Republic of Iraq
Ministry of Higher Education
And Scientific Research
University of Babylon
College of Engineering

Employing Different Techniques for Speaker Recognition System Based on Deep Learning Approach

A Thesis
Submitted to the Department of Electrical Engineering, Faculty of Engineering, University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master of Science (M.Sc.) in Electrical Engineering / Industrial Electronics

By

Huda Wasfi Hassoon

Supervised by

Assist. Prof. Dr. Ahmed Q. Aldhahab

Assist. Prof. Dr. Hanaa Mohsin Ali

2023A.D **1445A.H**

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

﴿وَعِندَهُۥ مَفَاتِحُ ٱلْغَيْبِ لَا يَعْلَمُهَآ إِلَّا هُوَۚ وَيَعْلَمُ مَا فِى ٱلْبَرِّ وَٱلْبَحْرِۚ وَمَا تَسْقُطُ مِن وَرَقَةٍ إِلَّا يَعْلَمُهَا وَلَا حَبَّةٍ فِى ظُلُمَٰتِ ٱلْأَرْضِ وَلَا رَطْبٍ وَلَا يَابِسٍ إِلَّا فِى كِتَٰبٍ مُّبِينٍ۝﴾

صَدَقَ اللَّهُ العلي العظيم

الأنعام (59)

Dedications

It was challenging, but I completed it. I want to express my heartfelt dedication to my amazing parents. They truly deserve all the appreciation for their unwavering support and guidance throughout my journey to obtaining my master's degree. Thank you very much!

Words can barely express how grateful and amazed I am.

It has been a great source of inspiration, encouragement, and direction for me. You have truly inspired me to strive for success in everything I do, persevere no matter what, and have unwavering confidence in myself. I appreciate all your kind wishes and prayers for me. They have always given me the courage I needed.

I would like to express my sincere gratitude to my parents for their unwavering spiritual support throughout my life, as well as to my sisters and brother who have consistently believed in me along the way.

Lastly, I would like to dedicate my work to Imam Mahdi (may God hasten his arrival) and all the scholars.

Huda Wasfi

Acknowledgments

I express my utmost gratitude and reverence to my God, "ALLAH," the Supreme Being, for bestowing upon me the ability to successfully accomplish this task, despite the numerous challenges encountered along the way.

I would like to extend my utmost gratitude to my supervisor, Assistant Professor Dr. Ahmed Q. Aldhahab, and Assistant Professor Dr. Hanaa Mohsin Ali for their unwavering support throughout the course of my thesis. Their exceptional comprehension of the subject matter and prompt responsiveness to the challenges I encountered were invaluable. Additionally, their guidance and advice have greatly contributed to my personal growth and knowledge acquisition. Furthermore, their unwavering patience, enthusiasm, motivation, and guidance proved invaluable throughout the entire duration of this thesis. They also imparted to me the necessary methodology to effectively present this thesis with utmost clarity.

I would like to extend my sincere gratitude to the numerous individuals who have provided me with their invaluable assistance and support. While it is impossible to list everyone here, please know that your contributions are deeply appreciated and will be remembered indefinitely. Additionally, I would like to express my heartfelt thanks to my friends for their unwavering encouragement and care throughout the duration of this project.

Abstract

Speaker recognition refers to the cognitive process of identifying the speaker's identity by analyzing and extracting certain distinct features included within the speech signal. The categorization of speaker recognition can be divided into two main categories: identification and verification. Speaker identification refers to the computational process of ascertaining the identity of a registered speaker who has produced a particular speech. In contrast, speaker verification refers to the validity of a speaker's identification assertion, leading to either acceptance or rejection.

This thesis investigates the subject of speaker identification systems, with a specific emphasis on the different methodologies used for extracting features from a speech signal.

The proposed system has three fundamental phases, consisting of: preprocessing, feature extraction, and classification. During the preprocessing phase, several techniques are used, such as noise removal, data augmentation, and segmentation of time duration into distinct lengths (0.5, 1, 2, 3, and 5 seconds), etc. These techniques are implemented to properly organize the data and get discriminant features during the feature extraction phase. In the feature extraction phase, various methods are employed, including the utilization of the Mel-frequency Cepstral Coefficient (MFCC), the integration of Mel-frequency Cepstral Coefficient with Principal Component Analysis (MFCC/PCA), the implementation of two-dimensional Discrete Wavelet Transform (2D-DWT), the combination of two-dimensional Discrete Wavelet Transform with Principal Component Analysis (2D-DWT/PCA), the integration of Mel-frequency Cepstral Coefficient with two-dimensional Discrete Wavelet Transform (MFCC/2D-DWT), and eventually the adoption of two-dimensional Discrete Multi-wavelet Transform (2D-DMWT). Finally, the extracted features are fed to

the third phase, which employs a deep learning algorithm that is based on Convolutional Neural Network (CNN) and used for classification purposes.

In this thesis, four different databases are used to evaluate the proposed approaches; namely, SALU-AC, ELSDSR, RAVDESS, and TIMIT. These databases have different speech variations, such as age, gender, number of speakers, etc.

The proposed system has demonstrated outstanding performance, achieving the highest accuracy rates of 99.82% and 99.22% for the ELSDSR and SALU-AC databases, respectively, within a time duration of 5 sec. based on the MFCC/2D-DWT approach. The RAVDESS database achieved a classification accuracy of 97.96% within a duration of 3 sec. based on the 2D-DMWT method. The TIMIT database demonstrated the highest accuracy of 96.02% when employing a duration of 2 sec. based on the MFCC/2D-DWT method. The length of the speech samples used in speaker identification systems significantly influences the system's overall performance. Typically, the use of longer phrases can convey a greater amount of information and reduce variation in the speaker's voice. These factors can enhance the accuracy and robustness of the speaker identification system.

The study conducted in this thesis accomplished the objectives of attaining successful results and decreasing the data dimensions, hence mitigating the system's complexity. The results accomplished by the proposed system are outperformed those results discussed in the previous works based on the same databases.

Table of contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Definition
Adam	Adaptive Moment Estimation
ANN	Artificial Neural Network
BFCC	Bark Frequency Cepstral Coefficient
BPNN	Back Propagation Neural Network
CG	Cochlea Gram
CLPC	Cepstral Linear Prediction Coefficient
CM	Confusion Matrix
CNN	Convolution Neural Network
CNN-VG	Convolution Neural Network Visual Geometry Group
CWT	Continuous Wavelet Transform
DA	Data Augmentation
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
Dim. R	Dimensionality Reduction
DTW	Dynamic Time Warping
DMWT	Discrete Multi-wavelet Transform
DNN	Deep Neural Network
DWT	Discrete Wavelet Transform
ELSDSR	English Language Speed Database for Speaker Recognition
EOF	Empirical Orthogonal Function
FC	Fully Connected
FFT	Fast Fourier Transform
FFNN	Feed Forward Neural Network
FFBPNN	Feed Forward Back Propagation Neural Network
FN	False Negative
FP	False Positive
GFCC	Gammatone Frequency Cepstral Coefficient
GHM	Geronimo, Hardian, and Mossopust
GMM	Gaussian Mixture Model
K-CV	K-Fold Cross Validation

Abbreviation	Definition
KNN	K-nearest Neighbour
LPC	Linear Predictive Coding
LPCC	Linear Predictive Cepstral Coefficient
LSH	Locality Sensitive Hashing
MFCC	Mel-frequency Cepstral Coefficient
MLP	Multi-layer Perceptron
MRA	Multi-resolution Analysis
OPSO	Optimized Particle Swarm Optimization
PCA	Principal Component Analysis
PIN	Personal Identification Number
PLP	Perceptual Linear Prediction
PP	Pitch Period
RAVDESS	Ryerson Audio-Visual Database of Emotional Speech and Song
RelU	Rectifier Linear Unit
ResNet	Simplified Residual Convolution Neural Network
RW-CNN	Row Wave Convolution Neural Network
SALU-AC	Salford University Anechoic Chamber
SC	Spectral Centroid
SE	Squeeze-and-Excitation
SDC	Shifted Delta Cepstral
Sgdm	Stochastic Gradient Descent with Momentum
SGD	Stochastic Gradient Descent
SI	Speaker Identification
SV	Speaker Verification
SVD	Singular Value Decomposition
SWT	Stationary Wavelet Transform
TIMIT	Texas Instruments Massachusetts Institute of Technology
TN	True Negative
TP	True Positive
VQ	Vector Quantization
WFT	Windowed Fourier Transform
WT	Wavelet Transform

List of Symbols

Item	Description
$X[n]$	The value of the input variable at time n.
$Y[n]$	The value of the output variable at time n.
$X[n-1]$	The value of the input variable at time n-1.
α	A constant that determines the weight given to the previous value of X in calculating the current value of Y.
$W(n)$	Hamming window.
N	The number of samples in each frame.
n	The index of the window.
$X(n)$	The input signal.
$Y(n)$	The output signal.
x_k	Aperiodic sequence with N value.
$e^{-2\pi jkn/N}$	a complex number that represents a rotation by an angle $-2\pi jkn/N$ of radians in the complex plane.
f_{Mel}	A unit of measure that represents the perceived pitch of a sound.
f_{LIN}	A unit of measure that represents the number of cycles per second of a sound wave.
C_n	The coefficient of the cepstrum mel frequency.
s_k	The coefficient of the mel power.
μ	The average value of the variable x.
D	Mean-Centered data.
D	A matrix whose columns are the deviations of the random variables from their means.
D^T	The transpose of matrix D.
λ	The eigenvalue.
V	The eigenvector.
*	Complex conjugation.
$\psi(t)$	The mother wavelet.
s	The scaling factor.
τ	The translation factor.
$\Psi_{j,k}(t)$	The CWT of the signal. It is a function of two variables, j and k.

s_0	This is a scaling factor that determines the size of the wavelet.
τ_0	This is a translation factor that determines the position of the wavelet.
t	The time variable.
$\Psi_{m,n}(t)$	Wavelet at scale m, and translation n.
$h(k)$	The low-pass filter.
$\phi(t)$	The scaling function.
$g(k)$	The high-pass filter.
$\Psi(t)$	The wavelet function.
$h(0), h(1), \ldots$	The low-pass filter coefficients.
$g(0), g(1), \ldots$	The high pass filter coefficients.
α	Is a parameter that determines the shape of the wavelet
$\Phi(t)$	Multi-scaling function.
$\Psi(t)$	Multi-wavelet function.
G_k	Filter matrices
H_k	Filter matrices
p	The amount of padding
f	The size of the filter
s	The stride
n	The number of filters
O	The size of the feature map
N	The number of neurons in the output layer.
a_i	Is the i^{th} element of the input vector
W_x^*	The new weights
W_x	The old weights
$\frac{\partial Error}{\partial W_x}$	The derivative of the error concerning weight

Chapter One
“Introduction”

Chapter one
Introduction

1.1 Introduction

Biometrics is the science of identifying individuals based on their physiological and behavioral features. Biometric identity systems, which are known as Identification based on biometric features, have gained popularity in ensuring access to physical and virtual resources and places. Biometric identification systems differ from conventional identification methods as they rely on an individual's unique characteristics to establish identity. A person may be determined by almost any physiological or behavioral trait; however, the most common biometric approaches automatically detect fingerprints, faces, iris, retina, hand geometry, voice, and signatures. In general, any physiological or behavioral trait might be used for people identification. The applications of biometrics technology are used in Government, banking, finance, consumer electronics, healthcare, transportation/logistics, and defense/security [1].

The most commonly used techniques at the present time are shown in Figure 1.1.

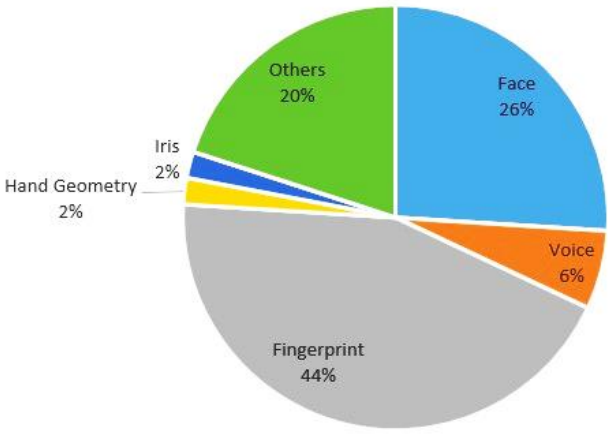


Figure 1.1 Biometric techniques.

The common characteristics of biometric technologies are outlined in Table 1.1, along with their respective performances in accuracy, user acceptability, simplicity of use, ease of implementation, and cost. According to the data in Table 1.1, voice is one of the most advantageous technologies since it is easy to use, and deploy, and customers generally accept it owing to its cheap cost. [2].

Table 1.1 An evaluation of several biometric traits.

Biometric Types	Characteristics of biometric technologies				
	Accuracy	Ease of use	User Acceptance	Ease of implementation	Cost
Voice	Medium	High	High	High	Low
Face	Low	Low	High	Medium	Low
Iris	Medium	Medium	Medium	Medium	High
Fingerprint	High	Medium	Low	High	Medium
Retina	High	Low	Low	Low	Medium
Hand geometry	Medium	High	Medium	Medium	High
Signature	Medium	Medium	High	Low	Medium

After face and fingerprint, the voice (speech signal) is one of the most widely biometrics techniques. However, the speech signal is consistently recognized as the most potent form. Aside from the speech text (word), the rich dimensions include a speaker's gender, emotion, health situation, attitude, and identity [3]. In terms of signal processing, speech can be defined as an information-carrying signal. In contrast, others define speech as speech itself. The waveform potentially serves as one of the representations of speech, and scientific research has demonstrated the

significant utility of such signals in practical contexts [3]. Speech signals can provide us with information about the following three categories: the Identification of speech text, speaker identity, and language as shown in Figure 1.2.

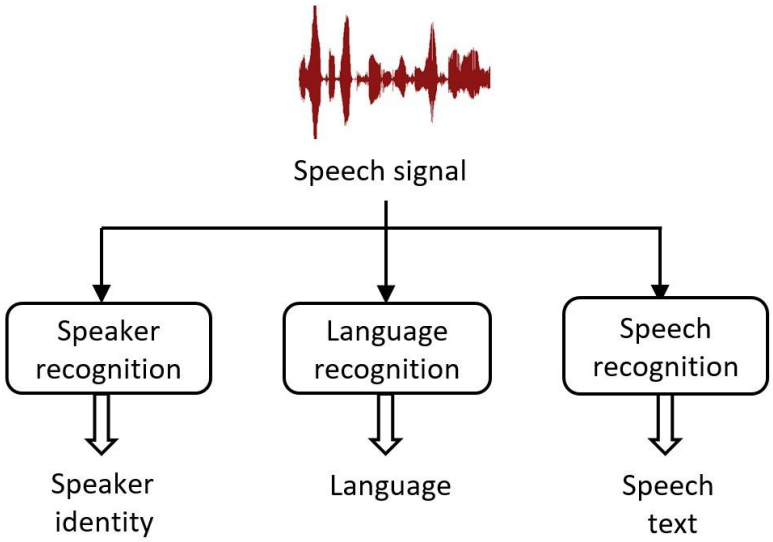


Figure 1.2 The extracted information of the speech signal.

Speech recognition, speaker recognition, and language recognition are the three main domains of recognition that can be utilized for analyzing a voice signal. These areas are often highly stimulating and have been the subject of much research for decades [3]. The primary focus of this thesis revolves around speaker recognition, particularly the speaker identification system, which commands an essential part of our academic research.

1.2 Speaker recognition

In literature, speaker recognition is commonly known as speech biometric recognition and voiceprint identification. Speaker recognition is a biometric identification system that utilizes speech parameters to determine the speaker's identity by analyzing behavioral traits present in the speech waveform [4]. Speaker recognition refers to recognizing the speaker by analyzing unique features within the speech signal. Therefore, this approach

allows the utilization of the speaker's vocal characteristics to authenticate their identity and regulate entry to various services. The services mentioned above include voice-activated dialing, telephone-based banking, telephonic shopping, database access services, information retrieval services, voice messaging, security management for limited information domains, remote computer connection, and other similar functionalities.

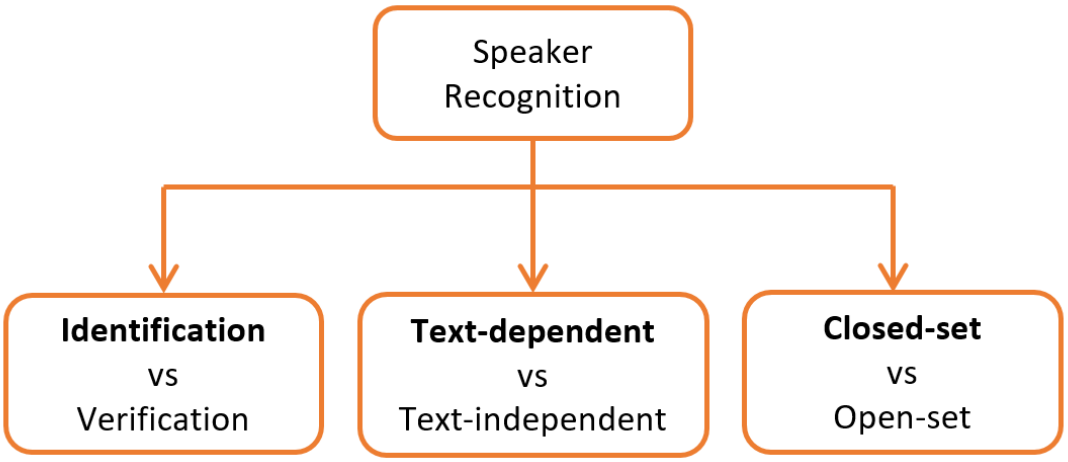


Figure 1.3 Classification of speaker recognition.

1.2.1 Speaker Recognition Classification

The speaker identification process can be divided into three different divisions depending on the type of applied criteria. Figure 1.3 explains the various approaches of speaker recognition. The following subsections provide clear explanations of the previously mentioned recognition strategies.

1) Identification and Verification

Identification and verification are two different types of speaker recognition strategies which can be explained as the following:

• Speaker identification (SI) refers to the systematic procedure of discerning the identity of an unidentified speaker by comparing their vocal characteristics with those of known individuals stored within a designated

database. This methodology is employed to ascertain the speaker's identity. The problem at hand pertains to a 1: N classification issue [5].

• Speaker verification (SV) refers to ascertaining the authenticity of a speaker's claimed identity. Additional words found in the academic literature with the same concept as SV include voice verification, voice authentication, talker verification, and speaker/talker authentication. Which is a 1:1 question of confirmation [6].

Figure 1.4 explains the main idea of SI and SV strategies.

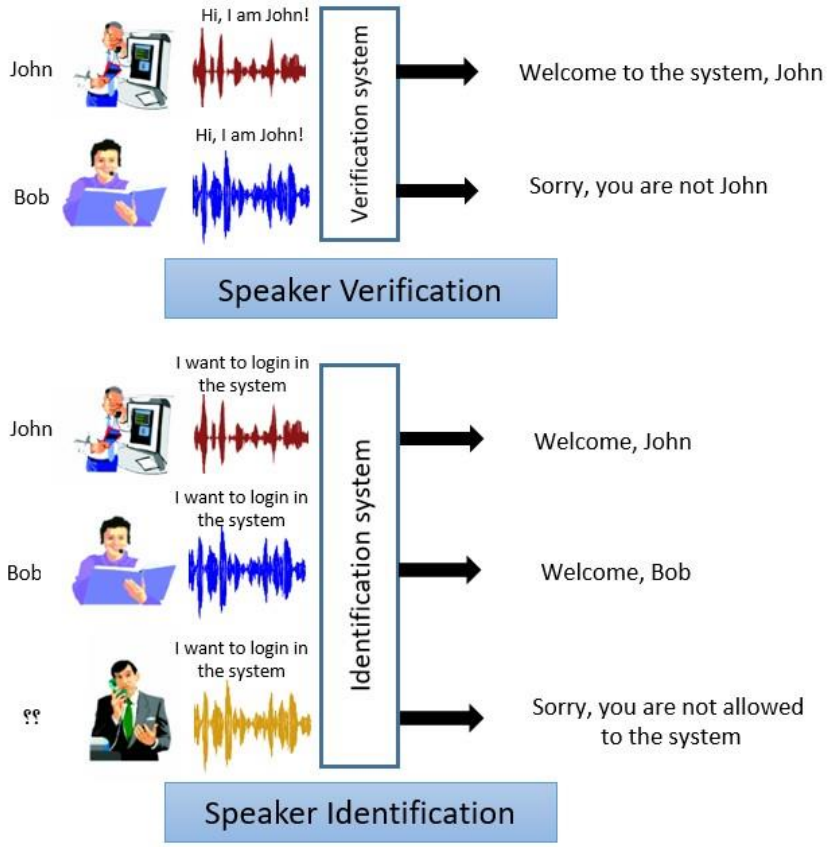


Figure 1.4 Practical examples of SI and SV.

The SI and SV speaker recognition criteria are be classified as text-independent or text-dependent, as described in the following subsection.

2) Text-independent and Text-dependent

Speaker recognition systems can also be classified according to the text type into Text-independent and Text-dependent methods. These categories are formed using the text uttered by the speakers during the identification step [7].

• Text-Dependent: Text-dependent systems are based on prior knowledge of the text to be uttered and employ the text during the training and testing phases [7].

• Text-Independent: In this type of speaker recognition method, the speakers have no prior knowledge of the training phases and can say whatever they like [7].

In contrast, text-independent speaker identification offers more convenience than text-dependent speaker recognition since it allows the speaker to communicate freely with the system without needing specific text prompts. To enhance the degree of accuracy, it is necessary to conduct more comprehensive training and testing of utterances [6].

3) Closed set and Open set

Speaker recognition systems can also be classified into closed-set and open-set systems. This type of classification is determined by the number of trained speakers available for the recognition process.

• Open set: Any number of trained speakers may be employed when utilizing an open-set system. This strategy is referred to as an open set, as the anonymous speech could originate from any one of a considerable number of distinct unknown speakers [6].

• Closed set: A closed set system pertains to a defined set of speakers. In this technique, the system determines the speaker's identity based on a collection of recorded voices [6].

A closed-set speaker identification system is studied in this thesis.

1.3 Speaker Identification System

The method of determining the identity of an unknown speaker by matching their voice to the sounds of registered speakers in a database is known as speaker identification. It is a comparison of one to many (1: N). As illustrated in Figure 1.5, the essential structure and components of speaker identification are divided into two phases: enrollment, also known as the training phase, and recognition, also familiar to the testing phase [6]. The following subsections outline the framework's primary stages.

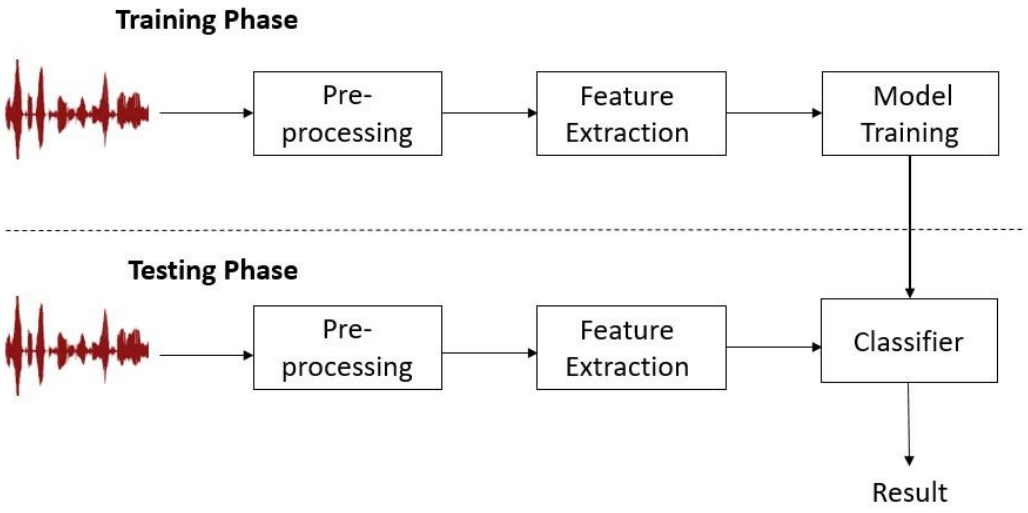


Figure 1.5 Speaker identification system.

Pre-processing: The initial stage in speech signal processing is pre-processing. The impact from noise often happens during speech recording, degrading performance. Thus, pre-processing is an essential step in creating a robust and efficient speaker recognition system, as weak pre-processing of recorded voice input will reduce classification performance. The primary goal of the pre-processing stage is to prepare the speech signal for feature extraction analysis [6].

Feature extraction: also defined as front-end pre-processing, is used in the training and testing of speaker recognition systems. The technique of conserving relevant information about a speech signal that contributes to a

speaker's identification while discarding redundant and undesired information is known as feature extraction [2]. It is used to transform speech signals into sets of feature vectors. These feature vectors include the critical qualities of the speaker's voice. The purpose of feature extraction is to minimize data size while preserving speech signal features.

Classification: is the act of recognizing, comprehending, and classifying objects and concepts into predefined categories using pre-categorized training datasets. Classification in machine learning and deep learning systems uses a diverse variety of algorithms to categorize future datasets. Classification algorithms employ input training data to estimate the likelihood or probability that subsequent data will fall into one of the established categories. One of the most prominent uses of classification is to categorize emails as "spam" or "non-spam," as employed by today's leading email service providers [2].

1.4 Literature survey

Speaker recognition is a vital aspect in ensuring optimal security measures. Due to this reason, it has garnered the attention of numerous researchers, prompting them to conduct extensive studies on the matter. Numerous publications have been produced regarding this subject matter, some are briefly explained below.

In 2014, M. D. Pawar et al. [8] proposed a text-dependent speaker identification system based on WT and ANN. Stationary Wavelet Transform was employed as a feature extraction phase. For the classification phase, Feed feed-forward propagation Neural Network was utilized. The authors collected the database from 40 speakers (20 males and 20 females) to evaluate their system. Their proposed approach was successful in achieving a recognition rate of 98%.

A speaker recognition system based on the MFCC and a neural network in a noisy environment was proposed by Paresh M. Chauhan et al. in [9]. For

the feature extraction phase, the authors utilized MFCC with a Wiener filter, an excellent filter for addressing noise in the speech signal. A Neural Network was used for the classification phase. The authors assessed their system using the NOIZEUS database, which contains six speakers, each with five samples. Their suggested approach achieved an accuracy level of 88.57%.

Tahira Mahboob et al. [10] proposed a speaker identification system based on MFCC and the GMM. The authors utilized MFCC for feature extraction, while GMM was used for the classification phase. To assess their method, the authors employed a database of 8 speakers. Their approach achieved an overall recognition rate of 87.5%.

In 2015, Juraj Kacur et al. [11] suggested a speaker recognition system that concentrates on the selection, adjustment, and overall performance of acoustic and prosodic speech parameters. For the feature extraction phase, the authors utilized several features such as PLP, MFCC, CLPC, and formant frequencies. The KNN and the GMM algorithms were applied in the classification phase. One hundred fourteen different speakers evaluated their method, and the length of each sample was around 30 sec. Their system obtained the most outstanding performance for the combination of CLPC with KNN, with a recognition rate of 99.1%.

In 2016, Amin A. Abdul Fattah et al. [12] suggested a speaker identification system based on DWT and ANN. One-dimensional Discrete Wavelet Transform was applied for the feature extraction step. For the classification step, ANN was implemented. To evaluate the system's performance, the system's authors utilized a database of 20 speakers (10 males and 10 females) with a Kurdish vocabulary. The system was able to achieve an identification rate of 80% for text-independent and a rate of 100% for text-dependent.

Faizan ur Rehman et al. [13] proposed a speaker recognition system that utilizes the MFCC and the BFCC. The authors employed MFCC and BFCC techniques to extract features from the speech signal. Vector quantization was utilized during the classification step. To evaluate their methodology, a total of twenty distinct speakers were utilized. The system demonstrated superior performance in the MFCC/VQ domain compared to BFCC/VQ.

Abrham Debasu Mengistu et al. [14] presented a speaker recognition system for the Amharic language in loud situations. A combination of MFCC, GFCC, and LPCC was employed for the feature extraction phase. Gaussian Mixture Models, Vector Quantization, and a combination of GMM and BPNN were used for the recognition phase. Their proposed system was tested using 300 speakers, and each speech sample has a length of 10sec. The combination of GMM and BPNN resulted in an identification rate of 93.7%, the best performance attainable.

A speaker identification system based on different feature extraction techniques was proposed by Avnish Bora et al. in [15]. For the feature extraction step, the authors utilized LPC, WT, and MFCC. Feed Forward Neural Network was used for the classification step. Their proposed system was tested by using the ELSDSR database which contains 22 speakers. The best recognition rate was obtained when WT and LPC were combined in the feature extraction step with an identification rate of 98%.

The authors in [16] suggested a method for a speaker identification system to overcome the disadvantages of short speech utterances. A combination of the LPC, SDC, CG, and Pitch related were employed for the feature extraction step. For the classification, a simplified version of the KNN was utilized. The authors evaluated the system by using the TIMIT database and accomplished an accuracy of 97.3 %.

The authors in 2018 [17] proposed a speaker identification system based on the multilinear decomposition of speaker adaptation. The authors

used MAP and Tensor adaptations via multilinear decomposition. GMM was utilized for the speaker recognition task. Their system was evaluated using TIMIT and NIST 2003 databases. Their system using a MAP adaptation accomplished higher rates compared with the rates obtained by Tensor adaptation.

A speaker recognition system based on MFCC and Locality Sensitive hashing was presented in 2018 by Ahmed Awais et al. [18]. The authors applied MFCC to extract the features for the feature extraction process. For the classification process, Locality Sensitive Hashing was utilized to solve the problems with an extensive database. To evaluate their system, 30 speakers (18 males and 12 females) from the TIMIT database were used. Their proposed system obtained a 92.66% recognition rate.

In 2019, Supaporn Bunrit et al. [19] suggested a text-independent speaker identification system based on CNN. The spectrogram image was used as an input to the CNN classifier. The database utilized to test their system was collected from YouTube and included five Thai-speaking people. Their suggested system obtained a classification rate of 95.83%.

A hybrid feature fusion strategy for a speaker identification system based on several feature extraction techniques and the Gaussian Mixture Model was introduced by Meixiang Dai et al. in [20]. For the feature extraction step, PLP and LPCC were used to extract the features, to gain more speech features, the PP and SC features were added. For the classification step, GMM was performed. To evaluate the proposed system, the authors used the TIMIT database which contains 630 speakers. The best performance was reached when all the features PLP, LPCC, PP, and SC were combined with a recognition rate of 94.37%.

In 2020, Minhui Qi et al. [21] proposed a speaker identification system based on a squeeze-and-excite component combined with a simplified residual convolutional neural network (ResNet). For the feature extraction

phase, MFCC was employed. A combination of squeeze-and-excitation components with a ResNet was used for the classification phase. To test their system, the authors used TIMIT and Librispeech databases. The identification rates were 95.83% and 93.92% for the TIMIT and Librispeech databases, respectively.

A speaker recognition system based on an Optimization-Based Support Vector Neural network was suggested by the authors in [22]. In the feature extraction phase, the authors employed frequency-dependent characteristics such as MKMFCC, spectral kurtosis, spectral skewness, and autocorrelation. An adaptive fractional bat-based support vector neural network (AFB-based SVNN) was used for the classification phase. The authors used the ELSDSR database to analyze their system. The results of their suggested approach attained a recognition rate of 95%.

The authors proposed a speaker identification system based on a modified Support Vector Machine as a classifier to enhance the degraded speaker identification performance for disguised voices under an extremely high-pitched condition in a neutral talking environment [23]. For the feature extraction phase, MFCC was employed. The authors used a modified SVM for the classification phase. Their suggested approach was assessed using different speech datasets, including an Arabic Emirati-accented database, the SUSAS English database, and the RAVDESS database. Their system achieved a good recognition rate.

The authors in [24] suggested a speaker identification system based on Random Forest. For the feature extraction phase, the authors used and compared two techniques, which are MFCC and Reconstructed Phase Space. Random Forest was used in the classification phase. They tested their approach on 38 speakers from the TIMIT database. The system demonstrated superior performance in the case of MFCC as compared to RPS.

The authors in [25] suggested a speaker recognition system relied on a deep learning algorithm. The authors presented two different strategies. In the first technique, the authors used the MFCC for the feature extraction phase and the CNN for the classification phase. Secondly, CNN used the row wave of the speech signal as an input to the network (RW-CNN). The authors selected 400 speakers from the VoxForge database to evaluate their system; each speaker has 10 samples. The MFCC-CNN and the RW-CNN techniques achieved a recognition rate of 96%.

Raghad Tariq Al-Hassani et al. [26] proposed a hybrid speaker identification system in 2021. Their approach was designed to have consistent speech features and achieved a high recognition rate. For the feature extraction phase, the authors relied on MFCC and improved upon it by adding a pitch frequency coefficient. The classification phase was completed with an FFNN trained using an OPSO approach. The authors collected 250 speech samples from different speakers and examined their system using 10-fold cross-validation. A recognition rate of 97.83% was achieved using their suggested approach.

In 2021, Kristiawan Nugroho et al. [27] presented a speaker identification system based on the MFCC and DNN. The authors used DA techniques to increase the database size, which included adding white noise, pitch shifting, and time stretching. For the feature extraction step, MFCC was applied. For the classification step, a seven-layer Deep Neural Network was utilized. Their proposed system was evaluated using a database of Indonesian Regional Language-301 Languages spoken in Indonesian, which includes speakers from different origins in Indonesia. Their system accomplished a 99.76% recognition rate.

Shirali Kadyrov et al. [28] suggested a speaker identification system based on Convolutional and Recurrent Neural Networks. In 2021, the authors used Long Short-Term Memory models to merge Convolutional and

Recurrent Neural Networks and construct the deep learning architecture. Their technique was evaluated using 77 different non-native speakers who read the same text in Turkish. Their method achieved a 98% recognition rate.

In 2022, Suci Dwijayanti et al. [29] proposed a speaker identification system based on a Convolution neural network. To identify speakers, the authors used a spectrogram, a graphical representation of speech in terms of raw features. These features were fed into a CNN, and the speakers were identified using a CNN-visual geometry group (CNN-VGG) architecture. To evaluate the proposed system, the authors used 78 speakers; each speaker has ten samples. Their system achieved a 98.78% identification rate.

In 2022, Fadwa Abakarim et al. [30] suggested a speaker recognition system based on adaptive orthogonal transformations. The authors developed an adaptive operator to extract essential features from input signals with a minimum dimension. For the classification step, DTW was used. The authors used a database containing 10 speakers to test their system. The highest accuracies accomplished using the technique proposed in [30] were 96.8% and 98.1% when using the Fourier transform and correlation method as a compression approach, respectively.

Saqlain Hussain Shah et al. [31] presented a two-branch network to enhance the speaker identification system's performance. This network could extract features from both the face and the speech signals. The VGGFace and VGGVox subnetworks were utilized, respectively, to extract features from face and voice. The author used SVM as a classifier. The authors used the VoxCeleb1 database, a large-scale database consisting of audio-visual human speaking recordings taken "in the wild" from YouTube, to evaluate their system. The authors found that by including information about the user's face in their system, the system performance improved to 97.2% accuracy.

1.5 Thesis Objectives

This thesis intends to accomplish the following goals primarily:

1. Investigate different preprocessing techniques, such as data augmentation, silent removal, and temporal duration splitting, to achieve discriminant features during the feature extraction phase.
2. Employ various feature extraction transforms to preprocessing features to accomplish high system performance while simultaneously reducing the number of data dimensions and the level of system complexity.
3. Apply a convolutional neural network (CNN) deep learning model for the classification phase, which is preferred over other machine learning and deep learning methods. CNNs are highly effective due to their ability to autonomously learn and extract pertinent features for speech signals. Additionally, they can effectively manage substantial volumes of data and demonstrate computational efficiency.
4. One of the objectives of this thesis is to conduct training on the database using varying time durations, to investigate the impact of these durations on the overall performance of the system.

1.6 Thesis organization

The thesis is organized in the following manner:

Chapter One: This chapter provides an overview of biometric technologies and speaker recognition systems, including the classification of each, in addition to the literature review, the aim of the thesis, and the thesis organization.

Chapter Two: This chapter presents a concept of the speaker recognition system, which comprises three primary components, as well as applications of speaker recognition and elements affecting the speaker recognition

system. Also, describe the methods used for the stage of feature extraction and the phase of classification.

Chapter Three: This chapter introduces the proposed system. This chapter incorporates various methodologies employed in the thesis. Furthermore, the databases used for the implementation of the proposed system.

Chapter Four: In Chapter Four, the performance parameter is introduced, and the experimental results of the ALU-AC, ELSDSR, RAVDESS, and TIMIT databases are presented based on the methods employed in this thesis.

Chapter Five: In this chapter, the conclusion and future works, are presented.

Chapter Two

“Background Methodologies Algorithms”

Chapter Two
Background Methodologies Algorithms

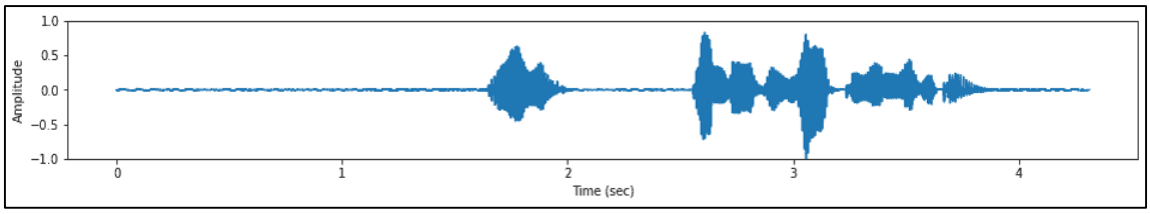
2.1 Introduction

Various applications throughout this chapter's discussion. An analysis of all the factors that may effect on the speaker recognition system. The preprocessing, feature extraction, and classification phases are identified as the three primary parts of the speaker identification system that will be proposed in this thesis. Describes the idea of the feature extraction techniques that are included in this thesis, such as Mel-frequency Cepstral Coefficient, Discrete Wavelet Transform, Principle Component Analysis, and Discrete Multi-Wavelet Transform feature. Additionally, CNN's deep learning concept applies to the classification phase.

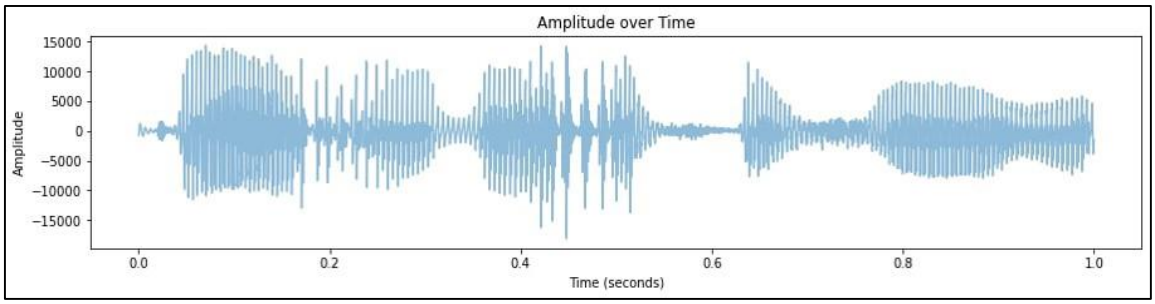
2.2 Silence Remove

The separation of the voiced area from the silence/unvoiced component of the recorded signal, known as pre-processing, is often regarded as an essential first step in creating an accurate speaker identification system. This is because the voiced portion of speech signals contains the majority of the speech or speaker-specific attributes and extracting the expressed portion of speech signals by marking and removing the silence and unvoiced region substantially reduces computational complexity at later stages [32,33].

Calculating the highest value for each frame and comparing it to a predetermined threshold value is how silence is removed from an audio file. If the maximum value for the one frame is more than the threshold, then the frame should be inserted as a speech frame; otherwise, the frame should be deleted as a silent frame [32]. Figure 2.1 illustrates an example of the speech signal both before and after the removal of silence.



(a)



(b)

Figure 2.1 The speech signal, (a) before and (b) after removing silence.

2.3 Mel Frequency Cepstral Coefficient (MFCC)

In the MFCC feature extraction procedure, discrete Fourier transforms are used as the foundation for extracting features. For speech processing, the MFCC algorithm is the one that is utilized the most frequently since it is seen to be adequate for presenting signals and features. The MFCC's operation captures sound and frequency signals in the same way as people detect sound features [34]. The MFCC process diagram is depicted in the Figure 2.2.

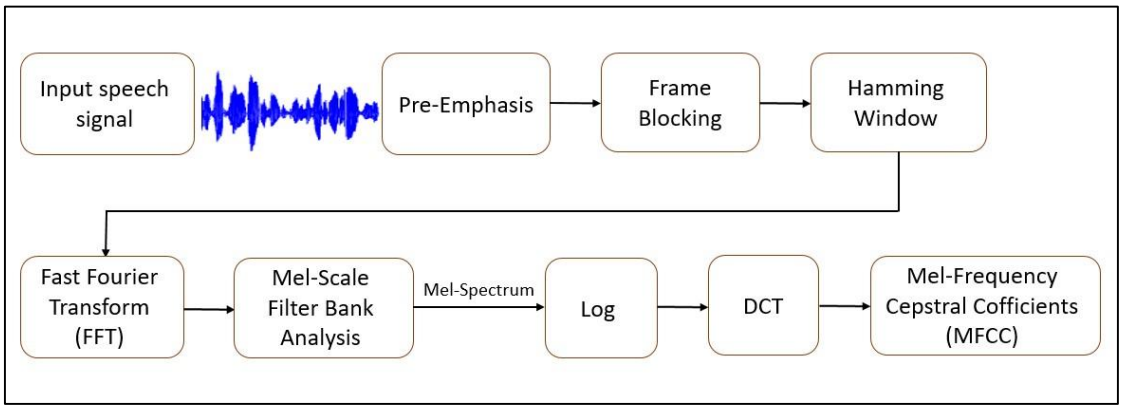


Figure 2.2 MFCC block diagram.

The following sections describe the process steps of the MFCC method, as seen in Figure 2.2.

2.3.1 Pre-emphasis

To achieve a flattening of the signal spectrum, an FIR high-pass filter with the transfer function described in the below Equation is utilized [35].

$$Y[n] = X[n] - \alpha X[n-1] \quad (2.1)$$

where α is most frequently seen within the range of 0.9 to 1. When the sound is transmitted through the lips, the high frequencies of the speech signal that was created in the vocal tract are lowered in volume. It is possible to generate a spectrum with a more even distribution of high-frequency and low-frequency information by reducing the amount of low-frequency information in the final speech signal [35].

2.3.2 Segmentation and Overlapping

During this stage, the continuous speech signal is segmented into frames of N samples, with an M (M < N) gap between each pair of consecutive frames. The very first frame is made up of the very first N samples. After M samples have passed, the second frame begins, overlaps the first frame with the N-M samples, and so on. This method will continue until the speech is accounted for inside a single frame or many frames. Typical values for N and M are N = 256 and M = 100 [35].

The speech signal cannot be termed a long-term stable signal since its qualities change dramatically. However, suppose such a signal is considered in a very short time (order of milliseconds). In that case, speech features do not vary much, and it may be termed a quasi-stationary signal. The movement of the articulators, which change position to create distinct phonemes, causes this lack of stability. The articulator organs move from one place to another during the transition between two phonemes. The waveform signal reflects the fact that this transition is not immediate. These

transitions are generally challenging to analyze in speech, especially when a speech frame is focused on that transition. Frame overlapping can be used on the voice signal to avoid this effect. Segmentation is required to separate the voice signal into small enough frames with quasi-stationary elements. Each frame will be evaluated independently and used to produce a feature vector [34].

The speech signal is windowed using a window size equal in length to the final frames to be obtained. This allows for segmentation as well as overlapping to take place. After a frame has been gained, the window is made longer to get the subsequent frame. The frame's duration, in conjunction with the window delay, will be used to determine the percentage of overlap. It seems like the most intuitively appropriate shape for windows would be a rectangle. On the other hand, as will be seen in a later section, feature vectors will be acquired from the frame spectrum, and there are other kinds of windows, such as the Hamming one, which provide superior spectral performance [34, 35].

2.3.3 Windowing

A speech signal is often segmented into temporal chunks using this method, which is a technique that is utilized in signal processing. The signal in the actual world has no connection to the signal occurring on the edges of the signal that is repeating itself. The windowing function is a smooth function that, results in zero when applied to extreme values. The goal is to make it appear like there is no break in the boundaries. There is signal change after applying the window function, but the influence on signal statistics is reduced. After applying the window function. The Hamming Window function is the one that is typically utilized in MFCC as the window function. It ensures that there is no break in continuity between the starting point and the finishing point in the frame. The Hamming window $W(n)$ is expressed as [36]:

$$W(n) = 0.54 - 0.46\cos(2\pi n/(N-1)) \quad (2.2)$$

where N is the number of samples in each frame, $W(n)$ = Hamming window, the signal's output is given as.

$$Y(n) = X(n) \times W(n) \quad (2.3)$$

where $X(n)$ is the input signal and $Y(n)$ is the output signal.

After determining which hammering window to use, the next question that has to be asked is, "What is the appropriate size for the window, W?" There is a tension here between the temporal and spectral resolutions. The window length is directly correlated with the spectral resolution, which is measured by how small the main segment is. Increasing the window size, however, results in the loss of stationary properties associated with short speech frames [36].

A reasonable compromise would be to pick a window with a duration between 10 and 20 milliseconds.

2.3.4 Fast Fourier Transform (FFT)

The FFT (Fast Fourier Transform) is a mathematical algorithm that decomposes a signal into sinusoidal components consisting of real and imaginary units. The FFT is commonly employed for frequency analysis, facilitating voice processing due to its resemblance to the auditory perception of humans. The Fast Fourier Transform (FFT) algorithm carries out the discrete Fourier transform (DFT). The discrete Fourier transform, sometimes known as DFT, converts data from the time domain to the frequency domain and is expressed as [37].

$$X(n) = \sum_{k=0}^{N-1} x_k e^{-2\pi jkn/N} \ , n = 0,1, \dots, N-1 \quad (2.4)$$

where: N= The number of frame segments, x_k is an aperiodic sequence with N value, k=0,1,2,3,…, N-1, and j = -1.

2.3.5 Mel-Scale Filter Bank

The frequency range represented by the FFT spectrum is quite broad, and the speech signal does not adhere to a linear scale. After that, the Mel scale-based filter bank depicted in Figure 2.3 is carried out. A series of triangle filters are depicted in this figure. These filters are utilized in computing a weighted sum of filter spectral components to ensure that the result of the procedure is comparable to a Mel scale. The magnitude frequency response of each filter has the shape of a triangle, and it is equal to unity at the center frequency. The magnitude frequency response of two neighboring filters decreases linearly to zero at the center frequency. After that, the output of each filter is the sum of the spectral components it has filtered. After that, the Mel is calculated by using equation (2.5) for a given frequency f expressed in Hz [38].

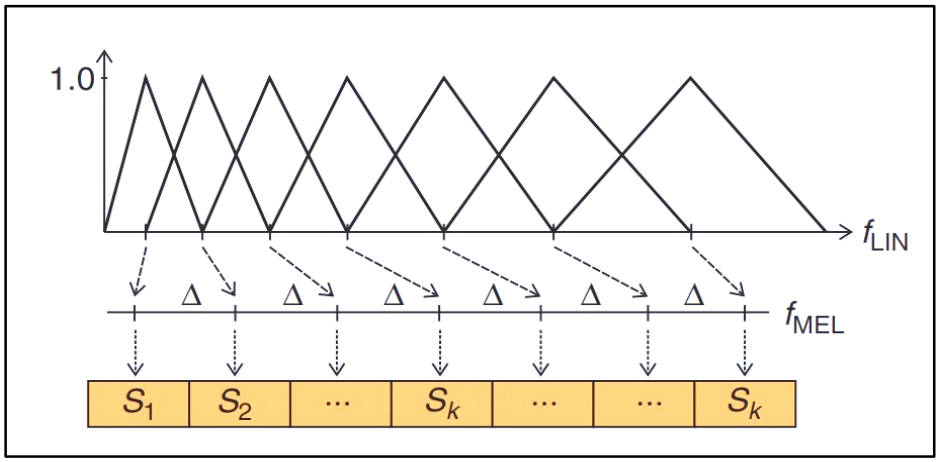


Figure 2.3 Mel scale filter bank [39].

$$f_{Mel} = 2595\, log_{10}\left(1 + \frac{f_{LIN}}{700}\right) \tag{2.5}$$

where f_{LIN} is the frequency in Hertz and f_{Mel} is the frequency in Mel scale. The numbers 2595 and 700 are constants used in the Equation to convert a frequency in Hertz to the Mel scale. Specifically, 2595 is the

constant that scales the logarithmic term and 700 is the frequency in Hertz that corresponds to 1000 Mel

2.3.6 Log

To obtain logarithm values, it is necessary to convert the DFT values into a singular value. By replacing each base logarithmic value, it is possible to reduce the overall value of the Mel filter bank. The logarithmic value of the mel filter needs to be extracted from the sound segment. The human ear requires less sensitivity to perceive sounds with low amplitude and frequency [36].

2.3.7 Discrete Cosine Transform (DCT)

The log Mel spectrum will be converted back to the time domain in this last step. The resulting quantities are referred to as the Mel frequency cepstrum coefficients. In the context of the analysis of the given frame, the cepstral representation of the speech spectrum offers a reliable illustration of the local spectral characteristics of the signal. The Mel spectrum coefficients, and by extension their logarithm, are real numbers; consequently, the Discrete Cosine Transform can be used to convert them to the time domain. The formula for DCT can be expressed as C_n [40].

$$C_n = \sum_{k=1}^{k} (logs_k) \cos\left(n \left(k - \frac{1}{2}\right)\frac{\pi}{k}\right) \quad (2.6)$$

where C_n denotes the coefficient of the cepstrum mel frequency, s_k is the coefficient of the mel power, and $n = 1,2, \dots, k$.

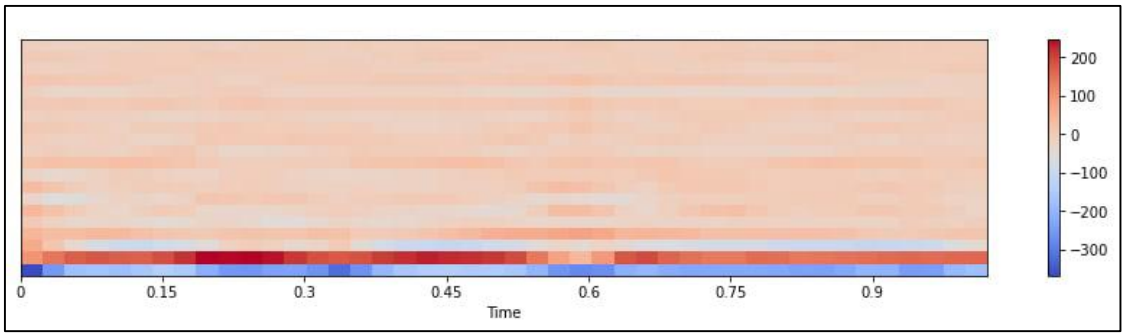


Figure 2.4 The cepstrum of MFCC.

2.4 Principal Component Analysis (PCA)

One of the most important discoveries to come out of applied linear algebra is something known as principal component analysis (PCA). Because it is a straightforward and non-parametric approach to extracting useful information from complicated datasets, principal component analysis (PCA) is widely utilized in many types of investigation, ranging from neurology to computer graphics. PCA gives a road map for how to reduce a complicated dataset to a lower dimension to show the often-hidden. These simpler dynamics often underlie it [41].

The challenges of conducting multivariate data analysis are referred to as increasing the dimensionality. While working with such enormous amounts of data requires dimensionality reduction, dimensionality reduction aims to represent the data in a lower-dimensional space while maintaining some of the data's original qualities [42].

PCA is a technique for the unsupervised reduction of dimensionality that also goes by the names discrete Karhunen–Loève transform (KLT), Hotelling transform singular value decomposition (SVD), and empirical orthogonal function (EOF). PCA is an approach to data analysis that aims to minimize the dimension of the data by identifying a small number of orthogonal linear combinations (the principle components PCs) of the variables that contributed the most variation to the database [42]. The number of primary components (PCs) corresponds to the total initial variables. For many databases, the first few principal components (PCs) explain most of the variation; hence, ignoring the remaining PCs might result in a modest loss of information. The purpose of principal component analysis (PCA) is to do the following:

(1) identify the most significant information from the data table.

(2) Reduce the size of the database by retaining just the most essential information.

(3) Simplify the description of the database.

(4) Analyze the structure of the observations and the variables.

Figure 2.5 depicts the transformation of the original data from their actual space (R^M) to the space of principal components analysis PCA (R^K). As a result of this, the PCA method is regarded as an orthogonal transformation because it utilizes orthogonal PCs [43].

The PCA is calculated in two steps depending on the method for obtaining the covariance matrix. In the first step of the process, the covariance matrix of the data matrix (X) is computed. In the second step of the process, the eigenvalues and eigenvectors of the covariance matrix are computed [43]:

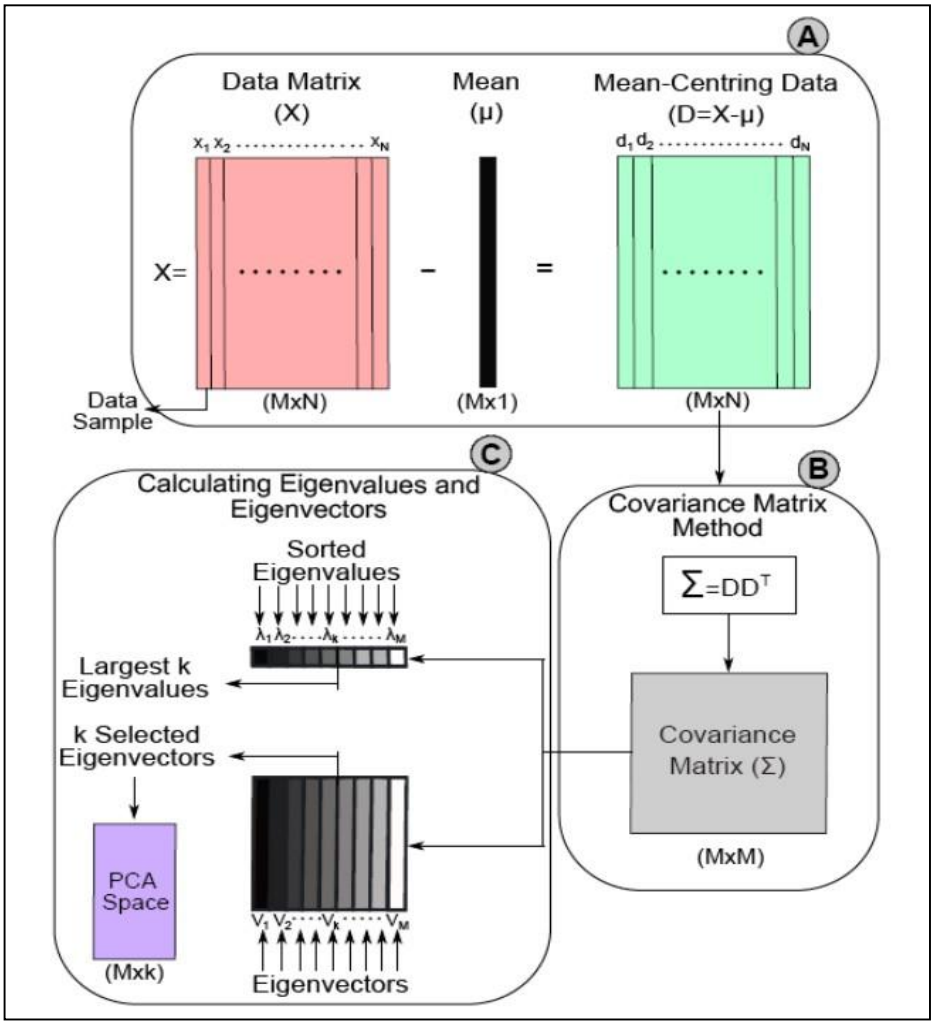


Figure 2.5 Covariance matrix method to calculate PCA [43].

To calculate the PCA space using the covariance matrix method, follow these steps [42, 43]:

1. Subtract each variable's mean from the data points. This is also known as data centering. Each variable's mean can be determined by averaging all of the data points for that variable [43].

$$\mathrm{D} = X - \mu \tag{2.7}$$

Where D denotes mean-centered data, X represents the data matrix and μ represents the mean.

2. Compute the covariance matrix of the data points that are centered. The covariance matrix is a square matrix containing the variances and covariance of every possible pair of variables in the data set [42].

$$\sum = DD^T \tag{2.8}$$

3. Calculate the covariance matrix's eigenvectors and eigenvalues. The eigenvectors represent the directions in which the data varies the most, whereas the eigenvalues represent the variance in each direction [42].

$$C = \lambda_1 > \lambda_2 > \cdots \lambda_M$$
$$C = V_1 > V_2 > \cdots V_M$$

4. Sort the eigenvectors in descending order by their corresponding eigenvalues. The eigenvector with the greatest eigenvalue indicates the direction in which the data varies the most, etc. [42]
5. To create a new subspace, choose the top k eigenvectors that map to the k greatest eigenvalues. The PCA space is the name given to this subset [42].

2.5 Wavelet Transform

In order to derive additional information from a signal that is not readily available in its unprocessed state, mathematical modifications are applied to the signal. Many different types of transformations may be utilized, but the Fourier transforms are perhaps the oldest common type of transformation.

The Fourier Transform (FT) is a mathematical technique that decomposes a signal into its constituent frequencies. It was first introduced by Joseph Fourier in 1822. Because the Fourier coefficients of the altered function indicate each sine's and cosine function's contribution at each frequency, the signal may then be evaluated for its frequency content [44].

The Discrete Fourier Transform (DFT) is a variant of the FT that is used to analyze discrete-time signals. If f(t) is a non-periodic signal, then the signal cannot be adequately represented by adding the sine and cosine functions since they are periodic. It is possible to lengthen the signal to make it periodic artificially; however, doing so will need more excellent continuity at the signal's ends [45]. One solution to the issue of adequately expressing non-periodic signals is the windowed Fourier transform (WFT). The Windowed Fourier Transform (WFT) is a mathematical technique that is used to analyze signals that are non-stationary, meaning that their frequency content changes over time. It was first introduced by Gabor in 1946 [44].

The Wavelet Transform (WT) is a mathematical technique that decomposes a signal into wavelets, which are small waves that are localized in both time and frequency. It was first introduced by Morlet in 1984 [46].
The wavelet transform is a mathematical tool used to analyze signals and data in a way that is similar to the Fourier transform. However, unlike the Fourier transform, which uses a fixed set of basis functions (sine and cosine waves), the wavelet transform uses a set of basis functions that are localized in both time and frequency. This allows for a more detailed analysis of signals that have both high and low-frequency components [45, 47].

There are two types of wavelet transforms: continuous wavelet transform (CWT) and discrete wavelet transform (DWT). The major difference between the two is the CWT is a continuous transform that is used to analyze continuous-time signals, while the DWT is a variant of the CWT that is used

to analyze discrete-time signals [46]. In the scope of this thesis, the DWT will be used.

2.5.1 The Discrete Wavelet

Using the wavelet transform in its current form presents three properties that render its direct application in the equation (2.9) format challenging.

$$\text{(CWT)} \qquad \Psi_x^{\psi}(s,\tau) = \frac{1}{\sqrt{s}} \int x(t)\psi^* \left(\frac{t-\tau}{s}\right) d(t) \tag{2.9}$$

where $x(t)$ is the signal being analyzed, ψ is the analyzing wavelet, s is the scale parameter, and τ is the translation parameter.

One issue to consider is the redundancy of the Continuous Wavelet Transform (CWT). The wavelet transform is computed by iteratively translating a continuously scalable function across a given signal and subsequently determining the correlation between the two. This process is described in (2.9). The scaled operation will not form an orthogonal basis, and as a result, the wavelet coefficients obtained will exhibit a high degree of redundancy [45].

Even though the CWT doesn't have any redundant parts, the wavelet transform can still handle an endless number of wavelets. It would be better to lower this number to a more doable level. At the moment, we are dealing with a second issue. Another issue is the need for more in-depth answers for the wavelet transforms that are used in many functions [49]. Most of the time, the CWT is not thought to be the best way to describe signals because its time-bandwidth product is the square of the signal's time-bandwidth product. Most of the time, these applications try to show the signal with as few parts as possible. Discrete wavelets have been used to help with these problems. Using discrete wavelets for continuous scaling and translation is not possible because scaling and translation need to be done in discrete steps [48, 49].

To get a genuine orthonormal basis, one must first discretize the time-scale parameters, τ, s acceptably, and then choose the proper mother wavelet $\Psi(t)$. The most logical approach is to discretize the scaling variable s in a logarithmic manner $\left(s = s_0^{-j}\right)$ and then discretize τ at each given scale $\left(\tau = ks_0^{-j}T\right)$ using the Nyquist sampling method, which is based on the spectrum of the function $x(t)$. The wavelet functions that are produced as a consequence are as follows [45]:

$$\Psi_{j,k}(t) = s_0^{j/2}\Psi\left(s_0^j t - k\tau_0\right) \tag{2.10}$$

In Equation (2.10), the variables j and k represent integers. If the initial value s_0 is close to one and the parameter t is sufficiently small, the wavelet functions exhibit an over-complete nature. Consequently, signal reconstruction takes place within a set of non-restrictive constraints. In contrast, in cases where the sampling is limited, such as when the calculation is performed octave by octave with a starting point of s_0 = 0, it is essential to note that a genuinely orthonormal basis can only be obtained for specific choices. The reason for this is that the computation is performed octave by octave. Taking into account the fact that wavelet functions are assumed to be orthonormal [45]:

$$\int \Psi_{j,k}(t)\Psi_{m,n}(t)dt = \begin{cases} 1 & if\ j = m\ and\ k = n \\ 0 & otherwise \end{cases} \tag{2.11}$$

In discrete-time conditions, Equation (2.11) is commonly employed with an initial value of $s_0 = 2$, and the calculation is performed incrementally. In this instance, basic scaling and translation generate the wavelet expansion system. The mother wavelet, denoted by Ψ(t)the generating wavelet symbol, yields the subsequent two-dimensional parameterization of $\Psi_{j,k}(t)$ [45].

$$\Psi_{j,k}(t) = 2^{j/2}\Psi\left(2^j t - k\right) \tag{2.12}$$

The factor $2^{j/2}$ in Equation (2.12) serves the purpose of normalizing every wavelet, thereby ensuring a consistent norm that remains unaffected by the scale j. In this instance, the discretization interval τ has been standardized to a value of one, and it is presumed to be equivalent to the sampling interval of the discrete signal $(\tau = k2^{-j})$. Multiresolution conditions are a requirement for all wavelet systems to be considered valid. The filter-bank algorithm, structured in a tree format, can be utilized to derive the lower-resolution coefficients from the higher-resolution coefficients. The discrete wavelet transform (DWT) is commonly referred to as such in the literature on wavelet transforms [45].

2.5.2 The Scaling and Wavelet Functions

The scaling function used in the Discrete Wavelet Transform (DWT) serves as a low-pass filter, allowing the decomposition of a signal into its constituent frequency components. The scaling function, abbreviated as $\phi(t)$, is mathematically defined as the integral of the low-pass filter $h(t)$ across the whole time domain t. The wavelet function $\psi(t)$ is mathematically described as the result of subtracting one scaling function from another, with one of the scaling functions being displaced by a certain amount. The discrete wavelet transform (DWT) employs a combination of wavelet and scaling functions to partition a signal into its constituent frequency components [49,50].

The scaling function can be expressed mathematically as:

$$\Phi(t) = \sum_k h(k)\sqrt{2}\ \Phi(2t - k) \tag{2.13}$$

where $h(k)$ is the low-pass filter, $\phi(t)$ is the scaling function, and $\sqrt{2}$ maintains the norm of the scaling function with the scale of two. The wavelet function can be expressed as [50]:

$$\Psi(t) = \sum_k g(k)\sqrt{2}\ \Phi(2t - k) \tag{2.14}$$

where $g(k)$ is the high-pass filter, and $\Psi(t)$ is the wavelet function.

The wavelet function $\Psi(t)$ is derived from the subtraction of two scaling functions, with one of them being subjected to a certain shift.

2.5.3 A DWT Computation Method

Figure 2.6 illustrates how the DWT may be used in the implementation. This diagram shows two different stages of the decomposition process. Low-pass and high-pass filters are denoted by the letters h and g, respectively, corresponding to the coefficients $h(n)$ and $g(n)$. A decimation, also known as a down-sampling by two, is represented by arrows heading downward. It is possible to use this splitting, filtering, and decimation process several times on the scaling coefficients to produce the two-scale structure. The spectrum of $X[n]$ was split into a low-pass band and a high-pass band during the initial step of the process, which led to the scaling coefficients and wavelet coefficients at lower scales. When it reaches the second stage, that low-pass band is subdivided into an even lower band of low-pass and a high-pass band [45].

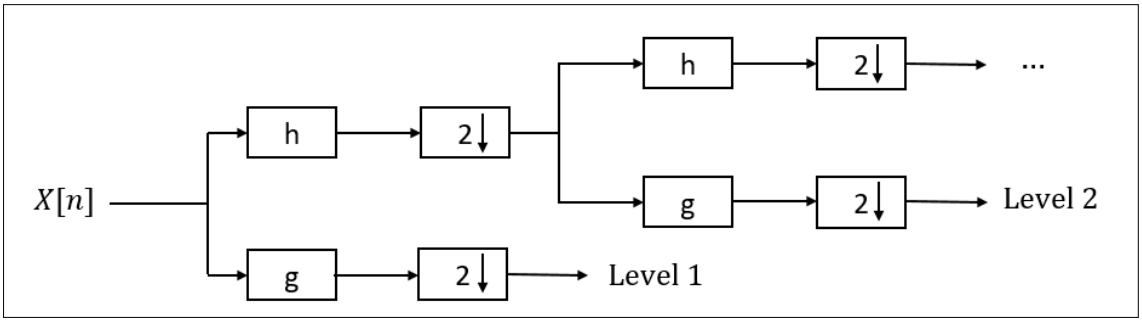


Figure 2.6 The filter bank for calculating the wavelet coefficients.

Consider the following transformation matrix for length-2 when performing the discrete wavelet transform (DWT) [45]:

$$T=\begin{bmatrix} h(0) & h(1) & 0 & 0 & \cdots & \cdots & & \\ 0 & 0 & h(0) & h(1) & \cdots & \cdots & & \\ \vdots & \vdots & \vdots & \vdots & \cdots & \cdots & & \\ 0 & 0 & 0 & 0 & \cdots & \cdots & h(0) & h(1) \\ g(0) & g(1) & 0 & 0 & \cdots & \cdots & & \\ 0 & 0 & g(1) & g(1) & \cdots & \cdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \cdots & \cdots & & \\ 0 & 0 & 0 & 0 & \cdots & \cdots & g(0) & g(1) \end{bmatrix} \quad (2.15)$$

as well as the transformation matrix for length-4 which is as follows:

$$T=\begin{bmatrix} h(0) & h(1) & h(2) & h(3) & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & h(0) & h(1) & h(2) & h(3) & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ h(2) & h(3) & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & h(0) & h(1) \\ g(0) & g(1) & g(2) & g(3) & \vdots & \vdots & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & g(0) & g(1) & g(2) & g(3) & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & g(0) & g(1) & g(2) & g(3) \\ g(2) & g(3) & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & g(0) & g(1) \end{bmatrix} \quad (2.16)$$

Here blank entries imply zeros. When it appears at the transform matrices of the scalar wavelet, which are presented in equations (2.15) and (2.16), respectively, one can see that the first row creates one component of the data that is convolved with the low-pass filter coefficients $(h(0), h(1), ...)$. Similarly, the second, third, and other rows are located in the top part of the matrix. The rows in the bottom part of the matrix carry out a different kind of convolution, using high pass filter coefficients $(g(0), g(1), ...)$. The general operation of the matrix is thus to carry out two related convolutions, then to decimate each of them by half (throw away half the data), and finally to interleave the remaining halves of the convolutions.

The transform matrices are as follows [45]:

$$T=\begin{bmatrix} h(0) & h(1) & 0 & 0 & \cdots & \cdots & & \\ 0 & 0 & h(0) & h(1) & \cdots & \cdots & & \\ \vdots & \vdots & \vdots & \vdots & \cdots & \cdots & & \\ 0 & 0 & 0 & 0 & \cdots & \cdots & h(0) & h(1) \\ h(1) & -h(0) & 0 & 0 & \cdots & \cdots & & \\ 0 & 0 & h(1) & -h(0) & \cdots & \cdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \cdots & \cdots & & \\ 0 & 0 & 0 & 0 & \cdots & \cdots & h(1) & -h(0) \end{bmatrix} \quad (2.17)$$

$$T=\begin{bmatrix} h(0) & h(1) & h(2) & h(3) & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & h(0) & h(1) & h(2) & h(3) & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ h(2) & h(3) & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & h(0) & h(1) \\ h(3) & -h(2) & h(1) & -h(0) & \vdots & \vdots & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & h(3) & -h(2) & h(1) & -h(0) & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & h(3) & -h(2) & h(1) & -h(0) \\ h(1) & -h(0) & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & h(3) & -h(2) \end{bmatrix} \quad (2.18)$$

It is helpful to conceive of the filter $(h(0), h(1), h(2), h(3), ...)$ as a smoothing filter denoted by the letter H, analogous to a moving average calculated using four points. The filter is denoted by the letters G and$(h(3), -h(2), h(1), -h(0), ...)$ is not a smoothing filter because of negative signs. H and G are referred to as quadrature mirror filters when used in signal-processing situations. The values are selected in such a way as to ensure that G returns a response of zero whenever it is presented with a suitably smooth data vector. Consequently, the information that is "smooth, approximately" in the data is correctly represented by the output of H, which has been decimated by half. The information referred to as the "detail" of the data is the output of G, which is likewise decimated [45].

For such a characterization to be of any value, it must be feasible to recreate the initial data vector of length N from its approximate and its detail components, each of which has $N/2$ elements. To do this, the matrices must be orthogonal since the inverse of an orthogonal matrix is just the transposed matrix [45].

$$T2=\begin{bmatrix} h(0) & 0 & \cdots & 0 & h(1) & 0 & \cdots & 0 \\ h(1) & 0 & \cdots & 0 & -h(0) & 0 & \cdots & 0 \\ 0 & h(0) & \cdots & 0 & 0 & h(1) & \cdots & 0 \\ 0 & h(1) & \cdots & 0 & 0 & -h(0) & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & h(0) & 0 & 0 & 0 & h(1) \\ 0 & 0 & 0 & h(1) & 0 & 0 & 0 & -h(0) \end{bmatrix} \quad (2.19)$$

$$T2=\begin{bmatrix}
h(0) & 0 & 0 & \cdots & \cdots & h(2) & h(3) & 0 & 0 & \cdots & 0 & h(1)\\
h(1) & 0 & 0 & \cdots & \cdots & h(3) & -h(2) & 0 & 0 & \cdots & 0 & -h(0)\\
h(2) & h(0) & 0 & \cdots & \cdots & 0 & h(1) & h(3) & 0 & \cdots & 0 & 0\\
h(3) & h(1) & 0 & \cdots & \cdots & 0 & -h(0) & -h(2) & 0 & \cdots & 0 & 0\\
0 & h(2) & h(0) & \cdots & \cdots & 0 & 0 & h(1) & h(3) & \cdots & 0 & 0\\
0 & h(3) & h(1) & \cdots & \cdots & 0 & 0 & -h(0) & -h(2) & \cdots & 0 & 0\\
0 & 0 & h(2) & \cdots & \cdots & 0 & 0 & 0 & h(1) & \cdots & 0 & 0\\
0 & 0 & h(3) & \cdots & \cdots & 0 & 0 & 0 & -h(0) & \cdots & 0 & 0\\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\
 & & & 0 & & & & & & & 0 & 0\\
 & & & 0 & h(0) & & & & & & h(3) & 0\\
 & & & 0 & h(1) & & & & & & -h(2) & 0\\
0 & 0 & 0 & 0 & h(2) & & & & & & h(1) & h(3)\\
0 & 0 & 0 & 0 & h(3) & & & & & & -h(0) & -h(2)
\end{bmatrix}$$

(2.20)

After completing all of the necessary requirements, no degrees of freedom are available for a length-2. The requirements are as follows [45]:

$$\begin{cases} h(0)+h(1)=\sqrt{2} \\ h^2(0)+h^2(1)=1 \end{cases} \tag{2.21}$$

It can only be realized in a certain way by:

$$h_{D2}=\{h(0),h(1)\}=\left\{\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right\} \tag{2.22}$$

These are the coefficients of the Haar scaling function, which are also the coefficients of the length-2 Daubechies scaling function.

One degree of freedom, also known as one parameter, for the length-4 coefficients sequence. This parameter supplies all of the coefficients that are suitable for the requirements that have been specified [45]:

$$\begin{cases} h(0)+h(1)+h(2)+h(3)=\sqrt{2} \\ h^2(0)+h^2(1)+h^2(2)+h^2(3)=1 \\ h(0)h(2)+h(1)h(3)=0 \end{cases} \tag{2.23}$$

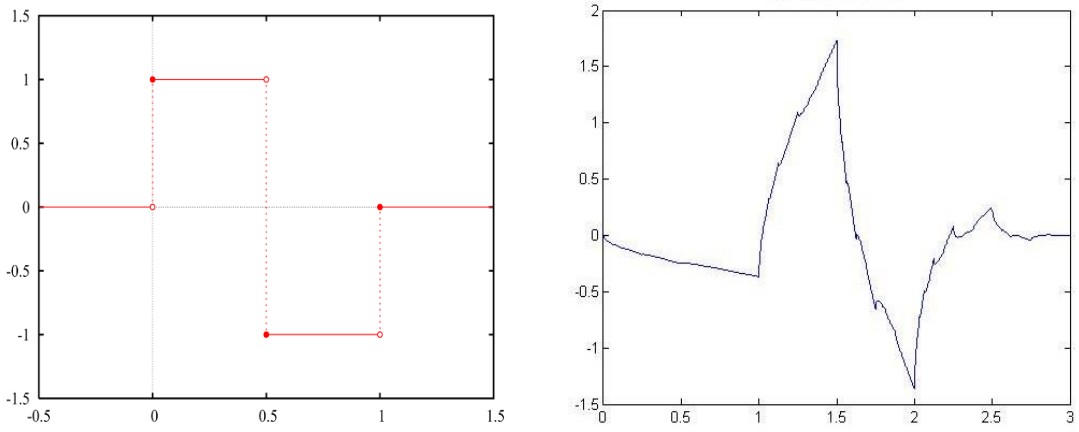
If we assume that the parameter is the angle α, then the coefficients are as follows:

$$\begin{aligned} h(0)&=(1-\cos(\alpha)+\sin(\alpha))/(2\sqrt{2}) \\ h(1)&=(1+\cos(\alpha)+\sin(\alpha))/(2\sqrt{2}) \\ h(2)&=(1+\cos(\alpha)-\sin(\alpha))/(2\sqrt{2}) \\ h(3)&=(1-\cos(\alpha)-\sin(\alpha))/(2\sqrt{2}) \end{aligned} \tag{2.24}$$

In addition, these equations provide the length-2 Haar coefficients (2.22) for the values $0, \pi/2, 3\pi/2$ and a degraded condition for the value equal to $\alpha = \pi$. When we solve for $\alpha = \pi/3$, we get the Daubechies coefficients. The form of these Daubechies-4 coefficients is much more apparent than others.

$$h_{D4} = \left\{\frac{1+\sqrt{3}}{4\sqrt{2}}, \frac{3+\sqrt{3}}{4\sqrt{2}}, \frac{3-\sqrt{3}}{4\sqrt{2}}, \frac{1-\sqrt{3}}{4\sqrt{2}}\right\} \quad (2.25)$$

Haar and Daubechies wavelets can see in Figure 2.7.



(a) (b)

Figure 2.7 Wavelet family, (a) Haar wavelet and (b) Daubechies wavelet.

2.5.3.1 Calculation of the DWT for a Two-Dimensional Signal

A wavelet transform that can separate the two-dimensional signals is used to do the analysis. A 2-D separable transform is analogous to a pair of 1-D transforms performed sequentially. It is implemented as a one-dimensional row transform, and then a one-dimensional column transform is performed on the data acquired from the row transform. Figure 2.8 depicts the construction of the filter bank that is used to compute a 2-D DWT [51].

There are two primary types of algorithms for calculating discrete wavelet transforms for two-dimensional signals. These types include separable algorithms and non-separable algorithms. Methods that may be separated out simply apply the same process to each dimension. The standard method involves first processing each row in the order that they appear,

followed by processing each column of the result. Ways that cannot be separated into separate stages function simultaneously in both picture dimensions. Non-separable techniques are often more challenging to put into practice, even though they may have advantages over separable approaches, such as a reduction in the amount of computing required [51].

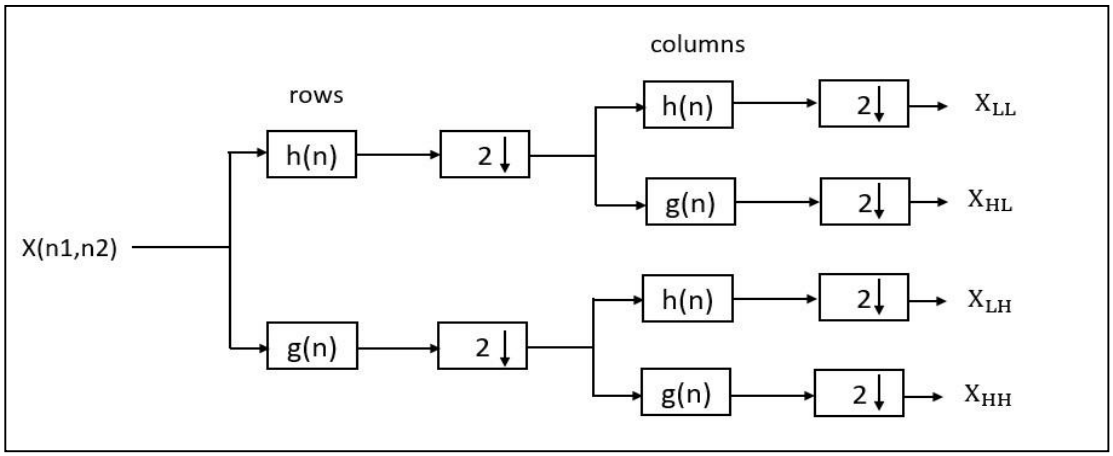


Figure 2.8 One-level filter bank for the 2-D discrete wavelet transform computation.

and Figure 2.9 displays the results of DWT's 1-level 2-D of decomposition. The LL band is repeatedly decomposed for many decomposition layers, which leads to a pyramid structure for the sub-bands, with the coarsest sub-band at the top and the finest sub-band at the bottom.

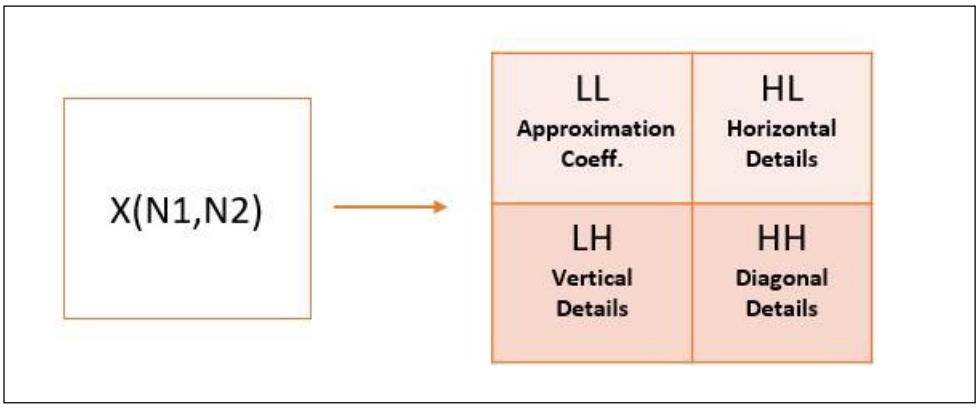


Figure 2.9 Output of 1-level 2-D decomposition.

Within the scope of this thesis, the separable method has been implemented; as a result, only separable methods will be explained.

2.5.3.1.1 The computation of the DWT for a Two-Dimensional Signal Applying the Separately Method

A 2-D separable transform is analogous to two 1-D transforms that are performed in sequence. It is implemented as a one-dimensional row transform, and then a one-dimensional column transform is performed on the data acquired from the row transform. To calculate a DWT for a 2-D signal using the separable approach, the following step has to be carried out [45]:

1. Checking the input dimensions: The input matrix has to be of length $N \times N$, with N being a power of two.
2. Generate a transformation matrix by using the transformation matrices provided in equations (2.17) and (2.18).
3. Transformation of input rows by applying matrix multiplication to the transformation matrix that was generated using the $N \times N$ input matrix.
4. The transformation of the input columns may be carried out in the following manner:
    i. Perform a transposition on the row-transformed $N \times N$ matrix produced in step 3.
    ii. Implement the $N \times N$ column matrix to perform matrix multiplication on the $N \times N$ produced transformation matrix.

2.6 Discrete Multi-Wavelet Transform

The wavelet transform is a signal transform based on multi-resolution analysis (MRA). It is used in computer vision, signal processing, pattern recognition, and image processing [52]. As a natural extension of wavelet, multi-wavelet is meant to be concurrently symmetric, orthogonal, and have short supports with high approximation power. This is something wavelets that only use one scaling function cannot do at the same time since they only use one scaling function. The solution is to increase the number of scaling

operations rather than relying on only one scaling function to improve the approximation power. It improves the overall performance of many wavelet applications [53]. The DWT is created with the assistance of two main functions: the wavelet function $\Psi(t)$ and the scaling function $\Phi(t)$.

In contrast to DWT, the DMWT uses several wavelet and scaling functions. This analysis method is also based on multi-resolution analysis. A vector notation may be used to denote the collection of scaling processes [53].

$$\Phi(t) = [\Phi_1(t), \Phi_2(t), \dots, \Phi_r(t)]^{\mathrm{T}} \quad (2.26)$$

Whereas $\Phi(t)$ is referred to as a multi-scaling function, the multi-wavelet function is defined by the collection of wavelet functions.

$$\Psi(t) = [\Psi_1(t), \Psi_2(t), \dots, \Psi_r(t)]^{\mathrm{T}} \quad (2.27)$$

Wavelets are referred to as scalar wavelets or just wavelets when r = 1. although, in theory, r may be any considerable value. Most studies done on multi-wavelets to this far have concentrated on r = 2. The following equations may be used to express multi-wavelet scaling functions $\Phi(t)$, as well as wavelet functions $\Psi(t)$ [54].

$$\Psi(t) = \sqrt{2}\ \sum_{k=-\infty}^{\infty} G_k\,.\,\Phi(2t-k) \quad (2.28)$$

$$\Phi(t) = \sqrt{2}\ \sum_{k=-\infty}^{\infty} H_k\,.\,\Phi(2t-k) \quad (2.29)$$

where G_k and H_k are the $r \times r$ filter matrices for each integer k. Compared to a standard scalar wavelet, the matrix components of these filters give more freedom. The valuable properties of the multi-wavelet filters, such as their orthogonality, symmetry, and a high degree of approximation, may be relied on.

Geronimo, Hardian, and Massopust proposed a useful multiwavelet filter that they came up with and called GHM. The GHM filter provides a combination of essential characteristics for signal processing, such as orthogonality and symmetry, in addition to its compact support. According

to equations (2.28) and (2.29), the GHM two scaling functions and the wavelet functions satisfy the following two-scale dilation equations [55]:

$$\begin{bmatrix}\Phi_1(t)\\ \Phi_{2(t)}\end{bmatrix} = \sqrt{2}\ \sum_k H_k \begin{bmatrix}\Phi_1(2t-k)\\ \Phi_2(2t-k)\end{bmatrix} \tag{2.30}$$

$$\begin{bmatrix}\Psi_1(t)\\ \Psi_2(t)\end{bmatrix} = \sqrt{2}\ \sum_k G_k \begin{bmatrix}\Psi_1(2t-k)\\ \Psi_2(2t-k)\end{bmatrix} \tag{2.31}$$

where H_k in the GHM system are four scaling matrices $H_0, H_1, H_2,$ and H_3 [54].

$$H_0 = \begin{bmatrix}\frac{3}{5\sqrt{2}} & \frac{4}{5}\\ -\frac{1}{20} & -\frac{3}{10\sqrt{2}}\end{bmatrix} \qquad H_1 = \begin{bmatrix}\frac{3}{5\sqrt{2}} & 0\\ \frac{9}{20} & \frac{1}{\sqrt{2}}\end{bmatrix}$$

$$H_2 = \begin{bmatrix}0 & 0\\ \frac{9}{20} & -\frac{3}{10\sqrt{2}}\end{bmatrix} \qquad H_3 = \begin{bmatrix}0 & 0\\ -\frac{1}{20} & 0\end{bmatrix} \tag{2.32}$$

Also, G_k in the GHM system are four wavelet matrices $G_0, G_1, G_2,$ and G_3 [54].

$$G_0 = \begin{bmatrix}-\frac{1}{20} & -\frac{3}{10\sqrt{2}}\\ \frac{1}{10\sqrt{2}} & \frac{3}{10}\end{bmatrix} \qquad G_1 = \begin{bmatrix}\frac{9}{20} & -\frac{1}{\sqrt{2}}\\ -\frac{9}{10\sqrt{2}} & 0\end{bmatrix}$$

$$G_2 = \begin{bmatrix}\frac{9}{20} & -\frac{3}{10\sqrt{2}}\\ \frac{9}{10\sqrt{2}} & -\frac{3}{10}\end{bmatrix} \qquad G_3 = \begin{bmatrix}-\frac{1}{20} & 0\\ -\frac{1}{10\sqrt{2}} & 0\end{bmatrix} \tag{2.33}$$

To design the wavelet and scaling function for the GHM filter, the iteration strategy is specified in equations (2.28) and (2.29). This approach may be found below. On the other hand, in this particular case, two wavelet functions

and two scaling functions are obtained from two box functions, as shown in Figure 2.10 [56].

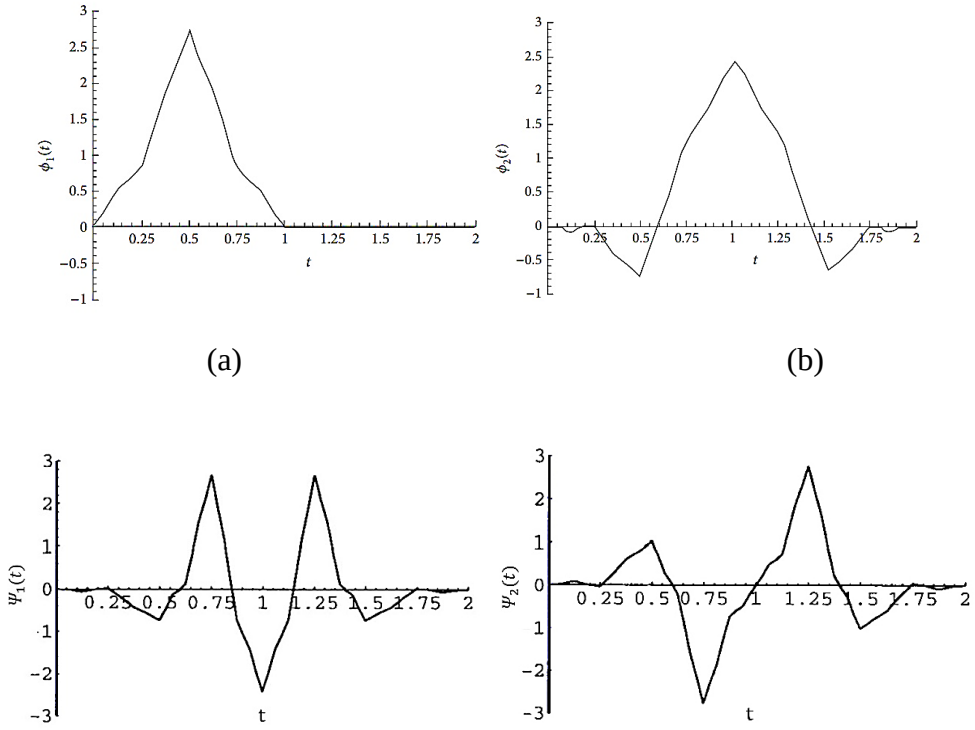


Figure 2.10 GHM pair of $\Phi(t)$ and $\Psi(t)$, (a) is $\Phi_1(t)$, (b) is $\Phi_2(t)$, (c) is $\Psi_1(t)$, and (d) is $\Psi_2(t)$.

Since the low-pass filter and the high-pass filter in the multi-wavelet filter bank are both 2×2 matrices, the convolution process requires the filters to multiply vectors (instead of scalars) when the input data is supplied to the low-pass filter and the high-pass filter. Because of this technique, multi-filter banks need two input rows; this is another challenge to overcome when using multi-wavelet transform. The act of transforming an input signal with a scalar value into an output signal with the proper vector value is referred to as pre-processing.

2.6.1 Preprocessing of DMWT

Pre-processing is a mapping procedure implemented with prefilter in the analysis phase. In the synthesis stage, a corresponding postfilter

operation occurs; this operation precisely reverses the effects of the prefilter. There are two available pre-processing techniques: oversampling (repeated rows) and critical sampling (approximation-based scheme). In oversampling pre-processing, the input data is multiplied by a constant and repeated. Pre-processing with oversampling doubles the number of input data symbols and raises the computational difficulty of the transform. The input signal is pre-processed in critically sampled pre-processing to obtain two vectors [54]. There are two approximation methods for critically sampled pre-processing: the first-order approximation method and the second-order approximation method.

Within the scope of this thesis, the critical sampling scheme-preprocessing based on the first order of approximation has been implemented; as a result, only critical sampling will be explained.

A first-order approximation is a method for approximating multi-wavelets that have been critically sampled. Following is a summary of how first-order approximation-based preprocessing operates (every two rows generate two new rows) as in (a) and (b) [55].

a- For any odd-row:

$$new\ odd-row = (0.373615)[same\ odd-row] + (0.11086198)[next\ even-\ row] + (0.11086198)[previous\ even-row] \quad (2.34)$$

b- For every even-row:

$$new\ even-row = \left(\sqrt{2}-1\right)[same\ even\ row] \quad (2.35)$$

When calculating the value of the first odd row in Equation (2.34), it is crucial to remember that the value of the preceding even row is zero. Likewise, when calculating the value of the final odd-row in Equation (2.34), the value of the subsequent even-row is zero.

To compute 2D-DMWT applying approximation-based preprocessing, the steps outlined in Figure 2.11 should be followed.

1. Checking the dimensions of the input matrices The length of the input matrix should be$N \times N$, where N is equal to 2^a. To produce a square matrix, it may be necessary to add zeros to individual rows or columns of the input matrix if it is not already square.
2. Constructing a transformation matrix H: The transformation matrix is represented by equation (2.36) [54, 55].

$$H = \begin{bmatrix} H_0 & H_1 & H_2 & H_3 & 0 & 0 & ... \\ G_0 & G_1 & G_2 & G_3 & 0 & 0 & ... \\ 0 & 0 & H_0 & H_1 & H_2 & H_3 & ... \\ 0 & 0 & G_0 & G_1 & G_2 & G_3 & ... \\ ... & ... & ... & ... & ... & ... & ... \end{bmatrix} \quad (2.36)$$

where H_i represents the impulse response of the low-pass filter and G_i represents the impulse response of the high-pass filter. Both the low-pass and high-pass matrices for the GHM filter are provided for you in Equations (2.32) and (2.33), respectively. These matrices should produce a transformation matrix of dimensions $N/2 \times N/2$. After that, modify the GHM matrix filter coefficients to correspond to the values provided by the subsequent matrix [55].

$$H = \begin{bmatrix} H_0 & H_1 & H_2 & H_3 & 0 & 0 & ... & 0 & 0 & 0 & 0 \\ 0 & 0 & H_0 & H_1 & H_2 & H_3 & ... & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ H_2 & H_3 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & H_0 & H_1 \\ G_0 & G_1 & G_2 & G_3 & \vdots & \vdots & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & G_0 & G_1 & G_2 & G_3 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & G_0 & G_1 & G_2 & G_3 \\ G_2 & G_3 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & G_0 & G_1 \end{bmatrix} \quad (2.37)$$

3. Preprocessing rows: This is accomplished by utilizing Equations (2.34) and (2.35) for the odd-even rows of the input N×N matrix, respectively, for the 1st-order approximation-based preprocessing. After row preprocessing, there will be no change to the initial matrix's dimension, which is $N \times N$.

4. The transformation of rows is possible by using the following method:
    I. Use matrix multiplication to apply Equation (2.37) to the preprocessed matrix with $N \times N$ rows.
    II. To perform a permutation on the rows of the resultant N by N matrix, first place the row pairs $1, 2$ and $5, 6 \ldots, N - 3, N - 2$ after each other in the top half of the rows of the resulting matrix, and then position the row pairs $3, 4$ and $7, 8, \ldots, N - 1, N$ below them in the following bottom half of the rows.
5. Perform pre-processing on the columns: To perform the procedure used in preparing the rows.
    I. It was transposing the row-transformed $N \times N$ matrix that was constructed in step 4.
    II. Repeat step 3 on the $N \times N$ matrix to produce the N×N columns preprocessed matrix.
6. Transformation of columns: To transform the columns, the following change is applied to the $N \times N$ column preprocessed matrix.
    I. Carry out matrix multiplication between the obtained matrix and the transformation matrix H.
    II. II. Perform a permutation on the rows of the resultant $N \times N$ matrix by first inserting the row pairs 1, 2 and 5, 6... N-3, N-2 after each other in the top half of the matrix, and then placing the row pairs $3, 4$ and $7, 8 \ldots N - 1, N$ below them in the following bottom half.

7. The final transformed matrix is as follows: To produce the final converted matrix, it is necessary to complete the following steps in order:

I. Once the column transformation step has been finished, the formed matrix has to have its rows inverted.

II. The resultant transpose matrix should then have the coefficients permuted applied to it.

III. The resultant DMWT matrix has the same dimensions ($N \times N$) as the input matrix when approximation-based pre-processing is utilized.

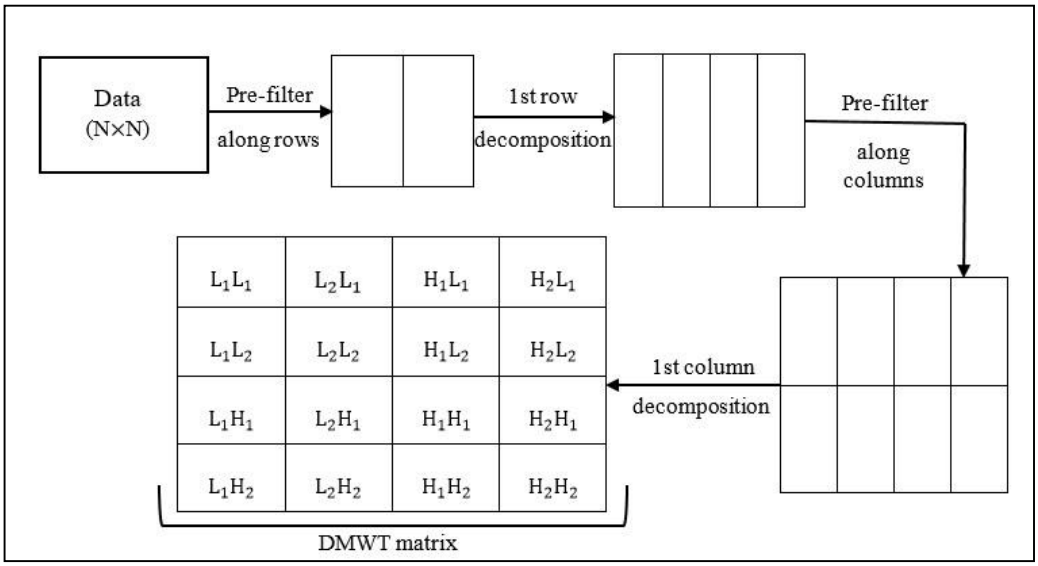


Figure 2.11 A single level of decomposition of 2D-DMWT.

As seen in Figure 2.10, the approximation-based pre-processing was used in the production of the final DMWT matrix yielded the same dimensions ($N \times N$) as the initial input matrix ($N \times N$).

2.6.2 Iteration of Decomposition

In the same way, as scalar wavelet theory is founded on the concept of multi-resolution analysis (MRA), the multi-wavelet idea does the same thing. The two-dimensional data is replaced with four blocks corresponding to the sub-bands representing either lowpass or high-pass filtering in both dimensions when a single level of decay is performed using a scalar wavelet transform as seen in Figure 2.12 (a). In this figure, the labels for the sub-bands explain how the data for that sub-bands were produced. For instance,

the information in the sub-band LH was made by first applying a high-pass filter to the rows and then applying a lowpass filter to the columns. Because the multi-wavelets have r scaling functions, and because the multi-wavelets utilized here have two channels (r=2), there will be two sets of scaling coefficients. There will also be two sets of wavelet coefficients.

As a result of the fact that it is desirable to do numerous iterations over the lowpass data, the scaling coefficients for the two channels are saved together [56, 57]. Similarly, the wavelet coefficients for the two channels are kept together in the location. Figure 2.12 (b) illustrates the multi-wavelet decomposition sub-bands for your reference. When dealing with multi-wavelets, the L and H labels each include subscripts that indicate the channel that the data refers to. For instance, the data from the second channel high-pass filter in the horizontal direction corresponds to the sub-band that is labeled L_1H_2, and the data from the first channel low-pass filter in the vertical direction corresponds to that sub-band.

(a) (b)

Figure 2.12 The output after a single level of decomposition, for (a) scalar wavelet and (b) multi-wavelet.

The figure above demonstrates how a single level of the decomposition process is carried out. In actual practice, more than one decomposition may be conducted on the data simultaneously. Iterations are carried out successively on the low pass coefficients from the stage before this one to decrease the total number of low pass coefficients even more [56].

2.7 Classification (Convolutional Neural Network (CNN))

CNN, also known as Convolutional Neural Network, is a highly renowned and extensively utilized algorithm in deep learning. In contrast to its predecessors, CNN has a notable advantage in its ability to autonomously identify significant attributes without requiring human supervision. CNNs have been widely utilized in diverse domains, such as computer vision, audio processing, and face recognition. The architecture of CNNs resembles conventional neural networks, as it draws inspiration from the neural structure observed in human and animal brains. The visual cortex in the feline brain consists of a complex arrangement of cells. This specific arrangement is what CNN aims to emulate [58].

The CNN network was formed based on sparse interaction, parameter sharing, and equivariant representation [59]. Convolutional neural networks differ from traditional fully connected (FC) networks by utilizing shared weights and local connections to optimize the analysis of 2D input-data structures. This procedure utilizes a limited number of parameters, facilitating the training process and enhancing the efficiency of the network's execution. The observed phenomenon is analogous to the cellular processes within the visual cortex. It is worth mentioning that these cells can perceive specific portions of a scene rather than the entire scene. In other words, they spatially extract the local correlation in the input, functioning as local filters over the input [58].

A commonly employed type of CNN involves arranging multiple convolution layers, which are subsequently followed by sub-sampling (pooling) layers. This configuration's final layer type typically comprises fully connected layers (FC).

2.7.1 Advantages of using CNN

The following is a list of the advantages that using CNNs in the computer vision environment offers in comparison to using other traditional neural networks [58]:

1) The primary advantage of CNN is its weight-sharing feature, which reduces the total number of trainable network parameters. As well as, in turn, assists the network in improving its generalization and preventing overfitting.
2) Because the model is learning the feature extraction layers and the classification layer simultaneously; the consequent result of the model is both highly methodical and highly reliant on the extracted features.
3) Compared to other neural networks, CNN's ease of use makes it much simpler to implement large-scale networks.

2.7.2 CNN layers

The architecture of CNN is made up of a few different levels, which are also referred to as multi-building blocks. Below is a detailed explanation of each layer in the CNN design, along with a description of each layer's purpose.

1. The Convolutional Layer: The most crucial aspect of a CNN's architecture is its design. The composition consists of a collection of convolutional filters, often called kernels. The process of convolving the input data, represented as N-dimensional metrics, with these filters leads to generating the output feature map [58].
    - Definition of the kernel: the kernel is represented by a grid of discrete numbers or values. The term "kernel weight" refers to each value. At the beginning of the CNN training process, the kernel weights are initially represented by random numbers that have been allocated. In addition, a variety of approaches may be used to perform the initialization of the weights. Next, throughout each

training epoch, these weights are altered; as a result, the kernel learns to extract essential features from the data [60].

- The initial steps of the convolutional operation include a description of the CNN input format. The input for the classic neural network is an image in the vector format, whereas the input for the CNN is an image that contains several channels. Let's look at an example of a 4×4 grayscale picture that has a 2×2 random weight-initialized kernel so that we can have a better understanding of the convolutional process [60]. To begin, the kernel will move over the whole picture in horizontal and vertical directions. Furthermore, the dot product is computed between the input picture and the kernel, whereby the relevant values are multiplied and afterward aggregated to get a singular scalar value, all performed simultaneously. The process above is repeated until the absence of more space for sliding is seen. Note that the output's feature map might be interpreted based on the generated dot product values. Figure 2.13 provides a graphical representation of the significant computations that are carried out at each stage. In this diagram, the 2×2 kernel is represented by the light green color, and the light blue color represents the equivalent-sized region of the input picture. Both are multiplied, and the final result after adding up the values of the products from the multiplication, provides an input value to the output feature map. The resultant product values are highlighted with a bright orange color. However, no padding is added to the input image in the preceding example. Padding enables you to regulate the size of the output. The convolution of input decreases the output size, resulting in information loss. We pad the input volume with zeros at the border to prevent this. Valid convolution and the same convolution are two popular options. The valid

convolution has no padding, and the same convolution has the same output size as the input size [61].

While a stride of one is added to the kernel (which is specified for the chosen step size total vertical or horizontal places). Take into consideration the fact that you may potentially utilize a different stride value. In addition to this, raising the stride value results in the production of, raising the stride value results in producing a feature map with reduced dimensionality [58].

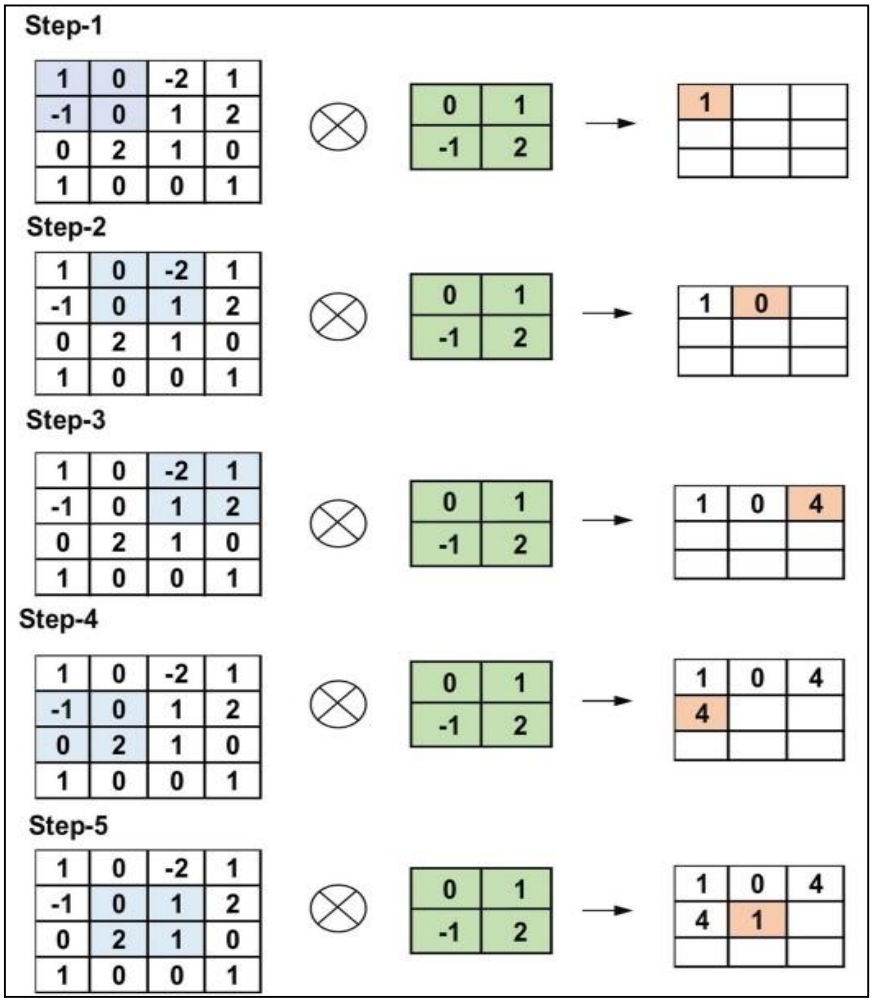


Figure 2.13 Providing examples for the first five steps of the convolution process [58].

The following formula is used to determine the size of the output [61]:

$$O = \frac{n+2p-f}{s} + 1 \tag{2.38}$$

Where n is the number of filters, p represents the amount of padding, f represents the size of the filter, and s represents s the stride.

The primary benefits of convolution layers include:

i. Sparse Connectivity: In a fully connected neural network, every neuron in one layer is coupled to every neuron in the subsequent layer. In a convolutional neural network (CNN), the number of weights between each pair of layers is restricted. As a consequence of this, the quantity of connections or weights required is minimal. As a result, the storage capacity needed for these weights is also very little, demonstrating an efficient use of memory resources. Moreover, it should be noted that the computational expense associated with a matrix operation in a Convolutional Neural Network (CNN) is much more significant compared to that of a dot (.) function, as highlighted in reference [58].

ii. Weight Sharing: is a characteristic of CNNs where there are no dedicated weights between individual neurons in adjacent layers. In contrast, each weight is applied to every pixel inside the input matrix. On the contrary, conventional neural networks provide weights to each pixel. This enables a substantial reduction in training time and associated costs. Instead of acquiring novel weights for each neuron, it is possible to acquire a single set of weights that applies to all inputs [60].

2. Pooling Layer: The sub-sampling of the feature maps is the primary responsibility of the layer that pools the data. To produce these maps, one must first complete the convolutional processes. this method reduces the size of enormous feature maps to build feature maps on a smaller scale. During this time, it keeps the vast majority of the dominant information (or characteristics) intact at each stage of the pooling process. Before the pooling process is carried out, an initial size assignment is given to both the stride and the kernel in a way analogous

to how the convolutional operation is carried out. Many different pooling algorithms are available, and they may be used in various pooling layers. The tree pooling method, the gated pooling method, the average pooling method, the min pooling method, the max pooling method, the global average pooling technique, and the global max pooling method are all included in these approaches. The most common and frequent strategy is known as Max Pooling [62]. The most significant disadvantage of the pooling layer is that it might sometimes bring the overall performance of CNN down. The explanation for this may be found in the fact that the pooling layer enables CNN to determine if a specific feature is present in the input picture that was sent to it without worrying about its precise location [60].

3. Activation Functions (Non-Linearity): In any neural network-based model, the primary responsibility of any activation function is to map the input to the output. The input value is produced by computing the weighted sum of the neuron's input and then adding bias (if there is a bias). The activation function generates the output corresponding to the given input to determine whether or not a neuron will fire in response to the input. After each learnable layer in a CNN architecture (layers with weights, such as the convolutional and FC layers), non-linear activation layers are used as the next step in the process. The CNN model can learn more complicated things because of the non-linear behavior of those layers, which also allows it to map inputs to outputs in a non-linear fashion. An activation function has to be differentiable for error backpropagation to be used when it is being used to train a model [63]. This is a crucial aspect of the function. Following is a description of some activation functions used most often in deep neural networks (including CNN).

- Sigmoid: Real numbers are used as input for the sigmoid activation function, and the function's output is bound to fall somewhere in the range [0,1]. The sigmoid function generates a 'S-shaped curve when plotted, as shown in Figure 2.14. The sigmoid curve may be represented mathematically as follows [60, 63]:

$$f(x)_{sigm} = \frac{1}{1+e^{-x}} \tag{2.39}$$

- Tanh: As shown in Figure 2.14, the Tanh activation function is used to bind the input values (real numbers) inside the range of [-1, 1]. The expression below is the mathematical representation of Tanh [60, 63]:

$$f(x)_{tanh} = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{2.40}$$

- ReLU: In CNN, the activation function known as the Rectifier Linear Unit (ReLU) is often used. Its purpose is to change all of the values that are entered into positive numbers, as can be seen in Figure 2.14. The benefit of using ReLU is that, compared to other algorithms, it needs a very minimum amount of compute burden. The ReLU expression, when represented mathematically, looks like this [60, 63]:

$$f(x)_{ReLU} = \max(0, x) \tag{2.41}$$

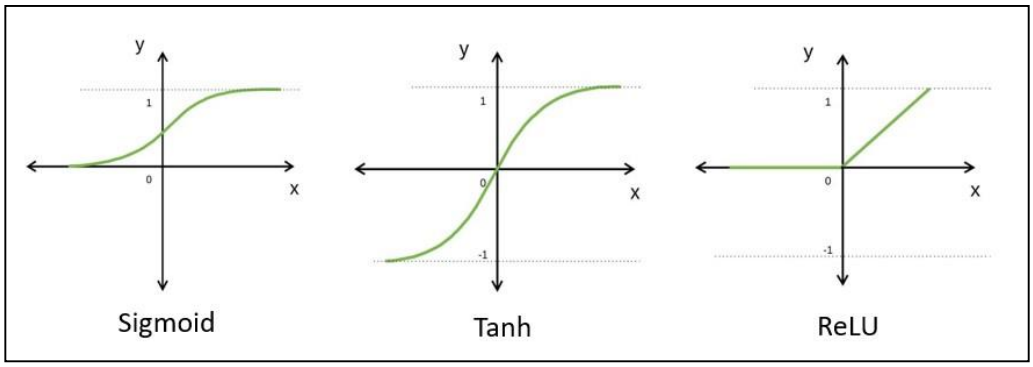


Figure 2.14 Common types of Non-Linearity.

4. Fully Connected (FC) Layer: Fully connected (FC) layers are often used as the concluding component or layers of CNN architectures for classification. Inside these hierarchical levels, every individual neuron inside a given layer is interconnected with each neuron from the

preceding layer. The CNN design's output layer, also known as the classifier, is the Fully-Connected layer that comes last in the stack. The Fully-Connected Layers are a sort of feed-forward artificial neural network (ANN), following the idea of the traditional multi-layer perceptron neural network (MLP). The FC layers get their input from the last convolutional or pooling layer, which is a collection of metrics (feature maps). The metrics undergo a flattening process to generate a vector upon receiving this input. This vector is then sent into the fully connected (FC) layer to generate the ultimate output of the convolutional neural network (CNN) [58].

5. Loss Functions: The previous section introduced many different types of layers that are used in CNN architecture. In addition, the output layer, the last layer of the CNN architecture, is responsible for producing the final classification. The CNN model employs several loss functions in its output layer to calculate the expected error incurred throughout the training dataset. The actual result is significantly different from the one that was expected, as seen by this mistake. It will go through the CNN learning process to be optimized. Nevertheless, to determine the error, the loss function makes use of two parameters. The CNN estimated output, also known as the prediction, is the initial value of the first parameter. The actual output, often known as the label, is the subject of the second parameter. A wide variety of problems call for using different kinds of loss functions. The following is a condensed explanation of the most popular loss function that is used in this thesis [58, 60].
   - ❖ Cross-Entropy or Soft-Max Loss Function: The log loss function, also called cross-entropy loss, is widely used for evaluating the performance of the CNN model. The output of the CNN model is represented by the probability p, which belongs to the set $\{0,1\}$. This function is often seen as a substitute for the squared error loss

function in multi-class classification problems. Softmax activations in the output layer are employed to generate output that conforms to a probability distribution. This distribution is denoted as i.e., $p, y \in R^N$, where p represents the likelihood of each output category, and y signifies the desired output. Using the technique described in [60] will accurately determine the probability of each output classification.

$$p_i = \frac{e^{a_i}}{\sum_{k=1}^{N} e^{a_k}} \quad (2.42)$$

Where N represents the number of neurons in the output layer, whereas e^{a_i} represents the unnormalized output of each neuron.

2.7.3 Optimizer selection

Optimizers are algorithms or approaches that are used to either maximize the effectiveness of production or minimize the impact of an error function (loss function). Optimizers are mathematical functions that rely on a model's learnable parameters (weights and biases). Optimizers assist in determining how the weights and learning rate of a neural network should be altered to minimize losses [64, 65].

Before mentioning the various optimizers, it is essential to first get familiar with some terminologies on deep learning.

- Epoch: The term "epoch" denotes the cumulative number of iterations in which the algorithm is applied to the training dataset.
- Batch: The word "batch" refers to the quantity of data required for parameter updates in a model.
- Learning rate: The learning rate is a parameter that provides the model with a scalar value to ascertain the extent to which the model weights should be adjusted.
- Weights/ Bias: The parameters that may be learned in a model that regulates the signal between two neurons.

Before looking into optimizers, it is critical to understand that the optimizer's primary role is to update the weights of the learning process to achieve the lowest cost function. Here is the formula that all optimizers utilize to update the weights with a certain learning rate value.

$$W_x^* = W_x - a\left(\frac{\partial Error}{\partial W_x}\right) \quad (2.43)$$

where W_x^* represented new weights, W_x referred to old weights, a means the learning rate, and $\frac{\partial Error}{\partial W_x}$ referred to the derivative of the error concerning weight.

2.7.3.1 Types of Optimizers:

The deep learning model includes many optimizers for changing the weights and learning rate. However, selecting the optimal optimizer is depending on the application [58, 60].

There are different types of optimizers such as Gradient Descent, Stochastic Gradient Descent with Momentum (sgdm), Adagrad, Adadelta, RMSprop, and Adam. The Adam and sgdm optimizers have been implemented within the scope of this thesis.

2.8 Factors Affecting the Speaker Recognition System

The speaker recognition system's performance is affected by several factors, some of which are listed below [66, 67].

A. **Microphones:** The microphone is crucial in any speaker recognition system. It is used to collect speech samples from a speaker. The microphone must be of high quality to obtain accurate results from the system.

B. **Noise:** It can affect the overall speaker recognition system. It is of the utmost significance to capture the samples using microphones of high quality in an atmosphere devoid of background noise.

C. **Length of a voice sample:** The applications of the speaker recognition system determine this. In other words, the utilization of lengthier speech samples tends to yield more precise and accurate findings. However, shorter speech samples produce more accurate findings when the environment is loud.

D. **Speaker database:** In a speaker recognition system, two phases are very significant, the training and testing phase. Using this procedure, a number of speakers are added to the system's database. It is necessary to enroll a large number of speakers in order to acquire reliable results.

E. **Health of a speaker:** During the process of obtaining samples, this may impact the system's reliability. if, for example, the person speaking has a fever, a stuffy nose, or a sore throat infection.

2.9 Performance Parameter

The evaluation of the DL-classifier approaches focuses on accurately identifying the speaker during the test phase. The evaluation of this proposed system utilizes the performance metric of accuracy. The evaluation metrics for the various classifiers are computed by using the confusion matrix (CM).

2.9.1 Confusion Matrix

The performance evaluation of a classification model involves using a confusion matrix., a matrix of dimensions N×N. In this context, N denotes the entire quantity of target classes. The verified goal values are contrasted with the predictions generated by the deep learning model inside the matrix. In the context of multi-class classification, the dimensions of the matrix will correspond to the number of classes being considered. Conversely, in the case of binary type, the matrix will have a fixed size of 2×2.

A confusion matrix is a tabular representation that enumerates the number of accurate and inaccurate predictions a classifier makes. The

purpose of this evaluation is to assess the efficacy of a classification model. The effectiveness of a classification model can be evaluated by measuring metrics such as accuracy. As shown in Figure 2.15, the matrix displays the quantities of true negatives (TN), false positives (FP), true positives (TP), and false negatives (FN) produced by the model using the test data [68].

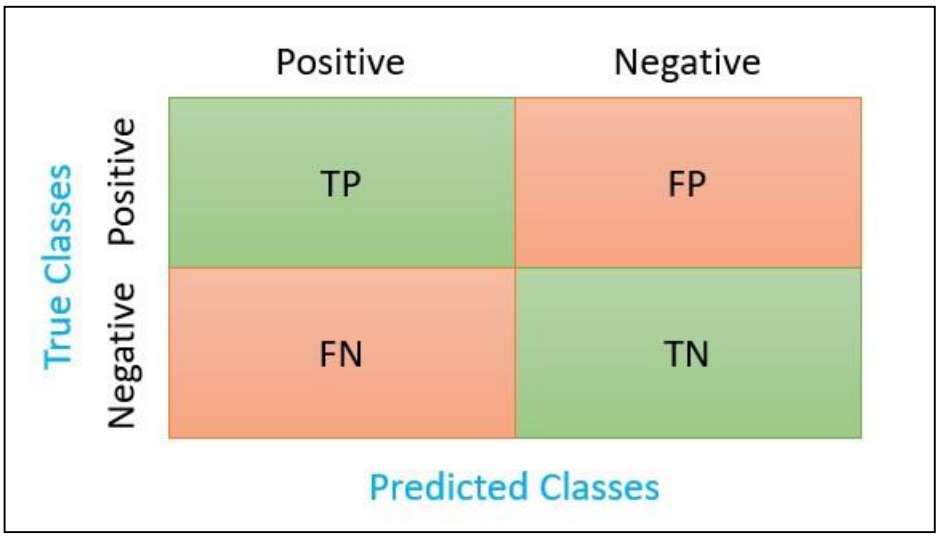


Figure 2.15 Confusion Matrix.

- TP are instances when the actual and projected values are positive.
- TN refers to instances when both the anticipated and actual values are negative.
- FP refers to instances when an optimistic forecast is made, but the actual result is negative. The phenomenon known as type 1 mistake is widely acknowledged within the context of statistical hypothesis testing.
- FN occurs when a forecast suggests a negative result, but the actual result is positive. Commonly known as the type 2 mistake, this statistical concept is frequently encountered in hypothesis testing.

A model with common FP and FN rates while having high TP and TN rates is regarded as an accurate model.

2.9.2 Accuracy

The measure of accuracy assesses the frequency with which the classifier makes accurate predictions. The calculation of accuracy involves dividing the number of accurate forecasts by the total number of predictions.

$$Accuracy = \frac{Number\ of\ samples\ correctly\ identified}{Total\ number\ of\ speakers\ samples} \times 100\% \quad (2.44)$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \times 100\% \quad (2.45)$$

2.10 K - Fold Cross Validation Technique Results

Cross-validation is a commonly employed technique in the domain of applied machine learning, serving as a mechanism for assessing the effectiveness of a machine learning system model when confronted with data that has not been previously encountered. In other words, the purpose is to utilize a limited subset of data to obtain an approximation of the model's overall performance when making predictions on unseen data, which was not included in the model's training process [69].

The process includes a singular parameter, denoted as "k" which indicates the number of distinct categories a given data sample will be partitioned into. As a result, the procedure is commonly denoted as k-fold cross-validation. Once a specific value for the variable k has been determined, it can be substituted instead of k in the reference to the model. For instance, k=5 can be referred to as 5-fold cross-validation, as illustrated in Table 2.1.

Table 2.1 K-CV when K=5.

Case 1	Case 2	Case 3	Case 4	Case 5
Test	Train	Train	Train	Train
Train	Test	Train	Train	Train
Train	Train	Test	Train	Train
Train	Train	Train	Test	Train
Train	Train	Train	Train	Test

This approach is commonly employed due to its inherent simplicity and tendency to yield less biased or overly optimistic estimates of the model's performance compared to alternative methodologies, such as a fundamental train-test split. The subsequent procedure is the conventional approach to adhering to the following:

1. Conduct a random shuffling of the database.
2. The dataset should be partitioned into k distinct categories.
3. For each unique group:
    - Use this group as a holdout or for the test database.
    - Utilize the remaining groups as a training database.
    - After fitting a model to the training set, assess the model using the test set.
    - The score from the assessment should be kept, but the model should be thrown out.
4. Incorporate a summary of the model's performance based on the model assessment score sample.

Chapter Three
“The Proposed System”

Chapter Three

The Proposed System

3.1 Introduction

Speaker Identification (SI) classifies each utterance according to the speaker's identity as a multi-class classification, where training and testing use the same predefined set of speakers.

Every speaker identification system has two modes: training and testing. In addition, every recognition system comprises three primary steps: pre-processing, feature extraction, and classification. The proposed system's first phase is pre-processing. It is a crucial procedure that can affect the effectiveness of deep learning algorithms. It can accelerate the discovery of knowledge from databases and can impact the accuracy of deep learning models in the long run. The step of feature extraction is essential to every recognition system. Rather than representing the speech signal with additional information, a few features are sufficient for improved speech representation. Finally, classification is defined as identifying, comprehending, and categorizing objects and concepts into predefined categories using pre-categorized training datasets. The proposed system is evaluated with the use of databases, namely SALU-AC, ELSDSR, TIMIT, and RAVDESS.

Six proposed systems (models) are employed in this thesis which are:

1. Model1: Mel-frequency Cepstral Coefficient / Convolution Neural Network.
2. Model2: Mel-frequency Cepstral Coefficient in conjunction with Principal Component Analysis / Convolution Neural Network.
3. 3-Model3: Two dimensional Discrete Wavelet Transform / Convolution Neural Network.

4. Model4: Two-dimensional Discrete Wavelet Transform in conjunction with Principal Component Analysis / Convolution Neural Network.
5. Model5: Mel-frequency Cepstral Coefficient in conjunction with Two-Dimensional Discrete Wavelet Transform / Convolution Neural Network.
6. Model6: Two-Dimensional Discrete Multi-Wavelet Transform / Convolution Neural Network.

And will be discussed later in detail.

3.2 Speech Databases

The SALU-AC, ELSDSR, TIMIT, and RAVDESS databases were used to evaluate the proposed system. The following is a description of the details of each of the databases that were used:

1- ELSDSR Database: an abbreviation for the English Language Speech Database for Speaker Recognition, is a curated collection of spoken language intended to provide speech data for the purpose of constructing and assessing a system for identifying and verifying speakers. The ELSDSR contains voice messages from 22 speakers (ten females and twelve males), ranging in age from 24 to 63 years. The language of communication is English, and 20 Danes, one Icelander, and one Canadian all contribute their voices. Each speaker has a total of nine utterances accessible to them. On average, the duration between samples varies from 6 to 20 seconds [3].

2- SALU-AC: "Salford University Anechoic Chamber," comprised a wide range of speech samples recorded from English native speakers as well as non-native speakers, and collected from a total of 110 speakers (55 males and 55 females). The speakers rely on reading texts from sources such as books, newspapers, and other publications. Every speaker was provided with three different recordings of their speech, each one lasting a different amount of time (the first sample lasted roughly 60 seconds, while the subsequent

examples lasted approximately 40 seconds each). This uses 104 speakers, with 55 females and 49 male contributing their voices [70].

3- RAVDESS: (Ryerson Audio-Visual Database of Emotional Speech and Song) is a database of emotional speech that was compiled using recordings of 24 professional speakers (12 females and 12 males) speaking two sentences that were lexically matched while using a neutral North American accent. Every speaker has 60 utterances, each lasting between 3-4 seconds. This database has eight distinct sorts of expressions, which are as follows: disgust, neutral, surprise, calm, afraid, angry, joyful, and sad [71].

4- TIMIT: which stands for (Texas Instruments Massachusetts Institute of Technology) is a collection of recorded speech used for acoustic-phonetic studies and speech recognition technology evaluation. The TIMIT database comprises high-quality recordings of 630 individuals who speak eight distinct dialects of American English, characterized by their wide frequency range. Each speaker produces 10 phonetically rich sentences, with each statement lasting around 2-3 seconds [72].

Table 3.1 presents the number of samples for each database, both before and after the implementation of data augmentation at durations of 0.5 sec., 1 sec., 2 sec., 3 sec., and 5 sec.

Table 3.1 The number of samples for each database.

Name of Database		Duration				
		0.5 sec.	1 sec.	2 sec.	3 sec.	5 sec.
SALU-AC	Before	32448	16224	8112	5408	2704
	After	64896	32448	16224	10816	5408
ELSDSR	Before	4400	2420	1298	880	572
	After	8800	4840	2596	1760	1144
RAVDESS	Before	9120	4560	2880	1680	__
	After	18240	9120	5760	3360	__
TIMIT	Before	18900	9450	6300	__	__
	After	37800	18900	12600	__	__

3.3 The proposed system

As mentioned before, the speaker identification system consists of three main phases: preprocessing, feature extraction, and classification. Figure 3.1 shows the proposed methodology for the steps utilized in this thesis. The first phase is pre-processing, and the second and third steps of the proposed system are feature extraction and classification respectively.

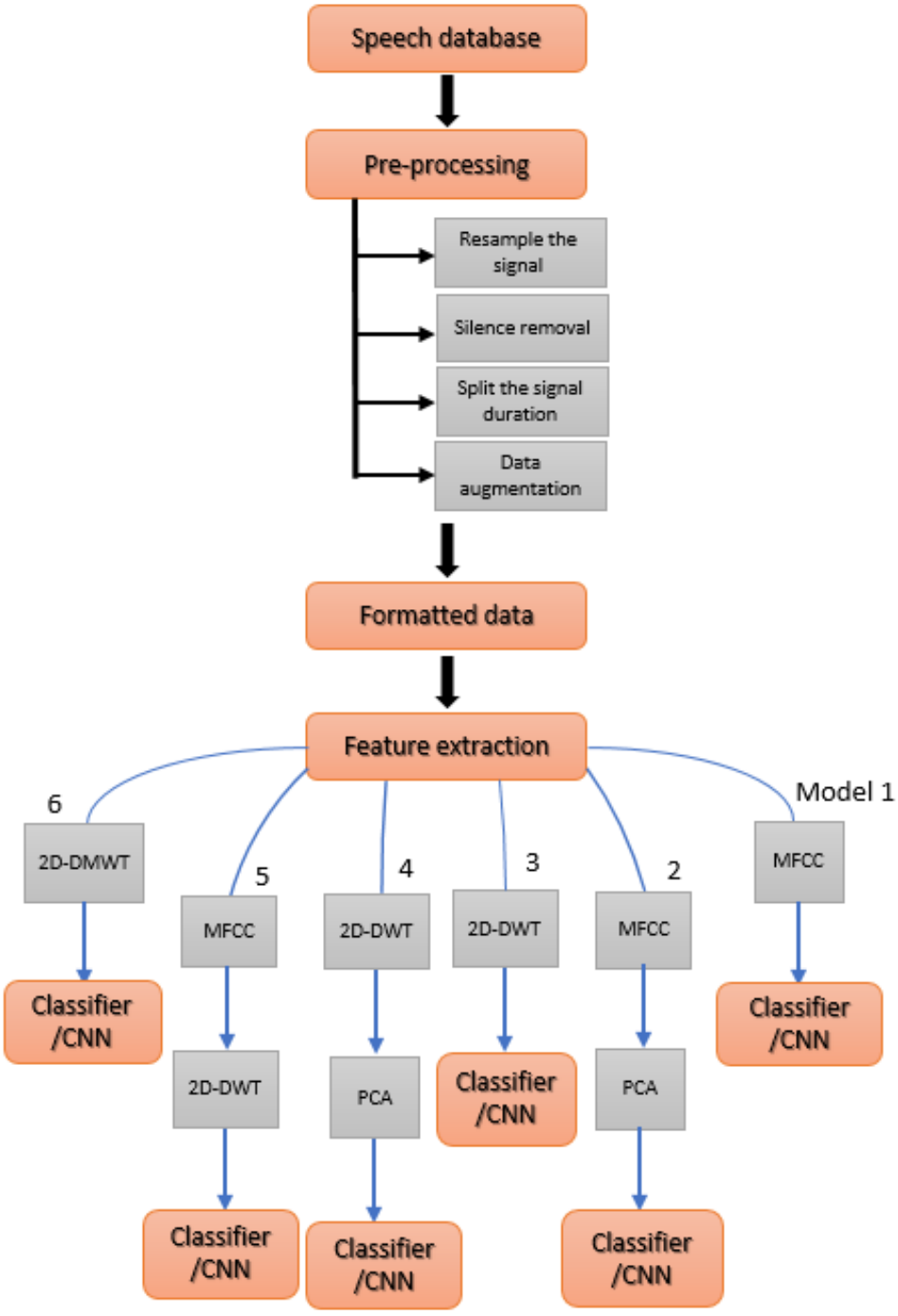


Figure 3.1 The proposed system.

3.3.1 Preprocessing

The first phase of the proposed system involves the preprocessing phase. The same preprocessing steps are applied to all methods proposed in this thesis. These steps are utilized to appropriately prepare the database for the feature extraction phase which are as follows:

1) Resample each voice sample channel to mono at a sampling frequency of 16kHz and a bit rate of 16 bits per sample.
2) Take the silence out of each spoken signal. Silence removal is essential in speaker and speech recognition since it increases system performance and reduces processing time. For example, a spontaneous speech sample was chosen to show the input speech signal after and before silence removal as shown in Figure 3.2.

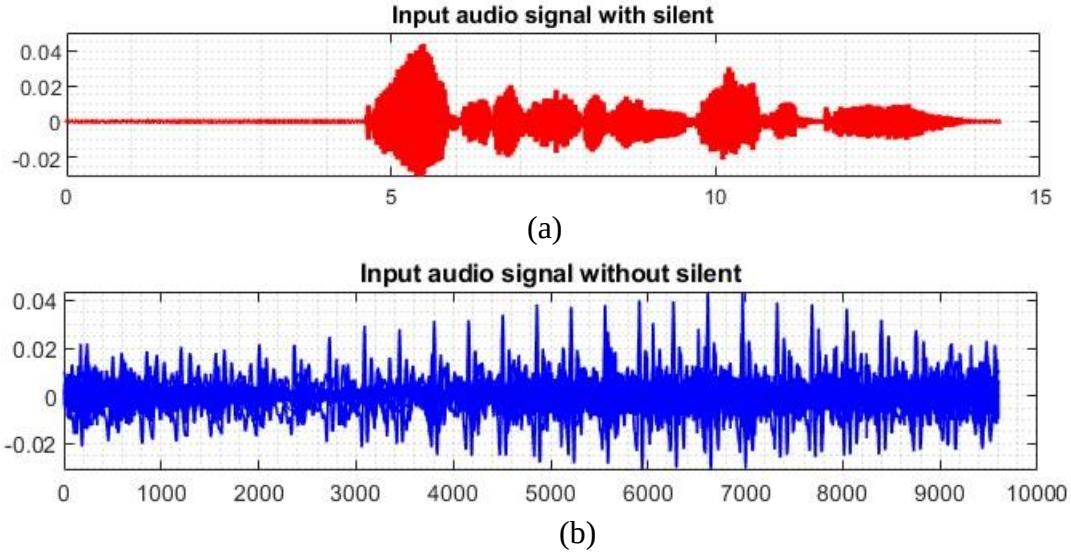


Figure 3.2 Speech signal (a) before removing silence, (b) after removing silence.

3) Split the time duration of each speech sample of each speaker into 0.5 sec., one sec., two sec., three sec., and five sec. to establish the appropriate length of time that leads to simply identifying the speakers' identities. Hence, most speakers speak at a rate of 2 words per half-second, 4 to 5 words per 1 second, 5 to 6 words per 2 second, seven words per 3 second, and 12 words per 5 second.

Depending on the duration of time that each database provides. The SALU-AC and ELSDSR databases each can divide their length into 5sec., while the RAVDESS database can split its duration into 3sec. Because each sample has a span of (3-4) seconds, whereas the samples in the TIMIT database each have a duration of (2-3) seconds, it is possible to break it up into 2 seconds.

4) Data augmentation is the process of expanding the amount of data that must be used to train the model. When it comes to deep learning models, huge data size is commonly required, but this is not always accessible. Not only did the addition of new data help to increase the size of the training set, but it also assisted in preventing models from becoming overfitting.

   In this thesis, data augmentation is accomplished by employing two techniques: time masking and frequency masking.

   i. Time Masking: The process of randomly selecting an uninterrupted portion of the audio file and replacing it with silence is known as time masking. By suppressing potentially distracting or uninformative parts, this strategy assists the model in concentrating on auditory properties pertinent to the analysis. The training of the model using time-masked data makes it more resistant to noise and more resilient to temporal fluctuations.

   ii. Frequency Masking: The audio spectrogram is divided into a series of frequency bins, and frequency masking includes covering up a portion of those bins. The model is better able to reject unimportant frequency components thanks to this method, which may have been caused by noise or background interference. The training of the model using

frequency-masked samples teaches it to concentrate on the fundamental frequency components, making it more resistant to spectrum perturbations.

3.3.2 The Feature Extraction and Classification for Models

The second and third phases of the PS are feature extraction (F.E.) and classification (Class.) respectively. Six algorithms are used for the F.E. phase and the CNN deep learning algorithm for Class. Each proposed system will be discussed separately as follows:

3.3.2.1 The F.E. and Class. for Model1: (MFCC/CNN)

The first model of the speaker identification system's performance employed the hybrid technique, which is comprised of the Mel-frequency Cepstral Coefficient as a feature extraction phase and Convolution Neural Network as a classifier.

The MFCC algorithm requires selecting a few parameters, such as the frame length, hop length, and mel filters of the MFCC algorithm. The size of the window applied to each signal segment is referred to as the frame length, and the amount of overlap between successive frames is referred to as the hop length. As indicated in Table 3.2, these parameters were used in this thesis.

Table 3.2 The parameters of MFCC.

Parameter	Value
Frequency Sampling Fs	16kHz
Frame Length	25 msec.
Hop length	10 msec.
Type of Window	Hamming window
FFT	1028
n-Mel	64
cepstral coefficients	128

This indicates that the frame length for a transmission with a frequency of 16 kHz would be 400 samples (16000×25msec =400), while the hop length would be 160 samples (16000×10msec =160). The number of frames required for each speech sample with a chosen time duration (0.5, 1, 2, 3, and 5 seconds) can be calculated as follows: for 0.5 seconds, the number of frames is 50 based on ((0.5×16000) ⁄160); similarly, for 0.5 seconds, the number of frames for the duration of the other can be calculated and it is equal to 100, 200, 300, and 500 for 1, 2, 3, and 5 seconds, respectively.

Then, for each speech sample, 64 mel filters and 128 cepstral coefficients were used to generate an MFCC matrix with the form (128, n), where "n" represents the total number of frames that were computed before. Thus, each frame would provide 128 features.

The dimensions of the input data for the CNN are determined by the specific convolution layer utilized and the MFCC matrix's structure. The convolution layers known as Conv1D and Conv2D are the most prevalent kinds used in speech processing. Conv1D receives its input as a two-dimensional shape of the form (f, n), where n presented as is the number of frames and f refers to the number of features. Conv2D receives its input as a three-dimensional shape of the form (f, n, c), where n and f represent the variables mentioned previously, and c represents the number of channels. Following the feature extraction phase, the CNN algorithm is used as a classifier.

The input dimension of a 3D CNN can be presented as (number of coefficients, number of frames, number of channels). The structure of the CNN algorithm is presented in Table 3.3.

Table 3.3 Structure of CNN layers

Layer No.	Layer Name	Detail
1	Input	(128×n×1)
2	Convolution	3×3, 64 filters, padding "same"
3	Tanh	$f(x)_{tanh} = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$
4	Max Pooling	Pool size = 2×2
5	Convolution	3×3, 128 filters, padding "same"
6	Tanh	$f(x)_{tanh} = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$
7	Max Pooling	Pool size = 2×2
8	Dropout	0.1, prevent overfitting
9	Dense (FC)	1024 neurons
10	softmax	Compute the probability of each label
11	Classification output	Number of classes

The CNN model underwent training for a total of 15 epochs, using a learning rate of 0.1 and employing a mini-batch size of 50. The optimization algorithm is employed by the Adam optimizer.

3.3.2.2 The F.E. and Class. for Model2: (MFCC-PCA/CNN)

The two techniques, MFCC and PCA, were combined to generate a hybrid methodology. Before using CNN as the classifier, PCA was employed to reduce the feature data dimension of the MFCC algorithm (explained in 3.3.2). One hundred twenty-eight features are extracted from each frame of the speech sample; after that, PCA is employed to reduce the dimensionality.

Remember that the Eigen decomposition of the transformation matrix, which in the case of principal component analysis (PCA) is a covariance

matrix, yields a collection of eigenvectors and associated eigenvalues. These eigenvectors are the primary components that hold most of the variation represented by features (independent variables) or information. The variation explained by a certain eigenvector is shown by the presented variance ratio. The first principal component signifies the most substantial variability within the data, whereas the subsequent principal components sequentially capture subsequent levels of variation.

"Explained variance" is a statistical measurement that refers to the amount of variation in a dataset that can be attributed to each of the principal components (eigenvectors) created by the principal component analysis (PCA) technique. This variation may be seen depicted in Figure 3.3. It is a simple means of referring to the proportion of the variability in a data set that can be attributed to each fundamental component. Moreover, it tells us how much of the total variance each component "explains" to us. This is significant because it helps to prioritize the most important elements when evaluating our investigation findings and rank the components in order of significance.

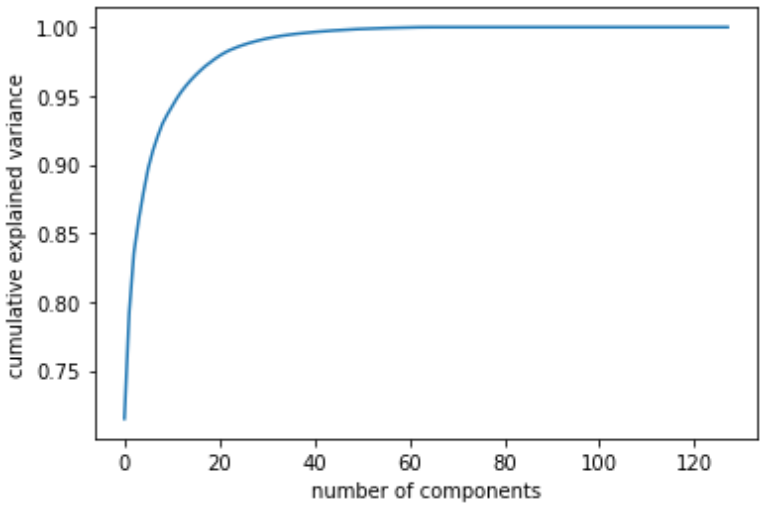


Figure 3.3 The explained variance.

As seen in Figure 3, which depicts the relationship between cumulative explained variance and the number of components (number of features), the curve became stable at the value of 30, and when numerous options for the right value were considered, the value of 32 was the best, which means (the number of features of each frame decreased from 128 to 32.

The structure of CNN layers was the same as in method one except for the change in the input dimensions with new input dimensions equal to (32×n×1)

3.3.2.3 The F.E. and Class. for Model3: (2D-DWT/CNN)

The third method used a 2D DWT in combination with the CNN algorithm. To start the 2D DWT process, the 1D speech signal must first be transformed into a format compatible with 2D. This implementation used the Daubechies wavelet coefficients (D4 or db2) as a low pass filter.

The size of the original input matrix is (128×128). After applying 2D DWT to a 2D processed speech signal with 128×128 dimensions and applying two levels of decomposition to it, the resultant processed speech-signal matrix is partitioned into four main sub-bands after the first level of decomposition, as shown in Figure 3.4; each sub-band has (64×64) dimensions. After processing level 2 on ($LL1$), the second level of decomposition has a size of (32×32) which is presented with the ($LL2$) sub-band. the following figure illustrates the procedures carried out to achieve the required dimensions.

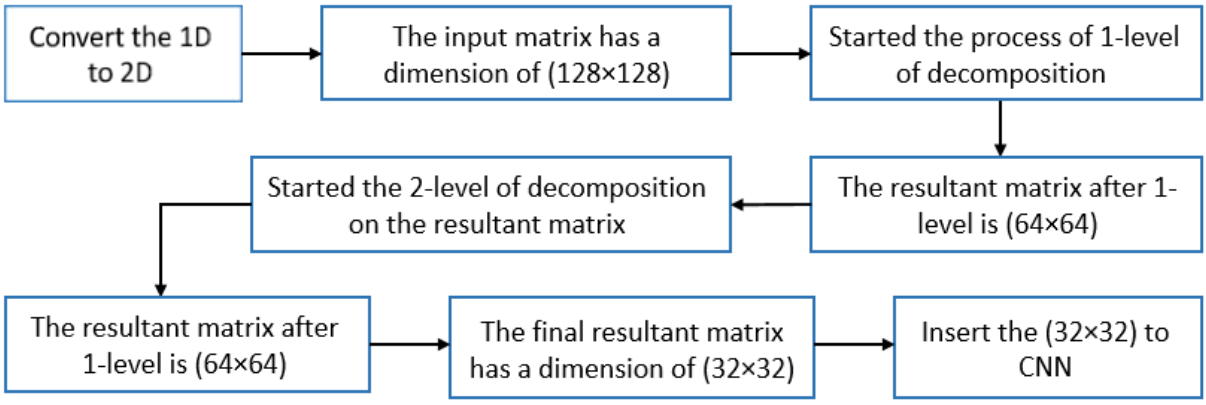


Figure 3.4 The proposed system of 2D-DWT.

Therefore, the dimensions of the resulting matrix features, which will be used as input to the classification step, will be (32×32). Consequently, both the extraction of features and the reduction of dimensionality are completed at this stage. Figure 3.5 illustrates the sub-bands that are present at each level of decomposition.

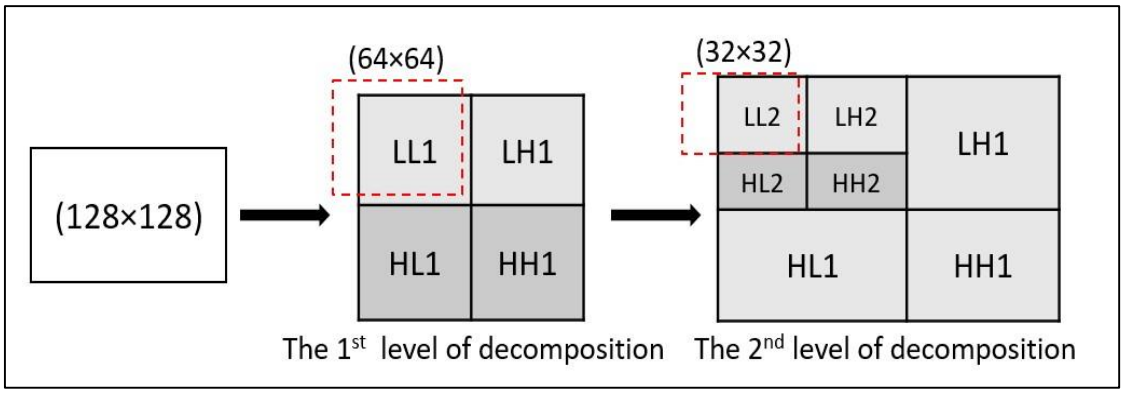


Figure 3.5 The 2 levels of decomposition of 2D DWT.

The structure of the Convolutional Neural Network algorithm is illustrated in Table 3.4.

Table 3.4 The architecture of CNN layers.

Layer No.	Layer Name	Detail
1	Input	32×32×1
2	Convolution	3×3, 64 filters, padding "same"
3	Tanh	$f(x)_{tanh} = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$
4	Max Pooling	Pool size = 2×2
5	Convolution	3×3, 128 filters, padding "same"
6	Tanh	$f(x)_{tanh} = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$
7	Max Pooling	Pool size = 2×2
8	Dropout	0.1, prevent overfitting
9	Dense (FC)	1024 neurons
10	softmax	Compute the probability of each label
11	Classification output	Number of classes

The CNN model was trained for 15 epochs at 0.1 learning rate and 50 mini-batch size. The Adam optimizer makes use of the optimization method.

3.3.2.4 The F.E. and Class. for Model4: (2D-DWT-PCA/CNN)

The fourth technique uses a hybrid system consisting of the combination of 2D-DWT and PCA for the feature extraction phase and the CNN as a classifier. In the 2D-DWT approach, a two-level decomposition was utilized, beginning with the dimensions (128×128) and ending with (32×32). This was indicated in the third method. PCA was used to reduce the dimensionality of (32×32), and the diagram helps explain how the PCA process works.

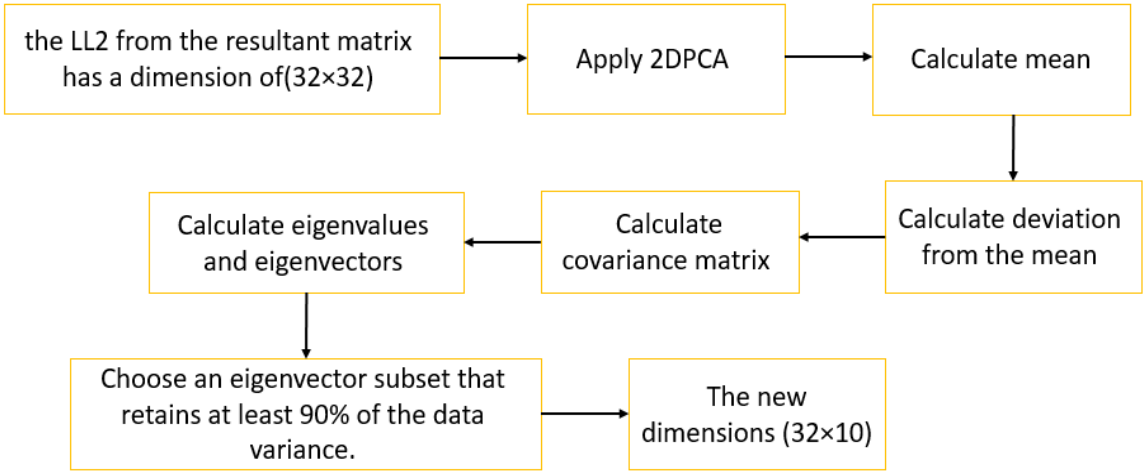


Figure 3.6 The steps of the PCA process.

According to the diagram, the resultant matrix from 2D-DWT with a dimensions of (32×32) used as an input to 2DPCA, and then the data must first be centered by doing the following operation: subtracting the mean of each column from each element in that column. The covariance matrix of the centered data should then be calculated, together with its eigenvectors and eigenvalues. In the last step, arrange the eigenvectors in decreasing order of their associated eigenvalues, choose a subset of the eigenvectors that preserves at least 90% of the variance in the data, and convert the data into the new coordinate system defined by the eigenvectors that were selected.

The dimensions (32×10) relate to the shape the data takes after PCA analysis with ten principal components has been performed. The number of rows in the original data is denoted by the value 32, and the number of columns in the data that has been modified is denoted by the value 10.

Consequently, the input for the classification stage will be comprised of the dimensions (32×10). At this stage, both feature extraction and dimensionality reduction are carried out as a result. The architecture of the CNN algorithm is presented in Table 3.5.

Table 3.5 CNN layer structure.

Layer No.	Layer Name	Detail
1	Input	32×10×1
2	Convolution	3×3, 64 filters, padding "same"
3	relu	$f(x)_{ReLU} = \max(0, x)$
4	Max Pooling	Pool size = 2×2
5	Convolution	3×3, 128 filters, padding "same"
6	relu	$f(x)_{ReLU} = \max(0, x)$
7	Max Pooling	Pool size = 2×2
8	Dropout	0.1, prevent overfitting
9	Dense (FC)	1024 neurons
10	softmax	Compute the probability of each label
11	Classification output	Number of classes

The CNN model was trained with 15 epochs at 0.1 learning rate and 50 mini-batch size. The Adam optimizer uses as the optimization algorithm.

3.3.2.5 The F.E. and Class. for Model5: (MFCC-2D-DWT/CNN)

For the fifth approach, a hybrid system was proposed. This system consisted of MFCC and two-dimensional DWT for the feature extraction phase, and the CNN was utilized as the classifier. As seen in the first method, MFCC was used for the feature extraction phase, and the form of the MFCC matrix was (128, n). Here, n refers to the number of frames associated with each time duration. The MFCC matrix was then subjected to a 2D-DWT transformation. To employ a 2D-DWT, the matrix has to be N×N, and the

value of N needs to be raised to the power of two. Therefore, more zeros need to be added to the matrix whose dimensions are (128, n) to express their dimensions as a power of two. The new sizes will be (128×64), (128×128), (128×256), (128×512), and (128×512) correspondingly for 0.5 seconds, 1 second, 2 seconds, 3 seconds, and 5 seconds respectively.

Two levels of decomposition were implemented on a two-dimensional matrix. In this implementation, the low pass filter utilized was the Daubechies wavelet coefficients (D4 or db2) wavelet coefficients. After the first level of decomposition, the resulting matrix is divided into four primary sub-bands, as illustrated in Figure 3.7. After performing level 2 processing on the (LL_1) sub-band, the (LL_2) sub-band is delivered as the output of the second level of decomposition.

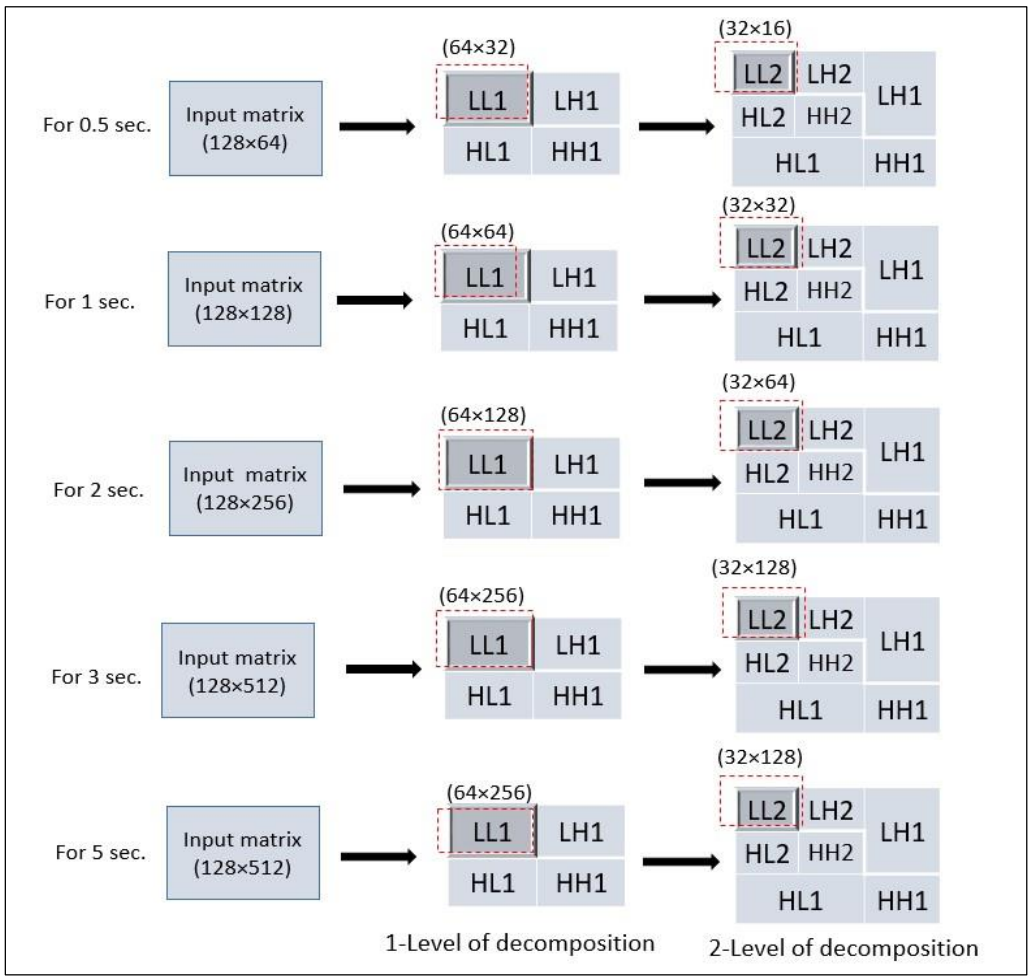


Figure 3.7 Two levels of decomposition on different input matrices.

The resulting matrix has a variable dimension based on the time durations, as seen in Figure 3.7; these dimensions were the input for the classification phase performed using CNN. Table 3.6 provides an illustration of the basic structure of the CNN algorithm.

Table 3.6 The structure layers of CNN

Layer No.	**Layer Name**	**Detail**
1	Input	For 0.5 sec. (32×16×1) For 1 sec. (32×32×1) For 2 sec. (32×64×1) For 3 sec. and 5 sec. (32×128×1)
2	Convolution	3×3, 64 filters, padding "same"
3	relu	$f(x)_{ReLU} = \max(0, x)$
4	Max Pooling	Pool size = 2×2
5	Convolution	3×3, 128 filters, padding "same"
6	relu	$f(x)_{ReLU} = \max(0, x)$
7	Max Pooling	Pool size = 2×2
8	Dropout	0.1, prevent overfitting
9	Dense (FC)	1024 neurons
10	softmax	Compute the probability of each label
11	Classification output	Number of classes

The CNN model was trained using 15 epochs, 0.1 learning rate, and 50 mini-batch sizes. The Adam optimizer uses the optimization algorithm in its processing.

3.3.2.6 The F.E. and Class. for PS6: (2D-DMWT/CNN)

The final approach used a two-dimensional DMWT in conjunction with the CNN algorithm. In this approach, the preprocessing phase contained all of the techniques with the exception of data augmentation. In order to initiate the two-dimensional DMWT procedure, it is necessary to convert the one-dimensional speech signal into a format that is suitable for two-dimensional analysis.

As illustrated in Figure 2.11, the approximation-based pre-processing utilized to generate the final DMWT matrix yields the exact dimensions (N×N) as the original input matrix (256×256). The resulting processed speech-signal matrix is partitioned into four primary sub-bands after applying 2D-DMWT to 2D processed speech-signal with 256×256 dimensions, as illustrated in Figure 3.8. Each sub-band has a size of 128×128. Furthermore, each major one (sub-band) is split into four sub-sub-bands, each of 64×64 dimensions. Because most of the discriminating elements of the speech signal are located in the low-low (LL) frequency sub-band, the primary one (LL-sub-band) is kept while the other sub-bands are omitted. The extracted speech matrix that results is the average matrix of the four sub-sub-bands located in the LL main sub-band.

Consequently, the generated matrix features have 64×64 dimensions when used as an input to the classification step.

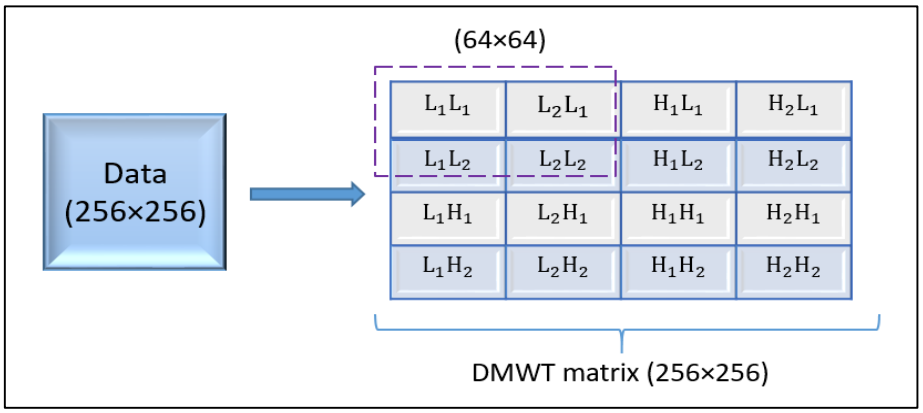


Figure 3.8 Single level of decomposition.

The CNN is made up of 15 layers, as shown in Table 3.7. To begin the classification process, the input picture for the CNN, which is a four-dimensional matrix with the dimensions (64×64×1×number of samples), is loaded into the CNN. A learning rate of 0.01 was employed throughout the CNN model's training procedure, which comprised 500 epochs. Additionally, the sgdm optimizer was responsible for applying the optimization approach.

Table 3.7 The layers of a CNN architecture.

Layer No.	**Layer Name**	**Detail**
1	Input	64×64×1×Number of samples
2	Convolution	3×3, 24 filters, padding "same."
3	Batch Normalization	improves the learning speed and provides regularization, avoiding overfitting
4	relu	$f(x)_{ReLU} = \max(0, x)$
5	Max Pooling	Pool size = 4×4
6	Convolution	3×3, 36 filters, padding "same"
7	Batch Normalization	improves the learning speed and provides regularization, avoiding overfitting
8	relu	$f(x)_{ReLU} = \max(0, x)$
9	Max Pooling	Pool size = 4×4
10	Convolution	3×3, 48 filters, padding "same"
11	Batch Normalization	improves the learning speed and provides regularization, avoiding overfitting
12	relu	$f(x)_{ReLU} = \max(0, x)$
13	Fully Connected Layer	Number of speaker
14	softmax	Compute the probability of each label
15	Classification output	Number of classes

Chapter Four
“Results and Discussion”

Chapter Four
Results and Discussion

4.1 Introduction

This chapter presents and explains the experimental results of the proposed system for four distinct databases: SALU-AC, ELSDSR, RAVDESS, and TIMIT. The suggested system's performance statistic is also explained. Additionally, each database's total number of samples will be shown in detail. The first five techniques were developed using Python, while the last method was done using MATLAB.

4.2 The Experimental Results

This section presents the results of experiments using the proposed system based on six methods which are MFCC-CNN, MFCC/PCA-CNN, 2D-DWT-CNN, 2D-DWT/PCA-CNN, MFCC/2D-DWT-CNN, and 2D-DMWT-CNN. The proposed system is tested based on four different databases: SALI-AC, ELSDSR, RAVDESS, and TIMIT. K - Fold CV technique is used to analyze the results of the proposed system for all the methods.

4.2.1 The Experimental Results for Model1

The first system proposed in this thesis involved the integration of MFCC-CNN. The experimental results of the model are explained in this section. The MFCC is employed for the feature extraction phase and CNN for the classification as discussed in Chapter Three. The model is applied to all databases utilized in this thesis, which are SALU-AC, ELSDSR, RAVDESS, and TIMIT. K-fold CV is employed to evaluate the model1 and k=2 and k=5 are used. The results of the model are illustrated in Table 4.1.

Table 4.1 The recognition rates for the Model1.

K-fold	Database	Accuracy (%)				
		0.5 sec.	**1 sec.**	**2 sec.**	**3 sec.**	**5 sec.**
K=2	SALU-AC	87.09	90.21	92.53	95.47	97.05
	ELSDSR	94.20	94.83	96.51	97.26	98.44
	RAVDESS	70.97	78.09	86.84	87.20	--
	TIMIT	68.42	74.07	78.45	--	--
K=5	SALU-AC	90.69	92.20	95.19	97.64	98.06
	ELSDSR	95.11	97.11	97.88	98.06	99.03
	RAVDESS	85.49	88.57	92.27	96.07	--
	TIMIT	75.36	88.03	92.19	--	--

According to the results presented in Table 4.1, it is obvious that using a value of k=5 yielded superior outcomes in comparison to k=2. This can be due to the fact that a higher value of k (k=5) allows for an accurate estimate of the model's performance. However, it is important to note that employing a higher value of k demands additional computational resources and time. On the other hand, employing a smaller value of k (k = 2) can offer computational efficiency, but it may lead to a less accurate estimation.

The high performance of the proposed system can be attributed, as can be seen in Table 4.1, to the MFCC algorithm, which leads to the learning of discriminative features from the speech signal and the classification of speakers based on the spectral and temporal characteristics they possess.

Because MFCC features are being used as the input to CNN, the speaker identification system can use both the spectrum information that MFCC provides and the feature ability to learn that CNN has. CNN can learn additional details and variances from the MFCC features and categorize speakers based on their speech patterns. In addition, using k-fold has contributed to the increased reliability of the results.

Given the exceptional outcomes achieved with a value of k equal to 5, the Figures will be only shown for the circumstance when k =5. Figure 4.1-

4.4 demonstrates the accuracy of the testing phase, illustrating the relationship between accuracy and the number of epochs for all databases used in this thesis.

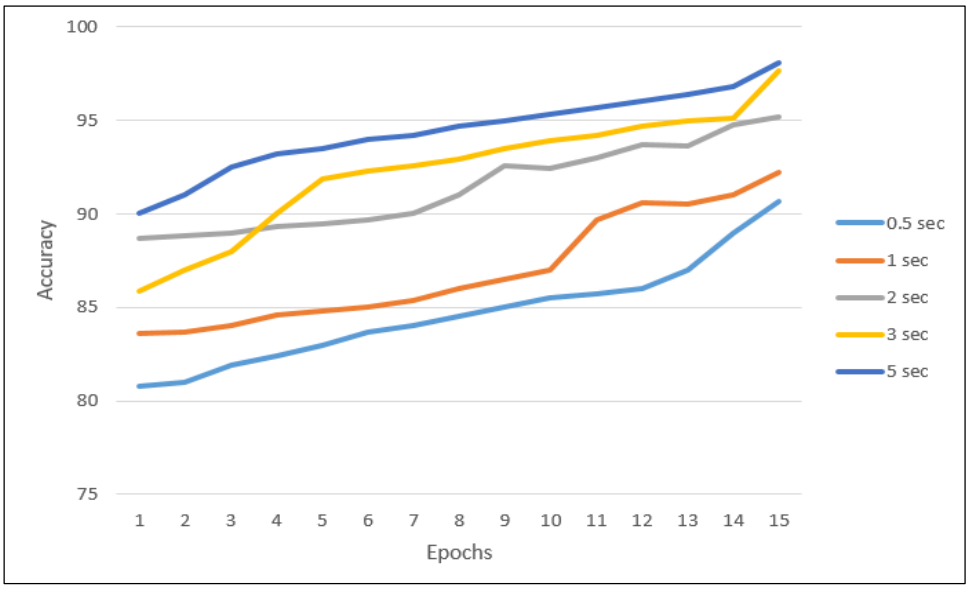


Figure 4.1 The performance of the SALU-AC database for the chosen durations for Model 1.

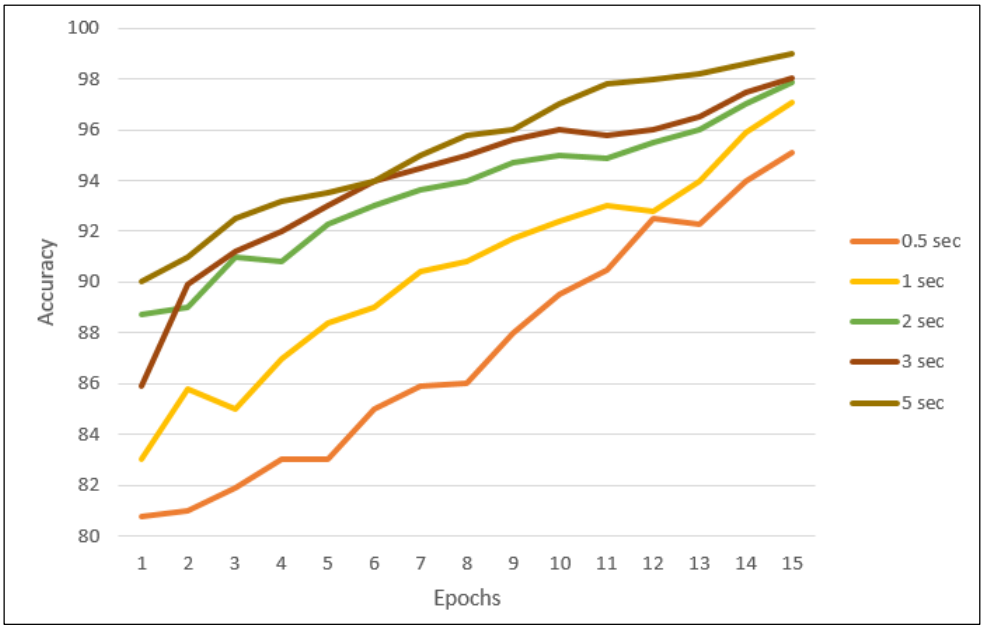


Figure 4.2 The performance of the ELSDSR database for the chosen durations for Model 1.

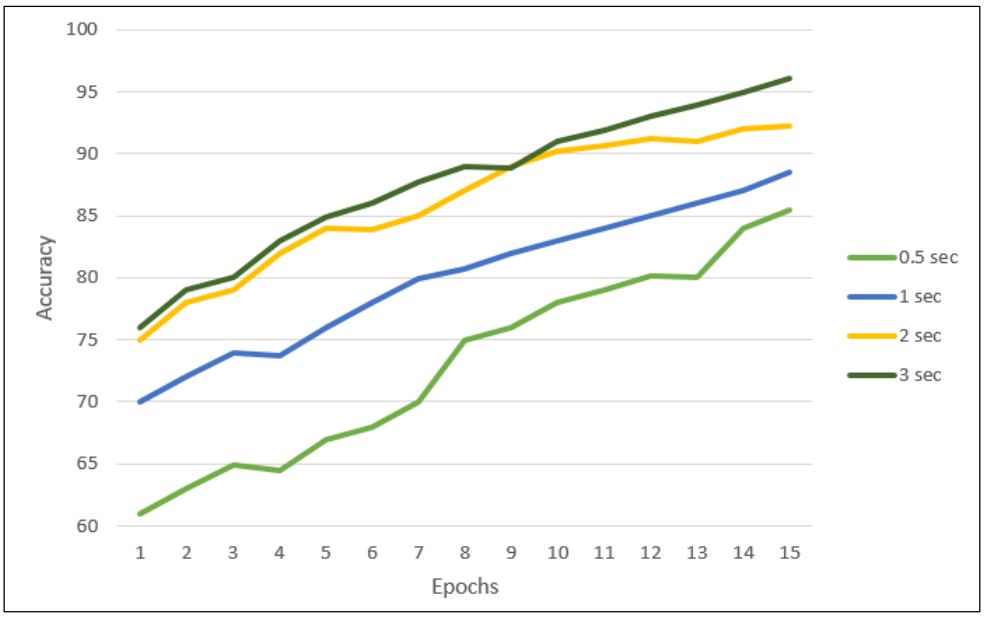


Figure 4.3 The performance of the RAVDESS database for the chosen durations for Model 1.

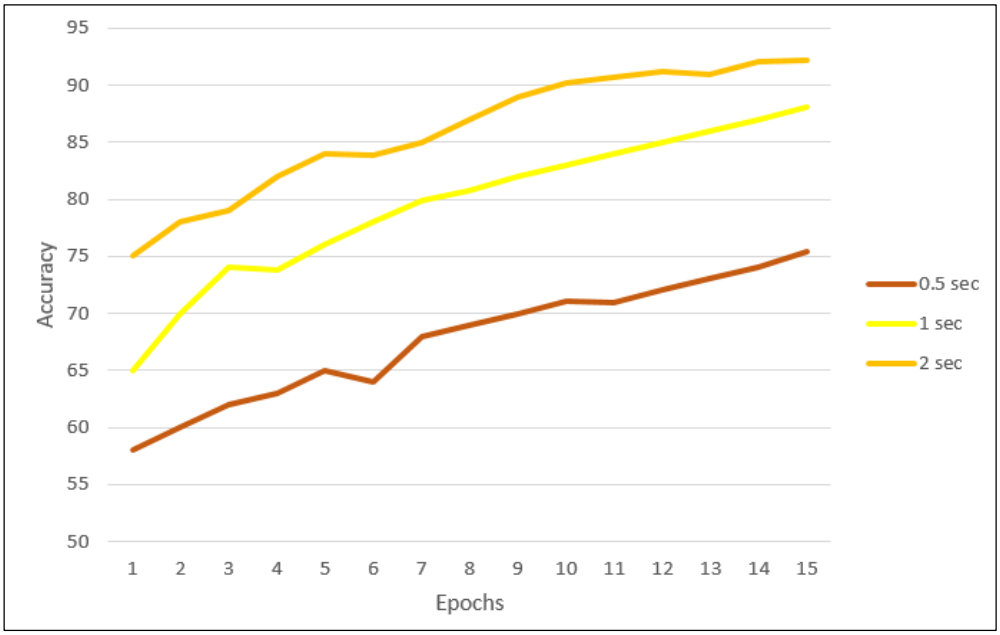


Figure 4.4 The performance of the TIMIT database for the chosen durations for Model 1.

Figures 4.5-4.7 show another way of exhibiting the results by the confusion matrix. An illustration of the accuracy calculation process for the SALU-AC, ELSDSR, and RAVDESS databases will be provided, with a time duration of 2 seconds. However, due to its large size (630×630), the TIMIT database cannot be accessed and therefore cannot be included in the

demonstration. Regarding the SALU-AC database, only a total of 24 speakers will be presented to ensure optimal clarity of the values.

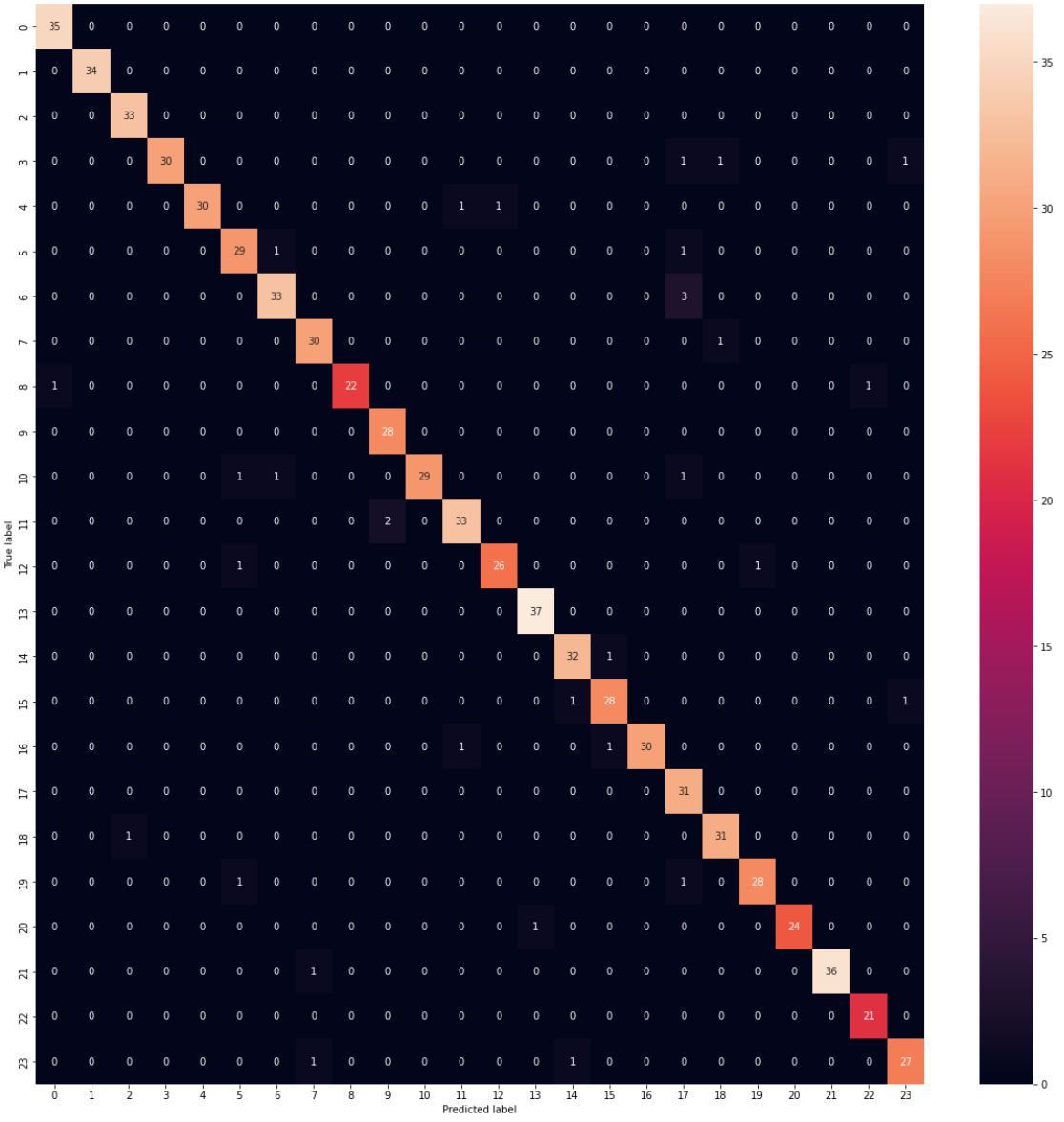


Figure 4.5 The confusion matrix of the SALU-AC database of Model 1.

Figure 4.5 displays the confusion matrix of the SALU-AC database at a duration of 2 seconds only for 24 speakers. During this time duration, the total number of evaluated speaker samples was 749 (TP+TN+FP+FN = 749). The true positive value, representing the sum of TP and TN, was 717. Based on Eq. 2.45, the accuracy can be expressed as follows:

$$Accuracy = \frac{717}{749} \times 100\% = 95.72\%$$

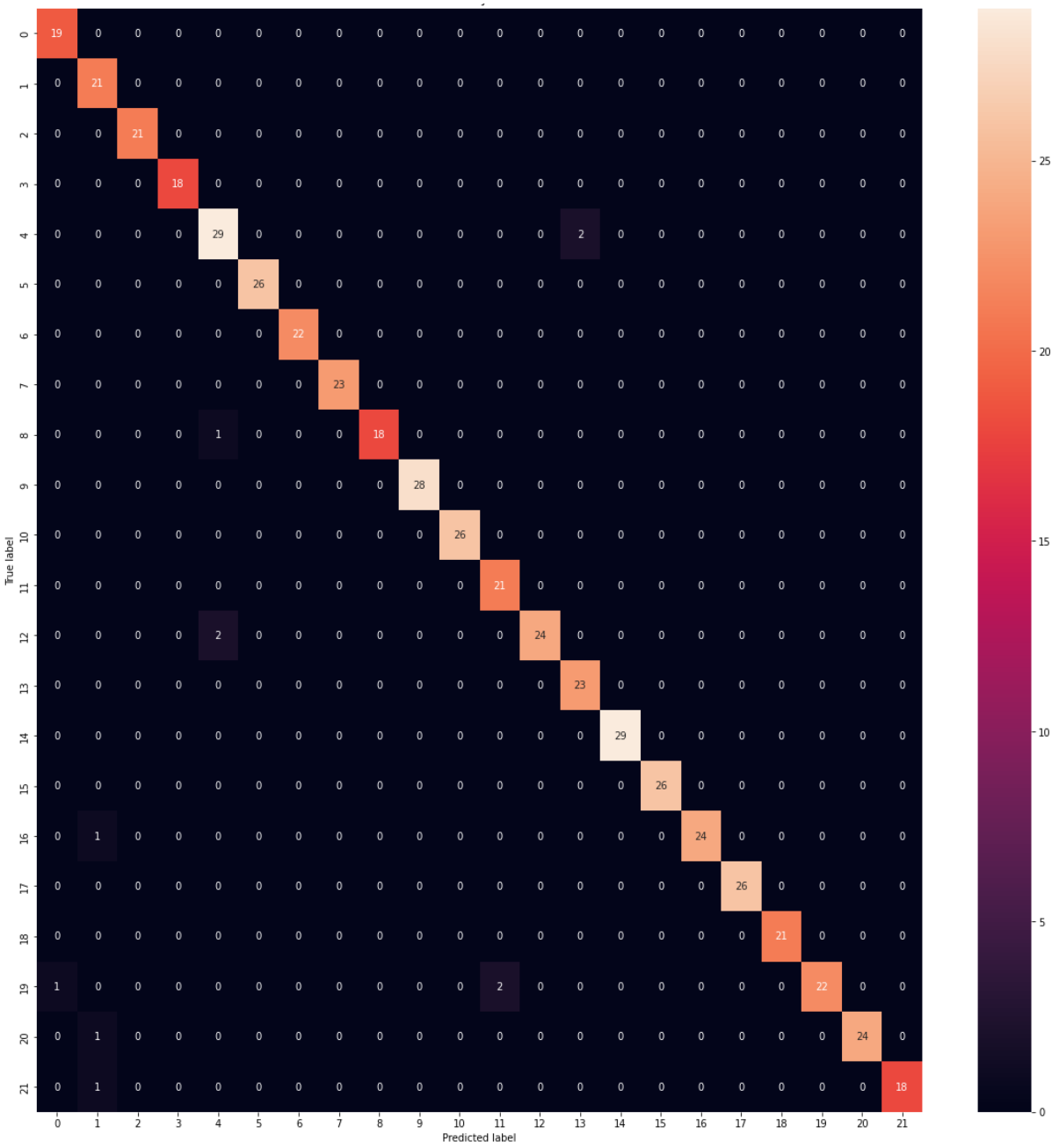


Figure 4.6 The confusion matrix of the ELSDSR database of Model 1.

Figure 4.6 shows the confusion matrix of the ELSDSR database at 2 sec., In the time length of 2 sec. the total number of speaker samples evaluated was equal to (TP+TN+FP+FN = 520), and the true positive was equal to (TP+TN = 509). Recognition rates can be found by Eq. 2.45 and express as:

$$Accuracy = \frac{509}{520} \times 100\% = 97.88\%$$

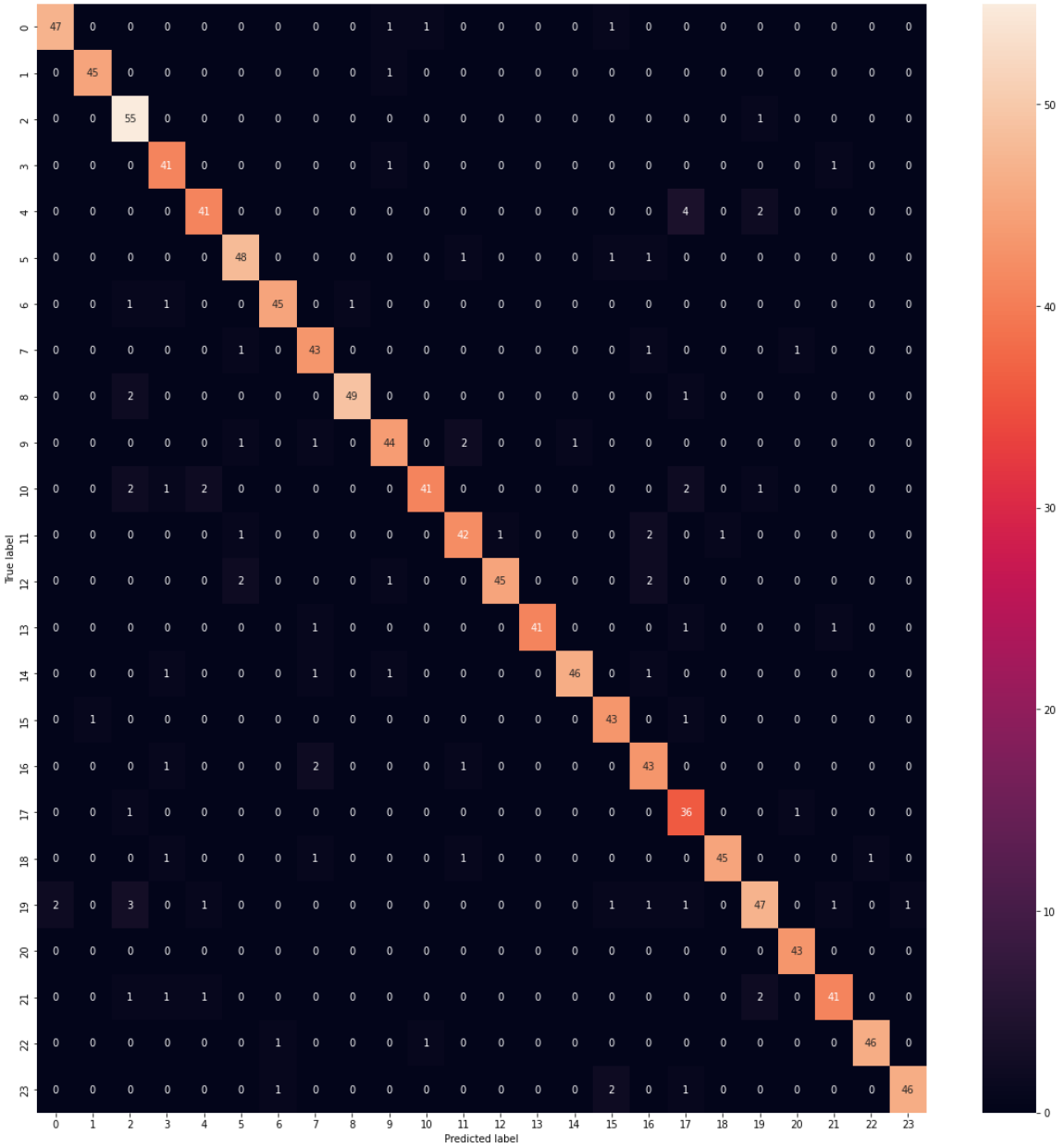


Figure 4.7 The confusion matrix of the RAVDESS database of Model 1.

Figure 4.7 presents the confusion matrix of the RAVDESS database at 2 sec., In the time length of 2 sec. the total number of speaker samples evaluated was equal to (TP+TN+FP+FN = 1152), and the true positive was equal to (TP+TN = 1063). The results are obtained by Eq. 2.45 and expressed as:

$$Accuracy = \frac{1063}{1152} \times 100\% = 92.27\%$$

4.2.2 The Experimental Results for Model2

The second proposed system presented in this thesis is the integration of MFCC with PCA for the feature extraction phase, and CNN for the classification phase, as discussed in Chapter Three. The model 2 methodology is used for all datasets utilized in this thesis, namely SALU-AC, ELSDSR, RAVDESS, and TIMIT. K-fold cross-validation is used to assess the performance of the model 2, with k values of 2 and 5 being applied. The findings of the model 2 experiment are shown in Table 4.2.

Table 4.2 The recognition rates for the Model2.

K-fold	Database	Accuracy (%)				
		0.5 sec.	1 sec.	2 sec.	3 sec.	5 sec.
K=2	SALU-AC	89.71	93.27	95.67	97.11	97.69
	ELSDSR	95.77	96.48	97.37	97.53	98.59
	RAVDESS	82.42	85	93.69	94.94	--
	TIMIT	69.46	77.25	85.08	--	--
K=5	SALU-AC	93.47	96.47	97.31	98.65	99.03
	ELSDSR	97.52	98.34	98.84	99.34	99.78
	RAVDESS	87.85	90.12	95.65	96.73	--
	TIMIT	77.60	90.59	93.84	--	--

Based on the findings presented in Table 4.2, it is evident that employing a value of k=5 resulted in superior outcomes compared to k=2. The Figures will only be presented for the specific case where the value of k is set to 5, as this particular setting has yielded exceptional results. Figures 4.8-4.11 present the findings of the testing phase, demonstrating the correlation between accuracy and the number of epochs across all databases examined in this thesis.

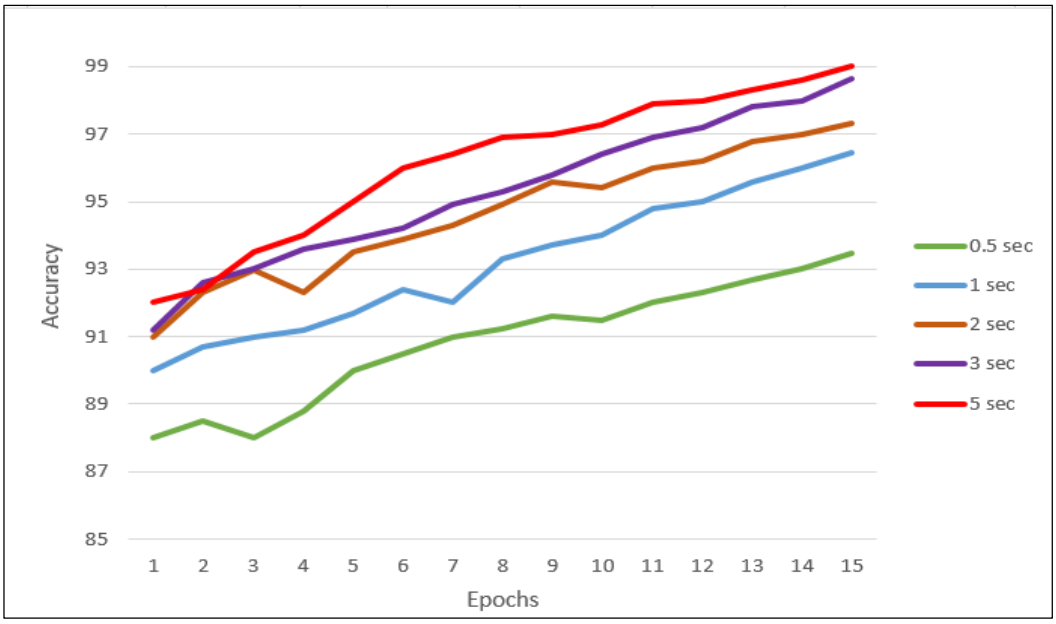


Figure 4.8 The performance of the SALU-AC database for the chosen durations for Model 2.

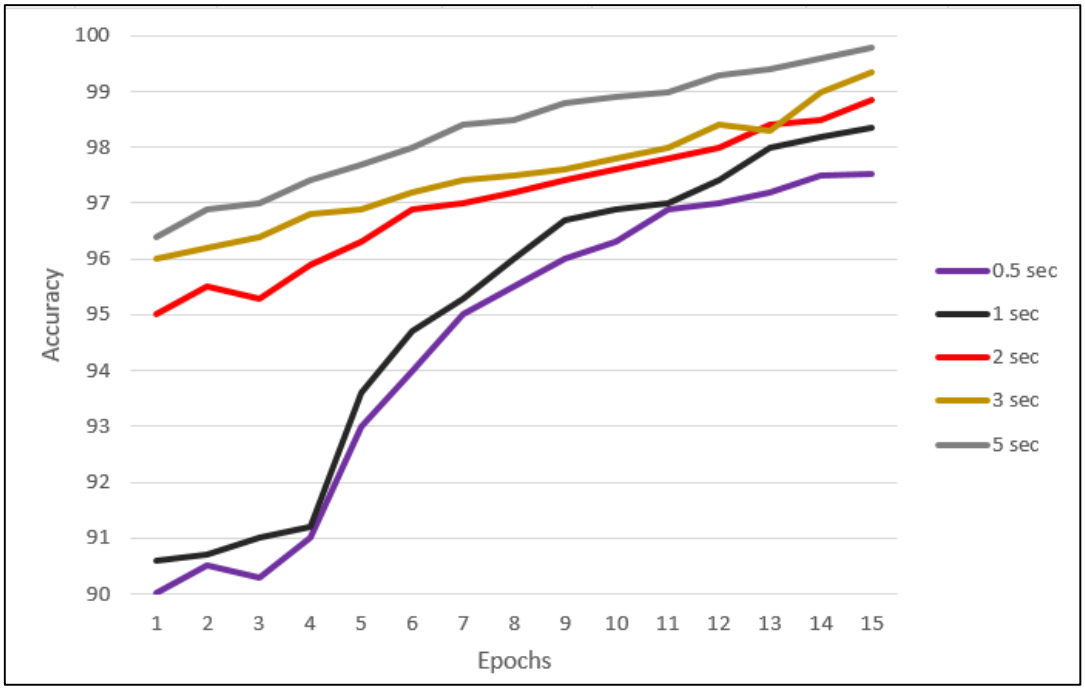


Figure 4.9 The performance of the ELSDSR database for the chosen durations for Model 2.

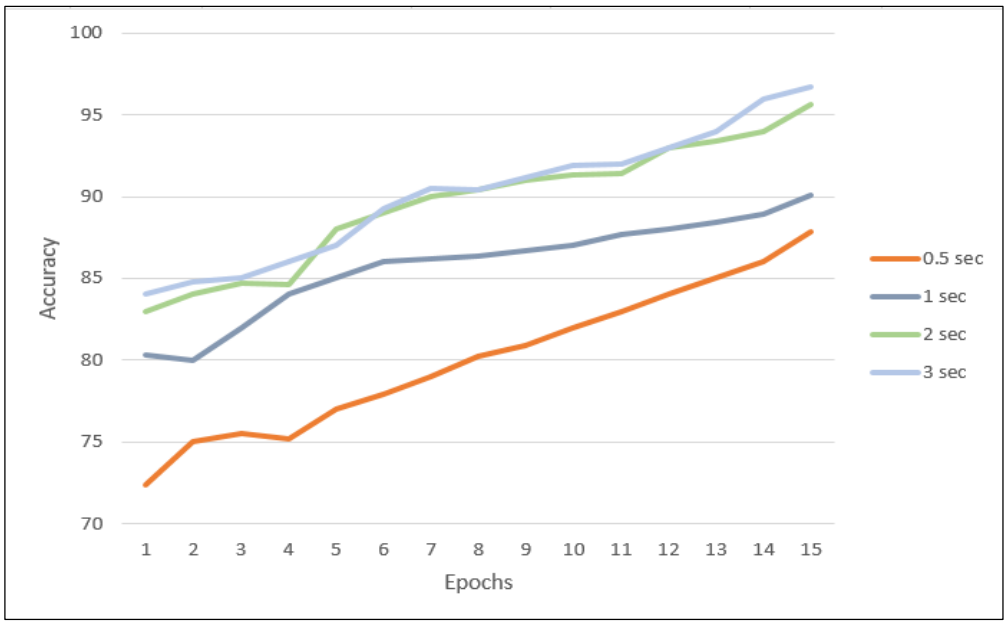


Figure 4.10 The performance of the RAVDESS database for the chosen durations for Model 2.

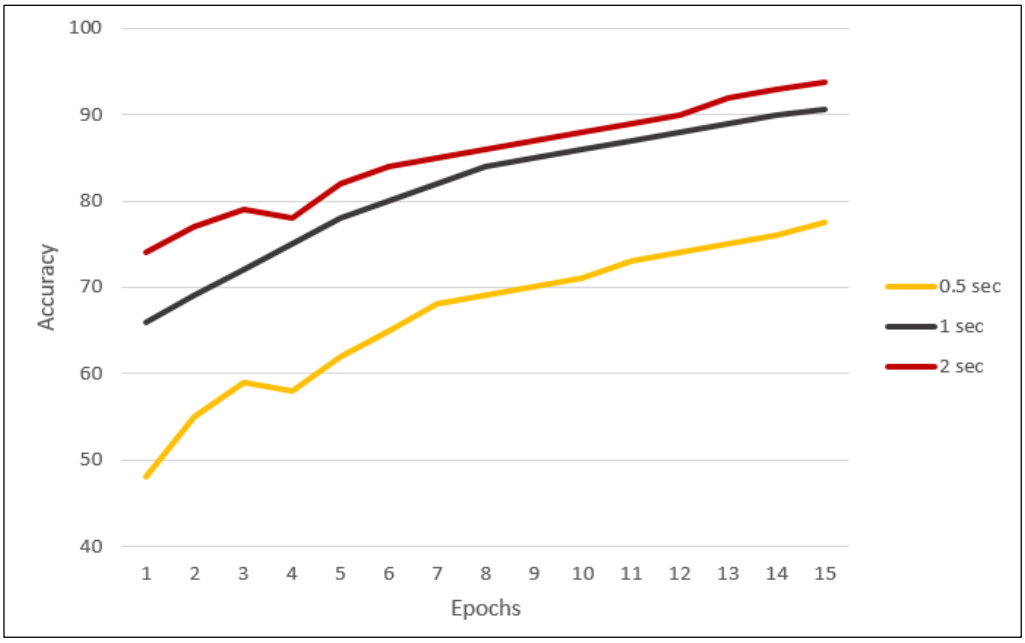


Figure 4.11 The performance of the TIMIT database for the chosen durations for Model 2.

The confusion matrix, which is another technique used for calculating accuracy, is seen in Figures 4.12-4.14. These figures aim to provide an illustration of the accuracy calculation procedure for the ELSDSR, SALU-AC, and RAVDESS databases. The length of the example will be limited to 2 seconds.

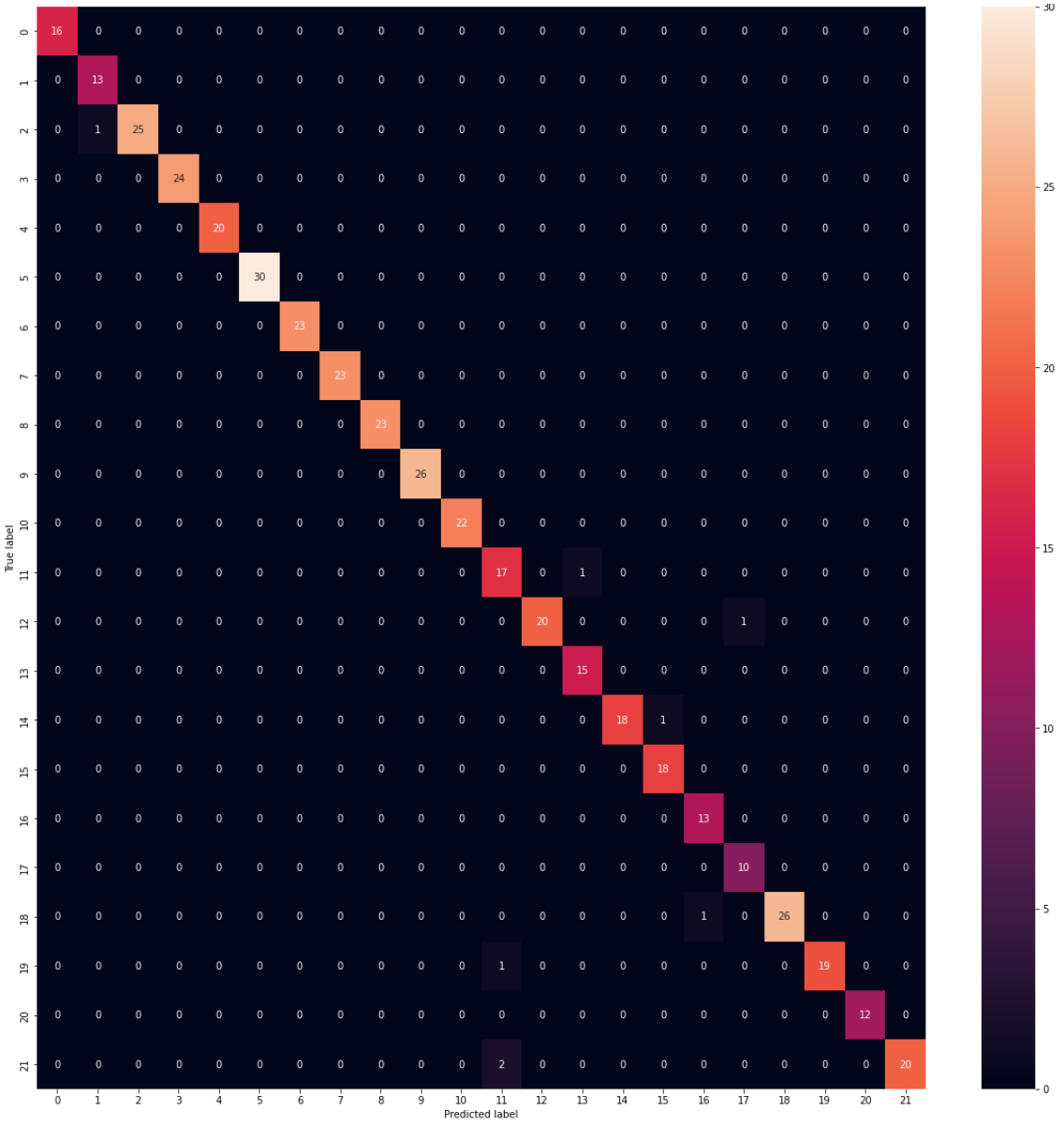


Figure 4.12 The confusion matrix of the ELSDSR database of Model 2.

Figure 4.12 depicts the ELSDSR database's confusion matrix at 2 seconds. Since the total number of speaker samples analyzed was equal to (TP+TN+FP+FN = 520) and the true positive was equal to (TP+TN = 514), the accuracy is calculated as follows:

$$Accuracy = \frac{514}{520} \times 100\% = 98.84\%$$

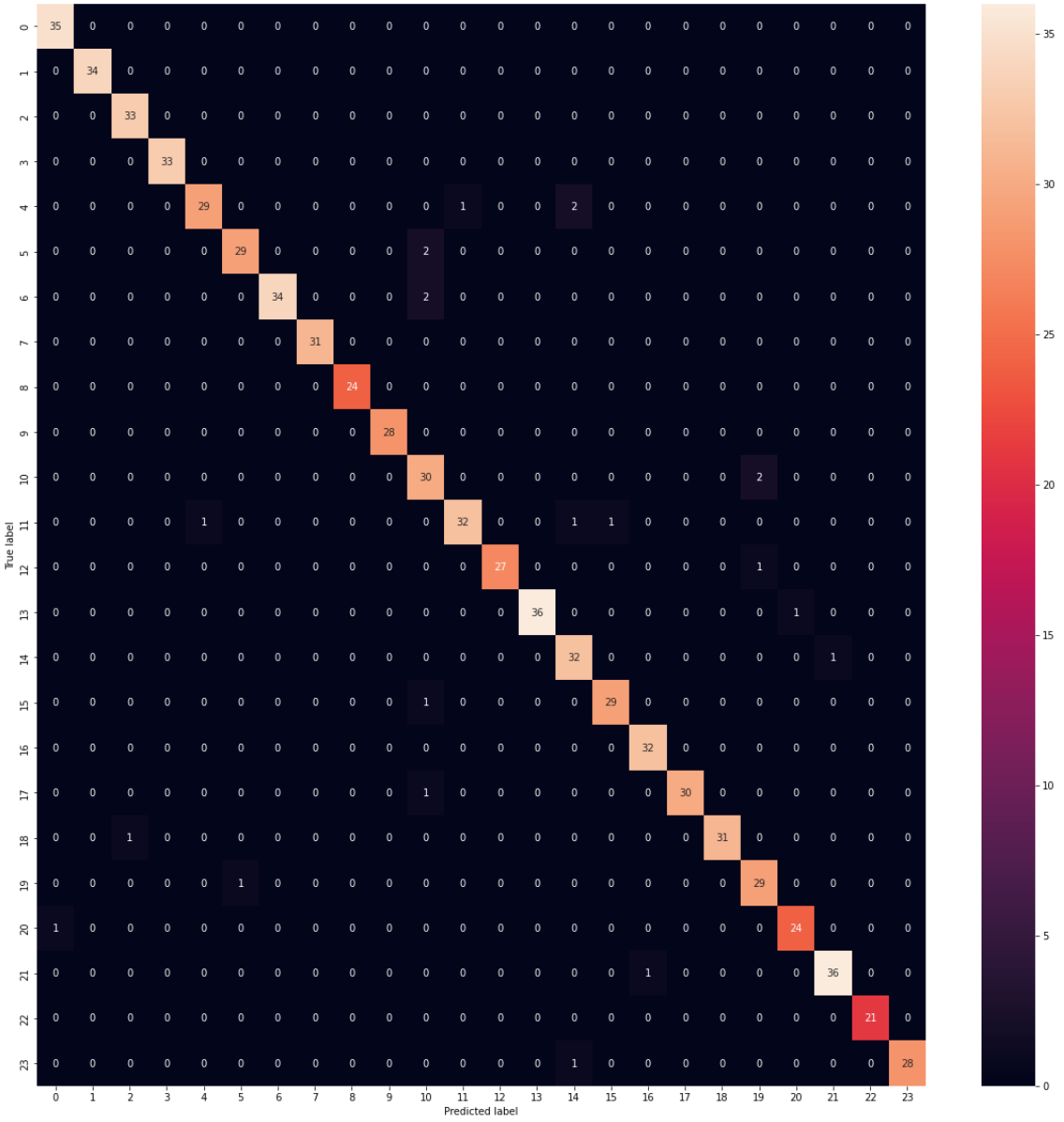


Figure 4.13 The confusion matrix of the SALU-AC database of Model 2.

Figure 4.13 shows the confusion matrix of the SALU-AC database for 2 seconds for 24 speakers. The total number of assessed speaker samples throughout this time period was 749 (TP+TN+FP+FN = 749). The real positive number (727) represented the sum of TP and TN. The results are found by:

$$Accuracy = \frac{727}{749} \times 100\% = 97.06\%$$

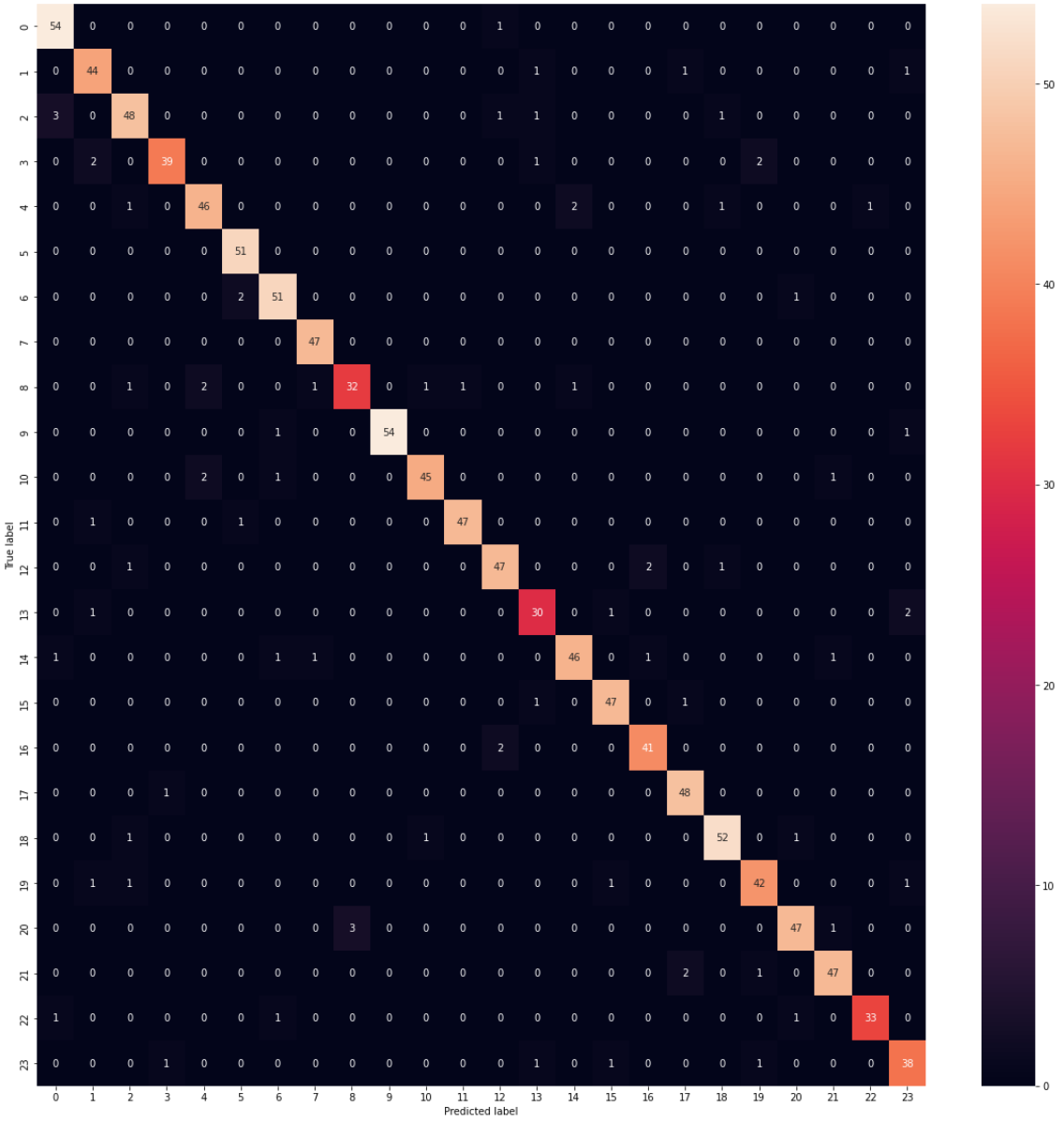


Figure 4.14 The confusion matrix of the RAVDESS database of Model 2.

Figure 4.14 depicts the confusion matrix of the RAVDESS database at 2 seconds. The total number of speaker samples analyzed was equal to (TP+TN+FP+FN = 1152), and the true positive was equal to (TP+TN = 1102). The accuracies are calculated by:

$$Accuracy = \frac{1102}{1152} \times 100\% = 95.65\%$$

As can see in Tables 4.1 and 4.2 the use of MFCC in conjunction with PCA as a technique of feature extraction is preferable to the use of MFCC separately as a method of feature extraction because it has the potential to increase the accuracy of the speaker identification system while simultaneously reducing the feature data dimension.

MFCC is a common feature extraction approach that transforms the speech signal into a set of cepstral coefficients that describe the signal's spectral envelope. These coefficients can then be used to analyze the speech signal. However, a few drawbacks are associated with the MFCC, one of which pertains to the substantial dimensionality of the feature vector. PCA is a technique that aims to reduce the dimensionality of a feature vector. by selecting the principal components that are the most important and represent the majority of the variation in the data.

The performance and speed of the classification process and the speaker identification system's accuracy may be improved by combining MFCC and PCA. This allows the feature vector to be reduced to a lower dimension without losing a significant amount of information, which in turn can enhance the accuracy of the speaker identification system.

Figure 4.15 illustrates the histograms indicating the difference in accuracy percentages for the first and second model for each type of databases utilized for each chosen time duration.

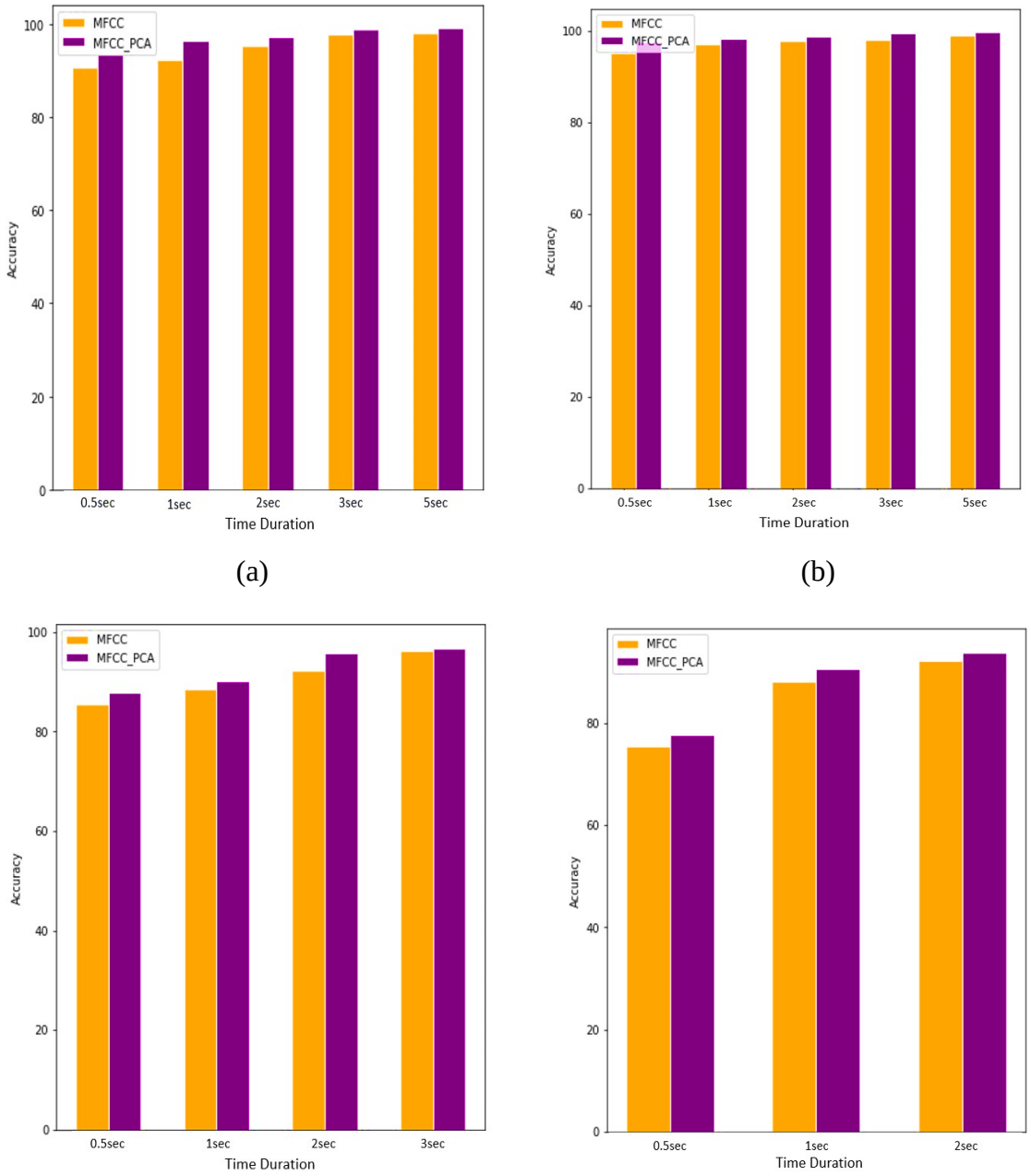


Figure 4.15 The histogram illustrates the difference between Model 1 and Model 2 of, (a) SALU-AC, (b) ELSDSR, (c) RAVDESS, and (d) TIMIT.

4.2.3 The Experimental Results for Model3

The third system described in this thesis involves the combination of two-dimensional DWT for the feature extraction phase and CNN for the classification phase, as explained in Chapter Three. The model 3 methodology is used for all datasets included in this thesis, namely SALU-

AC, ELSDSR, RAVDESS, and TIMIT. K-fold cross-validation is used to evaluate the performance of the model 3, with k values of 2 and 5 for examination. The results of the model 3 experiment are shown in Table 4.3.

Table 4.3 The recognition rates for the Model3.

K-fold	Database	Accuracy (%)				
		0.5 sec.	1 sec.	2 sec.	3 sec.	5 sec.
K=2	SALU-AC	67.17	73.16	79.23	80.38	85.46
	ELSDSR	74.44	85.40	89.17	91.83	93.90
	RAVDESS	63.46	67.55	70.79	82.01	--
	TIMIT	62.51	66.09	72.57	--	--
K=5	SALU-AC	86.08	88.35	91.61	92.41	93.85
	ELSDSR	91.73	92.45	93.49	93.54	94.14
	RAVDESS	78.48	86.02	89.56	90.53	__
	TIMIT	71.85	80	86.99	__	__

As shown in Table 4.3, the performance of 2D-DWT and CNN gives good accuracy in the proposed system. This is because these algorithms can minimize computational cost, dimensionality, and complexity and automatically extract robust and discriminative features. Although it would seem that DWT in two dimensions is capable of providing good accuracy, this may not be the best or only solution. Therefore, to get superior outcomes, it is recommended to combine DWT with an additional feature extraction approach, such as PCA.

Based on the results shown in Table 4.4, it is clear that using a value of k=5 yielded better results in comparison to k=2. The Figures will just be provided for the unique instance in which the value of k is fixed at 5 since this particular configuration has produced outstanding outcomes. The results of the testing phase are shown in Figures 4.16-4.19, illustrating the

relationship between accuracy and the number of epochs for all databases analyzed in this thesis.

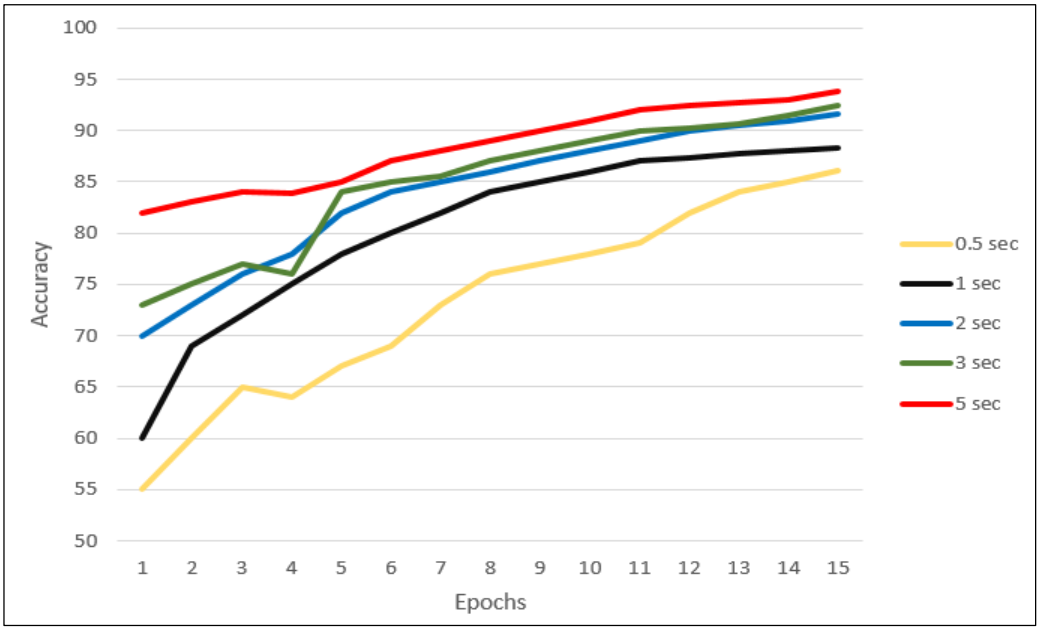


Figure 4.16 The performance of the SALU-AC database for the chosen durations for Model 3.

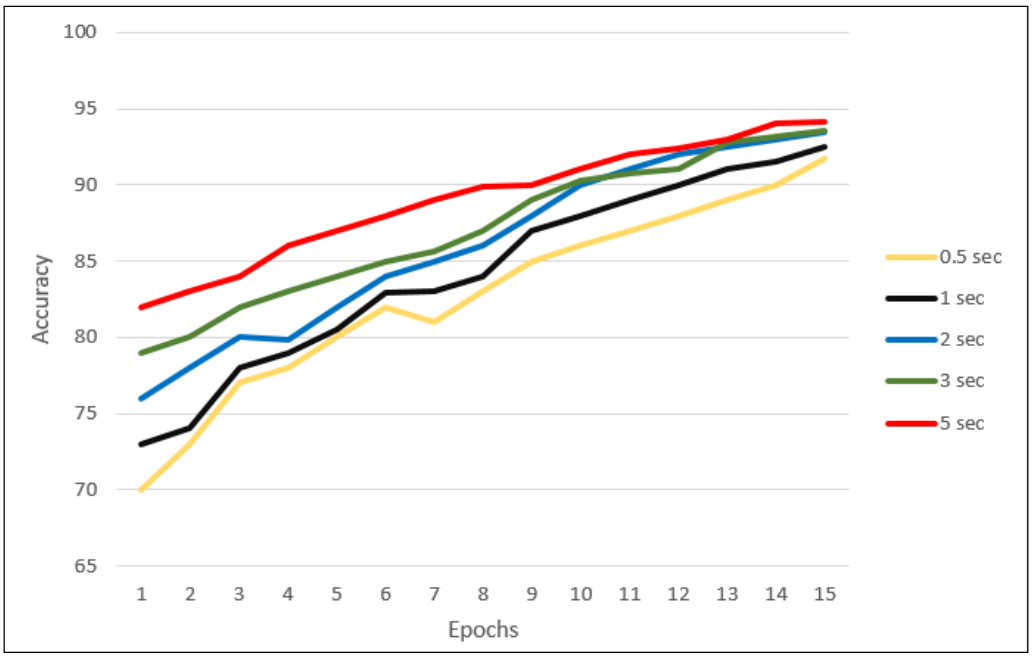


Figure 4.17 The performance of the ELSDSR database for the chosen durations for Model 3.

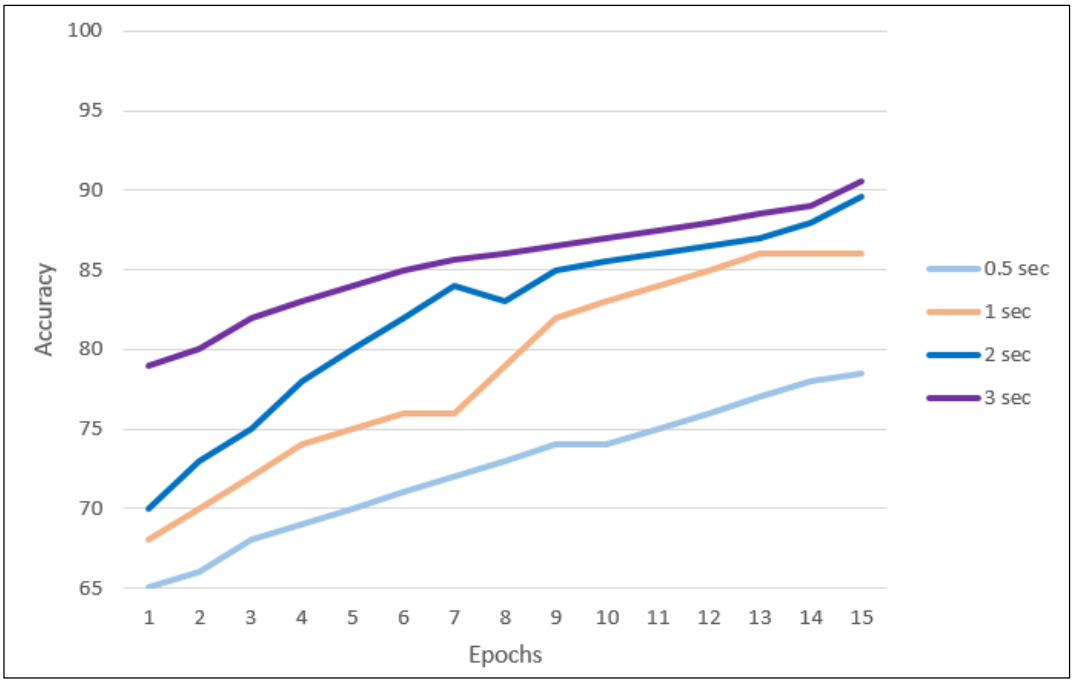


Figure 4.18 The performance of the RAVDESS database for the chosen durations for Model 3.

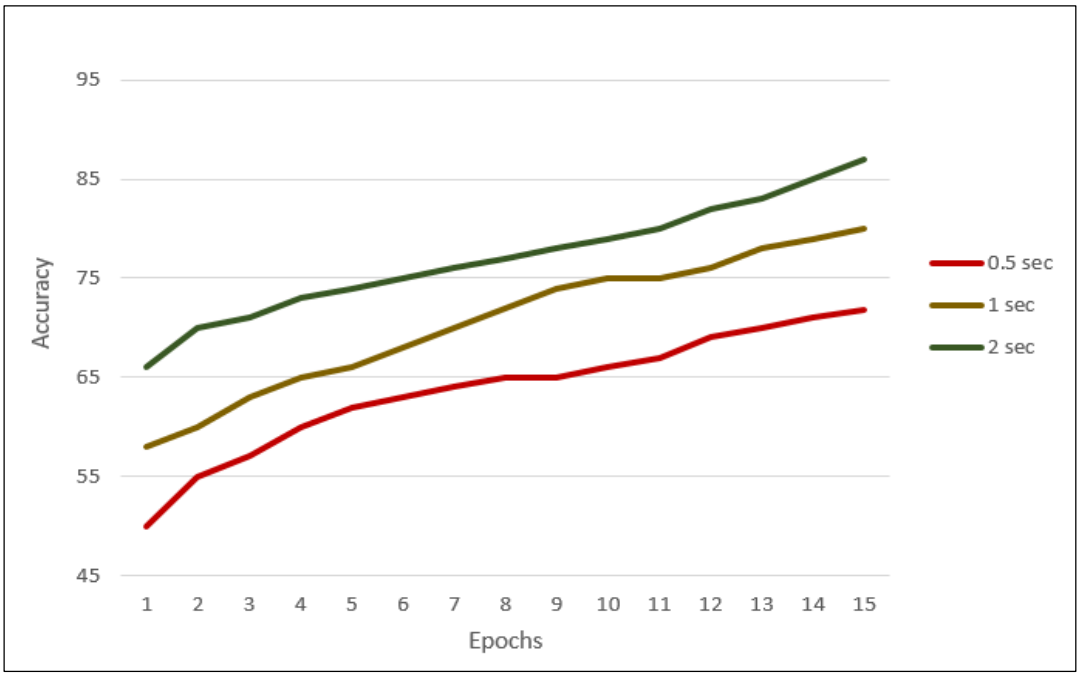


Figure 4.19 The performance of the TIMIT database for the chosen durations for Model 3.

The use of the confusion matrix, a method employed for accuracy computation, is shown in Figures 4.20-4.22. The purpose of these statistics is to visually demonstrate the technique for calculating accuracy in the

SALU-AC, ELSDSR, and RAVDESS databases. The duration of the example will be constrained to a time frame of one seconds.

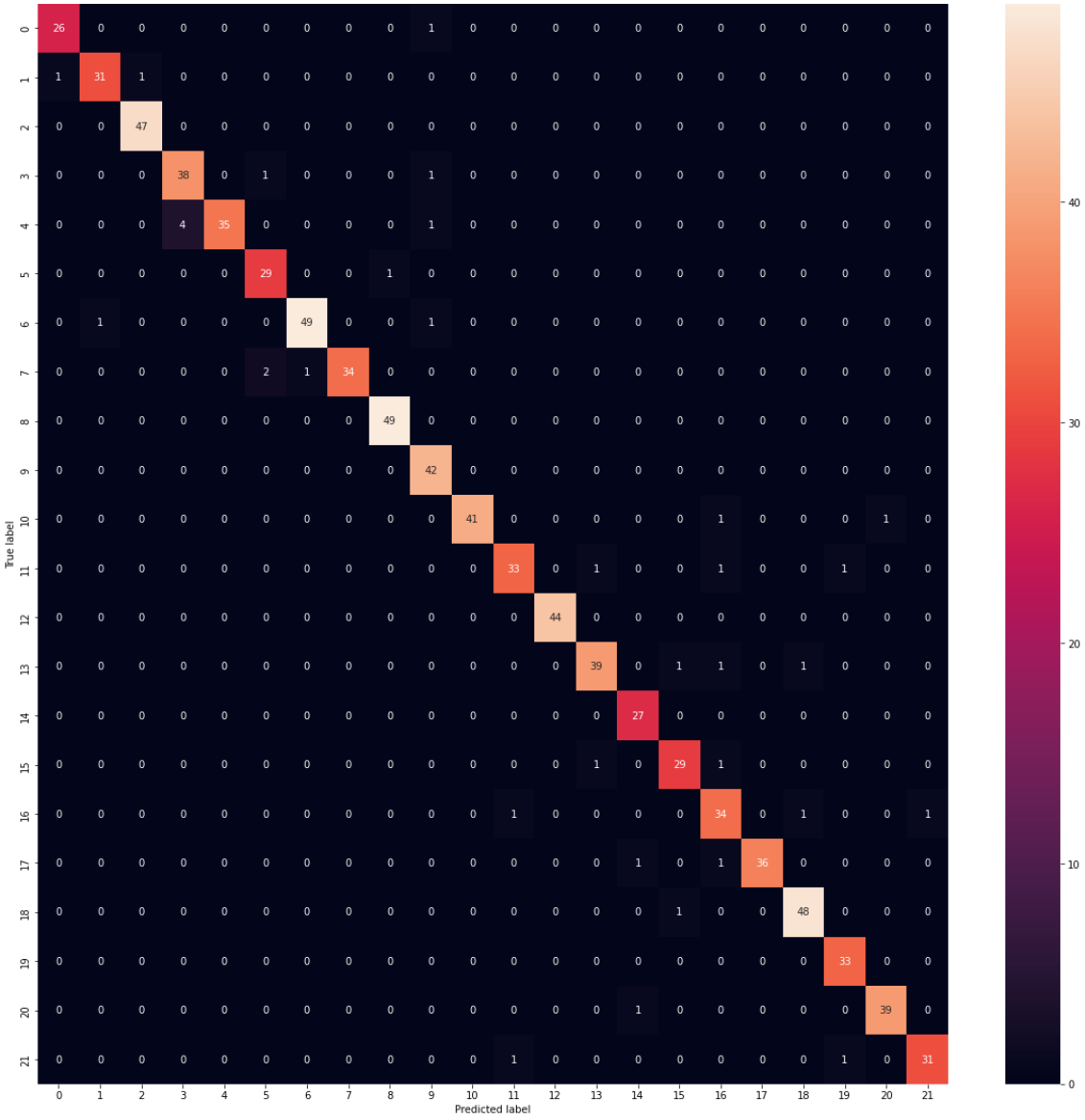


Figure 4.20 The confusion matrix of the ELSDSR database of Model 3.

Figure 4.20 depicts the confusion matrix of the ELSDSR database at 1 second. In 1 second, the total number of speaker samples evaluated was 968 (TP+TN+FP+FN = 968), and the number of true positives was 895 (TP+TN = 895) The accuracies are calculated as follows:

$$Accuracy = \frac{895}{968} \times 100\% = 92.45\%$$

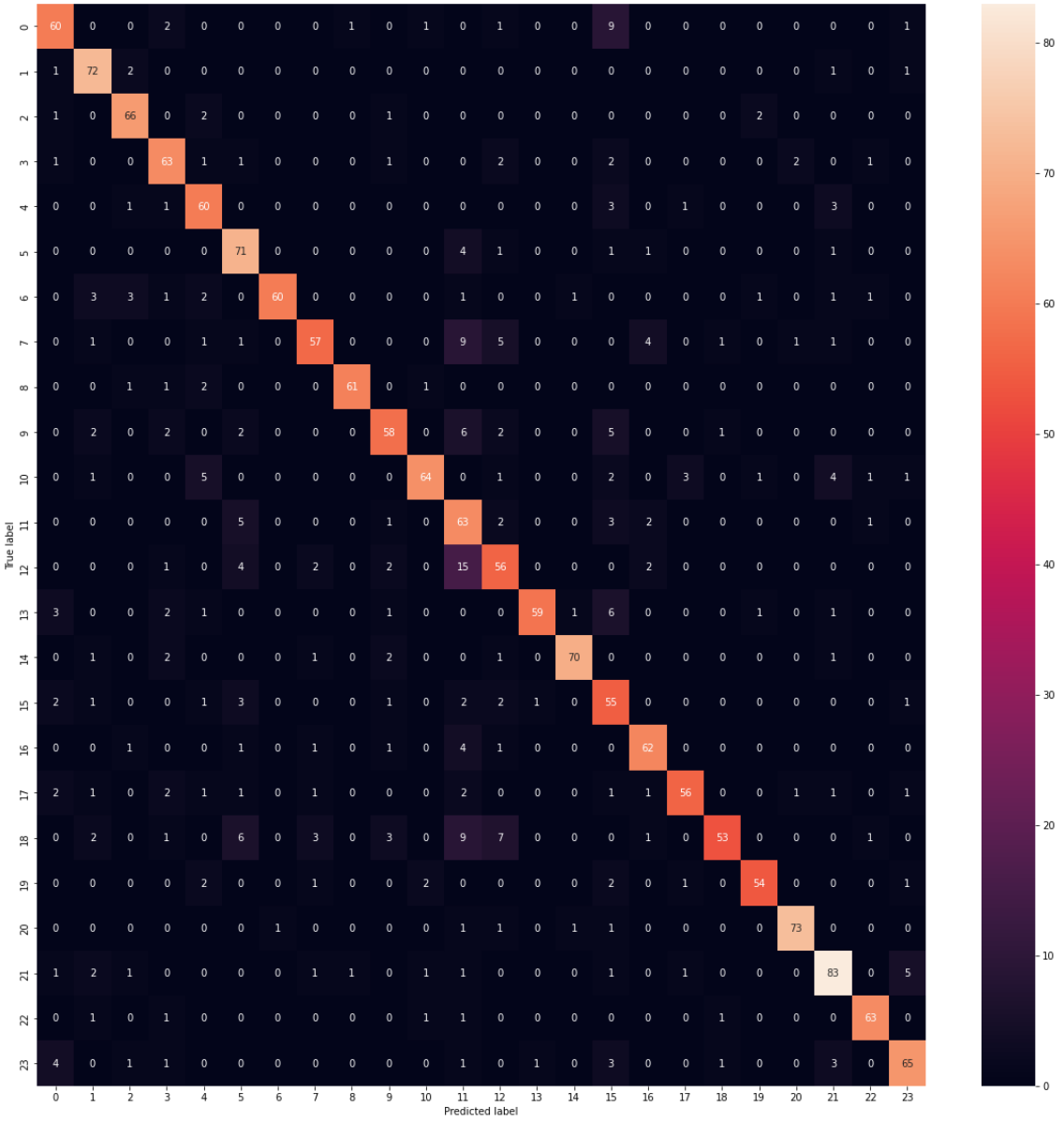


Figure 4.21 The confusion matrix of the RAVDESS database of Model 3.

Figure 4.21 shows the confusion matrix for the RAVDESS database at one second. Given that there were a total of (TP+TN+FP+FN=1824) speaker samples analyzed, and that the true positive was equivalent to 1569 true positives (TP+TN=1569). The rates are obtained by:

$$Accuracy = \frac{1569}{1824} \times 100\% = 86.02\%$$

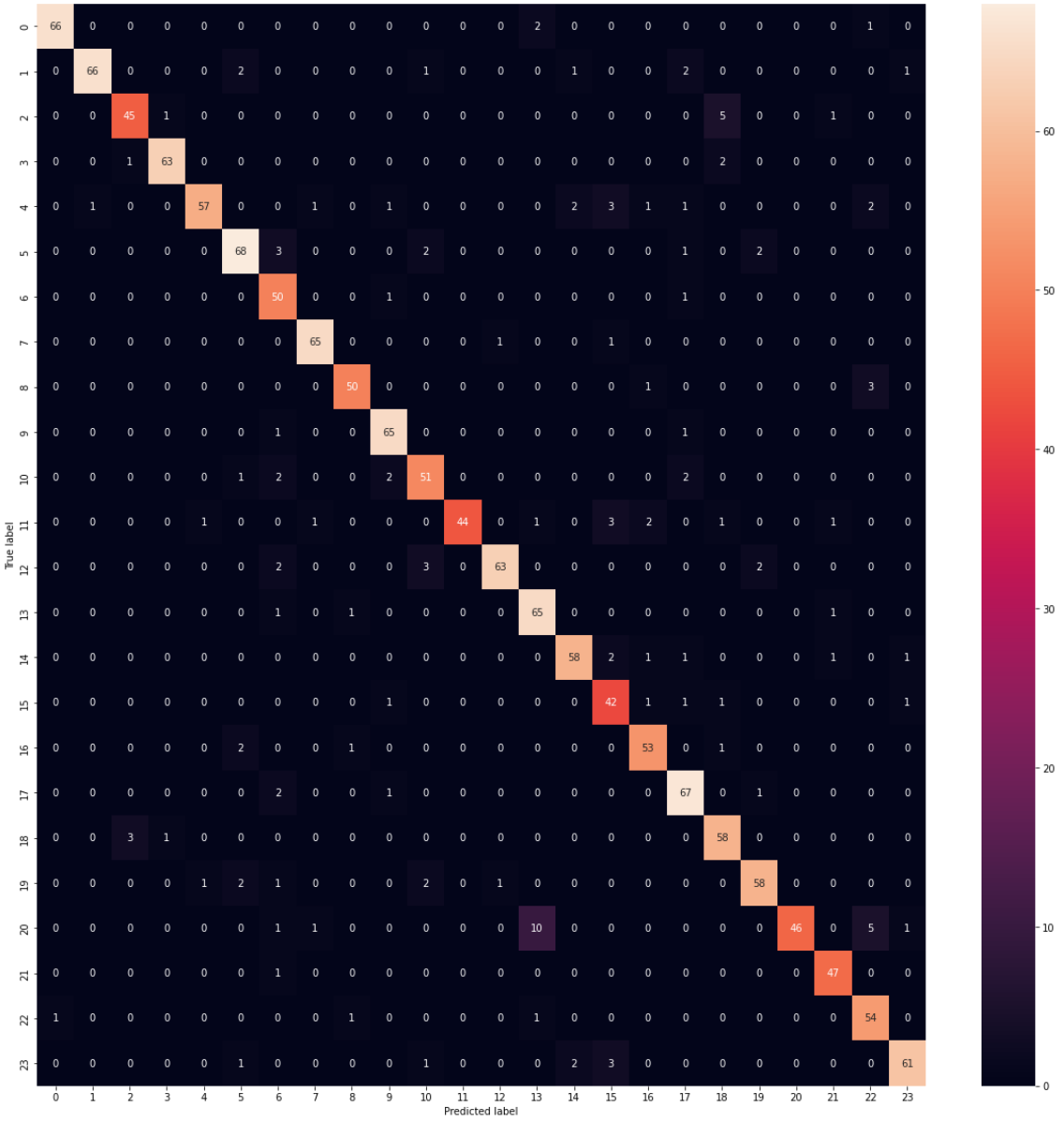


Figure 4.22 The confusion matrix of the SALU-AC database of Model 3.

The confusion matrix for the SALU-AC database at one second only for 24 speakers is shown in Figure 4.22. Since the total number of speaker samples examined at one second was (TP+TN+FP+FN =1498) and the true positive was (TP+TN =1362). The results are computed by:

$$Accuracy = \frac{1362}{1498} \times 100\% = 90.92\%$$

4.2.4 The Experimental Results for Model4

The fourth method offered in this thesis includes the integration of two-dimensional DWT with PCA for the feature extraction phase, followed by the utilization of CNN for the classification phase, as explained in Chapter Three. The PS4 approach is used for all datasets included in this thesis, namely SALU-AC, ELSDSR, RAVDESS, and TIMIT. The performance evaluation of the model 4 is conducted using k-fold cross-validation, with k-values of 2 and 5 being used for analysis. The results of the model 4 are shown in Table 4.4.

Table 4.4 The recognition rates for the Model4.

K-fold	Database	Accuracy (%)				
		0.5 sec.	**1 sec.**	**2 sec.**	**3 sec.**	**5 sec.**
K=2	SALU-AC	68.36	74.45	81.20	83.48	86.22
	ELSDSR	81.98	87.80	89.25	93	94.13
	RAVDESS	66.48	73.66	76.16	81.70	--
	TIMIT	65.30	71.19	76.66	--	--
K=5	SALU-AC	89.10	91.72	92.25	93.69	94.59
	ELSDSR	93.86	94.52	95.06	95.55	96.74
	RAVDESS	81.82	87.19	91.32	92.90	--
	TIMIT	74.28	84.88	89.04	--	--

Based on the results presented in Table 4.4, it is apparent that utilizing a value of k=5 yielded more favorable results in comparison to k=2. The Figures will be exclusively presented for the specific case where the value of k = 5. Figures 4.23-4.26 illustrate the results obtained during the testing phase, displaying the relationship between accuracy and the number of epochs across all databases analyzed in this thesis.

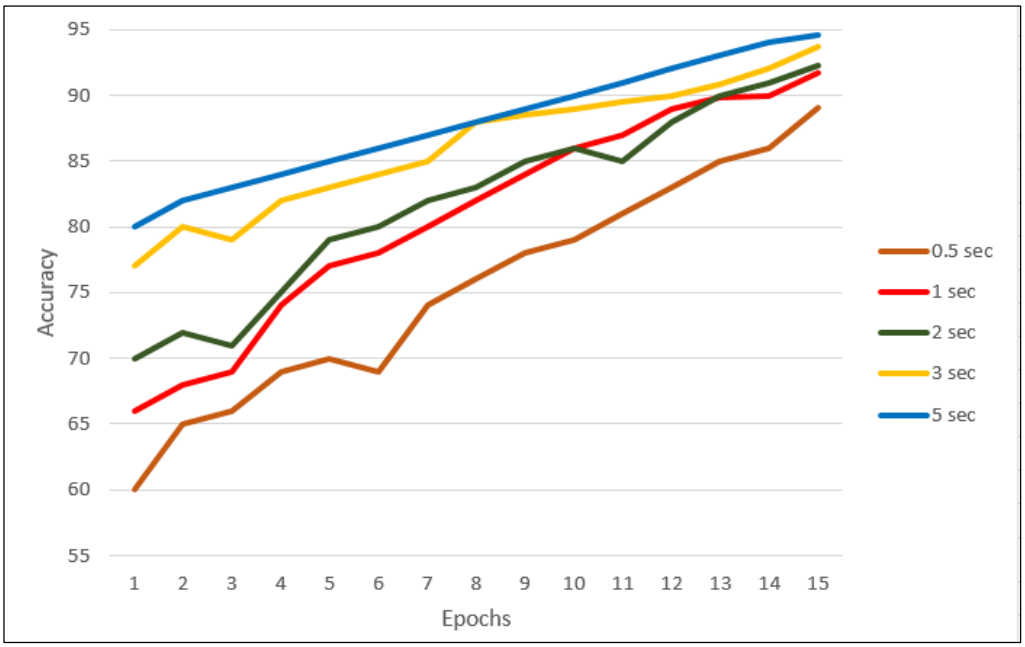


Figure 4.23 The performance of the SALU-AC database for the chosen durations for Model 4.

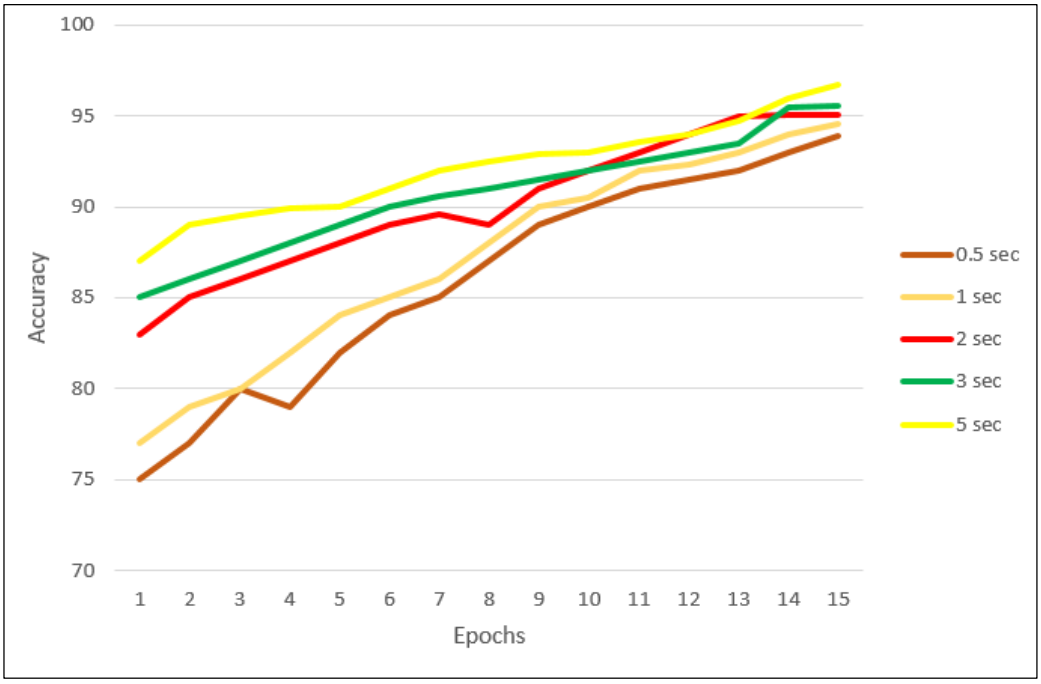


Figure 4.24 The performance of the ELSDSR database for the chosen durations for the Model 4.

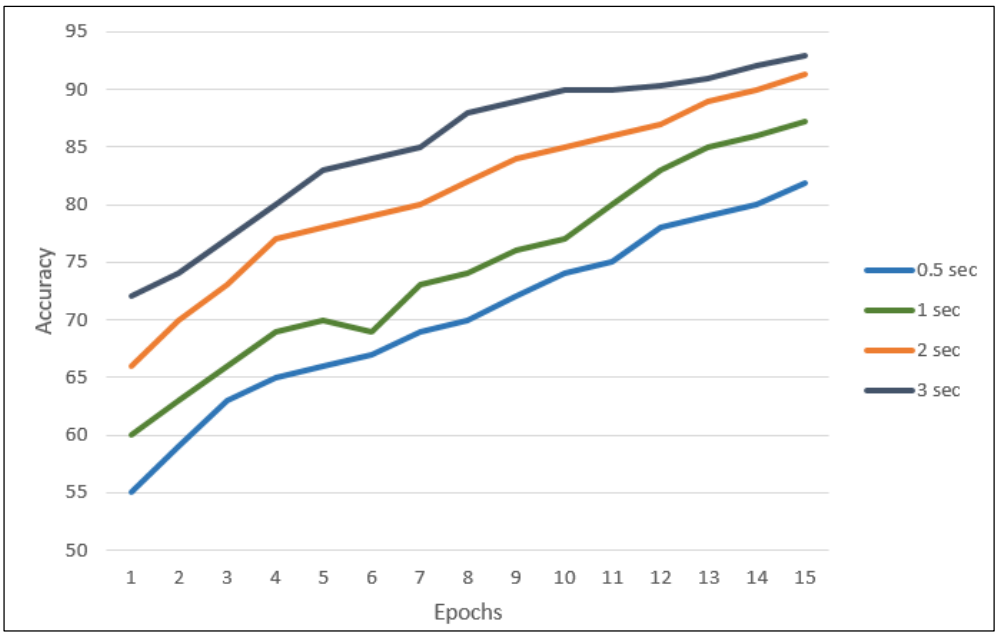


Figure 4.25 The performance of the RAVDESS database for the chosen durations for Model 4.

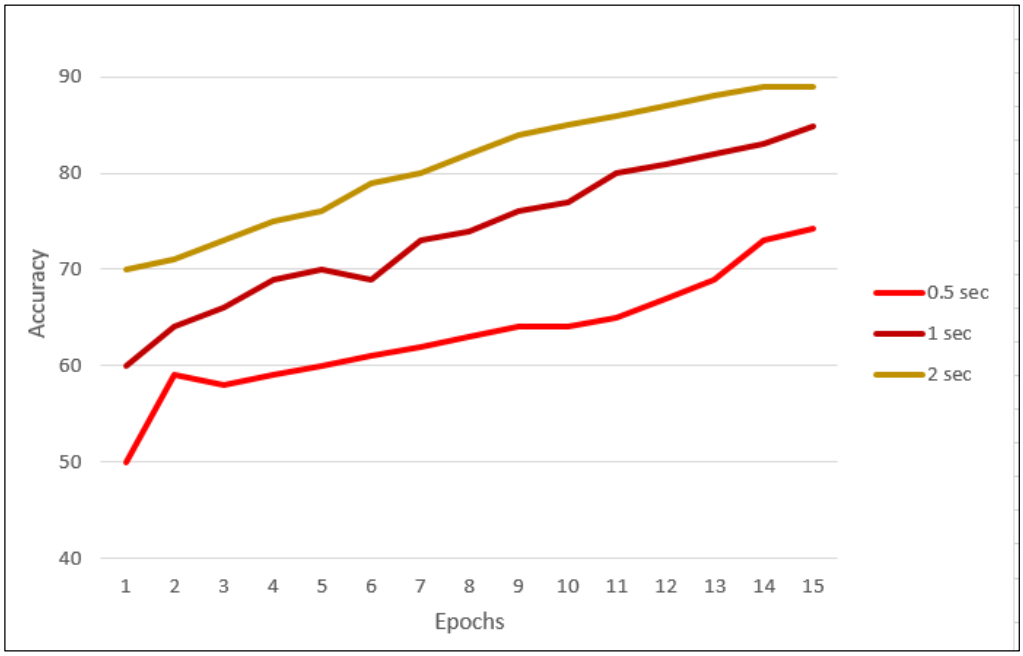


Figure 4.26 The performance of the TIMIT database for the chosen durations for Model 4.

The use of the confusion matrix, an additional method employed for evaluating the accuracy, is shown in Figures 4.27-4.29. The purpose of these Figures is to show the process of calculating accuracy for the SALU-AC,

ELSDSR, and RAVDESS databases. The duration of the given example will be constrained to a time interval of 0.5 sec.

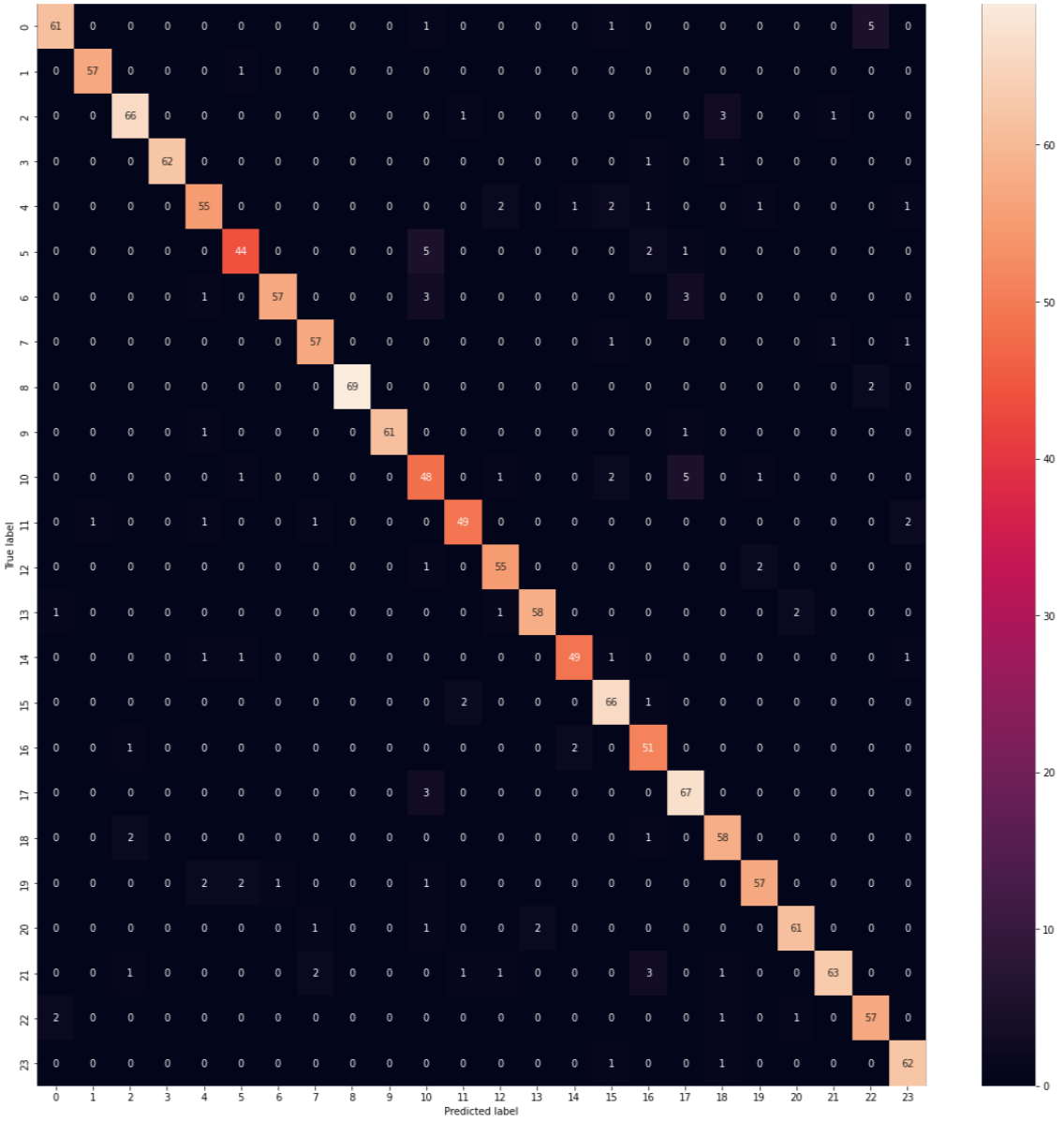


Figure 4.27 The confusion matrix of the SALU-AC database of Model 4.

Figure 4.27 displays the confusion matrix for the SALU-AC database at 0.5 sec for 24 speakers. Since there were 2995 speaker samples analyzed in total at 0.5 sec (TP+TN+FP+FN=2995). and 2780 true positives (TP+TN=2780). The accuracies are calculated as follows:

$$Accuracy = \frac{2780}{2995} \times 100\% = 92.82\%$$

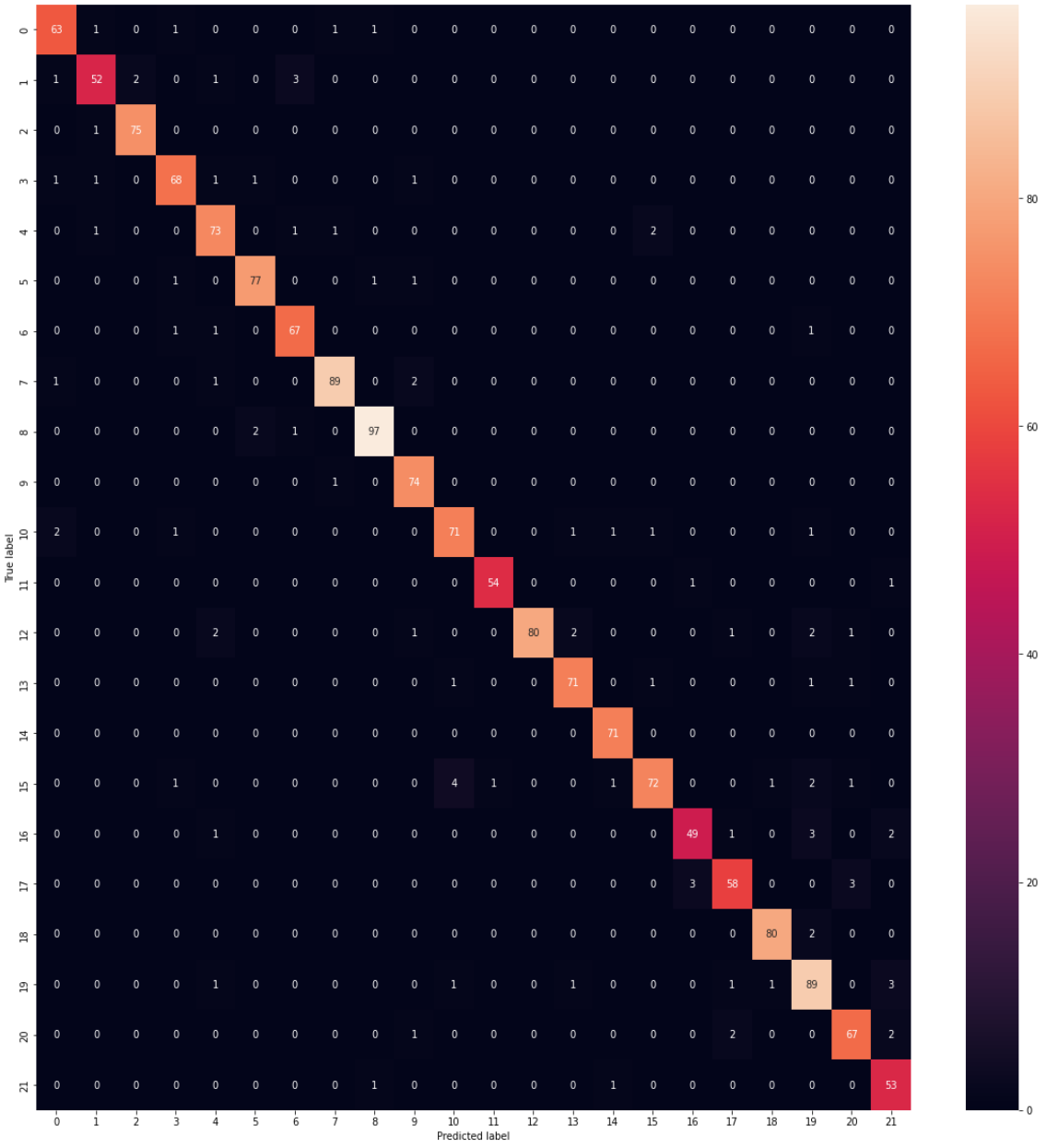


Figure 4.28 The confusion matrix of the ELSDSR database of Model 4.

The confusion matrix for the ELSDSR database is shown in Figure 4.28 at 0.5 seconds. The total number of speaker's sample of 0.5 sec, (TP+TN+FP+FN=1760). The true positives (TP+TN=1652). The accuracies are determined as follows:

$$Accuracy = \frac{1652}{1760} \times 100\% = 93.86\%$$

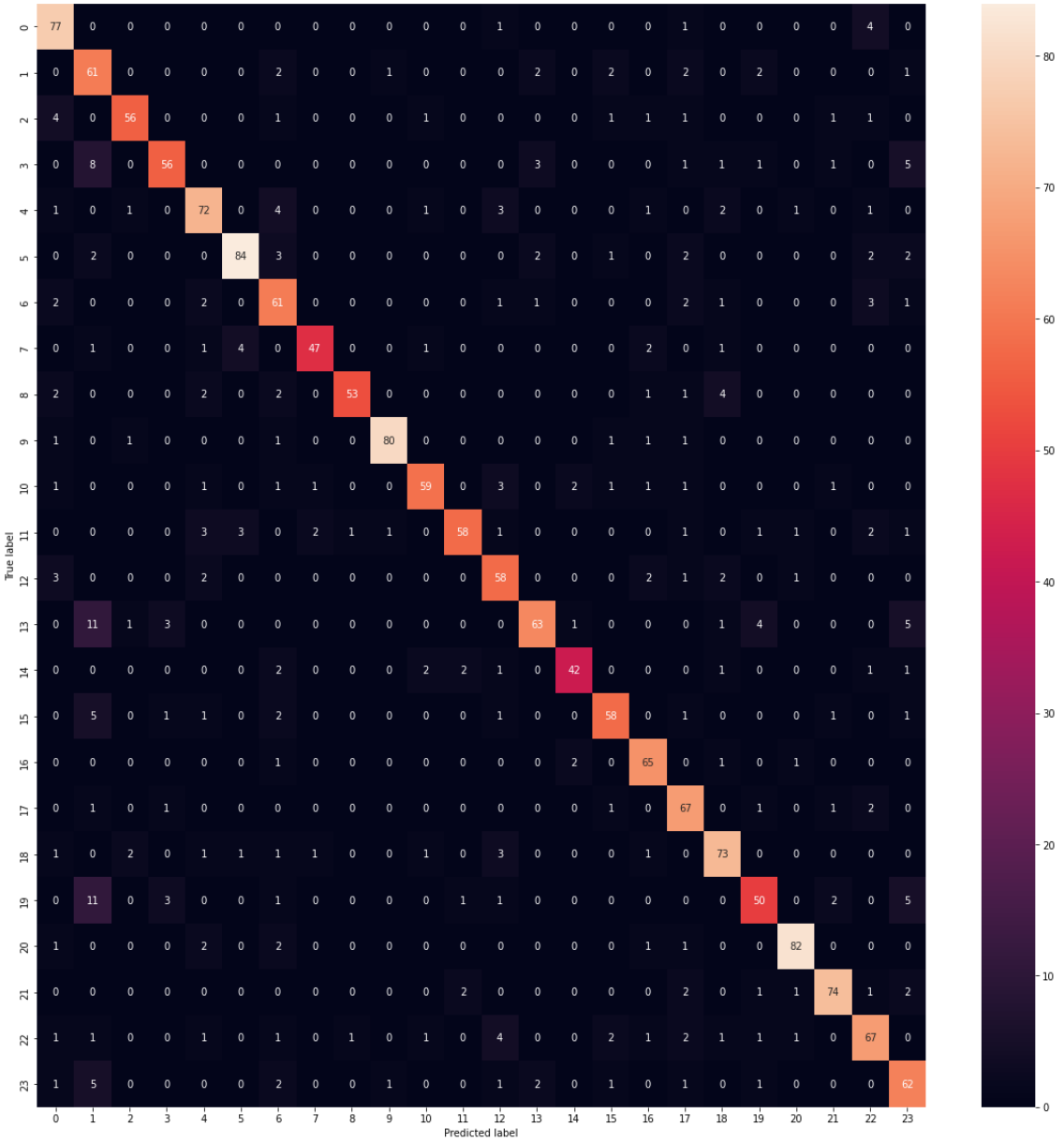


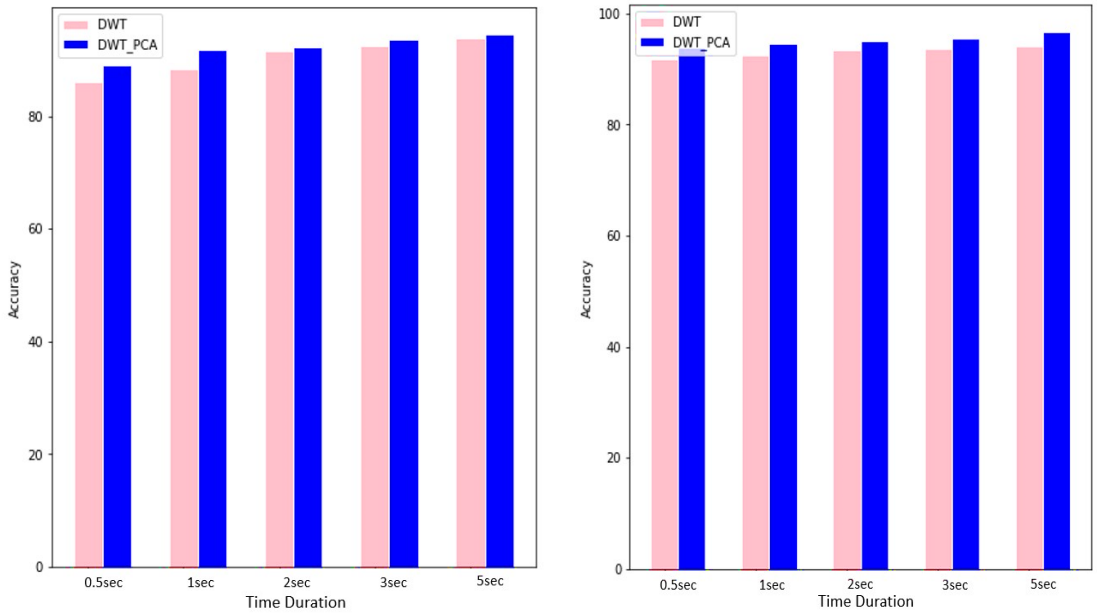
Figure 4.29 The confusion matrix of the RAVDESS database of Model 4.

The confusion matrix for the RAVDESS database is presented in Figure 4.29 at a duration of 0.5 sec. A comprehensive analysis was conducted on a total of 3648 speaker samples, each lasting 0.5 sec, as indicated by the equation TP+TN+FP+FN=3648. The accuracy can be determined by considering the number of true positives (TP+TN), which in this case is 2985. Based on Eq. 2.45, the accuracy can be expressed as follow:

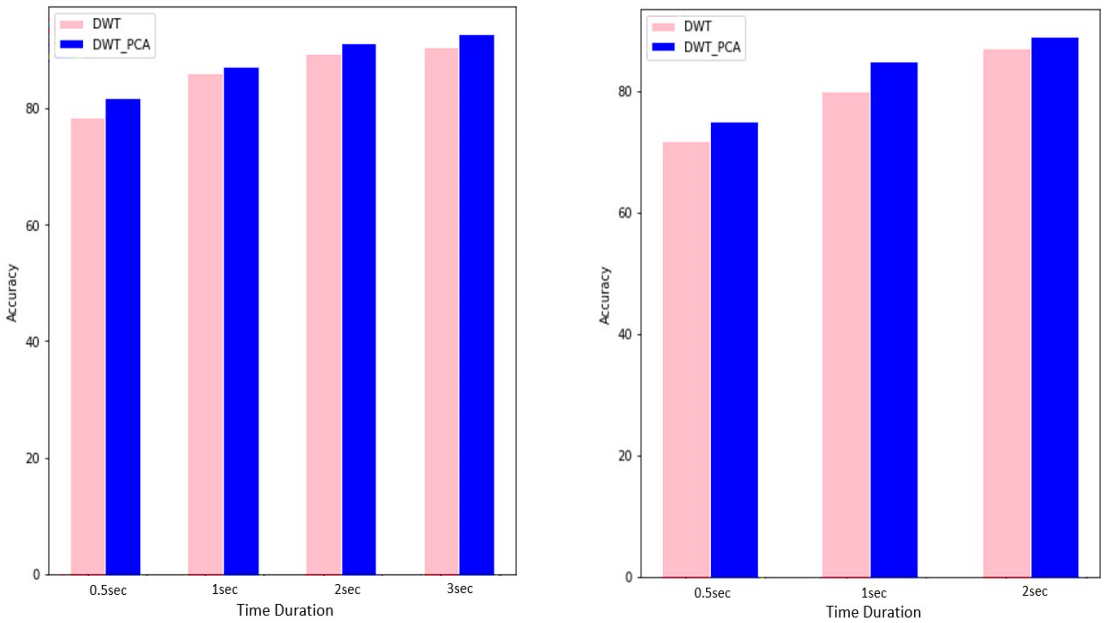
$$Accuracy = \frac{2985}{3648} \times 100\% = 81.8\%$$

As seen in the Table 4.3-4.4, the accuracy of the proposed hybrid scheme for the 2D DWT with PCA was superior to that of the 2D DWT scheme presented. Firstly, using 2D DWT in conjunction with PCA may simplify and minimize the dimensionality of the data, which may increase the computing efficiency and speed up the classification process. Secondly, 2D DWT with PCA may improve the classification's accuracy and performance by reducing data redundancy while maintaining the most significant discriminative features. These previously mentioned advantages can be accomplished by preserving the principal components of the principal component analysis. Finally, 2D DWT with PCA may retain the speech signals' spatial correlation and energy compaction, which may capture speaker-specific information more accurately than 2D DWT.

Figure 4.30 displays histograms illustrating the variation in accuracy percentages between the model 3 and model 4 across different types of databases for each specified period.



(a) (b)



(c) (d)

Figure 4.30 The histogram presents the contrast between Model 3 and Model 4 of, (a) SALU-AC; (b) ELSDSR; (c) RAVDESS; and (d) TIMIT.

4.2.5 The Experimental Results for Model5

The fifth proposed system in this thesis involves the use of a hybrid approach, combining MFCC with two-dimensional DWT for the purpose of feature extraction. Subsequently, a CNN is employed for the classification phase, as detailed in Chapter Three. The model 5 methodology is used for all datasets included in this thesis, namely SALU-AC, ELSDSR, RAVDESS, and TIMIT. The performance assessment of the model 5 model is carried out via the use of k-fold cross-validation, whereby k-values of 2 and 5 are employed for the purpose of analysis. The findings of the model 5 are shown in Table 4.5.

Table 4.5 The recognition rates for the Model5.

K-fold	Database	Accuracy (%)				
		0.5 sec.	**1 sec.**	**2 sec.**	**3 sec.**	**5 sec.**
K=2	SALU-AC	90.23	93.70	96.56	97.81	98.06
	ELSDSR	96.11	96.72	97.88	98.20	98.96
	RAVDESS	85.60	88.03	93.98	95.07	--
	TIMIT	71.22	80.46	86.37	--	--
K=5	SALU-AC	96.61	97.49	98.55	98.84	99.22
	ELSDSR	98.23	99.12	99.40	99.43	99.82
	RAVDESS	91.34	94.75	97.20	97.76	__
	TIMIT	90.38	93.62	96.02	__	__

According to the results presented in Table 4.5, the hybrid system, which incorporated a combination of MFCC and 2D-DWT as the feature extraction phase, along with CNN as the classifier, demonstrated an excellent level of performance. This can be attributed to the reduction in the dimensionality of the feature vector achieved by applying 2D-DWT on the MFCC matrix and selecting the most pertinent coefficients. To improve the feature extraction process against distortion, a technique involving the utilization of 2D-DWT is employed. This technique consists in decomposing the MFCC matrix into distinct frequency bands. The spatial and spectral information of the input signal can be captured by employing the MFCC technique to extract the cepstral features and utilizing the 2D-DWT method to extract the wavelet features.

Based on the findings presented in Table 4.5, it can be inferred that the utilization of MFCC in conjunction with 2D-DWT yields superior results compared to the use of MFCC independently and the combination of MFCC with PCA. This phenomenon arises due to its potential to enhance the

precision and resilience of the system under various circumstances, retain a more significant amount of speaker-specific details, amplify the discriminatory capability of the MFCC features, and effectively manage non-stationary signals. In contrast, using MFCC alone or in conjunction with PCA, which assumes linearity in data correlation, may not yield comparable outcomes.

According to the results offered in Table 4.5, it is obvious that utilizing a value of k=5 yielded more favorable results in comparison to k=2. The Figures will be exclusively presented for the specific case where the value of k = 5. Figures 4.31-4.34 illustrate the results obtained during the testing phase, displaying the relationship between accuracy and the number of epochs across all databases analyzed in this thesis.

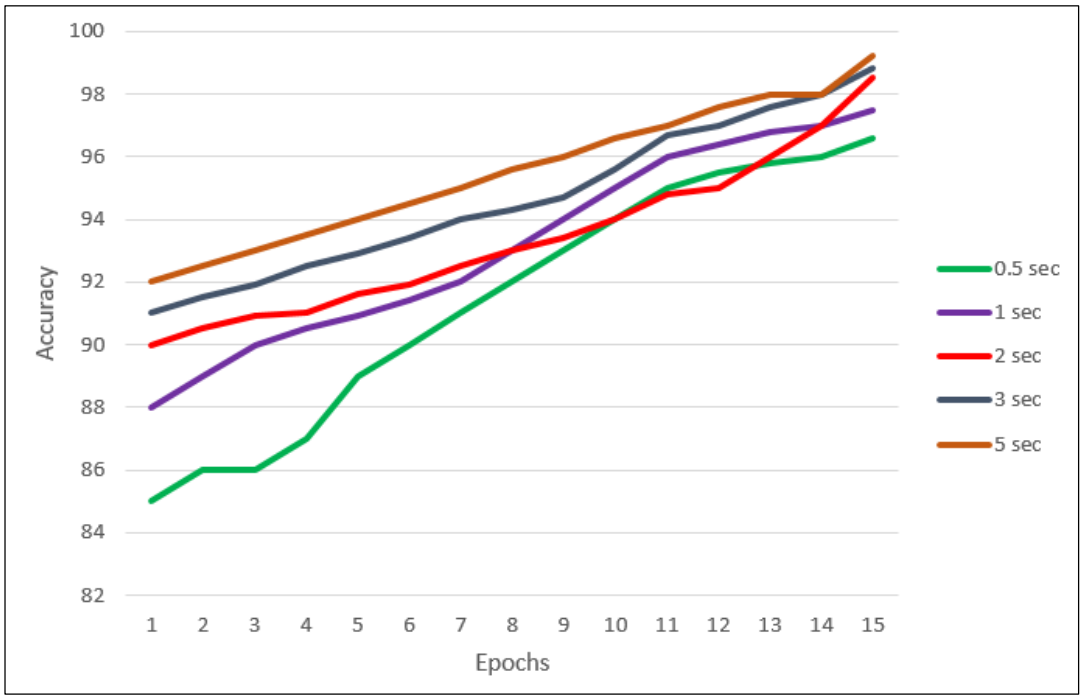


Figure 4.31 The performance of the SALU-AC database for the chosen durations for Model 5.

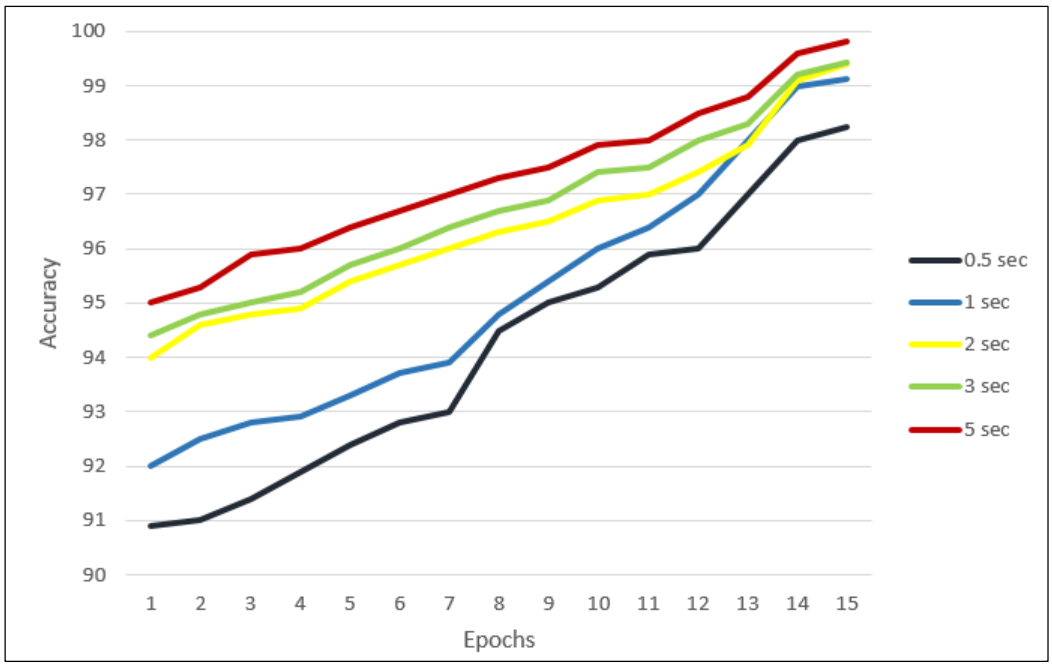


Figure 4.32 The performance of the ELSDSR database for the chosen durations for Model 5.

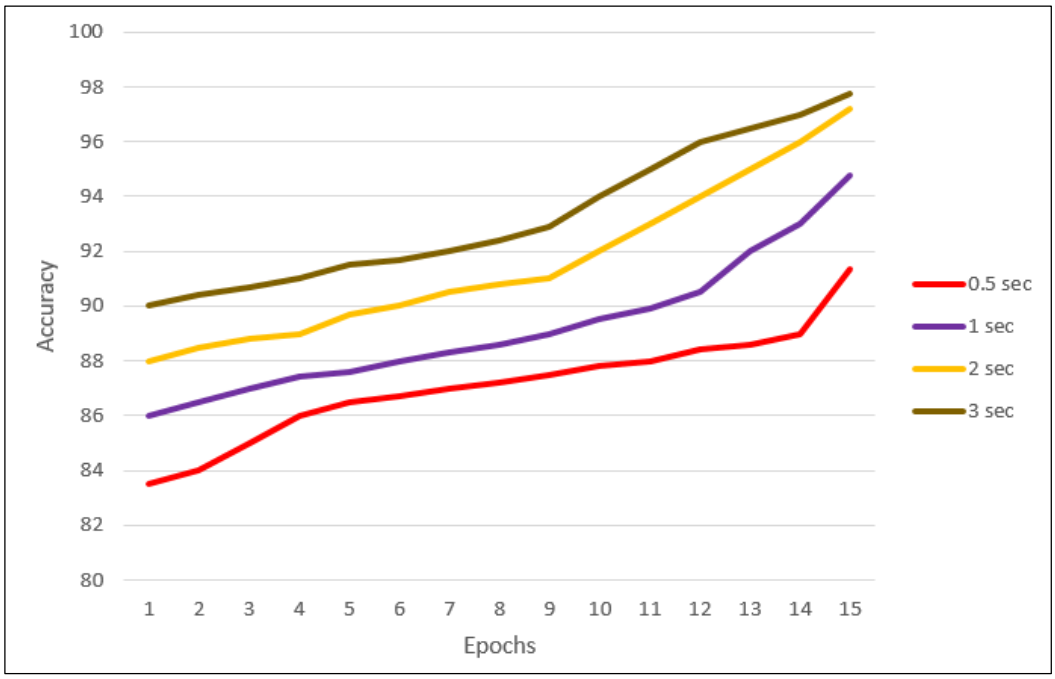


Figure 4.33 The performance of the RAVDESS database for the chosen durations for Model 5.

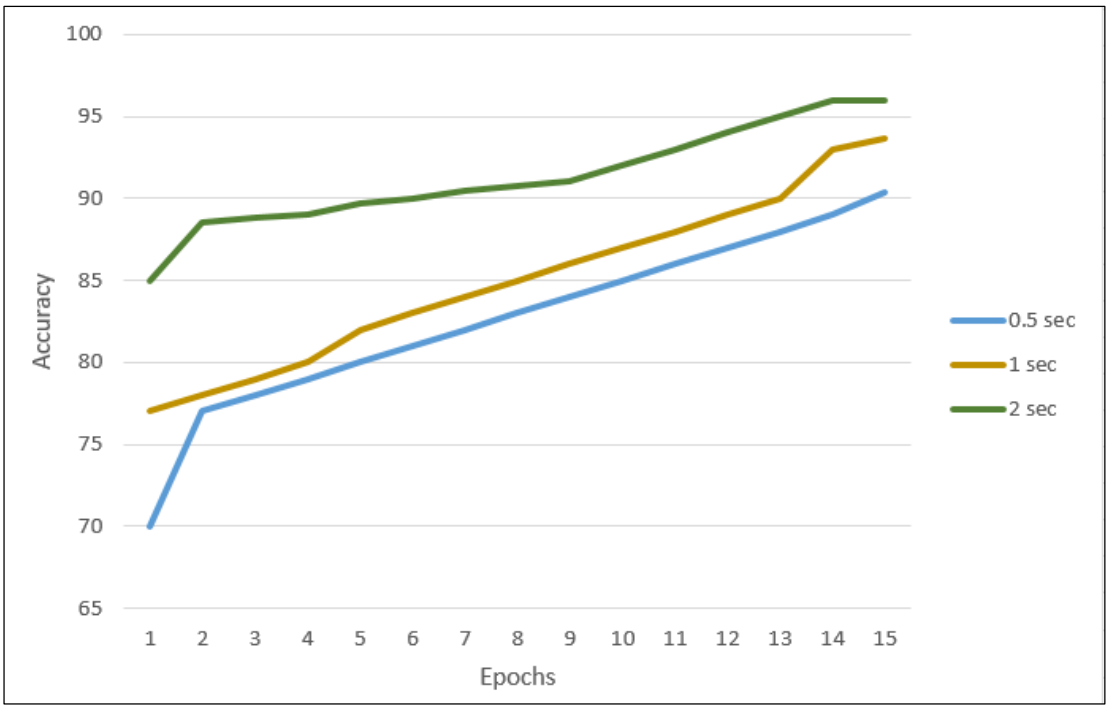


Figure 4.34 The performance of the TIMIT database for the chosen durations for Model 5.

The utilization of the confusion matrix, a supplementary technique utilized for assessing accuracy, is depicted in Figures 4.35-4.37. The objective of these Figures is to illustrate the procedure for calculating accuracy in the context of the ELSDSR, SALU-AC, and RAVDESS databases. The duration of the provided example will be limited to a time interval of 3 seconds.

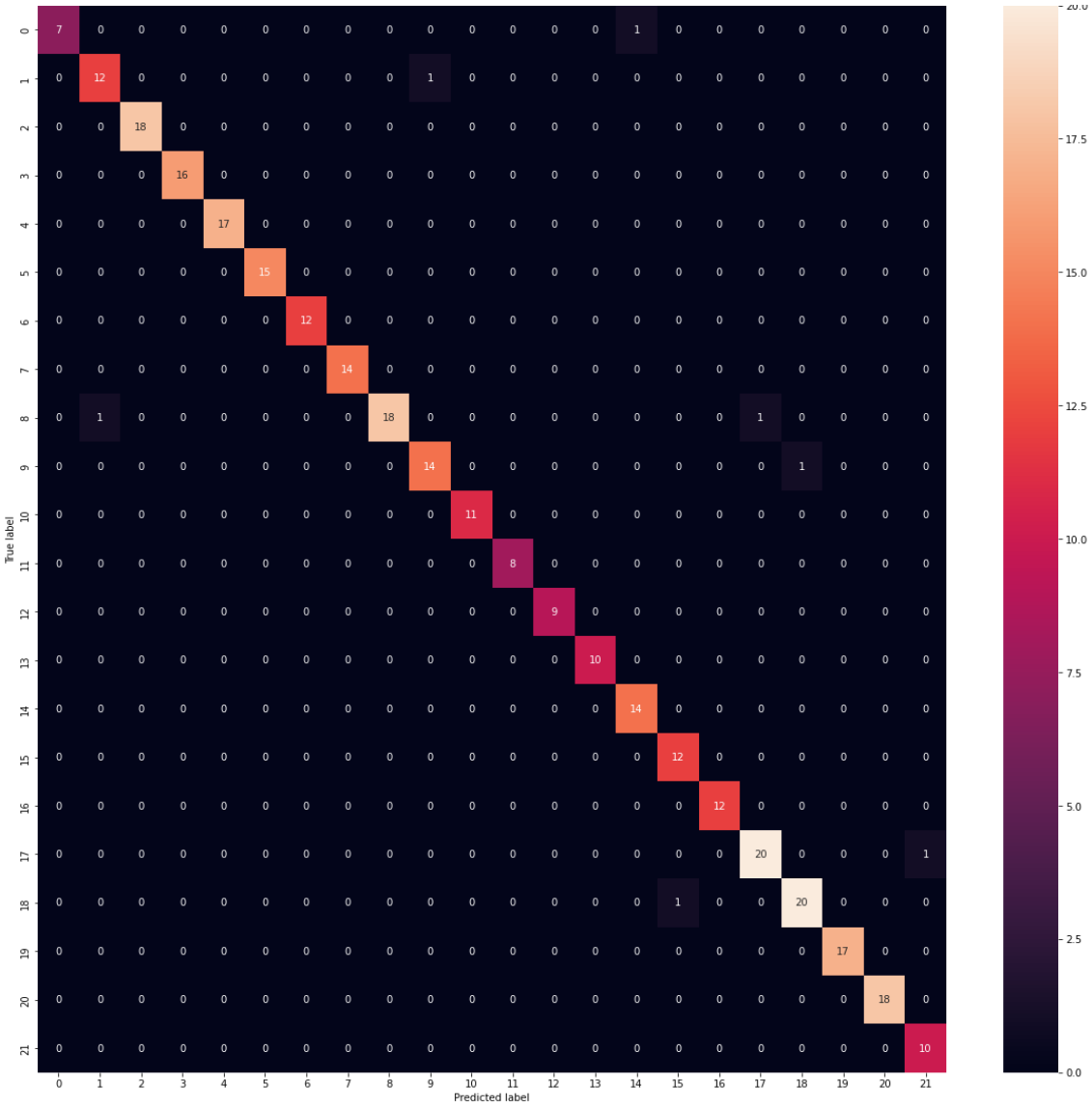


Figure 4.35 The confusion matrix of the ELSDSR database of Model 5.

Figure 4.35 displays the confusion matrix for the ELSDSR database over a length of 3 seconds. The equation TP+TN+FP+FN=352 indicates that a thorough analysis was performed on a total of 352 speaker samples, each lasting 3 seconds. The number of true positives (TP+TN), which in this example is 350. The accuracy can be calculated as:

$$Accuracy = \frac{350}{352} \times 100\% = 99.43\%$$

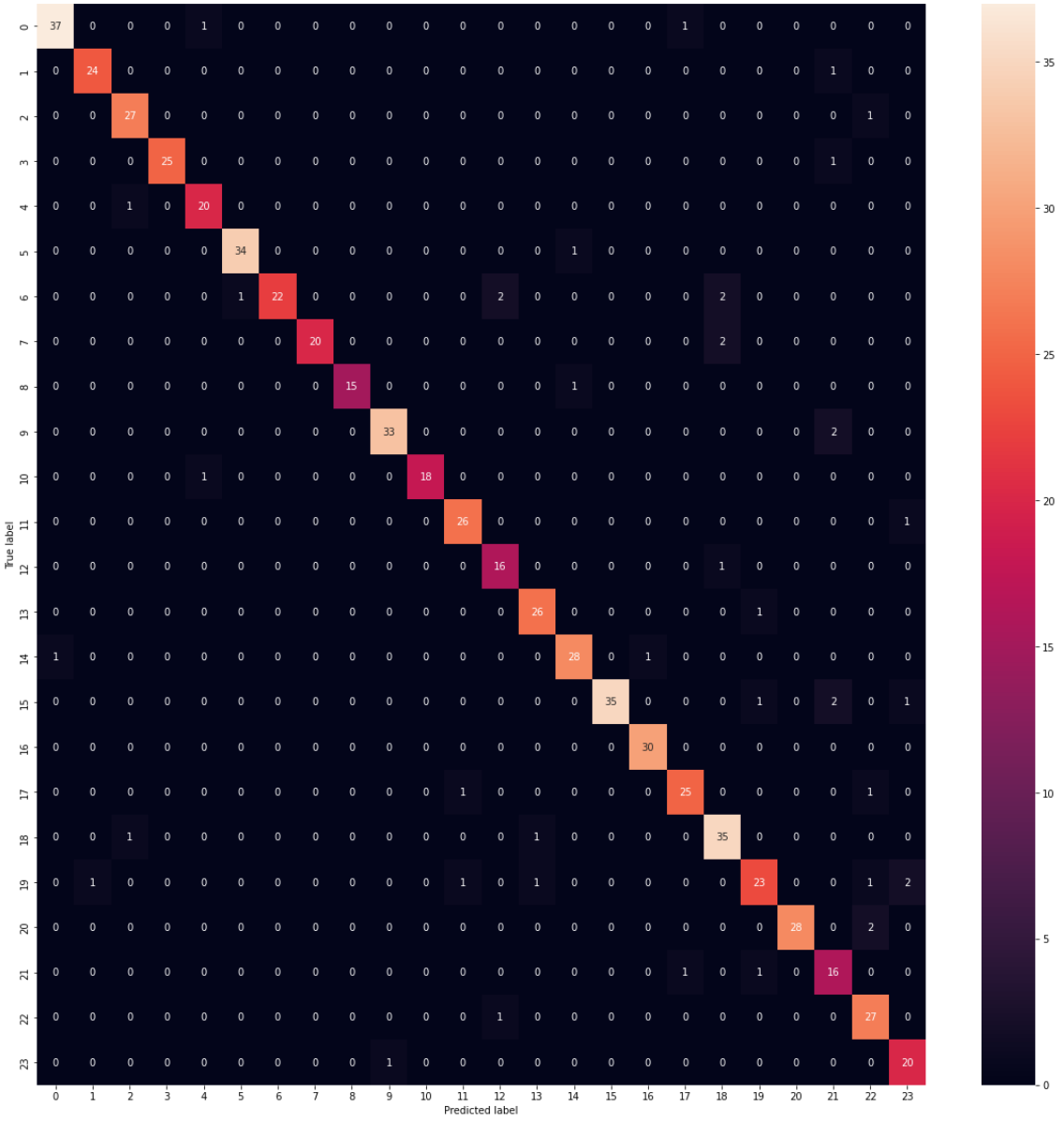


Figure 4.36 The confusion matrix of the RAVDESS database of Model 5.

The confusion matrix for the RAVDESS database is shown in Figure 4.36 for a duration of 3 seconds. A detailed study was conducted on a total of 672 speaker samples, each lasting 3 seconds, as shown by the equation TP+TN+FP+FN=672. Accuracy may be determined using the number of true positives (TP+TN= 657). The accuracies are calculated by:

$$Accuracy = \frac{657}{672} \times 100\% = 97.76\%$$

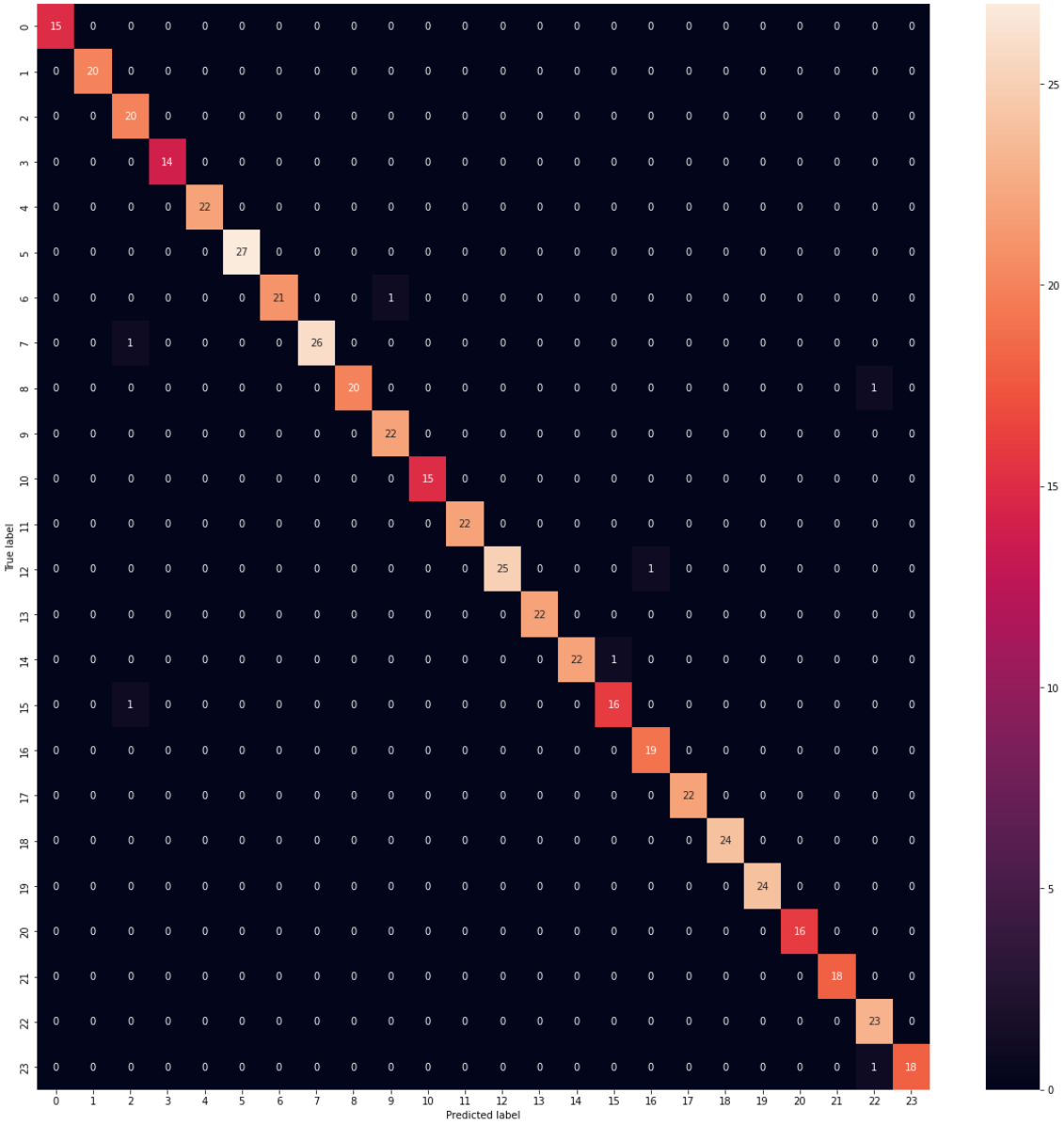


Figure 4.37 The confusion matrix of the SALU-AC database of Model 5.

Figure 4.37 displays the SALU-AC database's confusion matrix for three seconds for 24 speakers only. The equation TP+TN+FP+FN=499 indicates that a comprehensive analysis was performed on a total of 499 speaker samples, each lasting 3 seconds. The number of true positives (TP+TN= 493). Based on Eq. 4.2, the accuracy can be expressed as follows:

$$Accuracy = \frac{493}{499} \times 100\% = 98.79\%$$

4.2.6 The Experimental Results for Model6

The final system proposed in this thesis utilizes a two-dimensional DMWT for the purpose of extracting features. Following that, a CNN is used for the classification phase, as explained in Chapter Three. The model 6 methodology is applied to all datasets mentioned in this thesis, namely SALU-AC, ELSDSR, RAVDESS, and TIMIT. The performance evaluation of the PS6 model is conducted using k-fold cross-validation, with k-values of 2 and 5 employed for analysis purposes. The results of the model 6 are presented in Table 4.6.

Table 4.6 The recognition rates for the Model6.

K-fold	Database	Accuracy (%)				
		0.5 sec.	1 sec.	2 sec.	3 sec.	5 sec.
K=2	SALU-AC	90.08	93.20	94.40	96.05	96.88
	ELSDSR	93.55	94.49	95.17	96.12	97.31
	RAVDESS	86.51	89.90	92.04	94.78	--
	TIMIT	69.65	78.32	84.60	--	--
K=5	SALU-AC	95.86	96.30	96.63	97.04	97.56
	ELSDSR	97.61	97.81	98	98.25	98.70
	RAVDESS	96.05	97.04	97.81	97.96	--
	TIMIT	89.83	93.59	95.90	--	--

Regarding Table 4.6, it is obvious that the combination of 2D-DMWT and CNN resulted in notable recognition rates. This can be attributed to the inherent characteristics of discrete multi-wavelets, including compact support, orthogonality, symmetry, and high-order vanish moments. In contrast, a multi-wavelet system possesses the capability to provide instantaneous perfect reconstruction while preserving length (orthogonality), exhibiting beneficial performance at the boundaries (through linear-phase

symmetry), and achieving a high level of approximation due to its ability to maintain linear-phase balance (vanishing moments). Utilizing multi-wavelets presents a potential avenue for enhancing system performance across various signal-processing applications. And the arduous investigation of determining the optimal parameters for the convolutional neural network architecture.

Figures 4.38-4.41 illustrate the normalization process applied to several results obtained using the model 6. Each figure corresponds to a distinct database type. The duration for the normalization process is set at one second for illustrative purposes.

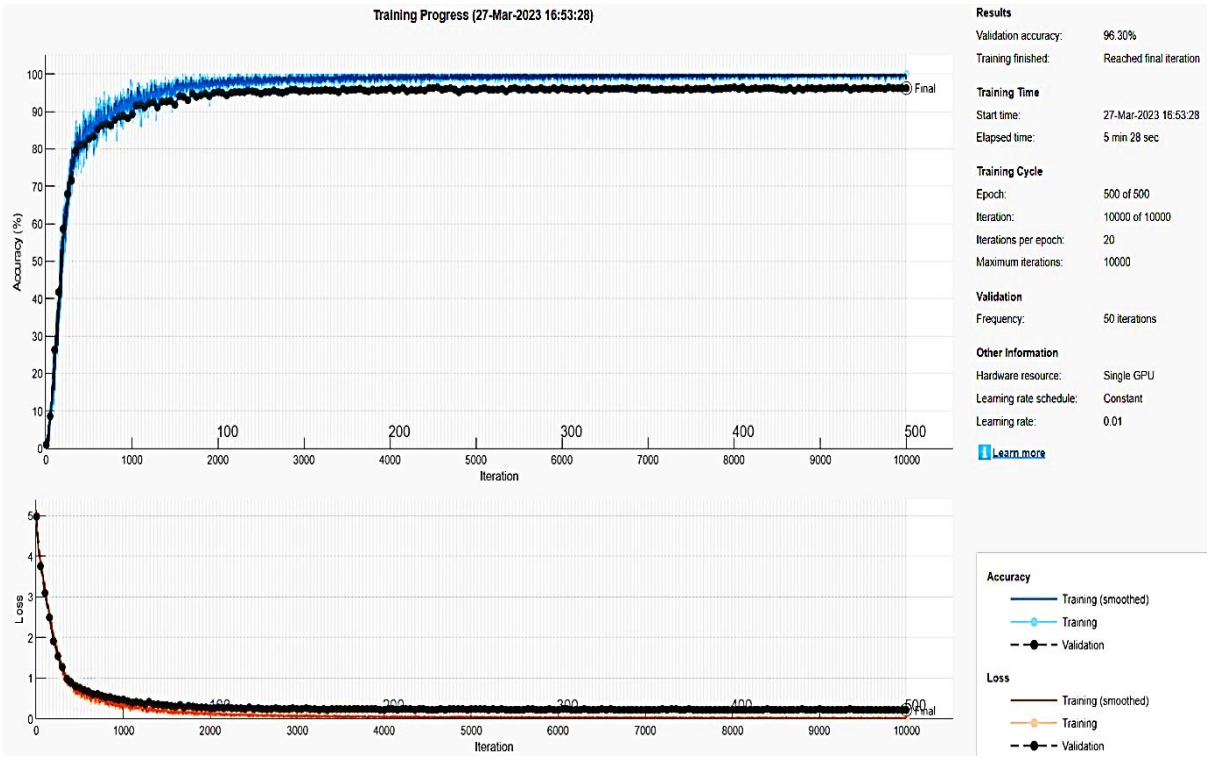


Figure 4.38 The normalization process of the SALU-AC database.

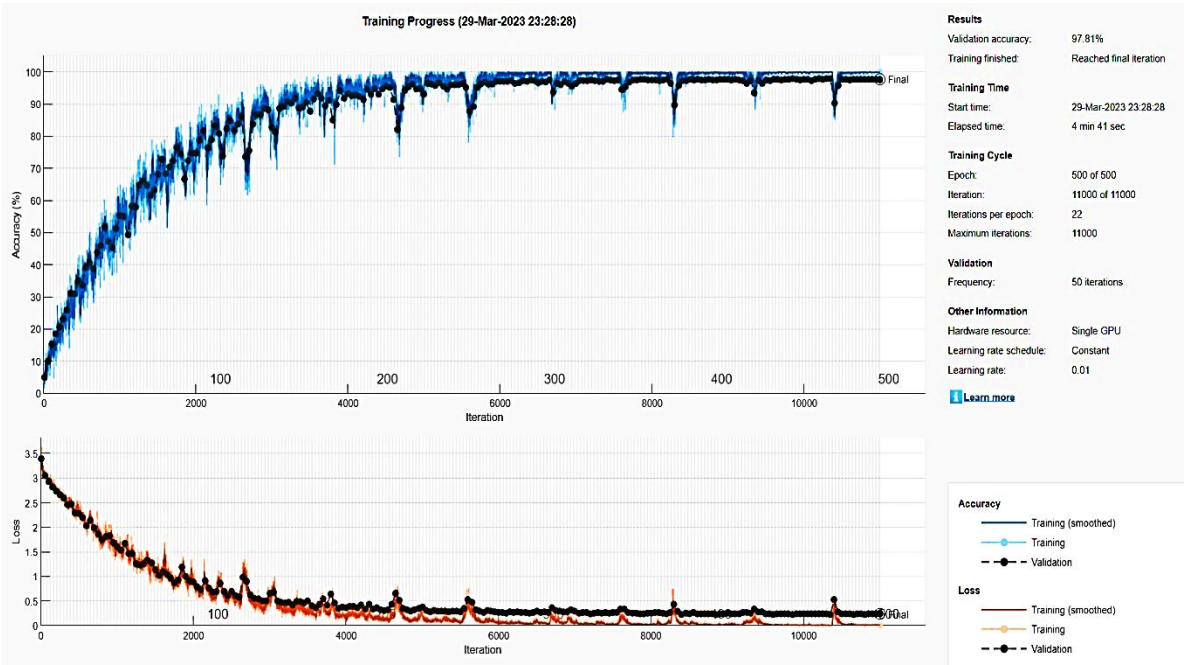


Figure 4.39 The normalization process of the ELSDSR database.

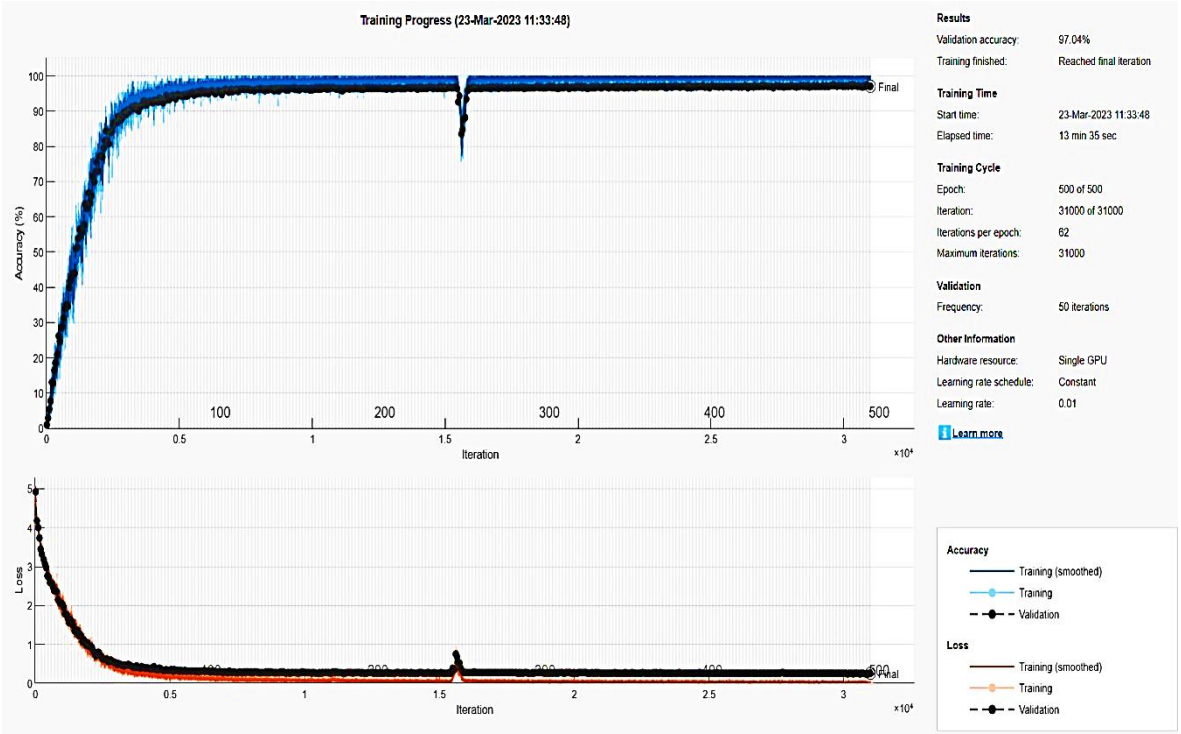


Figure 4.40 The normalization process of the RAVDESS database.

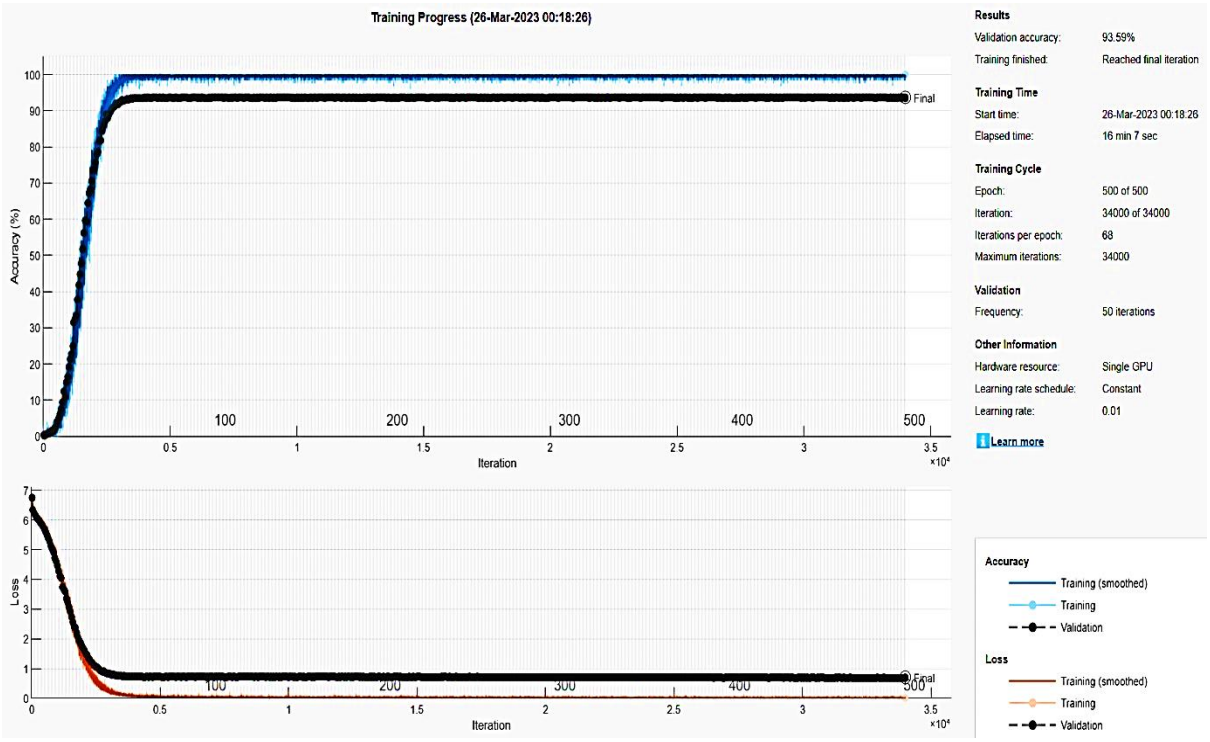


Figure 4.41 The normalization process of the TIMIT database.

Figures 4.42-4.44 show how the confusion matrix is used as an additional tool for evaluating accuracy. These Figures provide an example of how to calculate accuracy according to the ELSDSR, RAVDESS, and SALU-AC databases. The time frame of the given example will be limited to 0.5 sec. Unfortunately, the TIMIT database, which are of considerable size, are currently impossible and thus cannot be established in the demonstration.

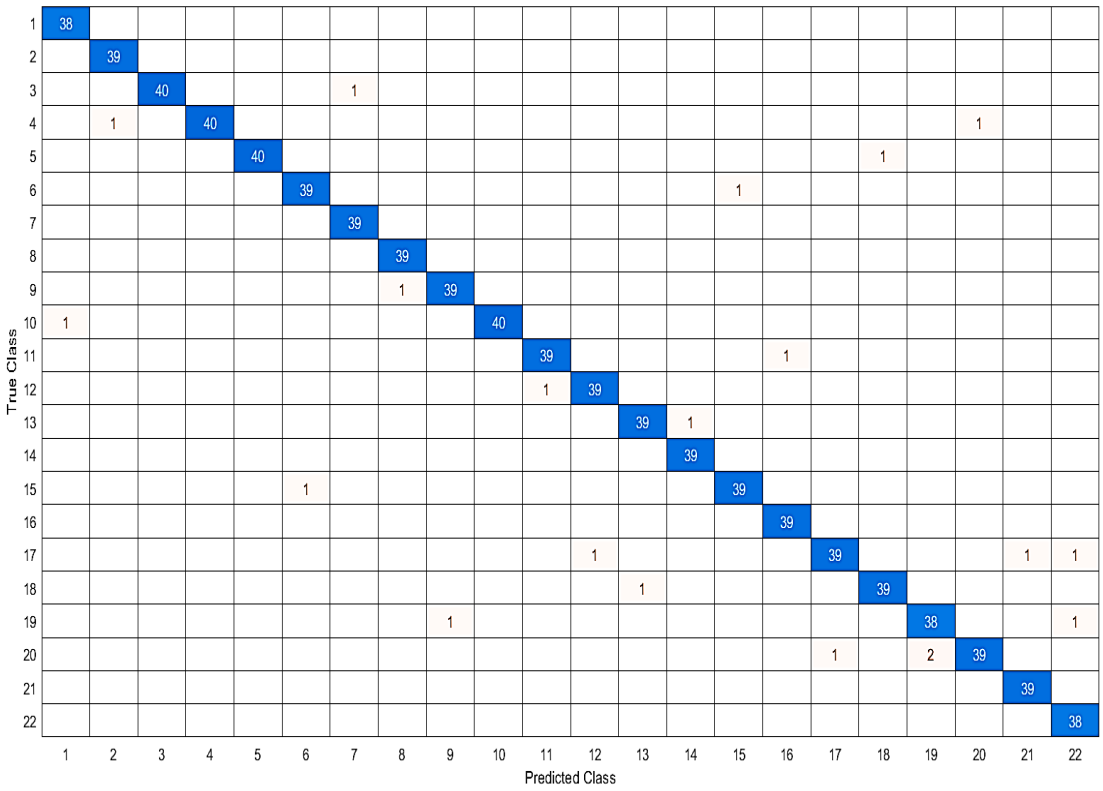


Figure 4.42 The confusion matrix of the ELSDSR of Model 6.

Figure 4.42 displays the ELSDSR database's confusion matrix for 0.5 seconds. The equation TP+TN+FP+FN=880 indicates that a comprehensive analysis was performed on a total of 880 speaker samples, each lasting 0.5 seconds. The number of true positives is (TP+TN= 859). Based on Eq. 2.45, the accuracy can be expressed as follows:

$$Accuracy = \frac{859}{880} \times 100\% = 97.61\%$$

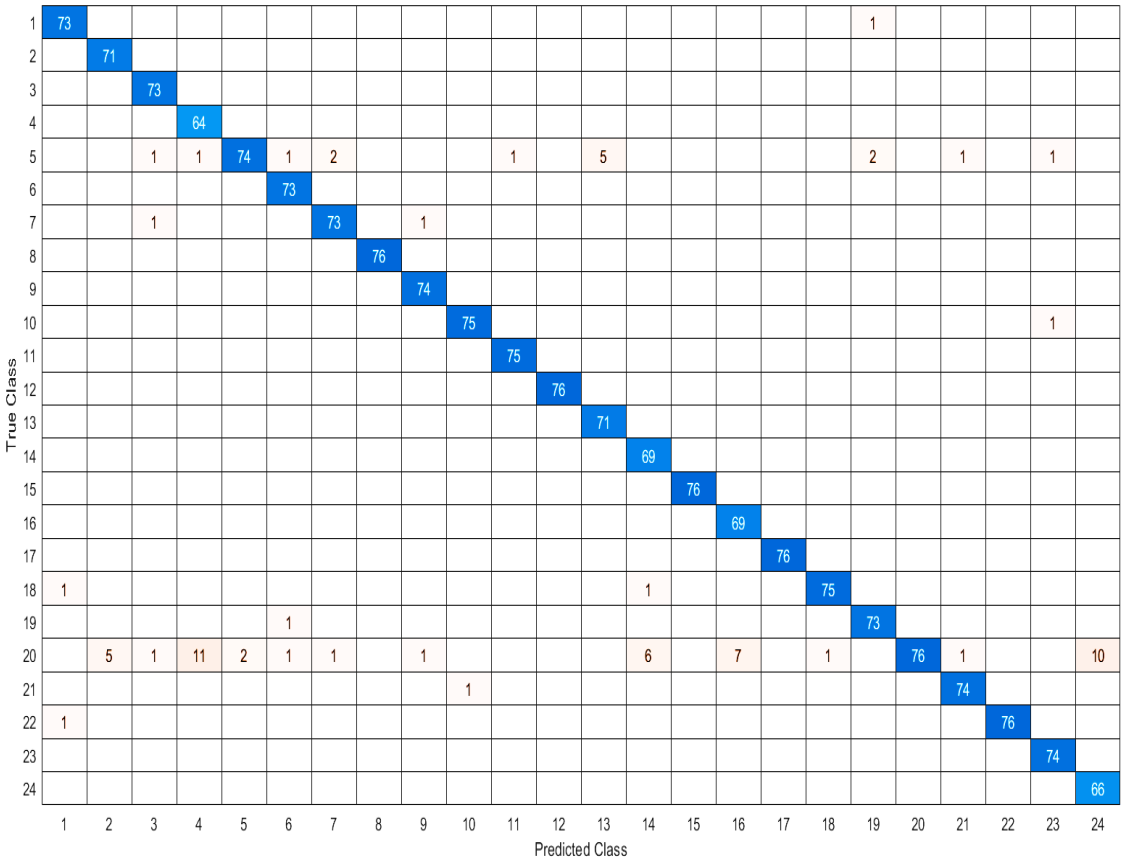


Figure 4.43 The confusion matrix of the RAVDESS of Model 6.

Figure 4.43 illustrates the RAVDESS database's confusion matrix for 0.5 seconds. The equation TP+TN+FP+FN=1824 shows that a full analysis was done on a total of 1824 speaker samples, each lasting 0.5 seconds. The number of true positives is (TP+TN= 1752). The results are computed by:

$$Accuracy = \frac{1752}{1824} \times 100\% = 96.05\%$$

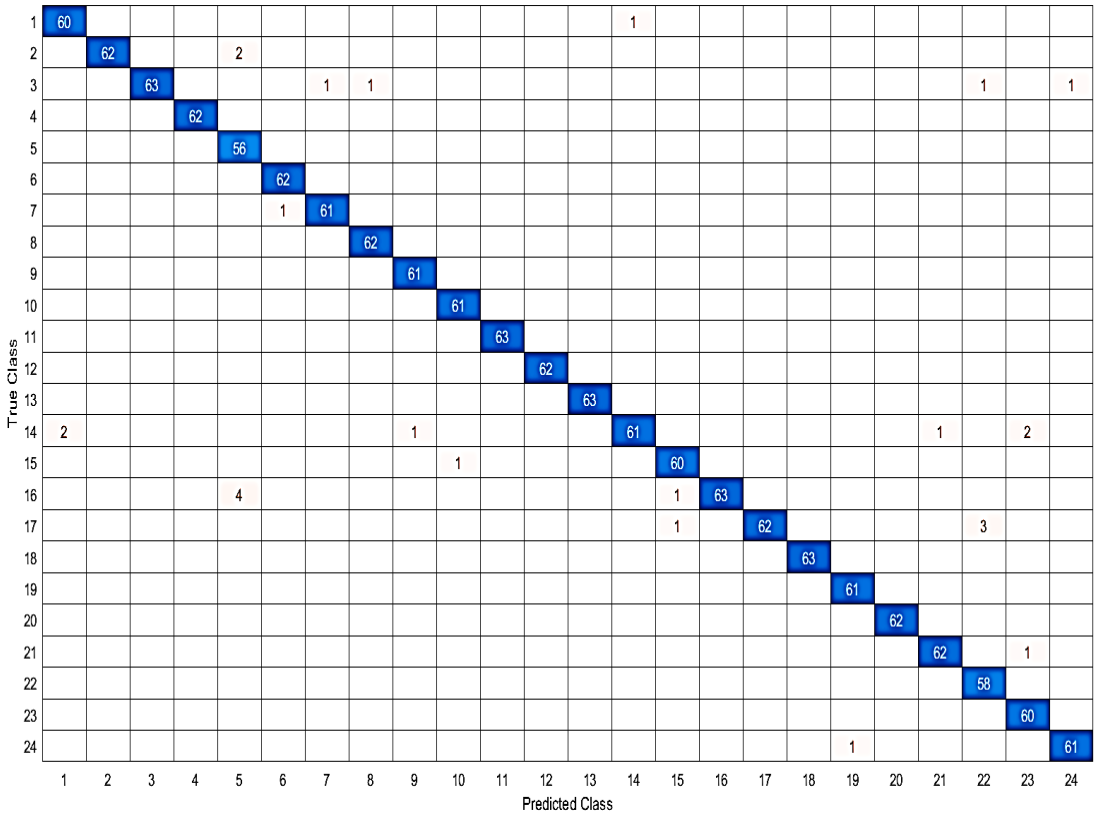


Figure 4.44 The confusion matrix of the SALU-AC of Model 6.

The confusion matrix of the SALU-AC database at 0.5 seconds is shown in Figure 4.44. Due to its extensive dimensions, the whole matrix including all 104 speakers cannot be exhibited. Therefore, a subset of 24 speakers will be used for analysis purposes. During a time, interval of 0.5 seconds, the total number of speaker samples assessed amounted to 1497 (TP+TN+FP+FN = 1497), with the true positive number being 1471 (TP+TN = 1471). The result can be expressed based on Eq. 2.45 as follow:

$$Accuracy = \frac{1471}{1497} \times 100\% = 98.26\%$$

4.3 Dimensionality Reduction

Dimensionality reduction is aimed at reducing the number of features present in a database, while simultaneously preserving the utmost significant information. Dimensionality reduction is used to convert high-dimensional data into a lower-dimensional space while retaining the basic features of the original data. The presence of high-dimensional data can result in overfitting, a phenomenon in which the model excessively conforms to the training data and fails to effectively generalize to unseen data. The utilization of dimensionality reduction techniques can effectively mitigate the complexity of the model and enhance its generalization performance.

One of the objectives of this thesis is to achieve optimal system performance by reducing dimensions and subsequently reducing system complexity. The dimensionality reduction (Dim. R) can be expressed as:

$$\text{Dimensionality reduction} = \left(1 - \frac{\text{Dimensions of the final feature extracted}}{\text{Input matrix dimensions}}\right) \times 100\% \quad (4.3)$$

Table 4.7 presents the percentage of (Dim. R) for all the databases utilized in this thesis, categorized by each model.

Table 4.7 The percentage of dimensionality reduction for each model.

Model2 (MFCC-PCA/CNN)					
Duration	**Feature Dim. R**	**SALU-AC**	**ELSDSR**	**RAVDESS**	**TIMIT**
0.5 sec.	**Dim. R Before**	128×50			
	Dim. R After	32×50			
	Dim. R	75%			
1 sec.	**Dim. R Before**	128×100			
	Dim. R After	32×100			
	Dim. R	75%			
2 sec.	**Dim. R Before**	128×200			
	Dim. R After	32×200			
	Dim. R	75%			

3 sec.	**Dim. R Before**	128×300		-
	Dim. R After	32×300		-
	Dim. R	75%		-
5 sec.	**Dim. R Before**	128×500	-	-
	Dim. R After	32×500	-	-
	Dim. R	75%	-	-
Model3 (2D-SWT/CNN)				
0.5 sec. **1 sec.** **2 sec.**	**Dim. R Before**	128×128		
	Dim. R After	32×32		
	Dim. R	93.75%		
3 sec.	**Dim. R Before**	128×128		-
	Dim. R After	32×32		-
	Dim. R	93.75%		-
5 sec.	**Dim. R Before**	128×128	-	-
	Dim. R After	32×32	-	-
	Dim. R	93.75%	-	-
Model4 (2D-DWT-PCA/CNN)				
0.5 sec. **1 sec.** **2 sec.**	**Dim. R Before**	32×32		
	Dim. R After	32×10		
	Dim. R	68.75%		
3 sec.	**Dim. R Before**	32×32		-
	Dim. R After	32×10		-
	Dim. R	68.75%		-
5 sec.	**Dim. R Before**	32×32	-	-
	Dim. R After	32×10	-	-
	Dim. R	68.75%	-	-
Model5 (MFCC-2D-DWT/CNN)				
0.5 sec.	**Dim. R Before**	128×64		
	Dim. R After	32×16		
	Dim. R	93.75%		
1 sec.	**Dim. R Before**	128×128		
	Dim. R After	32×32		
	Dim. R	93.75%		

2 sec.	**Dim. R Before**	128×256		
	Dim. R After	32×64		
	Dim. R	93.75%		
3 sec.	**Dim. R Before**	128×512		-
	Dim. R After	32×128		-
	Dim. R	93.75%		-
5 sec.	**Dim. R Before**	128×512	-	-
	Dim. R After	32×128	-	-
	Dim. R	93.75%	-	-
Model6 (2D-DMWT/CNN)				
0.5 sec. 1 sec. 2 sec.	**Dim. R Before**	256×256		
	Dim. R After	64×64		
	Dim. R	93.75%		
3 sec.	**Dim. R Before**	256×256		-
	Dim. R After	64×64		-
	Dim. R	93.75%		-
5 sec.	**Dim. R Before**	256×256	-	-
	Dim. R After	64×64	-	-
	Dim. R	93.75%	-	-

4.4 Overall performance

According to the proposed systems in this study, which are MFCC, MFCC-PCA, 2D-DWT, 2D-DWT-PCA, MFCC-2D-DWT, and 2D-DMWT, and whose results can be seen in Tables (4.1-4.6), this study used a variety of techniques to analyze the data in different time duration.

The duration of the speech samples that are utilized in speaker identification systems might have an impact on the system's overall performance. In general, longer phrases can convey more information and lessen the unpredictability of the speaker's voice, both of which may contribute to improvements in the accuracy and robustness of the speaker

identification system. Therefore, it seems that a duration of half a second may not be sufficient to capture the speaker-specific information in speech signals and to attain a high accuracy rate when compared to longer durations such as 5 seconds.

The histograms shown in Figures 4.45-4.48 illustrate the overall performance of each database for each of the methods used with the specified time durations.

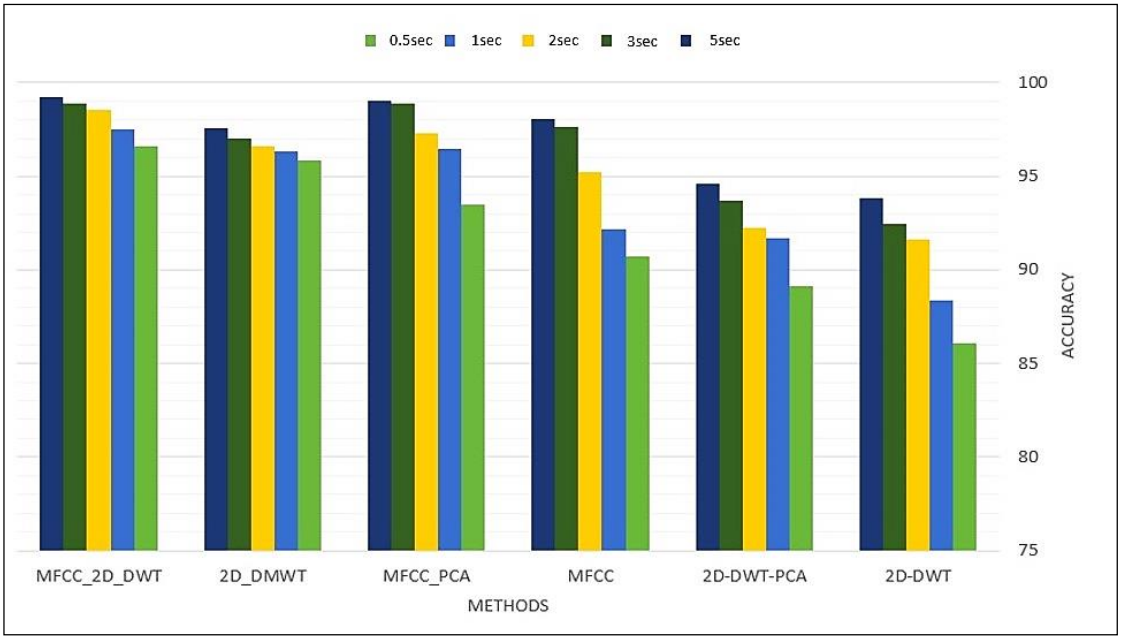


Figure 4.45 The overall performance of the SALU-AC database.

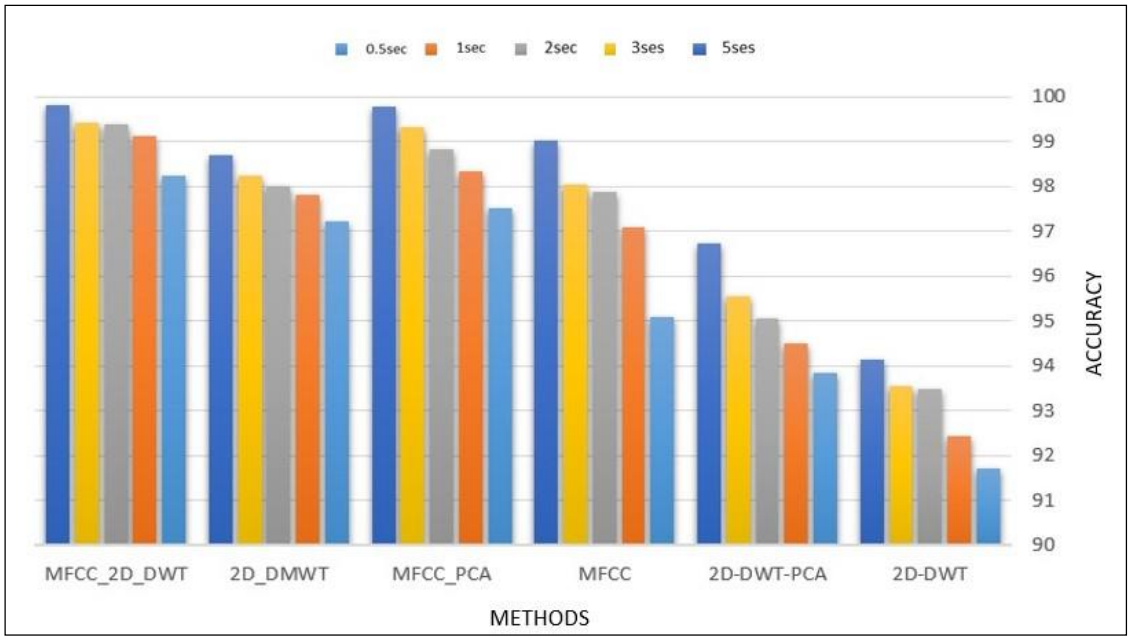


Figure 4.46 The overall performance of the ELSDSR database.

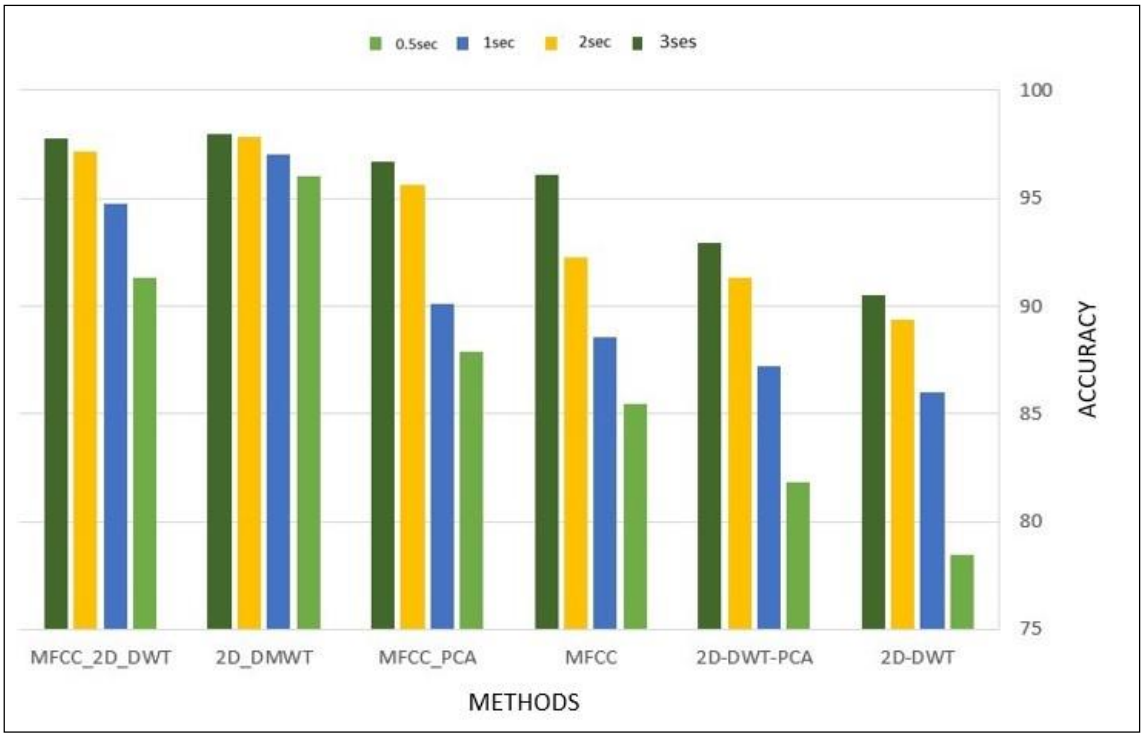


Figure 4.47 The overall performance of the RAVDESS database.

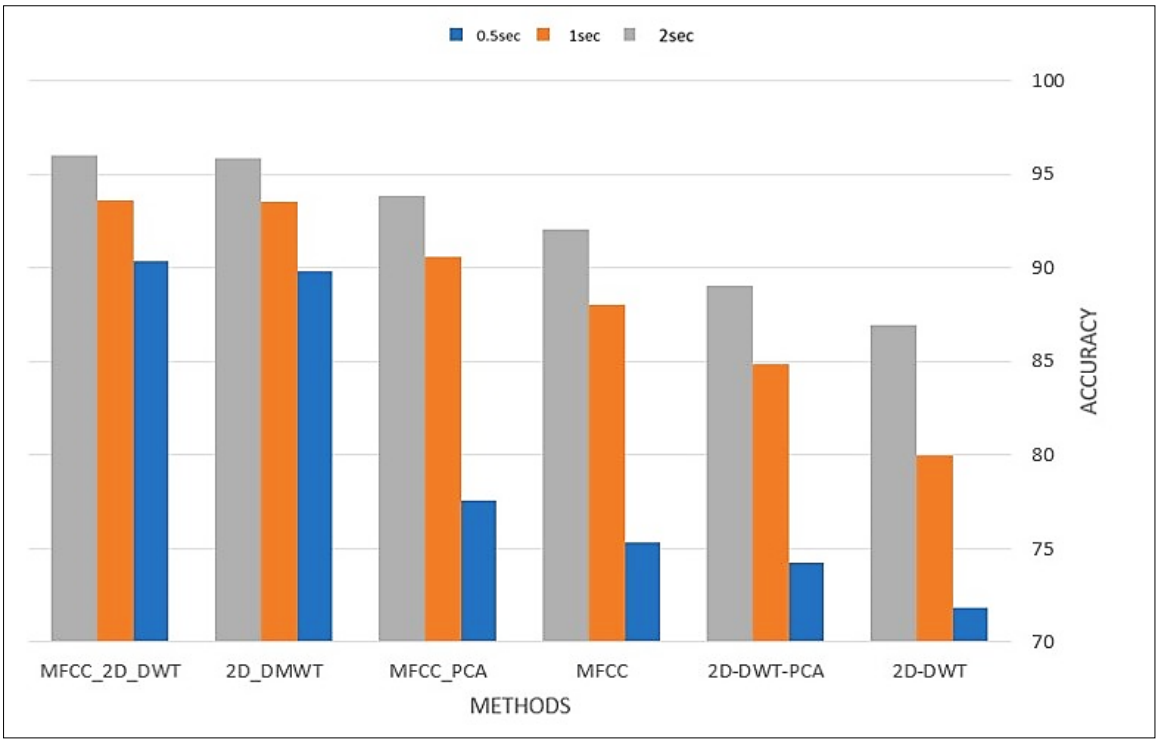


Figure 4.48 The overall performance of the TIMIT database.

4.5 The Compression with Related Work

The outcomes of the proposed method are compared with those obtained in the previous works [15-17, 21-24, 71, and 73] based on the same databases; namely, TIMIT, RAVDESS, and ELSDSR. Comparisons will be made between the approaches used in the feature extraction and classification. In order to establish which one of these approaches is the most efficient in terms of obtaining a high recognition rate.

The ELSDSR database is used to assess and test the proposed system. The outcomes of the proposed system are compared with those found in [15]. Table 4.8 displays the recognition rates.

The findings of the TIMIT database's suggested system in contrast to the approach described by [16] are shown in Table 4.8. A total of 38 speakers from the TIMIT database were involved in the evaluation of the method proposed by [16]. In order to ensure a fair comparison, a total of 38 speakers from the TIMIT database were selected to evaluate the proposed system.

Table 4.8 shows the results of the proposed system of the TIMIT database in comparison with the method presented by [17]. The same number of speakers used to evaluate the system proposed by [17], which were 70 speakers, is employed to test the system.

The recognition rate of the system proposed based on the TIMIT database is presented in Table 4.8. As can be shown in Table 4.8, the outcome achieved by the system proposed performs better than the one that was achieved by [21].

The comparison between the proposed system and the approach described in [22] is also shown in Table 4.8, which is based on the ELSDSR database.

Table 4.8 shows a comparison between the proposed system and the method provided in [23] based on the RAVDESS database. In [23], The authors selected only 12 samples for each speaker. Therefore, 12 samples

were chosen for each speaker to analyze the proposed approach for a fair comparison.

The recognition rate of the proposed system based on the TIMIT database is shown in Table 4.8. As seen in Table 4.8, the result of the system proposed outperforms the result accomplished by [24]. The approach presented in [24] was evaluated using 38 speakers of the TIMIT database. So, a total of 38 speakers were chosen from the TIMIT database to assess the proposed technique, ensuring a fair and equitable basis for comparison.

The experimental results of the RAVDESS database are shown in Table 4.8. T system proposed is compared with the approach presented by [71].

The result of the system proposed based on the ELSDSR database is shown in Table 4.8. As seen in Table 4.8, a comparison is made between the proposed system and the technique described in [73].

Table 4.8 The recognition rates of the proposed systems compared to the previous works.

Method	Database	Rates (%)
MFCC-CNN (proposed)	ELSDSR	**98.99**
MFCC/PCA-CNN (proposed)		**99.02**
2D-DWT-CNN (proposed)		**96.54**
2D-DWT/PCA-CNN (proposed)		**96.67**
MFCC/2D-DWT-CNN (proposed)		**99.43**
2D-DMWT-CNN (proposed)		**98.42**
MFCC-FFNN [15]		78
AFB-SVNN [22]		95
hybrid techniques-RF-SVM [73]		98.16
MFCC-CNN (proposed)	RAVDESS	**96**
MFCC/PCA-CNN (proposed)		**96.74**
2D-DWT-CNN (proposed)		**89.50**
2D-DWT/PCA-CNN (proposed)		**90.33**
MFCC/2D-DWT-CNN (proposed)		**96.82**
2D-DMWT-CNN (proposed)		**95.22**
MFCC-Modified SVM [23]		93.01

MFCC-CNN (proposed) **MFCC/PCA-CNN (proposed)** **2D-DWT-CNN (proposed)** **2D-DWT/PCA-CNN (proposed)** **MFCC/2D-DWT-CNN (proposed)** **2D-DMWT-CNN (proposed)** hybrid techniques-MLP [71]	RAVDESS	**96.38** **96.88** **93.09** **94.10** **98.52** **98.26** 92
MFCC-CNN (proposed) **MFCC/PCA-CNN (proposed)** **2D-DWT-CNN (proposed)** **2D-DWT/PCA-CNN (proposed)** **MFCC/2D-DWT-CNN (proposed)** **2D-DMWT-CNN (proposed)** LPC+SDC+CG with KNN [16] Random Forest using MFCC features [24]	TIMIT	**97.82** **98.20** **95.47** **95.77** **98.35** **97.46** 97.3 97
MFCC-CNN (proposed) **MFCC/PCA-CNN (proposed)** **2D-DWT-CNN (proposed)** **2D-DWT/PCA-CNN (proposed)** **MFCC/2D-DWT-CNN (proposed)** **2D-DMWT-CNN (proposed)** MAP adaptation [17] Tensor adaptation [17]		**95.99** **96.32** **91.08** **91.54** **96.76** **96.43** 93.04 92.86
MFCC-CNN (proposed) **MFCC/PCA-CNN (proposed)** **2D-DWT-CNN (proposed)** **2D-DWT/PCA-CNN (proposed)** **MFCC/2D-DWT-CNN (proposed)** **2D-DMWT-CNN (proposed)** MFCC-SECNN [21]		**92.19** **93.84** **86.99** **89.04** **96.02** **95.90** 95.83

According to the results shown in Table 4.8, the proposed systems that used multiple methods for feature extraction in conjunction with Convolutional Neural Networks accomplished higher recognition rates compared to the rates reached by previous studies in [15-17, 21-24, 71, and 73]. The superior performance of the proposed systems can be attributed to

the integration of robust techniques utilized during the preprocessing phase, such as silence removal, data augmentation, signal duration splitting, and signal resampling. These techniques directly impact the performance of the proposed system. Additionally, the feature extraction algorithms employed successfully extract significant features from the speech signal. Furthermore, the use of an exhaustive search method aids in selecting appropriate parameters for the convolutional neural network (CNN) structure. Moreover, a K-fold cross-validation with K=5 is used to analyze the outcomes of the parameter search. The use of k-fold cross-validation enhances the reliability of the results.

Chapter Five
“Conclusions and Future Works”

Chapter Five
Conclusions and Future Works

5.1 Conclusion

In this chapter, the conclusions of the proposed system for speaker identification system are discussed in the following points:

1. Different techniques for speaker identification systems were applied successfully.
2. Each proposed system consists of three phases: preprocessing, feature extraction, and classification.
3. Different approaches were employed in the preprocessing phase such as noise removal, data augmentation, and segmentation of time duration into distinct lengths (0.5, 1, 2, 3, and 5 seconds), etc.
4. Different methodologies were used in the feature extraction phase including MFCC, MFCC in conjunction with PCA, two-dimensional DWT, 2D-DWT combined with PCA, a fusion of MFCC and 2D-DWT, and 2D DMWT. These methodologies were utilized for feature dimension reduction, feature selection, noise reduction, classification purposes, etc.
5. For the classification phase, a deep neural network algorithm known as Convolutional Neural Network (CNN) was successfully employed.
6. Four different databases were used to evaluate the proposed systems; namely, SALU-AC, ELSDSR, RAVDESS, and TIMIT. These databases have different speech variations, such as age, gender, number of speakers, etc.
7. K-fold cross-validation was employed to evaluate the proposed systems.

8. The effectiveness of the MFCC technique in capturing the spectral envelope of the speech signal, which is closely associated with both the vocal tract shape and the unique characteristics of the speaker, was demonstrated.
9. Combining MFCC with PCA to decrease dimensions and reduce system complexity improved outcomes compared to using MFCC separately. Specifically, by reducing sizes by 75%, the results were enhanced.
10. The 2D-DWT technique effectively reduced dimensionality by 93.75% in the context of the speaker recognition system.
11. Utilizing both 2D-DWT and PCA yielded superior outcomes. The resulting matrix, obtained after two decomposition levels with input dimensions 32×32, was utilized in the subsequent PCA process, reducing sizes by 68.75%.
12. The fifth approach has outstanding results using MFCC and 2D-DWT. This improved system accuracy and robustness under various situations. It also preserved more speaker-specific information, improved MFCC feature discrimination, and addressed non-stationary signals. Also, feature dimensionality was decreased by 93.75%.
13. The 2D-DMWT approach for speaker identification was unique. Decomposition was done once using a GHM filter. Due to multi-wavelets' symmetry, orthogonality, and short support, the design performed well. Multi-wavelet systems can perform well near borders and provide important approximation with vanishing moments.
14. Speaker identification systems' performance depends on speech sample length. Longer sentences may provide more information and reduce the speaker's unpredictability, improving the speaker

identification system's accuracy and resilience. Compared to 5 seconds, half a second may not be long enough to acquire speaker-specific information in speech signals and achieve high accuracy.

15. The methods offered demonstrated higher performance when compared to other methodologies discussed in previous studies [15–17, 21–24, 71, and 73], as shown in Table 4.8 based on the same database.

5.2 Futures works

For future work, the primary emphasis will be placed on the DMWT methodology, as it has demonstrated high performance. The forthcoming techniques will be employed to ascertain the identities of the speakers:

- Applying a hybrid approach involving both techniques MFCC with 2D-DMWT for better classification performance.
- It may integrate the 2D-DMWT with principal component analysis (PCA) to effectively reduce the dimensions of the data.
- The amalgamation of two methodologies encompasses the utilization of both 2D-DMWT and 3D-DWT.

References

[1] R. d. L. Garc'a, C. A. L'opez, O. Aghzoutb, and J. R. Alzolab, "Biometric identification systems," *Signal Processing*, Vol. 83, Issue 12, pp. 2539-2557, December 2003.

[2] R. M. Hanifa, K. Isa, and S. Mohamad, "A review on speaker recognition: Technology and challenges," *Computers and Electrical Engineering*, Vol. 90, No. January, p. 107005, 2021.

[3] L. Feng, "Speaker Recognition," M.S. thesis, Dept. Inform. Math. Model., Tech. Univ. Denmark, Lyngby, Denmark, 2004.

[4] X. Yuan, G. Li, J. Han, D. Wang, and T. Zhi, "Overview of the development of speaker recognition," *Journal of Physics: Conference Series ICETIS 2021*, Vol. 1827, pp. 1-6, 2021.

[5] F. E. Abualadas, A. M. Zeki, M. S. Al-Ani, and A.-E. Messikh, "Speaker Identification based on Hybrid Feature Extraction Techniques," *(IJACSA) International Journal of Advanced Computer Science and Applications*, Vol. 10, No. 3, pp. 322-327, 2019.

[6] M. M. Kabir, M. F. Mridha, J. Shin, I. Jahan, and A. Q. Ohi, "A Survey of Speaker Recognition: Fundamental Theories, Recognition Methods and Opportunities," *IEEE Access*, Vol. 9, pp. 79236–79263, 2021.

[7] A. K. Panda, and A. K. Sahoo, "Study Of Speaker Recognition Systems," B.Tech. thesis, Dept. Electron. Commun. Eng., Natl. Inst. Technol., Rourkela, India, 2011.

[8] Ms. M. D. Pawar, Ms. S. C. Saraf, and Ms. P. P. Patil, "WAVELET ENTROPY AND NEURAL NETWORK FOR TEXT-DEPENDENT SPEAKER IDENTIFICATION," *INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS*, Vol.2 Issue 1, pp.178 – 184, January 2014.

[9] P. M. Chauhan, and N. P. Desai, "Mel Frequency Cepstral Coefficients (MFCC) based speaker identification in noisy environment using wiener

filter," in Conf. *Proceeding IEEE Int. Conf. Green Comput. Commun. Electr. Eng. ICGCCEE 2014*, 2014.

[10] T. Mahboob, M. Khanum, M. S. H. Khiyal, R. Bibi, "Speaker Identification Using GMM with MFCC," *IJCSI International Journal of Computer Science Issues,* Vol. 12, Issue 2, pp. 126-135, March 2015.

[11] J. Kacur, and P. Truchly, "Acoustic and auxiliary speech features for speaker identification system," in Conf. *2015 57th International Symposium ELMAR (ELMAR),* Zadar, Croatia, 2015, pp. 109-112.

[12] A. A. A. Fattah, and L. M. Salih, "Speaker Recognition Using Discrete Wavelet Transform and Artificial Neural Networks," *ZANCO Journal of Pure and Applied Sciences*, Vol. 28, Issue 3, pp.78-85, 2016.

[13] F. ur Rehman, C. Kumar, S. Kumar, A. Mehmood, and U. Zafar, "VQ based comparative analysis of MFCC and BFCC speaker recognition system," in Conf. *2017 International Conference on Information and Communication Technologies (ICICT)*, Karachi, Pakistan, 2017, pp. 28-32.

[14] A. D. Mengistu, "Automatic Text Independent Amharic Language Speaker Recognition in Noisy Environment Using Hybrid Approaches of LPCC, MFCC and GFCC," in Conf. *Int. J. Advnaced Stud. Comput. Sci. Eng.*, Vol. 6, Issue 5, pp. 8–12, 2017.

[15] A. Bora, B. C. G. Sanjay, and J. Vajpai, "Neural Network based Speaker Identification using Hybrid Features," in Conf. *NACNC 2017*, Jodhpur, India, 2017, pp. 1-5.

[16] S. V. Arora, and R. Vig, "Short Utterance Based Speaker Identification System For Resource Constrained Devices," in conf. 2018 2nd International Conference on Micro-Electronics and Telecommunication Engineering, Ghaziabad, India, 2018, pp. 246-249.

[17] S. Pandey, S. Jelil, S. R. M. Prasanna and H. S. Shekhawat, "Speaker Identification Using Tensor Decomposition of Acoustic Models," TENCON

2018 - 2018 IEEE Region 10 Conference, Jeju, Korea (South), 2018, pp. 1484-1488.

[18] A. Awais, S. Kun, Y. Yu, S. Hayat, A. Ahmed, and T. Tu, "Speaker recognition using mel frequency cepstral coefficient and locality sensitive hashing," in Conf. *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, 2018, pp. 271–276.

[19] S. Bunrit, T. Inkian, N. Kerdprasop, and K. Kerdprasop, "Text-independent speaker identification using deep learning model of convolution neural network," *International Journal of Machine Learning and Computing*, Vol. 9, No. 2, pp. 143–148, Apr. 2019.

[20] M. Dai, G. Dai, Y. Wu, Y. Xia, F. Shen, and H. Zhang, "An improved feature fusion for speaker recognition," in Conf. *2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC)*, Hangzhou, China, 2019, pp. 183–187.

[21] M. Qi, Y. Yu, Y. Tang, Q. X. Deng, F. Mai, and N. Zhaxi, "Deep CNN with SE Block for Speaker Recognition," in Conf. *2020 Information Communication Technologies Conference (ICTC)*, Nanjing, China, 2020, pp. 240–244.

[22] V. SRINIVAS, and CH. SANTHIRANI, "Optimization-Based Support Vector Neural network for Speaker Recognition", THE COMPUTER JOURNAL, Vol. 63, Issue. 1, pp. 151–167, 2020.

[23] N. A. Al Hindawi, I. Shahin and A. B. Nassif, "Speaker Identification for Disguised Voices Based on Modified SVM Classifier," in conf. 2021 18th International Multi-Conference on Systems, Signals & Devices (SSD), Monastir, Tunisia, 2021, pp. 687-691.

[24] K. K. Nawas, M. K. Barik, and A. N. Khan, "Speaker Recognition using Random Forest", In conf. of ITM Web Conf., Vol. 37, pp. 1-5, 2021.

[25] H. Y. Khdier, W. M. Jasim, and S. A. Aliesawi, “Deep Learning Algorithms based Voiceprint Recognition System in Noisy Environment,” *Journal of Physics: Conference Series*, Vol. 1804, No. 1, pp. 1-10, 2021.

[26] R. T. Al-Hassani, D. C. Atilla, and Ç. Aydin, “Development of High Accuracy Classifier for the Speaker Recognition System*,” Applied Bionics and Biomechanics*, Vol. 2021, pp. 1-10, 2021.

[27] K. Nugroho, E. Noersasongko, Purwanto, Muljono, and D. R. I. M. Setiadi, “Enhanced Indonesian Ethnic Speaker Recognition using Data Augmentation Deep Neural Network,” *Journal of King Saud University – Computer and Information Sciences*, Vol. 34, Issue 7, pp 4375–4384, Jul 2022.

[28] S. Kadyrov, C. Turan, A. Amirzhanov, and C. Ozdemir, “Speaker Recognition from Spectrogram Images,” in Conf. *2021 IEEE International Conference on Smart Information Systems and Technologies (SIST)*, Nur-Sultan, Kazakhstan, 2021, pp.1-4.

[29] S. Dwijayanti, A. Y. Putri, and B. Y. Suprapto, “Speaker Identification Using a Convolutional Neural Network,” *JURNAL RESTI (Rekayasa Sist. dan Teknol. Informasi)*, Vol. 6, No. 1, pp. 140–145, 2022.

[30] F. Abakarim, and A. Abenaou, “Comparative study to realize an automatic speaker recognition system,” *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 12, No. 1, pp. 376–382, February 2022.

[31] S. H. Shah, M. S. Saeed, S. Nawaz, and M. H. Yousaf, “Speaker Recognition in Realistic Scenario Using Multimodal Data,” in Conf. International Conference on Artificial Intelligence (ICAI'2023), 2023, pp. 209–213.

[32] A. A. Desta, “Text-Independent Speaker Identification for the Amharic Language,” M.S. thesis, Bahir Dar Institute of Technology, Bahir Dar University, Bahir Dar, Ethiopia, 2016.

[33] A. Keerio, B. K. Mitra, P. Birch, R. Young, and C. Chatwin, “On preprocessing of speech signals,” *International Journal of Signal Processing*, Vol. 5, Issue 3, pp.216–222, 2009.

[34] Y. Afrillia, H. Mawengkang, M. Ramli, F. Fadlisyah, and R. P. Fhonna, “Performance Measurement of Mel Frequency Ceptral Coefficient (MFCC) Method in Learning System of Al-Qur’an Based in Nagham Pattern Recognition,” *Journal of Physics: Conference Series*, Vol. 930, Iss. 1, pp. 1-6, Dec. 2017.

[35] S. Ali, Dr. S. Tanweer, S. S. Khalid, and Dr. N. Rao, Mel Frequency Cepstral Coefficient: A Review, in Conf. 2nd International Conference on ICT for Digital, Smart, and Sustainable Development (ICIDSSD 2020), New Delhi, India, 2020.

[36] B. M. Nema, and A. A. Abdul-Kareem, “Preprocessing signal for Speech Emotion Recognition,” *Al-Mustansiriyah J ournal of Science*, Vol. 28, Issue 3, pp. 157–165, 2018.

[37] K. S. Rao, and S. R. M. Prasanna, *Speech Recognition Using Articulatory and Excitation Source Features*. Cham: Springer International Publishing, 2017.

[38] L. Muda, M. Begam, and I. Elamvazuthi, “Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques,” *JOURNAL OF COMPUTING*, Vo. 2, Issue 3, pp. 138-143, MARCH 2010.

[39] R. Togneri, and D. Pullella, “An Overview of Speaker Identification: Accuracy and Robustness Issues,” *IEEE Circuits and Systems Magazine*, vol. 11, no. 2, pp. 23-61, 2011.

[40] I. D. G. Y. A. Wibawa and I. D. M. B. A. Darmawan, "Implementation of audio recognition using mel frequency cepstrum coefficient and dynamic time warping in wirama praharsini," *Journal of Physics: Conference Series*, Vol 1772, No. 1, pp. 1-8, Jan. 2021.

[41] J. Shlens, "A Tutorial on Principal Component Analysis: Derivation, Discussion and Singular Value Decomposition," 2003.

[42] T. ArchanaH, and D. Sachin, "Dimensionality Reduction and Classification through PCA and LDA," *International Journal of Computer Applications*, Vol. 122, No. 17, pp. 4–8, July 2015.

[43] A. Tharwat, "Principal component analysis - a tutorial," *Int. J. Appl. Pattern Recognit.*, vol. 3, no. 3, pp. 197-240, 2016.

[44] A. Graps, "An Introduction to Wavelets," *Institute of Electrical and Electronics Engineers*, Vol. 2, No. 2, pp. 1–18. 1995.

[45] H. N. Al-Taai, "Optical flow estimation using DSP technique," Ph.D. dissertation, Univ. Technol., Baghdad, Iraq, 2005.

[46] R. Polikar, (1996), *The Wavelet Tutorial* (2nd ed.) 329 Durham Computation Center Iowa State University. [Online]. http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html.

[47] S. G. MALLAT, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, Vol. 11, No. 7, pp. 674-693, JULY 1989.

[48] J. Chen, X. Ouyang, W. Zheng, J. Xu, J. Zhou, and S. Yu, "The application of symmetric orthogonal multiwavelets and prefilter technique for image compression," Multimed Tools App, Vol. 29, No. 2, pp. 175–187, May 2006.

[49] C. Valens, *A really friendly guide to wavelets*, Birkhäuser Boston, MA, 1999.

[50] C. S. Burrus, R. Gopinath, and H. Guo, *Wavelets and Wavelet Transforms-A primer*, Rice University, Houston, Texas, 1998.

[51] S. Rout, "Orthogonal vs. biorthogonal wavelets for image compression," M.S. thesis, Virginia Polytech. Inst. and State Univ., Blacksburg, VA, USA, 2003.

[52] A. Aldhahab, and W. B. Mikhael, "Employing Efficient Techniques based on 2D DMWT / FastICA for Supervised Facial Recognition," in conf. *FTC 2016 - Future Technologies Conference 2016*, San Francisco, United States, December 2016, pp. 1316-1323.

[53] A. H. Kattoush, W. A. Mahmoud, and A. Mashagbah, and A. Ghodayyah, "Multiwavelet Computed Radon-Based Ofdm Trasceiver Designed and Symulation under Different Channel Conditions," *Journal of Information and Computing Science*, Vol. 5, No. 2, pp. 133–145, 2010.

[54] S. A. Dawood, F. Malek, M. S. Anuar, and S. Q. Hadi, "Discrete Multiwavelet Critical-Sampling Transform-Based OFDM System over Rayleigh Fading Channels," *Mathematical Problems in Engineering, Hindawi*, Vol. 2015, pp. 1-10, May 2015.

[55] W. A. Mahmoud, Z. J. M. Saleh, and N. K. Wafi, "The Determination of Critical-Sampling Scheme of Preprocessing for Multiwavelets Decomposition as 1 st and 2 nd Orders of Approximations.," *Al-Khwarizmi Engineering Journal*, Vol.1, No.1, pp 26-37, 2005.

[56] H. H. Wang, J. Wang, and W. Wang, "Multispectral image fusion approach based on GHM multiwavelet transform," in conf. *2005 International Conference on Machine Learning and Cybernetics*, Guangzhou, China, 2005, pp. 5043–5049.

[57] M. B. Martin, "Applications of Multiwavelets to Image Compression," M.S. thesis, Virginia Polytech. Inst. and State Univ., Blacksburg, VA, USA, 1999.

[58] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, and O. Al-Shamma, et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, Vol. 8, Issue 1, pp 1-74, Jan. 2021.

[59] S. Bunrit, T. Inkian, N. Kerdprasop, and K. Kerdprasop, "Text-independent speaker identification using deep learning model of convolution neural network,", *International Journal of Machine Learning and Computing*, Vol. 9, No. 2, pp. 143–148, April 2019.

[60] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti and D. De, "Fundamental Concepts of Convolutional Neural Network," *in Advances in Intelligent Systems and Computing*, Vol. 1047, pp. 519-567, Jan. 2020.

[61] T. Bezdan, and N. B. Džakula, "Convolutional Neural Network Layers and Architectures," in conf. *Sinteza 2019 - International Scientific Conference on Information Technology and Data Related Research*, Belgrade, Singidunum University, Serbia, 2019, pp. 445-451.

[62] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in conf. *2017 International Conference on Engineering and Technology (ICET)*, Antalya, Turkey, 2017, pp. 1-6.

[63] M. M. Taye, "Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions," *Computation*, Vol. 11, No. 3, pp. 1-23, 2023.

[64] S. Ioffe, and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in conf. *32nd International Conference on Machine Learning*, Lille, France, 2015, pp. 1-9.

[65] Musstafa. (2021). Optimizers in Deep Learning [Online]. Available: https://medium.com/mlearning-ai/optimizers-in-deep-learning-7bf81fed78a0.

[66] K. A. Y. Al-Karawi, "Robust Speaker Recognition in Reverberant Condition-Toward Greater Biometric Security," Ph.D. dissertation, School of Computing, Science and Engineering, University of Salford, Salford, UK, 2018.

[67] A. Y. Niwatkar, and Y. K. Kanse, "Speaker Recognition System and it's Applications," *International Journal of Recent Technology and Engineering (IJRTE),* Vol. 8, Issue-2, pp. 6429–6432, July 2019

[68] M. Heydarian, T. E. Doyle, and R. Samavi, "MLCM: Multi-Leabel Confusion Matrix," *IEEE Access*, Vol. 10, pp. 19083-19095, 2022.

[69] R. Valavi, J. Elith, J. J. L-Monfort, and G. G-Arroita, "blockCV: An r package for generating spatially or environmentally separated folds for k-fold cross-validation of species distribution models," *Methods Ecol. Evol.*, Vol. 10, No. 2, pp. 225–232, 2019.

[70] H. Al-abboodi, Binaural Sound Source Localization Using Machine Learning with Spiking Neural Networks Features Extraction, Ph.D. dissertation, School of Computing, Science and Engineering, University of Salford, Salford, UK, 2018.

[71] T. J. Sefara, and T. B. Mokgonyane, "Emotional Speaker Recognition based on Machine and Deep Learning," in conf. *2020 2nd International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, Kimberley, South Africa, 2020, pp. 1-8.

[72] D. Ellis, "TIMIT Corpus," Kaggle, Kaggle Inc., 2018, https://www.kaggle.com/datasets/dodger23/timit-corpus.

[73] V. Karthikeyan, and S. S. Priyadharsini, "Hybrid machine learning classification scheme for speaker identification," *J. Forensic Sci.*, Vol. 67, No. 3, pp. 1033–1048, 2022.

الخلاصة

يشير التعرف على المتحدث إلى العملية المعرفية لتحديد هوية المتحدث من خلال تحليل واستخراج بعض الميزات المميزة المضمنة في إشارة الكلام. يمكن تقسيم تصنيف التعرف على المتحدث إلى فئتين رئيسيتين: التحديد والتحقق. يشير تعريف المتحدث إلى العملية الحسابية للتأكد من هوية المتحدث المسجل الذي أنتج خطابًا معينًا. في المقابل، يشير التحقق من المتحدث إلى صحة تأكيد هوية المتحدث، مما يؤدي إما إلى القبول أو الرفض.

تبحث هذه الأطروحة في موضوع أنظمة تعريف المتحدث، مع التركيز بشكل خاص على المنهجيات المختلفة المستخدمة لاستخراج الميزات من إشارة الكلام.

يحتوي النظام المقترح على ثلاث مراحل أساسية، تتكون من: المعالجة المسبقة، واستخراج الميزات، والتصنيف. خلال مرحلة المعالجة المسبقة، يتم استخدام العديد من التقنيات، مثل إزالة الضوضاء، وزيادة البيانات، وتجزئة المدة الزمنية إلى أطوال مختلفة (0.5، 1، 2، 3، و5 ثواني)، وما إلى ذلك. يتم تطبيق هذه التقنيات لتنظيم البيانات بشكل مناسب والحصول على الميزات المميزة خلال مرحلة استخراج الميزة. في مرحلة استخراج المعالم، يتم استخدام طرق مختلفة، بما في ذلك استخدام معامل سبيسترال ذو التردد ميل (MFCC)، وتكامل معامل سبيسترال ذو التردد ميل مع تحليل المكونات الرئيسية (MFCC/PCA)، وتنفيذ التحويل المويجي المنفصل ثنائي الأبعاد (2D-DWT)، مزيج من تحويل المويجات المنفصلة ثنائي الأبعاد مع تحليل المكونات الرئيسية (2D-DWT/PCA)، وتكامل معامل سبيسترال ذو التردد الميل مع تحويل المويجات المنفصلة ثنائي الأبعاد (MFCC/2D-DWT)، وفي النهاية اعتماد التحويل المنفصل متعدد المويجات ثنائي الأبعاد (2D-DMWT). وأخيرا، يتم تغذية الميزات المستخرجة إلى المرحلة الثالثة، والتي تستخدم خوارزمية التعلم العميق التي تعتمد على الشبكة العصبية التلافيفة (CNN) وتستخدم لأغراض التصنيف.

في هذه الأطروحة، يتم استخدام أربع قواعد بيانات مختلفة لتقييم الأساليب المقترحة؛ وهي SALU-AC، وELSDSR، وRAVDESS، و TIMIT. تحتوي قواعد البيانات هذه على اختلافات مختلفة في الكلام، مثل العمر والجنس وعدد المتحدثين وما إلى ذلك.

أظهر النظام المقترح أداءً متميزًا، حيث حقق أعلى معدلات دقة تبلغ 99.82% و99.22% لقاعدتي بيانات ELSDSR وSALU-AC، على التوالي، خلال مدة زمنية قدرها 5 ثوانٍ بناءً على نهج (MFCC/2D-DWT). حققت قاعدة بيانات RAVDESS دقة تصنيف تصل إلى 97.96% خلال مدة 3 ثوانٍ على أساس طريقة (2D-DMWT). أظهرت قاعدة بيانات TIMIT أعلى دقة بلغت 96.02% عند استخدام مدة 2 ثانية بناءً على طريقة (MFCC/2D-DWT). يؤثر طول

عينات الكلام المستخدمة في أنظمة التعرف على المتحدث بشكل كبير على الأداء العام للنظام. عادة، يمكن أن يؤدي استخدام العبارات الأطول إلى نقل قدر أكبر من المعلومات وتقليل التباين في صوت المتحدث. يمكن لهذه العوامل أن تعزز دقة ومتانة نظام التعرف على السماعات.

حققت الدراسة التي أجريت في هذه الأطروحة أهداف تحقيق نتائج ناجحة وتقليل أبعاد البيانات، وبالتالي التخفيف من تعقيد النظام. النتائج التي حققها النظام المقترح تفوق تلك النتائج التي تمت مناقشتها في الأعمال السابقة اعتمادا على نفس قواعد البيانات.

جمهوريـــة العراق
وزارة التعليم العالي والبحث العلمي
جامعـــــة بابـــل
كليـــة الهندســـة/قسم الهندســـة
الكهربائية

توظيف تقنيات مختلفة لنظام التعرف على المتكلم بالاعتماد على طريقة التعلم العميق

رسالة مقدمـــة إلى كليـــة الهندســـة في جامعـــــة بابـــل

كجزء من متطلبات نيل درجـــــــة الماجستير

الهندسة الكهربائية/ الالكترونيك الصناعي

من قبل:

هدى وصفي حسون

بأشراف:

أ.م.د. أحمد قاسم جمعة

أ.م.د. هناء محسن علي

2023م **1445ه**