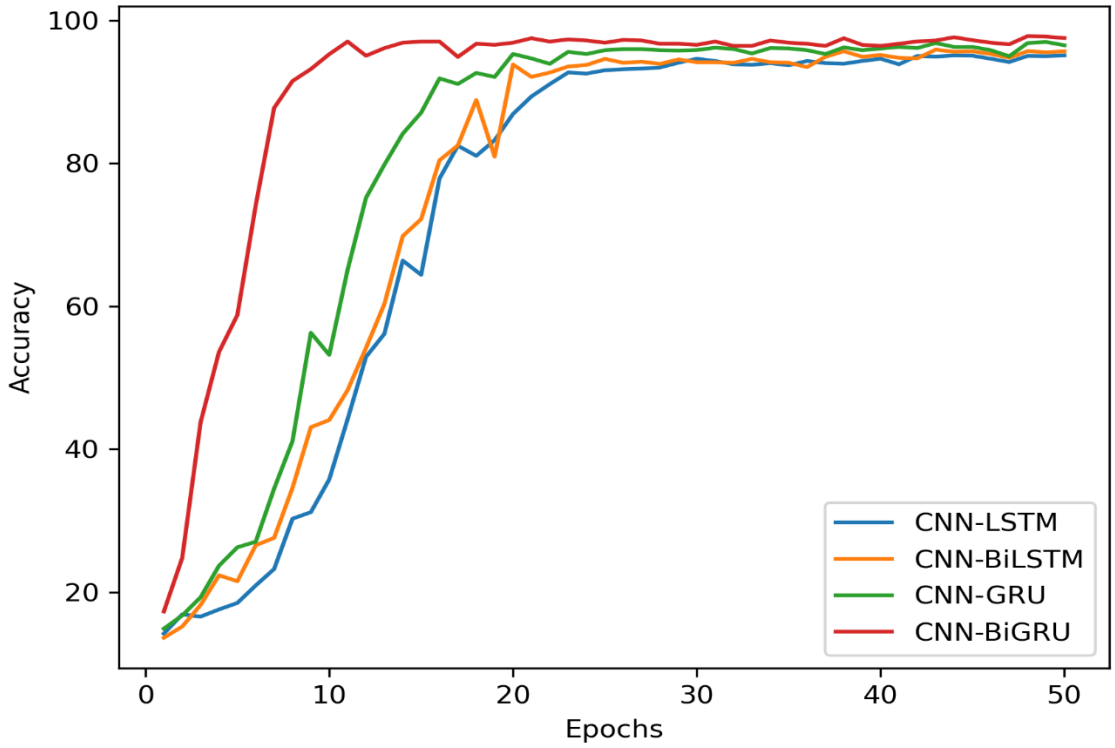

Figure 47: Speaker Identification Accuracy of the Models on the dataset with real world noise at SNR=10dB

Figure 48 show that the speaker identification loss of the models on the VoxCeleb1 dataset with real-world noise at SNR of 10dB. At this SNR level CNN-LSTM have shown the highest loss, whereas the CNN-BiGRU have achieved minimum loss in comparison with other models. Both CNN-BiLSTM and CNN-GRU have shown average loss, which was higher than the loss of BiGRU and lower than CNN-LSTM. The loss of the CNN-BiGRU model converged much faster than other models during training. The loss values in most of the epochs in the figure confirmed that CNN-BiGRU have superior performance than other model on the dataset with the real world noise at SNR=10dB.

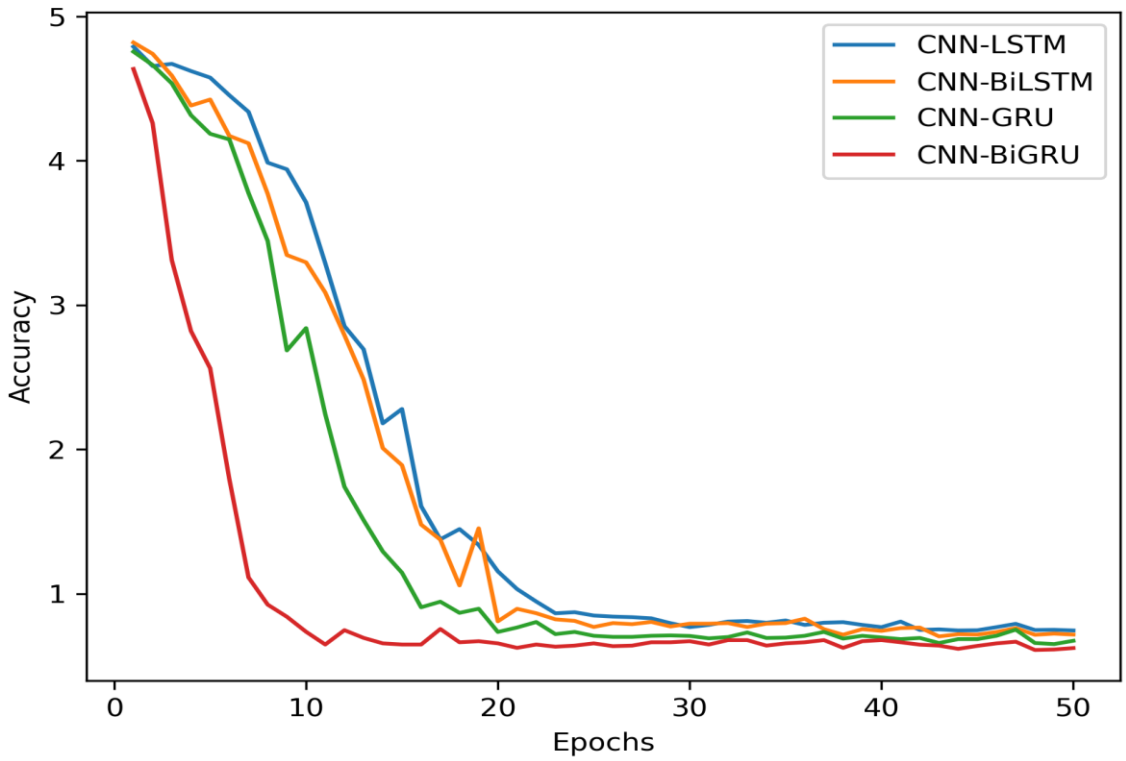

Figure 48: Speaker Identification Loss of the Models on the dataset with real world noises at SNR=10dB

Figure 49 presents speaker identification accuracy of the proposed and other models on the dataset with real world noises at SNR of 20dB. From the figure, we can observe that CNN-LSTM have the lowest accuracy than other models. Moreover, CNN-LSTM converges slower than other models during training. CNN-BiLSTM have lower accuracy than both CNN-GRU and CNN-BiGRU. During training, the learning curve of both CNN-GRU and CNN-BiGRU converged much faster than other models. For example, CNN-BiGRU converged after the 10^{th} epoch, likewise CNN-LSTM converged after the 24^{th} epoch. CNN-GRU have better accuracy than both CNN-LSTM and CNN-BiLSTM models. From the figures we can see that CNN-BiGRU model have highest accuracy and converges much faster than other models evaluated in this study.

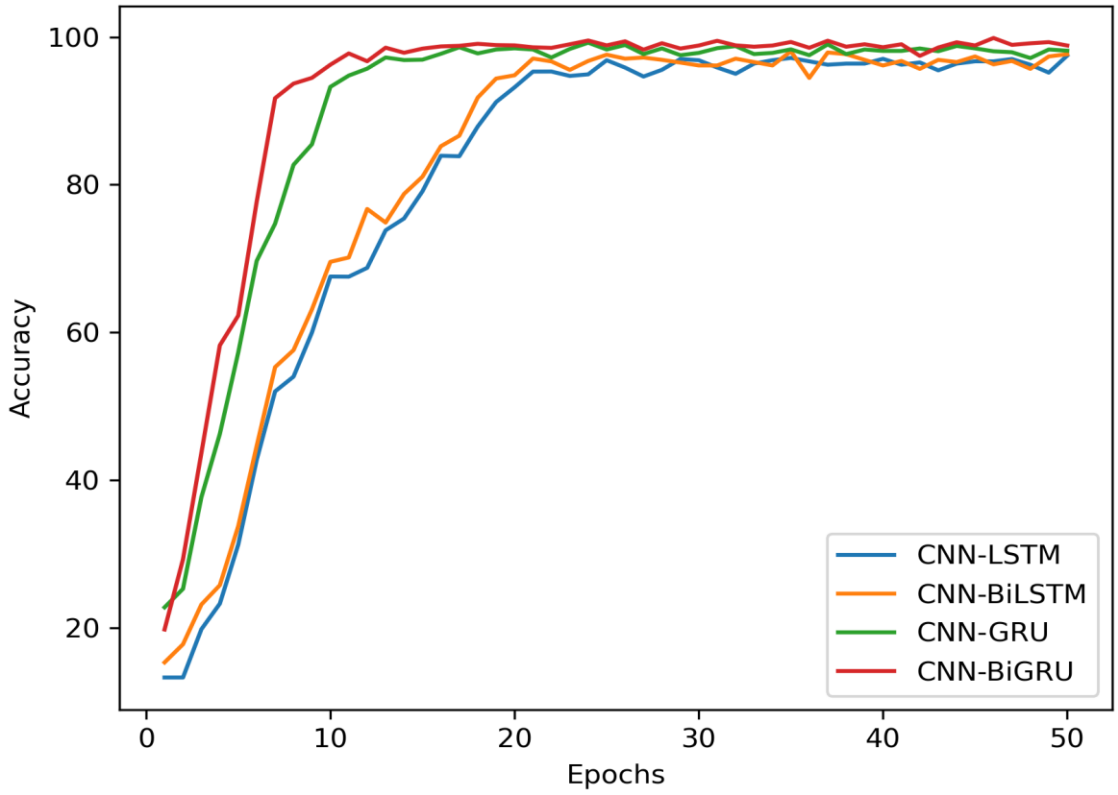

Figure 49: Speaker Identification Accuracy of the Models on the dataset with real world noise at SNR=20dB

In figure 50, the speaker identification loss of models such as CNN-LSTM, CNN-BiLSTM, CNN-GRU and CNN-BiGRU on the VoxCeleb1 with the real world noise at SNR level of 20dB were presented. The results indicate that the CNN-BiGRU model have the lowest loss and CNN-LSTM have highest loss than other models during training. The remaining models (CNN-BiLSTM and CNN-GRU) have average loss, but lower and higher than the loss of the CNN-LSTM and CNN-BiGRU models respectively. The loss of CNN-BiGRU model converges much faster than other models loss. The results of the loss in most of the epochs in the figure confirmed that CNN-BiGRU model have better performance than other models in speaker identification at SNR=20dB.

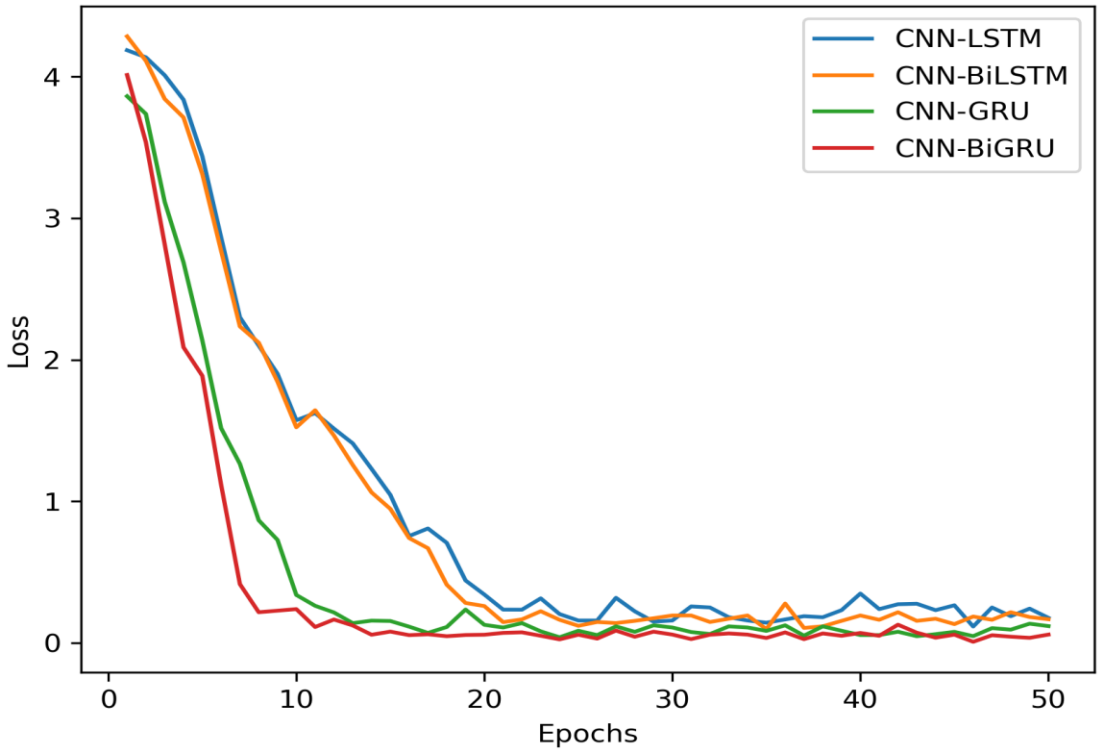

Figure 50: Speaker identification Loss of Models on the dataset with real-world noise at SNR=20dB

Table 13, presents an overall speaker identification accuracy of the models such as CNN-LSTM, CNN-BiLSTM, CNN-GRU and CNN-BiGRU on the VoxCeleb1 dataset with the white gaussian noise. Each model's accuracy at SNR level from -5dB to 20dB in the interval of 5dB were presented in this table.

At very high white gaussian noise on the dataset (at SNR=-5dB), an overall accuracy of the models ranges from 86.61% to 93.63%. At this SNR level CNN-LSTM had achieved the lowest accuracy than other models which was 86.61%, whereas CNN-BiGRU model have shown the highest accuracy than other models which was 93.63%. At this SNR level CNN-BiGRU model have shown an improvement ranging from 1.81% to 7.02%. The results in the table confirmed that CNN-BiGRU model have superior performance over the other model in speaker identification at the SNR level of -5dB.

An overall speaker identification accuracy of the models at the SNR level of 0dB range from 89.85 % to 96.24%. The minimum speaker identification performance was achieved by the CNN-LSTM model that was 89.85%, whereas the highest accuracy was achieved using the CNN-BiGRU model

that was 96.24%. At this SNR level CNN-BiGRU model have shown an improvement ranging from 2.02% to 6.39%. The results at SNR=0dB confirmed that CNN-BiGRU model have superior performance over the other models in speaker identification.

The overall speaker identification accuracy of the models at SNR=5dB indicated that CNN-BiGRU model have higher performance than other models. At this SNR level an overall accuracy of the models range from 93.86% to 97.41%. The CNN-LSTM model achieved the lowest accuracy than other models which was 93.86%, whereas CNN-BiGRU have shown the highest accuracy which was 97.41%. At this SNR level CNN-BiGRU have achieved an improvement ranging from 1.45% to 3.55%, which confirmed superior performance of the model.

At the SNR=10dB, an overall speaker identification accuracy of the models range from 96.35% to 97.83%. At this level of SNR, CNN-BiGRU have shown better performance than other models with the overall accuracy of 97.83%. The CNN-LSTM model have shown the lowest overall speaker identification accuracy than other models which was 96.35%. An improvement achieved by the CNN-BiGRU model over the other models range from 0.31% to 1.48% which was relatively smaller than previous improvements.

The speaker identification overall accuracy of the models at SNR=15dB ranges from 97.07% to 98.22%. The CNN-LSTM have also lowest overall accuracy which was 97.07% than other models. Moreover, highest overall accuracy at SNR=15dB was achieved by the CNN-BiGRU model. The results confirmed that CNN-BiGRU model have better performance than other model in speaker identification at SNR=15dB. This model have shown an improvement ranging from 0.17% to 1.15% which was relatively smaller than the improvements achieved by this model at higher noise ratio in the speech.

At the small noise ratio in the speech (at SNR=20dB) overall speaker identification accuracy of the models range from 97.36% to 98.73%. At this SNR level CNN-LSTM have shown smaller overall accuracy which was 97.36%, whereas CNN-BiGRU model have shown the highest overall accuracy which was 98.73%. The results in the table confirmed that CNN-BiGRU model have better performance than other models at the SNR level of 20dB. An improvement achieved using CNN-BiGRU model at this SNR level range from 0.13% to 1.07% which was relatively smaller than an improvements achieved at higher noise ratios.

The overall accuracy of the CNN-LSTM range from the 86.61% to 97.36% at the SNR level between -5dB to 20dB in interval of 5dB. The CNN-BiLSTM model have achieved the overall speaker identification accuracy ranging from 87.58% to 97.55% at the SNR level between -5dB to 20dB respectively, which were higher than the performance of the CNN-LSTM model. At the SNR level ranging from -5dB to 20dB in interval of the 5dB, the overall accuracy of the CNN-GRU model ranges from the 91.82% to 98.43%, which were better than the performance of the CNN-LSTM and CNN-BiLSTM at similar SNR levels. The overall accuracy of the CNN-BiGRU model at SNR level between -5dB to 20dB in interval of 5dB were range from 93.63% to 98.73%, which were the highest performance compared with other models at each similar SNR levels.

Table 13: Overall Speaker Identification Accuracy of the Models on the dataset with White Gaussian Noise

Method	Accuracy (in %) at SNR					
	-5dB	0dB	5dB	10dB	15dB	20dB
CNN-LSTM	86.61	89.85	93.86	96.35	97.07	97.36
CNN-BiLSTM	87.58	91.51	94.72	96.44	97.18	97.55
CNN-GRU	91.82	94.22	95.96	97.52	98.05	98.43
CNN-BiGRU (Proposed)	93.63	96.24	97.41	97.83	98.22	98.73

Table 14, reports an overall speaker identification accuracy of the models such as CNN-LSTM, CNN-BiLSTM, CNN-GRU and CNN-BiGRU on the VoxCeleb1 dataset with the real-world noise. Each model's accuracy at SNR level from -5dB to 20dB in the interval of 5dB.

At very high white gaussian noise on the dataset (at SNR=-5dB), an overall accuracy of the models ranges from 86.39% to 93.15%. At this SNR level CNN-LSTM had achieved the lowest accuracy than other models which was 86.39%, whereas CNN-BiGRU model have shown the highest accuracy than other models which was 93.15%. At this SNR level CNN-BiGRU model have shown an improvement ranging from 1.82% to 6.76%. The results in the table confirmed that CNN-BiGRU model have superior performance over the other model in speaker identification at the SNR level of -5dB.

An overall speaker identification accuracy of the models at the SNR level of 0dB range from 89.63% to 95.86%. The minimum speaker identification performance was achieved by the CNN-LSTM model that was 89.63%, whereas the highest accuracy was achieved using the CNN-BiGRU

model that was 95.86%. At this SNR level CNN-BiGRU model have shown an improvement ranging from 1.81% to 6.23%. The results at SNR=0dB confirmed that CNN-BiGRU model have superior performance over the other models in speaker identification.

The overall speaker identification accuracy of the models at SNR=5dB indicated that CNN-BiGRU model have higher performance than other models. At this SNR level an overall accuracy of the models range from 93.63% to 96.37%. The CNN-LSTM model achieved the lowest accuracy than other models which was 93.63%, whereas CNN-BiGRU have shown the highest accuracy which was 96.37%. At this SNR level CNN-BiGRU have achieved an improvement ranging from 0.61% to 2.74%., which confirmed superior performance of the model.

At the SNR=10dB, an overall speaker identification accuracy of the models range from 96.11% to 97.55%. At this level of SNR, CNN-BiGRU have shown better performance than other models with the overall accuracy of 97.55%. The CNN-LSTM model have shown the lowest overall speaker identification accuracy than other models which was 96.11%. An improvement achieved by the CNN-BiGRU model over the other models range from 0.24% to 1.44% which was relatively smaller than previous improvements.

The speaker identification overall accuracy of the models at SNR=15dB ranges from 96.84% to 98.07%. The CNN-LSTM have also lowest overall accuracy which was 96.84% than other models. Moreover, highest overall accuracy at SNR=15dB was achieved by the CNN-BiGRU model which was 98.07%. The results confirmed that CNN-BiGRU model have better performance than other model in speaker identification at SNR=15dB. This model have shown an improvement ranging from 0.22% to 1.35 % which was relatively smaller than the improvements achieved by this model at higher noise ratio in the speech.

At the small noise ratio in the speech (at SNR=20dB) overall speaker identification accuracy of the models range from 97.14% to 98.60%. At this SNR level CNN-LSTM have shown smaller overall accuracy which was 97.14%, whereas CNN-BiGRU model have shown the highest overall accuracy which was 98.60%. The results in the table confirmed that CNN-BiGRU model have better performance than other models at the SNR level of 20dB. An improvement achieved using CNN-BiGRU model at this SNR level range from 0.36% to 1.43 % which was relatively smaller than an improvements achieved at higher noise ratios.

The overall accuracy of the CNN-LSTM range from the 86.39% to 97.14% at the SNR level between -5dB to 20dB in interval of 5dB. The CNN-BiLSTM model have achieved the overall speaker identification accuracy ranging from 87.38% to 97.35% at the SNR level between -5dB to 20dB respectively, which were higher than the performance of the CNN-LSTM model. At the SNR level ranging from -5dB to 20dB in interval of the 5dB, the overall accuracy of the CNN-GRU model ranges from the 91.33% to 98.24%, which were better than the performance of the CNN-LSTM and CNN-BiLSTM at similar SNR levels. The overall accuracy of the CNN-BiGRU model at SNR level between -5dB to 20dB in interval of 5dB were range from 93.15% to 98.60%, which were the highest performance compared with other models at each similar SNR levels.

Table 14: Overall Speaker Identification Accuracy of the models on the dataset with real-world noises

Methods	Accuracy (in %) at SNR					
	-5db	0dB	5dB	10dB	15dB	20dB
CNN-LSTM	86.39	89.63	93.63	96.11	96.84	97.14
CNN-BiLSTM	87.38	91.31	94.51	96.22	96.97	97.35
CNN-GRU	91.33	94.05	95.76	97.31	97.85	98.24
CNN-BiGRU (Proposed)	93.15	95.86	96.37	97.55	98.07	98.60

Table 15, presents the speaker identification accuracy of the models such as CNN-LSTM, CNN-BiLSTM, CNN-GRU and CNN-BiGRU on the VoxCeleb1 dataset without additive noise. The overall accuracy of the models on the VoxCeleb1 dataset without additive noise range from 97.44% to 98.85%. The CNN-LSTM have shown the lowest accuracy (97.44%) in comparison with other models on the dataset without additive noise. The CNN-BiGRU model have shown the highest accuracy (98.85%) on the dataset without additive noise. Other models CNN-BiLSTM and CNN-GRU models have shown medium accuracy which were 97.79% and 98.52% respectively on the dataset without additive noises. The CNN-BiGRU models also have shown improvement over the other models using the dataset without additive noise. The results confirmed that CNN-BiGRU model performs better than other models developed in this study on the dataset without additive noise.

Table 15: Speaker identification accuracy of the models on the VoxCeleb1 dataset without additive noise

Methods	**Accuracy (in %)**
CNN-LSTM	97.44
CNN-BiLSTM	97.79
CNN-GRU	98.52
CNN-BiGRU (Proposed)	98.85

4.2.2. Speaker Verification Performance of the Models

In this section the speaker verification performance of the models such as CNN-LSTM, CNN-BiLSTM, CNN-GRU and CNN-BiGRU were presented. The equal error rate of each models on the VoxCeleb1 dataset without additive noise, with real-world noise and white gaussian noise at SNR of -5dB to 20dB in the interval of 5dB were presented.

Table 16, presents the speaker verification performance of the models such as CNN-LSTM, CNN-BiLSTM, CNN-GRU and CNN-BiGRU on the VoxCeleb1 dataset with the white gaussian noise at the SNR of the -5dB to 20dB in interval of 5dB.

At the very high gaussian noise in the speech (at SNR=-5dB), the EER of the models in speaker verification ranges from 10.51% to 14.27%. The lowest EER (highest performance) which was 10.51% was achieved using the CNN-BiGRU model, whereas the maximum EER which was 14.27% or lowest performance was achieved by CNN-LSTM model at the SNR level -5dB. In this SNR level the CNN-BiGRU model have shown an improvement (minimizes EER) from 0.65% to 3.76% over the other models employed in this study.

The EER of the models in speaker verification on the dataset with the white gaussian noise at SNR=0dB ranges from the 7.42% to 9.87%. The highest performance or minimum EER were achieved by CNN-BiGRU which was 7.42%, whereas the lowest performance or maximum EER were achieved using CNN-LSTM model. An improvement ranging from 0.50% to 2.45% were achieved by the CNN-BiGRU model over the other models.

On the dataset with white gaussian noise at the SNR=5dB, the CNN-BiGRU model have shown superior speaker verification performance over the other models. The performance or EER of the models in this SNR level ranges from 3.57% to 5.84% which was better than the models

performance at higher noise ratio. In this SNR level CNN-LSTM have achieved the highest EER which was 5.58% and CNN-BiGRU have shown the highest performance or minimum EER which was 3.57%. The CNN-BiGRU model have shown an improvement of 0.42% to 2.27% over the other models.

At medium white gaussian noise ratio (at SNR=10dB) in the speech, the speaker verification EER of the models range from 2.38 to 3.94% which was better than the performance in the higher noise ratio. The lowest performance or the highest EER which was 3.94% achieved by CNN-LSTM and the minimum EER which was 2.38% at this SNR level was achieved using CNN-BiGRU model. At this SNR level the CNN-BiGRU model have shown an improvement of 0.33% to 1.56 % over the other models.

The speaker verification EER of the models on the dataset with the white gaussian noise at SNR of 15dB ranges from 0.98% to 2.37% which were better than the performance of the models at the higher noise ratios. The CNN-LSTM have shown the highest EER which was 2.37% or the lowest performance and CNN-BiGRU model have shown the lowest EER which was 0.98% or the highest performance on the dataset with this level of SNR. In this SNR level CNN-BiGRU model have shown an improvement of 0.22% to 1.39 % over the other models.

At small white gaussian noise ratio in the speech (at SNR=20dB) the speaker verification EER of the models range from 0.46% to 1.51% which was much better than their performance at higher noise ratios. The maximum speaker verification performance or the lowest EER which was 0.46% was achieved by the CNN-BiGRU model. The minimum speaker verification performance or the highest EER which was 1.51% obtained by CNN-LSTM. At this SNR the CNN-BiGRU have also shown an improvement ranging from 0.19% to 1.05 % over the other models.

Moreover, the speaker verification EER of the CNN-LSTM ranges from 14.27% to 1.51% on the dataset with white gaussian noise at the SNR level ranges from -5dB to 20dB respectively, which were higher than EER of other models at each similar level of SNR. The CNN-BiLSTM model have shown the speaker verification EER ranging from 13.62% to 1.32% at the SNR level -5dB to 20dB respectively. The speaker verification performance or EER of the CNN-GRU ranges from the 11.62% to 0.64% on the dataset with white gaussian noise at SNR level of -5dB to 20dB respectively. The speaker verification EER of the CNN-BiGRU model ranges from 10.51% to 0.46% at SNR level of -5dB to 20dB respectively which were smaller than the EER of the other

models at similar SNR level. The results in the table at different level of SNR confirmed that the CNN-BiGRU model have better performance than other models at each SNR level.

Table 16: Speaker Verification performance of the models on the dataset with WGN

Model	**EER (%)**					
	-5dB	**0dB**	**5dB**	**10dB**	**15dB**	**20dB**
CNN-LSTM	14.27	9.87	5.84	3.94	2.37	1.51
CNN-BiLSTM	13.62	9.37	5.42	3.61	2.15	1.32
CNN-GRU	11.07	7.87	3.98	2.68	1.31	0.64
CNN-BiGRU (Proposed)	10.51	7.42	3.57	2.38	0.98	0.46

Table 17, presents the speaker verification performance or EER of the models such as CNN-LSTM, CNN-BiLSTM, CNN-GRU and CNN-BiGRU on the VoxCeleb1 dataset with real world noise at the SNR of -5dB to 20dB.

The speaker verification EER of the models on the dataset with very high real-world noise (at SNR=-5dB) ranges from 10.55% to 14.34%. At this SNR level the CNN-BiGRU model have shown superior performance (lowest EER which was 10.55%) than other models, whereas the CNN-LSTM model have achieved the lowest performance (highest EER which was 14.34%). The CNN-BiGRU model have shown maximum improvement ranging from 0.57% to 3.79% of EER over the other models.

At the SNR level of 0dB (where noise and signal ratio was equal), the speaker verification EER of the models range from 7.47% to 9.92%. The CNN-LSTM have shown the highest speaker verification EER which was 9.92% and the CNN-BiGRU model have achieved the lowest EER that was 7.47% which was highest performance at this specific SNR level. An improvement ranging from 0.44% to 2.45% of EER was achieved by the CNN-BiGRU model over the other models.

The EER of the models on the dataset with real-world noise at SNR=5dB ranges from 3.60% to 5.87% which was better performance (lower EER) than at SNR level of -5dB and 0dB. The speaker verification EER of the CNN-LSTM was higher than other models at this SNR level. The CNN-BiGRU model have shown higher performance or lower EER than other models at this SNR level.

An improvement of speaker verification EER ranging from 0.41% to 2.27% was achieved at this SNR level by CNN-BiGRU model over the other models.

At medium real world noise in the speech (at SNR=10dB), the speaker verification EER of the models ranges from 2.42% to 3.96% which was better than the models performance at the previous SNR levels. The maximum EER which was 3.96% achieved by the CNN-LSTM model which shows the model have lowest performance. The CNN-BiGRU model have achieved the lowest EER which was 2.42% or higher performance than other models at this specific SNR level. At this SNR level, CNN-BiGRU model have achieved an improvement of EER ranging from the 0.30% to 1.54% over the other models.

At the SNR=15dB, the speaker verification EER of the models ranges from the 1.02% to 2.38% which shows the models have better performance or lower EER at this SNR level than SNR level with higher noise ratio. The better performance or lower EER at this SNR level was achieved by CNN-BiGRU model which was 1.02%. Moreover, the CNN-LSTM model have shown the lowest performance or the higher EER which was 2.38% than other models at this specific SNR level. The CNN-BiGRU model have shown an improvement of EER ranging from 0.27% to 1.36% over the other models.

On the dataset with the small real world additive noise (at SNR=20dB), the speaker verification EER of the models range from the 0.47% to 1.53% which was smaller EER or better performance than the models performance on the higher noise ratio. At this specific SNR level CNN-LSTM have achieved the lowest performance or the higher EER which was 1.53% than other models, whereas the CNN-BiGRU model have achieved the higher performance or lower EER than other models which was 0.47%. An improvement of EER ranging from 0.18% to 1.06% was achieved over the other models by using CNN-BiGRU models.

Table 17: Speaker Verification performance of the models on the real-world noise added VoxCeleb1 dataset

Model	**EER (%)**					
	-5dB	**0dB**	**5dB**	**10dB**	**15dB**	**20dB**
CNN-LSTM	14.34	9.92	5.87	3.96	2.38	1.53
CNN-BiLSTM	13.67	9.43	5.45	3.63	2.18	1.35
CNN-GRU	11.12	7.91	4.01	2.72	1.29	0.65
CNN-BiGRU (Proposed)	10.55	7.47	3.6	2.42	1.02	0.47

In table 18, the speaker verification EER of the models such as CNN-LSTM, CNN-BiLSTM, CNN-GRU and CNN-BiGRU models were presented on the dataset without additive noise. The EER of the models on the dataset without additive noise ranges from the 0.37% to 1.16%. The CNN-LSTM model have achieved the lowest performance or the maximum EER than other models on the dataset without additive noise which was 1.16%. The CNN-BiGRU model have shown the better performance or minimum EER than other models on the dataset without additive noise which was 0.37%. An improvement of EER ranging from 0.08% to 0.79% was achieved by CNN-BiGRU model over the other models.

Table 18: Speaker Verification performance of the models on the dataset without additive noise

Model	**EER (%)**
CNN-LSTM	1.16
CNN-BiLSTM	1.04
CNN-GRU	0.45
CNN-BiGRU (Proposed)	0.37

Table 19, presents the comparisons of the highest performing model (proposed model) in speaker identification of this study with the existing works to show the effectiveness of the proposed model. The baseline speaker identification models for the comparison includes the study in ((Nagrani, Joon, & Zisserman, 2018), (Cai, Chen, & Li, 2018), and (Ding, Chen, Gong, Zha, & Wang, 2020). These baseline models were experimeted on similar dataset which was VoxCeleb1 dataset using deep learning models. The results in the table confirmed that the proposed model (CNN-BiGRU) in speaker identification have shown better performance than the existing works. The proposed model have shown an improvement ranging from 2.84% to 6.75% over the exisint works.

Table 19: Comparison of speaker identification performance of the proposed model with the existing works

Model	**Feature type**	**Dataset**	**Accuracy (%)**
CNN+embedding in ((Nagrani, Joon, & Zisserman, 2018)	Spectrogram	VoxCeleb1	92.10
Adaptive VGG-M in (Kim & Park, 2021)	Spectrogram	VoxCeleb1	95.31
CNN-LDE in (Cai, Chen, & Li, 2018)	Spectrogram	VoxCeleb1	95.70
AutoSpeech in (Ding, Chen, Gong, Zha, & Wang, 2020)	Spectrogram	VoxCeleb1	96.01
CNN-BiGRU (Proposed)	Cochleogram	VoxCeleb1	98.85

In table 20, the model which have highest speaker verification EER in this study was compared with the existing works to show the effectiveness of the proposed model. The baseline works selected for comparison includes the study in (Salehghaffari, 2018), (Kim & Park, 2021), (Cai, Chen, & Li, 2018), (Desplanques, Thienpondt, & Demuynck, 2020), and (Koluguri, Park, & Ginsburg, 2021) which were employed the VoxCeleb1 dataset for speaker verification using deep learning models. The comparison results in the table show that the speaker verification performance of the proposed model (CNN-BiGRU) was better than existing works.

Table 20: Comparison of Speaker Verification performance of proposed model with the existing works

Model	**Feature type**	**Dataset**	**EER (%)**
CNN-256 + Pair selection in (Salehghaffari, 2018)	Spectrogram	VoxCeleb1	10.5
Adaptive VGG-M in (Kim & Park, 2021)	Spectrogram	VoxCeleb1	7.8
CNN-LDE in (Cai, Chen, & Li, 2018)	Spectrogram	VoxCeleb1	4.56
ECAPA-TDNN in (Desplanques, Thienpondt, & Demuynck, 2020)	Spectrogram	VoxCeleb1	0.87
TitaNet in (Koluguri, Park, & Ginsburg, 2021)	Spectrogram	VoxCeleb1	0.68
CNN-BiGRU (Proposed)	Cochleogram	VoxCeleb1	0.37

Generally, our model has achieved better accuracy than other models evaluated in this study at different levels of SNR. Our model has also achieved superior accuracy than other models on the dataset without additive noises. The comparison also confirmed that our model has better accuracy than previous works used as baseline. The main reason for the enhancement of the accuracy is that the model has integrated the advantages of 2DCNN and BiGRU networks. Moreover, using cochleogram of the speech as an input improved the efficiency of the models under noisy conditions.

Conclusions and Recommendations

In this dissertation, the speaker recognition model for the noisy conditions was conducted by using the deep learning models. To select the better features for speaker recognition under noisy conditions, the noise robustness analysis of cochleogram and spectrogram features in speaker recognition were conducted. Analysis of both features was conducted using the basic 2DCNN, VGG-16, ResNet50, ECAPA-TDNN and TitaNet models. The experiments were conducted on the VoxCeleb1 dataset with real-world noise at SNR of -5dB to 20dB in intervals of 5dB and without additive noises. At each SNR level and without additive noise the noise robustness of both features was analyzed by using each type of deep learning model. The robustness of both features in speaker identification and verification was evaluated at each level of SNR. The analysis results indicate that cochleogram features are better for speaker recognition under noisy conditions at different levels of SNR. At very high noise ratio cochleogram achieved superior performance than spectrogram features in both speaker identification and verification. To develop the speaker recognition models for noisy conditions, hybrid models of CNN and enhanced RNN variants have been employed. The enhanced RNN variants employed in this study include LSTM, BiLSTM, GRU, and BiGRU models. The cochleogram were employed as an input at each speaker recognition models developed in this study. Each model was evaluated by using the VoxCeleb1 dataset with real-world noise, white Gaussian noise and without additive noise. The models were evaluated for speaker identification and verification under noisy conditions at the SNR of -5 dB to 20dB. The speaker recognition model using hybrid CNN and BiGRU on the cochleogram input model were proposed in this study because it has achieved better performance than other models on the dataset with real-world noise and white Gaussian noise at each SNR level and without additive noise in both speaker identification and verification. To show the effectiveness of the proposed model, the performance of the proposed model in speaker identification and verification was compared with the existing works. The comparison results also confirmed that the speaker identification and verification of the proposed model have performed better than existing works. The main reason for the performance enhancement was that the proposed model integrated the advantages of CNN and BiGRU models and cochleogram feature. The CNN model has advantages in extracting short-term correlation between the features of the speaker and it automatically extracts and adaptively learns features at each layer. The BiGRU model has an advantage in

extracting long-term feature dependency both in forward and backward directions. Cochleogram features are rich in acoustic features and more robust than other features for noise. Therefore, speaker recognition under noisy conditions using the CNN-BiGRU model and cochleogram input was proposed in this study. Several future research opportunities arise from the work in this dissertation. The researchers presented some of the future works which can be conducted by other researchers. In speech processing applications including speaker recognition, cochleogram performed better in noisy condition, whereas spectrogram also shown better performance in clean environment. The reserachers believe that identifying the SNR level at which spectrogram could have better performance than cochleogram and vice versa is very important to employ each feature in the appropriate SNR level for speech processing applications. In the speaker recognition using machine learning models, fusion of MFCC and GFCC have shown better performance than the separate features in some speech processing applications. In the speaker recognition using deep learning model, the reserachers believe that fusion of cochleogram and spectrogram features could enhance the performance of the models at some specific SNR level. The researchers have shown that hybrid models of CNN and RNN variants with two convolutional, two RNN variants layer and one fully connected layers could enhance the speaker recognition performance. The researchers believe that employing hybrid models of CNN with enhanced RNN variants with various number of layers and different types of architectures can improve the performance of the speaker recognition. Moreover, hybrid models of CNN with RNN or other deep learning and machine learning model could enhance the performance of the speaker recognition. Speaker recognition is very easy and interesting to employ together with other biometric techniques. Using more than one modality can enhance the performance and applicability in various real world applications. The researchers in the area can conduct multimodal speaker recognition using speech and other modalities such as face, iris, finger and gait recognition. In the future, localization of the speaker recognition using deep learning architectures could be conducted.

References

A., K., & Al-Karawi. (2020). Mitigate the reverberation effect on the speaker verification performance using different methods. *International Journal of Speech Technology, 24*, 143–153.

Abd, S., Nassar, M., Dessouky, M., Ismail, N., ElFishawy, A., & Abd, F. (2020). Text-independent speaker recognition using LSTM-RNN and speech enhancement. *Multimedia Tools and Applications,* 24013–24028.

Abdel-Hamid, O., Mohamed, A.-r., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014). Convolutional Neural Networks for Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* , 1533 - 1545.

Abdul, R., Setianingsih, C., & Nasrun, M. (2021). Speaker Recognition for Device Controlling using MFCC and GMM Algorithm. *2020 2nd International Conference on Electrical, Control and Instrumentation Engineering (ICECIE).* Kuala Lumpur, Malaysia.

Ahmad, K. S., Thosar, A. S., Nirmal, J. H., & Pande, V. S. (2015). A unique approach in text independent speaker recognition using MFCC feature sets and probabilistic neural network. *2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR).* Kolkata, India.

Ahmed, S., Mamun, N., & Hossain, A. (2021). Cochleagram Based Speaker Identification Using Noise Adapted CNN. *2021 5th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT).*

Ajgou, R., Sbaa, S., Ghendir, S., Chamsa, A., & Taleb-Ahmed. (2014). Robust remote speaker recognition system based on AR-MFCC features and efficient speech activity detection algorithm. *2014 11th International Symposium on Wireless Communications Systems (ISWCS).* Barcelona, Spain.

Alabbasi, H., Jalil, A., & Hasan, F. (2020). Adaptive wavelet thresholding with robust hybrid features for text-independent speaker identification system. *International Journal of Electrical and Computer Engineering* , 5208~5216 .

Alaliyat, S., Waaler, F., Dyvik, K., Oucheikh, R., & Hameed, I. (2021). Speaker Verification Using Machine Learning for Door Access Control Systems. *Proceedings of the International Conference on Artificial Intelligence and Computer Vision* (pp. 689–700). Springer Link.

Alam, J., Fathan, A., & Hyun, W. (2021). Text-Independent Speaker Verification Employing CNN-LSTM-TDNN Hybrid Networks. *International Conference on Speech and Computer* (pp. 1-13). Springer Nature.

Alam, J., Kinnunen, T., Kenny, P., Ouellet, P., & O'Shaughnessy, D. (2013). Multitaper MFCC and PLP features for speaker verification using i-vectors. *Speech Communication, 55*(2), 237-251.

Aldhaheri, R., & AlSaadi, F. (2004). Robust Text-independent Speaker Recognition with Short Utterance in Noisy Environment Using SVD as a Matching Measure. *Journal of King Saud University - Computer and Information Sciences*, 25-44.

Alegre, F., Soldi, G., Evans, N., Fauve, B., & Liu, J. (2014). Evasion and obfuscation in speaker recognition surveillance and forensics. *IEEE 2nd International Workshop on Biometrics and Forensics.* Valletta, Malta.

Ali, A., & Kumar, S. (2021). Automatic Speaker Recognition using Deep Neural Network Classifiers. *2021 2nd International Conference on Computation, Automation and Knowledge Management (ICCAKM).* Dubai, Arab Emirates: IEEE.

Al-Kaltakchi, M., Woo, W., Dlay, S., & Chambers, J. (2017). Speaker identification evaluation based on the speech biometric and i-vector model using the TIMIT and NTIMIT databases. *5th International Workshop on Biometrics and Forensics (IWBF).*

Ashar, A., Shahid, M., & Mushtaq, U. (2020). Speaker Identification Using a Hybrid CNN-MFCC Approach. *2020 International Conference on Emerging Trends in Smart Technologies (ICETST).* Karachi, Pakistan.

Ayadi, M., Hassan, A., Abdelnaby, A., & Elgendy, O. (2017). Text-independent speaker identification using robust statistics estimation. *Speech Communication, 92,* 52-63.

Ayoub, B., Jamal, K., & Arsalane, Z. (2016). Gammatone frequency Cepstral coefficients for speaker identification over VoIP networks. *2016 International Conference on Information Technology for Organizations Development (IT4OD).* Fez, Morocco.

B, N., Anees, M., & Yadava, T. (2023). Speech coding techniques and challenges: a comprehensive literature survey. *Multimedia Tools and Applications,* 29859–29879.

Bader, M., Shahin, I., Ahmed, A., & Werghi, N. (2022). Hybrid CNN-LSTM Speaker Identification Framework for Evaluating the Impact of Face Masks. *2022 International Conference on Electrical and Computing Technologies and Applications (ICECTA).* Ras Al Khaimah, United Arab Emirates.

Banjara, J., Mishra, K. R., Rathi, J., Karki, K., & Shakya, S. (2021). Nepali Speech Recognition using CNN and Sequence Models. *2020 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT).* Hyderabad, India: IEEE.

Belayneh G., Urgessa T., GopiKrishna T. (2019). Artificial Neural Network Based Amharic Language Speaker Recognition. Turkish Journal of Computer and Mathematics Education, 5105-5116

Ben, M. (2008). Display and Analysis of Speech. In D. K. Havelock, *Handbook of Signal Processing in Acoustics* (pp. 449–481). New York: Springer.

Benhafid, Z., Yasmine, K., & Amrouche, A. (2021). A Study of Acoustic Features in Arabic Speaker Identification under Noisy Environmental Conditions. *arXiv:2110.12304*.

Bhattacharya, G., Alam, J., Stafylakis, T., & Kenny, P. (2016). Deep Neural Network based Text-Dependent Speaker Recognition:Preliminary Results. *Odyssey*, 1-16.

Boussaa, M., Atouf, I., Atibi, M., & Bennis, A. (2016). ECG signals classification using MFCC coefficients and ANN classifier. *2016 International Conference on Electrical and Information Technologies (ICEIT).* Tangiers, Morocco: IEEE Access.

Bunrit, S., Inkian, T., Kerdprasop, N., & Kerdprasop, K. (2019). Text Independent Speaker Identification Using Deep Learning Model of Convolutional Neural Network. *International Journal of Machine Learning and Computing, 9*(2), 143-148.

Cai, W., Chen, J., & Li, M. (2018). Exploring the Encoding Layer and Loss Function in End-to-End Speaker and Language Recognition System. *arXiv:1804.05160v1 [eess.AS]*, 1-8.

Campbell, J. (1997). Speaker recognition: a tutorial. *Proceedings of the IEEE* (pp. 1437–1462). IEEE.

César, B., Mendes, M., Torres, J., & Assis, R. (2021). Comparing LSTM and GRU Models to Predict the Condition of a Pulp Paper Press. *energies, 14*, 1-21.

Chakroun, R., Beltaïfa, L., Frikha, M., & Ben, A. (2016). A hybrid system based on GMM-SVM for speaker identification. *2015 15th International Conference on Intelligent Systems Design and Applications (ISDA).* Marrakech: IEEE.

Chauhan, N., & Chandra, M. (2017). Speaker recognition and verification using artificial neural network. *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET).* Chennai, India: IEEE.

Chen, W.-C., Hsieh, C.-T., & Lai, E. (2004). Multiband Approach to Robust Text-Independent Speaker Identification. *Computational Linguisticsand Chinese Language Processing*.

Choudhary, H., Sadhya, D., & Vinal Patel. (2021). Automatic Speaker Verification using Gammatone Frequency Cepstral Coefficients. *2021 8th International Conference on Signal Processing and Integrated Networks (SPIN).* Noida, India.

Chowdhury, L., Zunair, H., & Mohammed, N. (2020). Robust Deep Speaker Recognition: Learning Latent Representation with Joint Angular Margin Loss. *applied sciences, 10*(21), 1-17.

Costantini, G., Cesarini, V., & Brenna, E. (2023). High-Level CNN and Machine Learning Methods for Speaker Recognition. *Sensors, 23*(7).

Delna, P. (2019, February 1). *Why are speech, language and communication so important?* Retrieved from https://beamservices.com.au/: https://beamservices.com.au/blog/why-are-speech-language-and-communication-so-important/

Desai, D., & Joshi, M. (2014). Speaker Recognition Using MFCC and Hybrid Model of VQ and GMM. *Advances in Intelligent Systems and Computing*, 53–63.

Desplanques, B., Thienpondt, J., & Demuynck, K. (2020). ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification. *arXiv:2005.07143v3 [eess.AS]*, 1-5.

Ding, S., Chen, T., Gong, X., Zha, W., & Wang, Z. (2020). AutoSpeech: Neural Architecture Search for Speaker Recognition. *arXiv:2005.03215v2 [eess.AS] 31*.

Dobie RA, V. H. (2004). *Hearing Loss: Determining Eligibility for Social Security Benefits.* Washington (DC): National Academies Press (US).

Dong, Y., Zhou, S., Xing, L., Chen, Y., Ren, Z., Dong, Y., & Zhang, X. (2022). Deep learning methods may not outperform other machine learning methods on analyzing genomic studies. *Front Genet*.

Dua, M., Kumar, R., & Biswas, M. (2018). Performance evaluation of Hindi speech recognition system using optimized filterbanks. *Engineering Science and Technology, an International Journal*, 389-398.

Ellis, D. (2002, 06 20). *Sound Examples/Noise*. Retrieved from Columbia University: https://www.ee.columbia.edu/~dpwe/sounds/noise/

Emre, S., Soufleris, P., Duan, Z., & Heinzelman, W. (2018). Front-end speech enhancement for commercial speaker verification systems. *Speech Communication, 99*, 101-113.

Eva, K., & Jozef, J. (2015). Speaker Recognition for Surveillance Application. *Journal of Electrical and Electronics Engineering*, 19-22.

Farsiani, S., Izadkhah, H., & Lotfi, S. (2022). An optimum end-to-end text-independent speaker identification system using convolutional neural network. *Computers and Electrical Engineering*.

Group, V. G. (2022, February 25). *The VoxCeleb1 Dataset*. Retrieved from Information Engineering: https://www.robots.ox.ac.uk/~vgg/data/voxceleb/vox1.html

Guennouni, S., Mansouri, A., & Ahaitouf, A. (2019, Oct 19). *Biometric Systems and Their Applications*. Retrieved from https://www.intechopen.com/: DOI: 10.5772/intechopen.84845

Guo, Y., Qiao, Y., Sukkarieh, S., Chai, L., & He, D. (2021). BiGRU-Attention Based Cow Behavior Classification Using Video Data for Precision Livestock Farming. *American Society of Agricultural and Biological Engineers*.

Gurbuz, S., J.Gowdy, & Tufekci, Z. (2002). Speech spectrogram based model adaptation for speaker identification. *Proceedings of the IEEE SoutheastCon 2000. 'Preparing for The New Millennium'*. Nashville, TN, USA.

Gustavo, A. (2007). Modeling prosodic differences for speaker recognition. *Speech Communication, 49*(4), 77-291.

Han, K., Omar, M., Pelecanos, J., Pendus, C., Yaman, S., & Zhu, W. (2011). Forensically inspired approaches to automatic speaker recognition. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* Prague, Czech Republic.

Hanifa, R. M., Isa, K., & Mohamad, S. (2020). Comparative Analysis on Different Cepstral Features for Speaker Identification Recognition. *IEEE Student Conference on Research and Development (SCOReD).* Batu Pahat, Malaysia.

Hansen, J. H., & Hasan, T. (2015). Speaker Recognition by Machines and Humans: A tutorial review. *IEEE Signal Processing Magazine*, 74 - 99.

Hasan, T., L, J. H., & Hansen. (2019). Acoustic Factor Analysis for Robust Speaker Verification. *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*.

Hossan, A., Memon, S., & Gregory, M. (2010). A novel approach for MFCC feature extraction. *2010 4th International Conference on Signal Processing and Communication Systems.* Gold Coast, QLD, Australia.

Hourri, S., & Kharroubi, J. (2019). A deep learning approach for speaker recognition. *International Journal of Speech Technology*, 123-131.

Hu, Z., Si, X., Luo, Y., Tang, S., & Jian, F. (2021). Speaker Recognition Based on 3DCNN-LSTM. *Engineering Letters*, 1-8.

Huapeng Wang, C. Z. (2020). The application of gammatone frequency cepstral coefficients for forensic voice comparison under noisy conditions. *Australian journal of Forensic Science*, 553-558.

India, M., Safari, P., & Hernando, J. (2019). Self Multi-Head Attention for Speaker Recognition. *INTERSPEECH*.

Institute, B. (2024). *What is Biometrics*. Retrieved from https://www.biometricsinstitute.org/: https://www.biometricsinstitute.org/what-is-biometrics/

Isam, A., John, P., David, L., & Victor, M. (2019). Speaker recognition using PCA-based feature transformation. *Speech Communication*, 33-46.

Islam, A., Jassim, W., Cheok, S., Shamsul, M., & Zilany, A. (2017). A Robust Speaker Identification System Using the Responses from a Model of the Auditory Periphery. *IEEE Access*.

Islam, S., Islam, N., Hashim, N., Rashid, M., & Bari, B. (2022). Janardan Banjara; Kaushal Raj Mishra; Jayshree Rathi; Karuna Karki; Subarna Shakya. *IEEE Access*, 58081 - 58096.

Ittichaichareon, C., Suksri, S., & Yingthawornsuk, T. (2012). Speech Recognition using MFCC. *International Conference on Computer Graphics, Simulation and Modeling (ICGSM'2012)*, (pp. 135-138). Thailand.

Jacob Benesty, M. M. (2008). *Springer Handbook of Speech Processing.* Springer Nature.

Jakubec, M., Lieskovska, E., & Jarina, R. (2021). Speaker Recognition with ResNet and VGG Networks. *2021 31st International Conference Radioelektronika (RADIOELEKTRONIKA).* Brno, Czech Republic: IEEE.

Joseph, T., & Billson, T. (2020). Emotional Speaker Recognition based on Machine and Deep Learning. *2020 2nd International Multidisciplinary Information Technology and Engineering Conference (IMITEC).* Kimberley, South Africa: IEEE.

Jung, J., Heo, H., Kim, J., Shim, H., & Yu, H. (2019). RawNet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification. *Electrical Engineering and Systems Science*, 1-5.

Kabir, M. M., Mridha, M. F., Shin, J., Jahan, I., & Ohi, A. Q. (2021). A Survey of Speaker Recognition: Fundamental Theories, Recognition Methods and Opportunities. *IEEE Access, 9*, 79236 - 79263.

Kanervisto, A., Vestman, V., Sahidullah, M., Hautamäki, V., & Kinnunen, T. (2017). Effects of gender information in text-independent and text-dependent speaker verification. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* New Orleans, LA, USA.

Karthik, S., Aju, J., & Anish, B. (2014). Speaker recognition system for security applications. *2013 IEEE Recent Advances in Intelligent Computational Systems (RAICS).* Trivandrum, India: IEEE.

Kaur, G., Bhushan, S., & Singh, D. (2016). Secure Speaker Biometric System using GFCC with Additive White Gaussian Noise and Wavelet Filter. *International Journal of Computer Science and Information Security*.

Khalid, S., Khalil, T., & Nasreen, S. (2014). A Survey of Feature Selection and FeatureExtraction Techniques in Machine Learning. *Science and Information Conference*, 1-8.

Kim, S.-H., & Park, Y.-H. (2021). Adaptive Convolutional Neural Network for Text-Independent Speaker Recognition. *INTERSPEECH*.

Kinkiri, S., & Keates, S. (2020). Speaker Identification: Variations of a Human voice. *2020 International Conference on Advances in Computing and Communication Engineering (ICACCE).* Las Vegas, NV, USA: IEEE.

Kinnunen, T., & Li, H. (2010). An overview of text-independent speaker recognition: from features to supervectors. *Speech Communication, 52*(1), 12-40.

Koluguri, N. R., Park, T., & Ginsburg, B. (2021). TITANET: NEURAL MODEL FOR SPEAKER REPRESENTATION WITH 1D DEPTH-WISE SEPARABLE CONVOLUTIONS AND GLOBAL CONTEXT. *arXiv:2110.04410v1 [eess.AS]*, 1-5.

KUMAR, S., RAJU, P., Rao, M., & Satheesh. (2010). SPEAKER RECOGNITION USING GMM. *International Journal of Engineering Science and Technology*.

Kumar, T., & Bhukya, R. (2022). Mel Spectrogram Based Automatic Speaker Verification Using GMM-UBM. *2022 IEEE 9th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON).* Prayagraj, India.

Labied, M., & Belangour, A. (2021). Automatic Speech Recognition Features Extraction Techniques: A Multi-criteria Comparison. *International Journal of Advanced Computer Science and Applications*, 177-182.

Leu, F.-Y., & Lin, G.-L. (2017). An MFCC-Based Speaker Identification System. *2017 IIEEE 31st International Conference on Advanced Information Networking and Applications (AINA).* Taipei, Taiwan.

Li, H., Ma, B., & Aik, K. (2013). Spoken Language Recognition: From Fundamentals to Practice. *Proceedings of the IEEE* (pp. 1136 - 1159). IEEE.

LI, J., ZHANG, X., XU, J., MA, S., & GAO, W. (2021). Learning to Fool the Speaker Recognition. *ACM Trans. Multimedia Comput. Commun. Appl*, 1-20.

Li, L., Liu, R., Kang, J., Fan, Y., Cui, H., Cai, Y., . . . Wang, D. (2022). CN-Celeb: Multi-genre speaker recognition. *Speech Communication*.

Li, Q., & Huang, Y. (2011). An Auditory-Based Feature Extraction Algorithm for Robust Speaker Identification Under Mismatched Conditions. *IEEE Transactions on Audio, Speech, and Language Processing, 19*(6), 1791 - 1801.

Liu, & K., G. (2018). Evaluating Gammatone Frequency Cepstral Coefficients with Neural Networks for Emotion Recognition from Speech. *arXiv:1806.09010v1 [cs.SD]*.

Liu, C., Yin, Y., Sun, Y., & Ersoy, O. (2022). Multi-scale ResNet and BiGRU automatic sleep staging based on attention mechanism. *PLOS ONE*, 1-20.

Liu, M., Xie, Y., Yao, Z., & Dai, B. (2006). A New Hybrid GMM/SVM for Speaker Verification. *18th International Conference on Pattern Recognition (ICPR'06).* Hong Kong, China: IEEE.

Liu, Y., Liu, X., Fan, W., Zhong, B., & Du, J. (2017). Efficient Audio-Visual Speaker Recognition via Deep Heterogeneous Feature Fusion. *Biometric Recognition*, 575–583.

Liu, Z., Wu, Z., Li, T., Li, J., & Shen, C. (2018). GMM and CNN Hybrid Method for Short Utterance Speaker Recognition. *IEEE Transactions on Industrial Informatics*, 3244 - 3252.

Mansour, A., & Lachiri, Z. (2017). A Comparative Study in Emotional Speaker Recognition in Noisy Environment. *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA).* Hammamet, Tunisia.

Marcela Hernandez-de-Menendez, 1. R., Escobar, C., & Arinez, J. (2021). Biometric applications in education. *Int J Interact Des Manuf*, 365–380.

Mashao, D. (2005). A hybrid GMM-SVM speaker identification system. *2004 IEEE Africon. 7th Africon Conference in Africa (IEEE Cat. No.04CH37590).* Gaborone, Botswana: IEEE.

MathWorks. (2023, March 21). *Feature Extraction*. Retrieved from https://www.mathworks.com/: communication using speech

Maurya, A., Kumar, D., & Agarwal, R. (2018). Speaker Recognition for Hindi Speech Signal using MFCC-GMM Approach. *Procedia Computer Science*, *125*, 880-887.

McLaren, M., Lei, Y., & Ferrer, L. (2015). Advances in deep neural network approaches to speaker recognition. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* South Brisbane, QLD, Australia: IEEE .

Menaka, R., Karthik, R., & Kabilan, S. (2024). An Improved AlexNet Model and Cepstral Coefficient-Based Classification of Autism Using EEG. *Clinical EEG and Neuroscience*, 43-51.

Mian, T., Choudhary, A., & Fatima, S. (2022). SOUND SIGNAL BASED GEAR FAULT DIAGNOSIS UNDER VARYING WORKING CONDITIONS. *The 28th International Congress on Sound and Vibration (ICSV28)*, (pp. 1-8). New Delhi, India.

Mikel, H. (2024, February 15). *5 Reasons to Use Biometrics to Attract More Business*. Retrieved from https://www.aware.com/: https://www.aware.com/blog-5-reasons-to-use-biometrics-to-attract-more-business/

Mobiny, A., & Najarian, M. (2018). Text Independent Speaker Verification Using LSTM Networks. *ArXiv:1805.00604v3 [eess.AS]*.

Moinuddin, A. K. (2014). Speaker Identification based on GFCC using GMM. *IJIRAE*.

Mróz-Gorgoń, B., Wodo, W., Andrych, A., Caban-Piaskowska, K., & Kozyra, C. (2022). Biometrics Innovation and Payment Sector Perception. *Sustainability*.

Muayad, A., Sahib, F., & Adnan, H. (2020). Speaker identification using convolutional neural network for clean and noisy speech samples. *2019 First International Conference of Computer and Applied Sciences (CAS).* IEEE.

Nagrani, A., Joon, C., & Zisserman, A. (2018). A large-scale speaker identification dataset. *arXiv:1706.08612v2 [cs.SD]*, 1-6.

Nagrani, A., Son, J., Xie, W., & Zisserman, A. (2020). Voxceleb: Large-scale speaker verification in the wild. *Computer Speech & Language*.

Nainan, S., & Kulkarni, V. (2021). Enhancement in speaker recognition for optimized speech features using GMM, SVM and 1-D CNN. *International Journal of Speech Technology* , 809–822.

Nakagawa, S., Wang, L., & Ohtsuka, S. (2012). Speaker Identification and Verification by Combining MFCC and Phase Information. *IEEE Transactions on Audio, Speech, and Language Processing* .

Nammous, M., Saeed, K., & Kobojek, P. (2022). Using a small amount of text-independent speech data for a BiLSTM large-scale speaker identification approach. *Journal of King Saud University - Computer and Information Sciences*, 764-770.

Naoyuki Kanda, Y. G. (2020). Joint Speaker Counting, Speech Recognition, and Speaker Identification for Overlapped Speech of Any Number of Speakers. *Audio and Speech Processing*.

Nasersharif, B., & Akbari, A. (2007). SNR-dependent compression of enhanced Mel sub-band energies for compensation of noise effects on MFCC features. *pattern recognition letters*, 1320-1326.

Nayana, P., Dominic, M., & Abraham, T. (2017). Comparison of Text Independent Speaker Identification Systems using GMM and i-Vector. *ICACC-2017*.

Nhat, M. D. (2024, April 29). *Applications of Biometric in Banking*. Retrieved from https://blog.nashtechglobal.com/: https://blog.nashtechglobal.com/applications-of-biometric-in-banking/

Nirvana, N., Mahmud, F., Habibullah, & Khan, R. (2022). Deep Learning Based Speaker Recognition System with CNN and LSTM Techniques. *2022 Interdisciplinary Research in Technology and Management (IRTM)* (pp. 1-6). IEEE.

O'Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. *arXiv:1511.08458v2 [cs.NE]*, 1-11.

Omid, S., & Hansen, J. (2014). Blind Spectral Weighting for Robust Speaker Identification under Reverberation Mismatch. *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*.

Patni, H., Jagtap, A., Bhoyar, V., & Gupta, A. (2021). Speech Emotion Recognition using MFCC, GFCC, Chromagram and RMSE features. *2021 8th International Conference on Signal Processing and Integrated Networks (SPIN).* Noida, India: IEEE Access.

Paulose, S., Mathew, D., & Thomas, A. (2017). Performance Evaluation of Different Modeling Methods and Classifiers with MFCC and IHC Features for Speaker Recognition. *Procedia Computer Science, 115*, 55-62.

Prachi, N. N., Nahiyan, F. M., Habibullah, M., & Khan, R. (2022). Deep Learning Based Speaker Recognition System with CNN and LSTM Techniques. *2022 Interdisciplinary Research in Technology and Management (IRTM).* Kolkata, India: IEEE.

Qi, J., Wang, D., Xu, J., & Tejedor, J. (2013). Bottleneck Features based on Gammatone Frequency Cepstral Coefficients. *INTERSPEECH*, 1751-1755.

R, S., & Patilkulkarni, S. (2021). Visual speech recognition for small scale dataset using VGG16 convolution neural network. *Multimedia Tools and Applications*, 28941–28952.

Rajesh, G. (2016). ANALYSIS OF MFCC FEATURES FOR EEG SIGNAL CLASSIFICATION. *INTERNATIONAL JOURNAL OF ADVANCES IN SIGNAL AND IMAGE SCIENCES*, 14-20.

Ramasubramanian, V. (2012). Speaker Spotting: Automatic Telephony Surveillance for Homeland Security. In A. P. Neustein, *Forensic Speaker Recognition* (pp. 427–468). New York: Springer.

Rao, N., Li, J., Lavrukhin, V., & Ginsburg, B. (2020). SPEAKERNET: 1D DEPTH-WISE SEPARABLE CONVOLUTIONAL NETWORK FOR TEXT-INDEPENDENT SPEAKER RECOGNITION AND VERIFICATION. *arXiv:2010.12653v1 [eess.AS]*.

Ravanelli, M., & Bengio, Y. (2019). Speaker Recognition from Raw Waveform with SincNet. *2018 IEEE Spoken Language Technology Workshop (SLT).* Athens, Greece: IEEE.

Rhynearson, C. (2024, February 23). *Biometrics vs Passwords: The Battle for Authentication Dominance*. Retrieved from https://www.techlocity.com/: https://www.techlocity.com/blog/biometrics-vs-passwords#:~:text=Biometric%20traits%20are%20unique%20to%20each%20individual%20and%20significantly%20more,such%20as%20PINs%20or%20tokens.

Rodríguez, E., Ruíz, B., García, Á., & García, F. (2005). Speech/speaker recognition using a HMM/GMM hybrid model. *Audio- and Video-based Biometric Person Authentication*, 227-234.

Ruiz, P. (2018, October 8). *Understanding and visualizing ResNets*. Retrieved from Towards Data Science: https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8

Safavi, S., Russell, M., & Jančovič, P. (2018). Automatic speaker, age-group and gender identification from children's speech. *Computer Speech & Language*, 141-156.

Sahidullah, & Saha, G. (2012). Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition. *Speech Communication*, 543-565.

Salehghaffari, H. (2018). Speaker Verification using Convolutional Neural Networks. *arXiv:1803.05427v2 [eess.AS]*, 1-6.

Salvati, D., Drioli, C., & Foresti, G. (2019). End to End Speaker Identification in noisy and reverberant environments using raw waveform convolutional neural networks. *InterSpeech* (pp. 4339-4335). Graz, Austria: 2019.

Salvati, D., Drioli, C., & Luca, G. (2023). A late fusion deep neural network for robust speaker identification using raw waveforms and gammatone cepstral coefficients. *Expert Systems with Applications*, 1-12.

Saritha, B., Azharuddin, M., Hussain, R., & Choudhury, M. (2022). Raw Waveform Based Speaker Identification Using Deep Neural Networks. *2022 IEEE Silchar Subsection Conference (SILCON).* Silchar, India: IEEE.

Schafer, R. (1994). Scientific Bases of Human-Machine Communication by Voice. *National Academies Press*, 15-33.

Sekkate, S., Khalil, M., & Adib, A. (2017). Speaker identification: A way to reduce call-sign confusion events. *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP).* Fez, Morocco.

Selvan, K., Joseph, A., & Babu, A. (2013). Speaker recognition system for security applications. *IEEE Recent Advances in Intelligent Computational Systems (RAICS).* Trivandrum, India.

Shah, V., & Chandra, M. (2020). Speech Recognition Using Spectrogram-Based Visual Features. *Advances in Machine Learning and Computational Intelligence* , 695–704.

Sharan, R., & Moir, T. (2019). Acoustic event recognition using cochleagram image and convolutional neural networks. *Applied Acoustics*, 62-66.

Sharma, D., & Ali, I. (2015). A modified MFCC feature extraction technique For robust speaker recognition. *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI).* Kochi, India.

Shekhar, H., & Roy, P. (2021). A CNN-BiLSTM based hybrid model for Indian language identification. *Applied Acoustics, 182*.

Shekhar, H., & Roy, P. (2021). A CNN-BiLSTM based hybrid model for Indian language identification. *Applied Acoustics*.

Shon, S., Tang, H., & Glass, J. (2019). Frame-Level Speaker Embeddings for Text-Independent Speaker Recognition and Analysis of End-to-End Model. *2018 IEEE Spoken Language Technology Workshop (SLT).* Athens, Greece.

Shrawankar, U., & Thakare, V. M. (2013). Techniques for Feature Extraction In Speech Recognition System : A Comparative Study. *arxiv.org*, 1-9.

Singh, G., Sharma, S., Kumar, V., Kaur, M., Baz, M., & Masud, M. (2021). Spoken Language Identification Using Deep Learning. *Computational Intelligence and Neuroscience*.

Singh, N., R.A.Khan, & Shree, R. (2012). Applications of Speaker Recognition. *Procedia Engineering, 38*, 3122-3126.

Soleymani, S., Dabouei, A., Mehdi, S., Kazemi, H., & Dawson, J. (2019). Prosodic-Enhanced Siamese Convolutional Neural Networks for Cross-Device Text-Independent Speaker Verification. *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS).* Redondo Beach, CA, USA.

Solutions, S. (2023, Nov 17). *AI vs Machine Learning vs Deep Learning: Know the Differences*. Retrieved from https://www.simplilearn.com/:

https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/ai-vs-machine-learning-vs-deep-learning

Soori, M., Arezoo, B., & Dastres, R. (2023). Artificial intelligence, machine learning and deep learning in advanced robotics, a review. *Cognitive Robotics*, 54-70.

Taherian, H., Wang, Z., Chang, J., & Wang, D. (2020). Robust Speaker Recognition Based on Single-Channel and Multi-Channel Speech Enhancement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 1293 - 1302.

Tazi, E. B., Benabbou, A., & Harti, M. (2012). Efficient text independent speaker identification based on GFCC and CMN methods. *2012 International Conference on Multimedia Computing and Systems.* Tangiers, Morocco.

Toledano, D., Ramos, D., Gonzalez-Dominguez, J., & González-Rodríguez, J. (2009). Speech Analysis. *Encyclopedia of Biometrics*, 1284–1289.

Torfi, A., Dawson, J., & Nasrabadi, N. (2018). Text Independent Speaker Verfication using 3D Convolutional Neural Network. *2018 IEEE International Conference on Multimedia and Expo (ICME).* San Diego, CA, USA.

Valero, X., & Alias, F. (2012). Gammatone Cepstral Coefficients: Biologically Inspired Features for Non-Speech Audio Classification. *IEEE Transactions on Multimedia, 14*(6), 1684 - 1689.

Wang, H., & Zhang, C. (2020). The application of Gammatone frequency cepstral coefficients for forensic voice comparison under noisy conditions. *Australian Journal of Forensic Sciences, 52*(5), 553-568.

Wang, J., Wang, C., Chin, Y., Liu, Y., Chen, E., & Chang, P. (2017). Spectral-temporal receptive fields and MFCC balanced feature extraction for robust speaker recognition. *Multimedia Tools and Applications, 76*, 4055–4068.

Wang, J.-C., Chin, Y.-H., Hsieh, W.-C., Lin, C.-H., Chen, Y.-R., & Siahaan, E. (2015). Speaker Identification With Whispered Speech for the Access Control System. *IEEE Transactions on Automation Science and Engineering* , 1191 - 1199.

Wang, J.-C., Wang, C., Chin, Y., Liu, Y., Chen, E., & Chang, P. (2017). Spectral-temporal receptive fields and MFCC balanced feature extraction for robust speaker recognition. *Multimedia Tools and Applications, 76*, 4055–4068.

Wang, M., Sirlapu, T., Kwasniewska, A., Szankin, M., Bartscherer, M., & Nic, R. (2018). Speaker Recognition Using Convolutional Neural Network with Minimal Training Data for Smart Home Solutions. *2018 11th International Conference on Human System Interaction (HSI).* Gdansk, Poland: IEEE.

Wang, X., Xue, F., Wang, W., & Liu, A. (2020). A network model of speaker identification with new feature extraction methods and asymmetric BLSTM. *Neurocomputing*, 167-181.

Wang, Y., Tang, F., & Zheng, J. (2012). Robust Text-independent Speaker Identification in a Time-varying Noisy Environment. *JOURNAL OF SOFTWARE*, 1975-1980.

Weng, Z., Li, L., & Guo, D. (2010). Speaker recognition using weighted dynamic MFCC based on GMM. *2010 International Conference on Anti-Counterfeiting, Security and Identification.* Chengdu, China.

Wilkinson, N., & Niesler, T. (2021). A Hybrid CNN-BiLSTM Voice Activity Detector. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* Toronto, ON, Canada: IEEE.

Wu, G., Ning, X., Hou, L., He, F., Zhang, H., & Shankar, A. (2023). Three-dimensional Softmax Mechanism Guided Bidirectional GRU Networks for Hyperspectral Remote Sensing Image Classification. *Signal Processing*.

Xie, W., Nagrani, A., Son, J., & Zisserman, A. (2019). Utterance-level Aggregation for Speaker Recognition in the Wild. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* Brighton, UK: IEEE.

Yadav, S., & Rai, A. (2018). Learning Discriminative Features for Speaker Identification and Verification. *InterSpeech*, 2237-2241.

Yağmur, S., & ÖZKURT, N. (2019). ECG Arrhythmia Classification By Using Convolutional Neural Network And Spectrogram. *2019 Innovations in Intelligent Systems and Applications Conference (ASYU).* Izmir, Turkey: IEEE Access.

Yan, F., Men, A., Yang, B., & Jiang, Z. (2016). An Improved Ranking-Based Feature Enhancement Approach for Robust Speaker Recognition. *IEEE Access*.

Ye, F., & Yang, J. (2021). A Deep Neural Network Model for Speaker Identification. *applied sciences, 11*(8), 1-18.

Demile Y. and Mulatu A. (2020). Frequency-domain Features for Environmental Accident Warning Recognition. *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, London, UK, 2020, pp. 40-46,.

Yenigalla, P., Kumar, A., Tripathi, S., Singh, C., Kar, S., & Vepa, J. (2018). Speech Emotion Recognition Using Spectrogram & Phoneme Embedding. *Interspeech*, 3688-3692.

Zhao, X., & Wang, D. (2013). Analyzing noise robustness of MFCC and GFCC features in speaker identification. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing.* Vancouver, BC, Canada: IEEE.

Zhao, Z., Duan, H., Min, G., Wu, Y., Huang, Z., Zhuang, X., . . . Fu, M. (2019). A lighten CNN-LSTM model for speaker verification on embedded devices. *Future Generation Computer Systems, 100*, 751-758.

ZRAR, K., & ABDULBASIT, K. (2022). Mel Frequency Cepstral Coefficient and Its Applications. *IEEE Access*, 122136 -122158.

APPENDICES

A. Important Packages for Cochleogram and Spectrogram Generation

```
import os
import random
import math
import librosa
import pylab
import cv2
from pathlib import Path
import numpy as geek
import numpy as np
import spafe
import matplotlib.pyplot as plt
from scipy.io import wavfile
from scipy.io.wavfile import write
from spafe.fbanks.gammatone_fbanks import gammatone_filter_banks
from spafe.features.mfcc import mel_spectrogram
from spafe.features.gfcc import erb_spectrogram
from spafe.utils.vis import show_spectrogram
from scipy.io.wavfile import read
from spafe.utils.vis import show_features
from IPython.display import Audio, IFrame, display
import spafe.utils.vis as vis
from pylab import rcParams
from spafe.utils.preprocessing import pre_emphasis, framing, windowing, zero_handling
from scipy.fftpack import dct
```

B. Sample Code for Spectrogram Generation

```
def spectrogram (sig,fs, pre_emph=0, pre_emph_coeff=0.97, win_len=0.030, win_hop = 0.015,
win_type="hamming", nfilts=128, nfft=2048, low_freq=0, high_freq=8000, scale = "constant",
fbanks=None, conversion_approach="Oshaghnessy",):
        if fbanks is None:
            mel_fbanks_mat, _ = mel_filter_banks(nfilts=nfilts, nfft=nfft, fs=fs,
            low_freq=low_freq, high_freq=high_freq, scale=scale,
            conversion_approach=conversion_approach,)
                fbanks = mel_fbanks_mat
        if pre_emph:
                sig = pre_emphasis(sig=sig, pre_emph_coeff=0.97)
        frames, frame_length = framing(sig=sig, fs=fs, win_len=win_len, win_hop=win_hop)
        windows = windowing(frames=frames, frame_len=frame_length, win_type=win_type)
        fourrier_transform = np.absolute(np.fft.fft(windows, nfft))
        fourrier_transform = fourrier_transform[:, : int(nfft / 2) + 1]
        abs_fft_values = (1.0 / nfft) * np.square(fourrier_transform)
        features = np.dot(abs_fft_values, fbanks.T)  # dB
        return features, fourrier_transform


parent_list = os.listdir(voxceleb1)
L1=len(parent_list)
for i in range(0,L1):
        par_list1=os.listdir(os.path.join(voxceleb1 +'/'+ parent_list[i]))
        L2=len(par_list1)
        for j in range(L2):
                par_list2=os.listdir(os.path.join(voxceleb1 +'/'+ parent_list[i]+'/'+par_list1[j]))
                L3=len(par_list2)
                for k in range(L3):
                        file_path = os.path.join(voxceleb1 +'/'+ parent_list[i]+'/'+par_list1[j],
                        par_list2[k])
```

```
#print(par_list2[k])
sig, fs = librosa.load(file_path)
file_stem = Path(file_path).stem
target_dir=parent_list[i]
dist_dir = os.path.join(os.path.join(Vox1_Mel, 'Clean'), target_dir)
file_dist_path = os.path.join(dist_dir, file_stem)
if not os.path.exists(file_dist_path + '.png'):
    if not os.path.exists(dist_dir):
        os.mkdir(dist_dir)
mSpec, _ = spectrogram(sig, fs=fs, pre_emph=0, pre_emph_coeff=0.97,
win_len=0.030, win_hop=0.015, win_type="hamming", nfilts=128,
nfft=2048, low_freq=0, high_freq=fs/2)
amin = 1e-10
magnitude = np.abs(mSpec.T)
ref_value = np.max(magnitude)
log_spec = 10.0 * np.log10(np.maximum(amin, magnitude) /
np.maximum(amin, ref_value))
log_spec = np.maximum(log_spec, log_spec.max() - 80.0)
pylab.figure(figsize=(14, 4))
pylab.imshow(log_spec,origin="lower",aspect="auto",cmap="jet",)
pylab.savefig(f'{file_dist_path+par_list1[j]}.png')
pylab.close()
```

C. Sample Code for Cochleogram Generation

```
def cochleogram(sig, fs, pre_emph=0, pre_emph_coeff=0.97, win_len=0.030, win_hop=
0.015, win_type="hamming", nfilts=128, nfft=2048,low_freq=50, high_freq=8000, scale=
"constant", fbanks=None, conversion_approach="Glasberg",):
    if fbanks is None:
        gamma_fbanks_mat, _ = gammatone_filter_banks(nfilts=nfilts, nfft=nfft, fs=fs,
        low_freq=low_freq, high_freq=high_freq, scale=scale, conversion_approach=
        conversion_approach,)
```

```
            fbanks = gamma_fbanks_mat
        if pre_emph:
            sig = pre_emphasis(sig=sig, pre_emph_coeff=0.97)
        frames, frame_length = framing(sig=sig, fs=fs, win_len=win_len, win_hop=win_hop)
        windows = windowing(frames=frames, frame_len=frame_length, win_type=win_type)
        fourrier_transform = np.absolute(np.fft.fft(windows, nfft))
        fourrier_transform = fourrier_transform[:, : int(nfft / 2) + 1]
        abs_fft_values = (1.0 / nfft) * np.square(fourrier_transform)
        features = np.dot(abs_fft_values, fbanks.T)
        return features, fourrier_transform
parent_list = os.listdir(voxceleb1)
L1=len(parent_list)
for i in range(0,L1):
        par_list1=os.listdir(os.path.join(voxceleb1 +'/'+ parent_list[i]))
        L2=len(par_list1)
        for j in range(L2):
            par_list2=os.listdir(os.path.join(voxceleb1 +'/'+ parent_list[i]+'/'+par_list1[j]))
            L3=len(par_list2)
            for k in range(L3):
                file_path = os.path.join(voxceleb1 +'/'+ parent_list[i]+'/'+par_list1[j],
                par_list2[k])
                sig, fs = librosa.load(file_path)
                file_stem = Path(file_path).stem
                target_dir=parent_list[i]
                dist_dir = os.path.join(os.path.join(Vox1_Coch, 'Clean'), target_dir)
                file_dist_path = os.path.join(dist_dir,file_stem)
                if not os.path.exists(file_dist_path + '.png'):
                   if not os.path.exists(dist_dir):
                      os.mkdir(dist_dir)
```

```
                gSpec, gfreqs = cochleogram(sig, fs=fs, pre_emph=0, pre_emph_coeff =
                0.97, win_len=0.030, win_hop=0.015, win_type="hamming", nfilts=128,
                nfft=2048, low_freq=0, high_freq=fs/2)
                amin = 1e-10
                magnitude = np.abs(gSpec.T)
                ref_value = np.max(magnitude)
                log_spec = 10.0 * np.log10(np.maximum(amin, magnitude) /
                np.maximum(amin, ref_value))
                log_spec = np.maximum(log_spec, log_spec.max() - 80.0)
                pylab.figure(figsize=(14, 4))
                pylab.imshow(log_spec,origin="lower",aspect="auto",cmap="jet",)
                pylab.savefig(f'{file_dist_path+par_list1[j]}.png')
                pylab.close()
```

D. Sample Code for Cochleogram Generation at the SNR=5dB

```
def get_noise_from_sound(signal,noise,SNR):
        RMS_s=math.sqrt(np.mean(signal**2))
        RMS_n=math.sqrt(RMS_s**2/(pow(10,SNR/10)))#required RMS of noise
        RMS_n_current=math.sqrt(np.mean(noise**2))#current RMS of noise
        noise=noise*(RMS_n/RMS_n_current)
        return noise
noise_file='E:/Datasets/audio/Noise/babble.wav'
parent_list = os.listdir(voxceleb1)
L1=len(parent_list)
for i in range(0,L1):
        print(parent_list[i])
        par_list1=os.listdir(os.path.join(voxceleb1 +'/'+ parent_list[i]))
        L2=len(par_list1)
        for j in range(L2):
                par_list2=os.listdir(os.path.join(voxceleb1 +'/'+ parent_list[i]+'/'+par_list1[j]))
                L3=len(par_list2)
                for k in range(L3):
                        file_path = os.path.join(voxceleb1 +'/'+ parent_list[i]+'/'+par_list1[j],
                        par_list2[k])
                        sig, fs = librosa.load(file_path)
                        noise, sr = librosa.load(noise_file)
```

```
SNR=5
if((len(noise)<len(sig))):
        noise=geek.concatenate((noise,noise), axis = 0)
        noise=geek.concatenate((noise,noise), axis = 0)
        noise=geek.concatenate((noise,noise), axis = 0)
        noise=noise[0:len(sig)]
elif((len(noise)>len(sig))):
        noise=noise[0:len(sig)]
else:
        noise=noise
noise=get_noise_from_sound(sig,noise,SNR)
signal_noise=sig+noise
file_stem = Path(file_path).stem
target_dir=parent_list[i]
dist_dir = os.path.join(os.path.join(Vox1_Coch, 'nsr05'), target_dir)
file_dist_path = os.path.join(dist_dir, par_list1[j]+file_stem)
if not os.path.exists(file_dist_path + '.png'):
   if not os.path.exists(dist_dir):
      os.mkdir(dist_dir)
gSpec, gfreqs = erb_spectrogram(signal_noise, fs=fs, pre_emph=0,
pre_emph_coeff= 0.97, win_len=0.030,win_hop=0.015,
win_type="hamming", nfilts=128, nfft=2048, low_freq=0, high_freq=fs/2)
amin = 1e-10
magnitude = np.abs(gSpec.T)
ref_value = np.max(magnitude)
log_spec = 10.0 * np.log10(np.maximum(amin, magnitude) /
np.maximum(amin, ref_value))
log_spec = np.maximum(log_spec, log_spec.max() - 80.0)
pylab.figure(figsize=(14, 4))
pylab.imshow(log_spec, origin="lower",aspect="auto",cmap="jet",)
pylab.savefig(f'{file_dist_path}.png')
pylab.close()
```

E. Important Packages of Deep Learning Models for Speaker Recognition

```
import numpy as np
import pandas as pd
import os, pathlib, wave, scipy,pylab
import tensorflow as tf
import keras
import matplotlib.pyplot as plt
import pylab as pl
```

```
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential,load_model,Model
from keras.layers import GRU,LSTM,RNN,Conv2D,add,
concatenate,BatchNormalization,Bidirectional,MaxPooling2D,
AveragePooling2D,Input,Dense,Activation, Reshape
from keras.layers import
GlobalAveragePooling2D,Flatten,Dropout,TimeDistributed,Embedding,SimpleRNN
from keras.constraints import maxnorm
from keras.utils import np_utils
from keras import backend as K
from keras.wrappers.scikit_learn import KerasClassifier
from random import shuffle
from tqdm import tqdm
import skimage
from skimage import transform
from sklearn.model_selection import GridSearchCV
import cv2
```

F. Sample code for Speaker Recognition using basic 2DCNN model

```
TRAIN_DIR='E:/Datasets/Cochleogram/'
TEST_DIR='E:/Datasets/Cochleogram/'
def get_data(Dir):
        X = []
        y = []
        i=0
        for nextDir in os.listdir(Dir):
        label=i
        temp = Dir + nextDir
        for file in tqdm(os.listdir(temp)):
                img = cv2.imread(temp + '/' + file)
                if img is not None:
                        img = skimage.transform.resize(img, (224, 224, 3))
                        img = np.asarray(img)
                        X.append(img)
                        y.append(label)
        i=i+1
   X = np.asarray(X)
   y = np.asarray(y)
   return X,y

X_train, y_train = get_data(TRAIN_DIR)
```

```
X_test , y_test = get_data(TEST_DIR)
print(X_train.shape,'\n',X_test.shape)
print(y_train.shape,'\n',y_test.shape)
from keras.utils.np_utils import to_categorical
y_train = to_categorical(y_train,1251)
y_test = to_categorical(y_test,1251)
print(y_train.shape,'\n',y_test.shape)
def Basic_2DCNN(img_h, channel, num_classes):
        input_shape = (224, 224, 3)
        kernel_size=(3, 3)
         pool_size=2
        act='relu'
        input_data = Input(name='the_input', shape=input_shape, dtype='float32')
        inner = Conv2D(64, kernel_size, padding='same',activation=act, kernel_initializer =
        'he_normal', name='conv1')(input_data)
        inner = MaxPooling2D(pool_size=(pool_size, pool_size), name='max1')(inner)
        inner = BatchNormalization()(inner)
        inner = Conv2D(128, kernel_size, padding='same',activation=act, kernel_initializer=
        'he_normal', name='conv2')(inner)
        inner = MaxPooling2D(pool_size=(pool_size, pool_size), name='max2')(inner)
        inner = BatchNormalization()(inner)
        inner = Conv2D(256, kernel_size, padding='same',activation=act, kernel_initializer=
        'he_normal',name='conv3')(inner)
        inner = MaxPooling2D(pool_size=(pool_size, pool_size), name='max3')(inner)
        inner = BatchNormalization()(inner)
        inner = Conv2D(512, kernel_size, padding='same',activation=act,
        kernel_initializer='he_normal',name='conv4')(inner)
        inner = MaxPooling2D(pool_size=(pool_size, pool_size), name='max4')(inner)
        inner = BatchNormalization()(inner)
        inner = Flatten()(inner)
        inner = Dense(1251, kernel_initializer='he_normal',name='dense2')(inner)
        y_pred = Activation('softmax', name='softmax')(inner)
        model =  Model(inputs=input_data, outputs=y_pred)
        return model
img_h = 224
models = Basic_2DCNN (img_h, 3, 1251)#sgd = SGD(lr=0.02, decay=1e-6, momentum=0.9,
nesterov=True, clipnorm=5)
models.compile(optimizer=tf.keras.optimizers.RMSprop(), loss='categorical_crossentropy',
metrics=['accuracy'])
history = models.fit(X_train, y_train, validation_data = (X_test , y_test) ,epochs=50)
```

G. Sample Code for Speaker Recognition using Hybrid CNN and BiGRU

```
def CNN_BiGRU(img_h, channel, num_classes):

        kernel_size = (3, 3)
        pool_size = 2
        time_dense_size = 128
        rnn_size = 256
        minibatch_size = 32
        act = 'relu'
        img_w = 224
        img_h = 224
        input_shape = (img_w, img_h, channel)
        input_data = Input(name='the_input', shape=input_shape, dtype='float32')
        inner = Conv2D(64, kernel_size, padding='same',activation=act,
        kernel_initializer='he_normal',name='conv1')(input_data)
        inner = MaxPooling2D(pool_size=(pool_size, pool_size), name='max1')(inner)
        inner = BatchNormalization()(inner)
        inner = Conv2D(128, kernel_size, padding='same',activation=act,
        kernel_initializer='he_normal',name='conv2')(inner)
        inner = MaxPooling2D(pool_size=(pool_size, pool_size), name='max2')(inner)
        inner = BatchNormalization()(inner)
        conv_to_rnn_dims = (img_w // (pool_size ** 2),(img_h // (pool_size ** 2)) *128)
        inner = Reshape(target_shape=conv_to_rnn_dims, name='reshape')(inner)
        BiGRU1 = Bidirectional(GRU(rnn_size, return_sequences=False ,kernel_initializer=
        'he_normal', name='gru2'))(inner)
        BiGRU2 = Bidirectional(GRU(rnn_size, return_sequences=False, go_backwards= True,
        kernel_initializer= 'he_normal', name='gru2_b'))(inner)
        inner = Dense(num_classes, kernel_initializer='he_normal', name='dense2')
        (concatenate([BiGRU1, BiGRU2]))
        y_pred = Activation('softmax', name='softmax')(inner)
        model =  Model(inputs=input_data, outputs=y_pred)
        return model
img_h = 224
model = CNN_BiGRU(img_h, 3, 1251)#sgd = SGD(lr=0.02, decay=1e-6, momentum=0.9,
nesterov=True, clipnorm=5)
model.compile(optimizer=tf.keras.optimizers.RMSprop(), loss='categorical_crossentropy',
metrics=['accuracy'])
model.summary()
```

H. Sample Screenshot of Speaker Recognition using CNN-LSTM at SNR=-5dB

```
Epoch 1/50
117/117 [==============================] - 444s 2s/step - loss: 3.7664 - accuracy: 0.0534 - val_loss: 3.4768 - val_accuracy: 0.0812
Epoch 2/50
117/117 [==============================] - 59s 505ms/step - loss: 3.3154 - accuracy: 0.1146 - val_loss: 3.1699 - val_accuracy: 0.1363
Epoch 3/50
117/117 [==============================] - 61s 522ms/step - loss: 2.9264 - accuracy: 0.2129 - val_loss: 2.8742 - val_accuracy: 0.2205
Epoch 4/50
117/117 [==============================] - 60s 512ms/step - loss: 2.4462 - accuracy: 0.3152 - val_loss: 2.3684 - val_accuracy: 0.3461
Epoch 5/50
117/117 [==============================] - 58s 496ms/step - loss: 1.9880 - accuracy: 0.4384 - val_loss: 1.9518 - val_accuracy: 0.4564
Epoch 6/50
117/117 [==============================] - 59s 508ms/step - loss: 1.6214 - accuracy: 0.5275 - val_loss: 1.6275 - val_accuracy: 0.5804
Epoch 7/50
117/117 [==============================] - 59s 506ms/step - loss: 1.2497 - accuracy: 0.6392 - val_loss: 1.4502 - val_accuracy: 0.6309
Epoch 8/50
117/117 [==============================] - 58s 496ms/step - loss: 0.9339 - accuracy: 0.7342 - val_loss: 1.1874 - val_accuracy: 0.7106
Epoch 9/50
117/117 [==============================] - 58s 498ms/step - loss: 0.6590 - accuracy: 0.8150 - val_loss: 1.3150 - val_accuracy: 0.6784
Epoch 10/50
117/117 [==============================] - 59s 500ms/step - loss: 0.4079 - accuracy: 0.8974 - val_loss: 0.9344 - val_accuracy: 0.8162
Epoch 11/50
117/117 [==============================] - 58s 493ms/step - loss: 0.2564 - accuracy: 0.9399 - val_loss: 0.8910 - val_accuracy: 0.8315
Epoch 12/50
117/117 [==============================] - 58s 497ms/step - loss: 0.1676 - accuracy: 0.9643 - val_loss: 1.0326 - val_accuracy: 0.8009
Epoch 13/50
117/117 [==============================] - 58s 499ms/step - loss: 0.1067 - accuracy: 0.9785 - val_loss: 1.3576 - val_accuracy: 0.7060
Epoch 14/50
117/117 [==============================] - 58s 497ms/step - loss: 0.0794 - accuracy: 0.9836 - val_loss: 1.1239 - val_accuracy: 0.7825
Epoch 15/50
117/117 [==============================] - 58s 499ms/step - loss: 0.0553 - accuracy: 0.9863 - val_loss: 0.9216 - val_accuracy: 0.8530
Epoch 16/50
117/117 [==============================] - 59s 502ms/step - loss: 0.0436 - accuracy: 0.9922 - val_loss: 0.9077 - val_accuracy: 0.8530
Epoch 17/50
117/117 [==============================] - 58s 496ms/step - loss: 0.0525 - accuracy: 0.9863 - val_loss: 0.9116 - val_accuracy: 0.8698
Epoch 18/50
117/117 [==============================] - 58s 496ms/step - loss: 0.0434 - accuracy: 0.9893 - val_loss: 0.9261 - val_accuracy: 0.8515
Epoch 19/50
117/117 [==============================] - 58s 496ms/step - loss: 0.0278 - accuracy: 0.9949 - val_loss: 0.9066 - val_accuracy: 0.8668
Epoch 20/50
117/117 [==============================] - 58s 497ms/step - loss: 0.0336 - accuracy: 0.9919 - val_loss: 0.9764 - val_accuracy: 0.8484
Epoch 21/50
117/117 [==============================] - 58s 497ms/step - loss: 0.0219 - accuracy: 0.9946 - val_loss: 0.9779 - val_accuracy: 0.8668
Epoch 22/50
117/117 [==============================] - 58s 497ms/step - loss: 0.0316 - accuracy: 0.9919 - val_loss: 1.0231 - val_accuracy: 0.8560
Epoch 23/50
117/117 [==============================] - 58s 500ms/step - loss: 0.0348 - accuracy: 0.9911 - val_loss: 0.9682 - val_accuracy: 0.8576
Epoch 24/50
117/117 [==============================] - 58s 495ms/step - loss: 0.0211 - accuracy: 0.9949 - val_loss: 0.9496 - val_accuracy: 0.8760
Epoch 25/50
117/117 [==============================] - 58s 499ms/step - loss: 0.0262 - accuracy: 0.9917 - val_loss: 1.0140 - val_accuracy: 0.8668
Epoch 26/50
117/117 [==============================] - 58s 498ms/step - loss: 0.0184 - accuracy: 0.9965 - val_loss: 1.0227 - val_accuracy: 0.8729
Epoch 27/50
117/117 [==============================] - 58s 497ms/step - loss: 0.0109 - accuracy: 0.9976 - val_loss: 1.1078 - val_accuracy: 0.8606
Epoch 28/50
117/117 [==============================] - 59s 504ms/step - loss: 0.0127 - accuracy: 0.9968 - val_loss: 1.0535 - val_accuracy: 0.8683
Epoch 29/50
117/117 [==============================] - 58s 499ms/step - loss: 0.0163 - accuracy: 0.9954 - val_loss: 1.0559 - val_accuracy: 0.8591
Epoch 30/50
117/117 [==============================] - 58s 495ms/step - loss: 0.0166 - accuracy: 0.9965 - val_loss: 1.0968 - val_accuracy: 0.8622
```

```
Epoch 31/50
117/117 [==============================] - 58s 500ms/step - loss: 0.0131 - accuracy: 0.9968 - val_loss: 1.0759 - val_accuracy: 0.8606
Epoch 32/50
117/117 [==============================] - 58s 499ms/step - loss: 0.0224 - accuracy: 0.9930 - val_loss: 1.0471 - val_accuracy: 0.8714
Epoch 33/50
117/117 [==============================] - 58s 495ms/step - loss: 0.0205 - accuracy: 0.9936 - val_loss: 1.0324 - val_accuracy: 0.8606
Epoch 34/50
117/117 [==============================] - 58s 498ms/step - loss: 0.0169 - accuracy: 0.9965 - val_loss: 1.0929 - val_accuracy: 0.8622
Epoch 35/50
117/117 [==============================] - 58s 498ms/step - loss: 0.0076 - accuracy: 0.9989 - val_loss: 1.2588 - val_accuracy: 0.8270
Epoch 36/50
117/117 [==============================] - 58s 495ms/step - loss: 0.0104 - accuracy: 0.9962 - val_loss: 1.0754 - val_accuracy: 0.8591
Epoch 37/50
117/117 [==============================] - 58s 496ms/step - loss: 0.0052 - accuracy: 0.9984 - val_loss: 1.1191 - val_accuracy: 0.8637
Epoch 38/50
117/117 [==============================] - 57s 491ms/step - loss: 0.0193 - accuracy: 0.9962 - val_loss: 1.0997 - val_accuracy: 0.8606
Epoch 39/50
117/117 [==============================] - 59s 504ms/step - loss: 0.0131 - accuracy: 0.9946 - val_loss: 1.1114 - val_accuracy: 0.8668
Epoch 40/50
117/117 [==============================] - 58s 496ms/step - loss: 0.0198 - accuracy: 0.9930 - val_loss: 1.0590 - val_accuracy: 0.8729
Epoch 41/50
117/117 [==============================] - 57s 490ms/step - loss: 0.0056 - accuracy: 0.9987 - val_loss: 1.0561 - val_accuracy: 0.8729
Epoch 42/50
117/117 [==============================] - 58s 496ms/step - loss: 0.0129 - accuracy: 0.9952 - val_loss: 1.2170 - val_accuracy: 0.8652
Epoch 43/50
117/117 [==============================] - 57s 492ms/step - loss: 0.0084 - accuracy: 0.9970 - val_loss: 1.1431 - val_accuracy: 0.8545
Epoch 44/50
117/117 [==============================] - 58s 494ms/step - loss: 0.0078 - accuracy: 0.9973 - val_loss: 1.1973 - val_accuracy: 0.8576
Epoch 45/50
117/117 [==============================] - 58s 494ms/step - loss: 0.0063 - accuracy: 0.9981 - val_loss: 1.1038 - val_accuracy: 0.8683
Epoch 46/50
117/117 [==============================] - 58s 493ms/step - loss: 0.0190 - accuracy: 0.9949 - val_loss: 1.1417 - val_accuracy: 0.8744
Epoch 47/50
117/117 [==============================] - 58s 495ms/step - loss: 0.0122 - accuracy: 0.9962 - val_loss: 1.2546 - val_accuracy: 0.8591
Epoch 48/50
117/117 [==============================] - 58s 495ms/step - loss: 0.0134 - accuracy: 0.9962 - val_loss: 1.1774 - val_accuracy: 0.8652
Epoch 49/50
117/117 [==============================] - 58s 492ms/step - loss: 0.0135 - accuracy: 0.9962 - val_loss: 1.1750 - val_accuracy: 0.8668
Epoch 50/50
117/117 [==============================] - 58s 498ms/step - loss: 0.0114 - accuracy: 0.9979 - val_loss: 1.1658 - val_accuracy: 0.8652
```