Journal of Intelligent Systems and Internet of Things
Vol. 15, No. 01, PP. 105-121, 2025
DOI: https://doi.org/10.54216/JISIoT.150109

A Hybrid Speech Recognition System Using Deep Learning Methods

Hadeel Luhaib Fouad[1,*], Husam Ali Abdulmohsin[1]
[1] Computer science department, college of science, university of Baghdad, Iraq
Emails: Hadeel.Fouad2301@sc.uobaghdad.edu.iq; Husam.a@sc.uobaghdad.edu.iq

Abstract

Speech-to-text Conversion is a type of Speech Recognition Program that effectively takes audio content as input and transcribes it into written words. With increasing technologies and large data corpus, the importance of speech recognition has increased. Now everyone seems to be exploitation Speech Recognition Technology for users to work a tool, perform commands, or write while not having to use a keyboard, mouse, or press any buttons. It is also easy for everyone to utter sound or speak than using hands to be work done and it is also convenient to use. In this paper, a system capable of converting audio files to text has been developed. The proposed system consists of a set of algorithms for processing audio files, where the MFCC algorithm combine with standard deviation was adopted to extract the features of the audio file and convert it into an image. The features of audio files are stored as images because deep learning algorithms can be trained on images better than CSV files. The second part of the proposed system is the design of a deep learning model in which two algorithms, Convolutional Neural Network (CNN) and Deep Neural Network (DNN) are combined to predict words. The model consists of a set of layers to extract the features from the images, choose the best features, then train and classify them based on the proposed DNN model. In this thesis, three types of datasets (Arabic, English, and Real) were adopted to test the proposed system in speech prediction and the accuracy of the proposed system has reached more than 95%.

Received: June 30, 2024 Revised: September 26, 2024 Accepted: December 25, 2024

Keywords: Speech Recognition; Convolutional Neural Network; Deep Neural Network; MFCC

1. Introduction

Automatic speech recognition is the method of translating a speech signal into a series of words using a computer program and its algorithms. The main goal of speech recognition is to allow machines to recognize sounds and act on them. The ability of a computer to identify "receive and interpret" speech and translate it into readable form or text is known as automatic speech recognition. Automatic speech recognition is the ability of a computer to understand speech as well as execute an action based on the human's instructions [1].

Nowadays, automatic speech recognition (ASR) has become a significant field in artificial intelligence (AI) research. ASR tasks are employed for translating speech waves, or signals, to word-sequence representation based on smart computations [1]. There is a wide area of Automatic speech recognition applications in many fields, such as voice applications, automatic language translation, human-computer interactions (HCI), and many via-voice systems [2]. However, most ASR works have been applied in the English language, and limited studies have utilized the Arabic language in the ASR field. Arabic is one of the complicated languages in computerizing works; hence, developing Arabic speech recognition systems has a tangible level of hardness for many reasons; the richness and sparseness nature of Arabic vocabulary data, lexical diversity, the non-discretized huge amount of available text, and the number of distinct alive verbal dialects in the world. Besides, the complexity of Arabic language written words morphology. Although, it is very rich in vocabulary [3]. So, several challenges for speech recognition research have been presented due to the large vocabulary of the Arabic language [4].

Recently, there are a growth of studies that have developed powerful Arabic automatic speech recognition (AASR) systems [4, 5]. The written words in the Arabic language have diacritics (Arabic haricot); a number of distinct symbols that are positioned above the Arabic letters [3]. The Arabic diacritics denote sounds corresponding to

English vowels and Chinese tones, they are important for recognizing the word's and sentence's sense, as another Arabic ASR challenge [4]. Furthermore, the dialectal Arabic ASR acoustic model building is a challenging task. The training task of these models entails the correct Arabic dialect. So, working with Arabic dialectal ASR has many challenges. To obtain a good and accurate model, a large dataset has to be gathered due to the lack of training data benchmarks [3]. ASR is a basic component of a virtual assistant. It works by processing a human voice and training a system to recognize vocabulary in that voice. ASR has many applications ranging from speech-based controls to online gaming to deliver commands to Iota devices. In the last five decades, ASR became an active research area. ASR is important for human and human-machine communication [1]. Single-word speech recognition can be used in voice interfaces for applications with keyword detection, which can be useful on mobile and embedded devices.

Through previous studies on converting audio files to text, note many problems and challenges that can face research in this field and can be summarized as follows:

a. In previous research, systems and algorithms have been developed that can recognize one language, and the process of designing a system that works on more than one language at the same time is one of the challenges facing research.
b. The number of words recognized in previous research is mostly small, which is why creating a language dataset that Arabic contains many words is a second challenge in this study.
c. The third challenge is that most of the previous research is working on recognizing individual words.

Through the previous problems and challenges, the objectives of the proposed system for speech to text recognition can be summarized as a following:

- Develop a model through which words can be recognized for more than one language (Arabic and English) at the same time.
- Create a dataset for more than 200 words and up to 20 speakers.
- Developing a deep learning model to reach high accuracy in recognizing words, in addition to developing the system to work on words and sentences at the same time.

2. Related Work

The process of converting audio to written text is one of the important topics that fall into different fields, and there are many previous studies that used different algorithms and techniques. The most important researches can be summarized as follows:

In 2022 [6] this paper proposed approach deals with recognition of audio queries which contain a mixture of words in two different languages - Kannada and English. The novelty in the approach presented, is the use of a next Word Prediction model in combination with a Deep Learning speech recognition model to accurately recognize and convert the input audio query to text. This paper used MFCC to features extraction from audio and proposed RNN (LSTM) model for words prediction. The dataset generates using most frequently used sentences were chosen from this group to generate the speech data. A total of 64 words were chosen and recorded by three different people, with each word being recorded ten times by each person, totaling 1920 recordings. The accuracy of this proposed model was 71% for words recognition.

In 2021 [7] this paper aims to establish a Formal Malayalam Speech to Text converter for the language of Malayalam. The system considers only isolated words with constrained vocabulary. The word which is spoken by the speaker is given as the input to the system is presented in the display as the output. The input audio word dataset was collated with these stored words. Pre-processing process includes the transformation of speech signal into digitized format. This digital signal is passed to the first order filters for the smoothening signals, which would help in the rise of signal's energy at a higher frequency. MFCC is the systematic technique for feature extraction. Following the pre-processing, syllabification, and feature extraction procedure, HMM is used to identify the speech and training. The speech recognition system using LSTM. The system is giving an accuracy of about 91% when modelled using HMM classification and LSTM training.

In 2021 [8] the paper worked on designing a word-tracking model by applying speech recognition features with deep convolutional neuro-learning. Six control words are used (start, stop, forward, backward, right, left). Words from people of different ages. Two equal parts, men and women, contribute to speech dataset which is used to train and test proposed deep neural networks. Collect data in different places in the street, park, laboratory and market. Words ranged in length from 1 to 1.30 seconds for thirty people. Convolutional Neural Network (CNN) is applied as advanced deep neural networks to classify each word from pooled data set as a multi-class classification task. The proposed deep neural network returned 97.06% as word classification accuracy with a completely unknown speech sample.

In 2021 [9] this paper focused on single word Arabic automatic speech recognition (AASR). Two techniques are used during the feature extraction phase; Log frequency spectral coefficients (MFSC) and Gammatone-frequency cepstral coefficients (GFCC) with their first and second-order derivatives. The convolutional neural network (CNN) is mainly used to execute feature learning and classification process. CNN achieved performance enhancement in automatic speech recognition (ASR). Local connectivity, weight sharing, and pooling are the crucial properties of CNNs that have the potential to improve ASR. The dataset used contains 20 words spoken by 50 native male Arabic speakers. It was found that the maximum accuracy obtained when using GFCC with CNN is 99.77 %. The outcome results of this work are compared to previous reports and indicate that CNN achieved better performance in AASR.

In 2020[10] this paper proposed speech recognition system using Convolutional 1Dimensional Neural Networks (CNN) to increase efficiency and accuracy. The speech recognition model selects the best speech signal illustration by feature extraction of the audio signal within the Time domain as speech is single-dimensional will be sometimes processed victimisation sliding windows that are fed into a network. Conv1D handles speech signals by providing a full frequency feature vector at every instant which completely describe the sample. The dataset contains audio samples as deals with speech. Audio samples are simply English spoken words that are one-second long utterances of 10 short words. The system gives 81.49% test accuracy for words recognition.

In 2023 [11] this paper, five neural network models, namely, CNN, LSTM, Bi-LSTM, GRU, and CONV-LSTM, for speech to text recognition. This paper used MFCC for audio features extraction. The models trained the networks using Audio MNIST dataset. MNIST dataset consists of 10000 audio samples of spoken English digits (0-9) of 60 different speakers. Experimentally, CNN and Conv LSTM network model consistently offers the best performance with accuracy 98.6%. In 2023 [12] this paper is used to create a system for automatic recognition of user commands for a graphical editor. The automatic speech recognition system is used as a recognition module in a plug-in for a graphics editor. The list of commands contains 20 commands, the use of the Mel-frequency cepstral coefficients and Dynamic time warping algorithm is justified by the fact that the vocabulary is limited and the commands are short. The accuracy of command recognition was evaluated for various speakers. The average recognition accuracy is 93%.

3. Proposed System

In this section the main architecture of the proposed system will be illustrated. The proposed system consists of four main steps, as illustrate in the Figure (1). Each of these steps has an important role in the proposed system where the system begins to reduction the noise from the audio file and then extracts the fragmentation of the audio file depending on the interval between words, in the next step the features of each fragment are extracted and a word prediction is made.

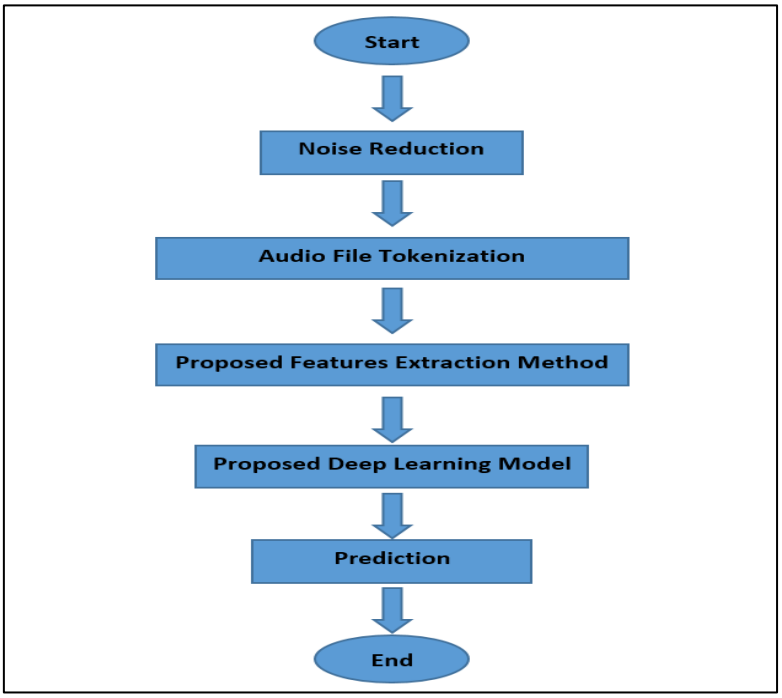


Figure 1. Proposed System Architecture

In this study, three datasets were relied upon, two of which were collected and downloaded from the Internet and the third was real and created by a group of people.

a. Arabic Dataset

This dataset was downloaded from the Internet from (https://www.kaggle.com/ datasets/abdulkaderghandoura/arabic-speech-commands-dataset) that called Arabic Speech Commands Dataset which contains 12,000 audio files divided into 40 classes each class contains 300 audio files. The dataset was divided into two parts, training and testing as a following:
Training: This part represents 80% of the dataset, which contains 9,600 audio files that are used to train the system to recognize Arabic words.
Testing: This part represents 20% of the dataset, which contains 2400 audio files that are used to test the system to predict Arabic words.

b. English Dataset

This dataset was downloaded from the Internet from (https://www.kaggle.com/datasets/neehakurelli/google-speech-commands) that called Google Speech Commands which contains 60,000 audio files divided into 30 classes each class contains 2000 audio files. The dataset was divided into two parts, training and testing as a following:

Training: This part represents 80% of the dataset, which contains 48000 audio files that are used to train the system to recognize English words.

Testing: This part represents 20% of the dataset, which contains 12000 audio files that are used to test the system to predict English words.

c. Real Dataset

In this section the real dataset generated will be illustrated. This dataset contains 4000 audio files divided into 200 classes each class contains 20 audio files recorded by 20 persons. This dataset was created for the purpose of training the proposed system to be able to convert audio files containing a set of words into text by merged dataset was recorded by our self that contain 160 classes with Arabic dataset 40 classes. The dataset was divided into two parts, training and testing, which are as follows:
Training: This part represents 80% of the dataset, which contains 3200 audio files that are used to train the system to recognize real Arabic words.
Testing: This part represents 20% of the dataset, which contains 800 audio files that are used to test the system to predict real Arabic words.

3.1. Noise Reduction

The process of removing noise from audio files is one of the important processes because the recording of audio is done by using a microphone. The audio file is exposed to different types of noise such as loud and low sound in addition to some background sounds. This noise negatively affects the audio files and word recognition The Algorithm (1) illustrates the steps to remove the noise.

Algorithm (1): Noise Reduction
Input: Audio File
Output: Audio File without noise
Process: ad audio file in wav format omputing a spectrogram of a signal atistics are calculated over spectrogram of the noise (in frequency) threshold is calculated based upon the statistics of the noise (and the desired sensitivity of the algorithm) mask is determined by comparing the signal spectrogram to the threshold. e mask is smoothed with a filter over frequency and time. e mask is applied to the spectrogram of the signal, and is inverted If the noise signal is not provided, the algorithm will treat the signal as the noise clip, which tends to work pretty well. duces noise in time-domain signals like speech, bioacoustics, and physiological signals. **End**

3.2. Audio Tokenization

The process of splitting the audio file into small files is one of the important steps in analyzing the audio file and converting it into a written sentence. The process of dividing the audio file into a set of clips is done through a series of steps, which is to calculate the number of zeros in the audio file, which represent the places of silence, and then generate a mask through which the places of cutting the audio file are determined. The following Algorithm (2) explains the steps for breaking down an audio file into a group of files.

Algorithm (2): Audio Tokenization

Input: Audio File without noise

Output: Series of fragment file

Process:

tialize the following parameters:
Min_silence_lenght = 0.08 second
Percentage_for_silence = 0.01 second
mputing sample rate of audio file.
reate mask of silence as a following:

$$eps = waveform.max() \times percentage\_for\_silence$$

$$silence\_mask = |waveform| < eps$$

lit audio file by the following:
Specify the first silence location
Loop from 0 to length of waveform
    start = location of silence
    end = start + silence_mask
    next_pos = start + end
    part = waveform[prev_pos:next_pos]
    prev_pos = next_pos
    if len(part) > 0
        chunks.append(part)
End loop
turn series of fragment file
End

3.3. Proposed Features Extraction Method

In order to obtain a high-accuracy word prediction, an algorithm must be developed to extract features from audio files. An algorithm for extracting features based on the MFCC algorithm with standard deviation has been proposed. These techniques are considered one of the most powerful techniques that extract features from audio files and clearly distinguish sound waves. The Figure (2) illustrate proposed features extraction method architecture.

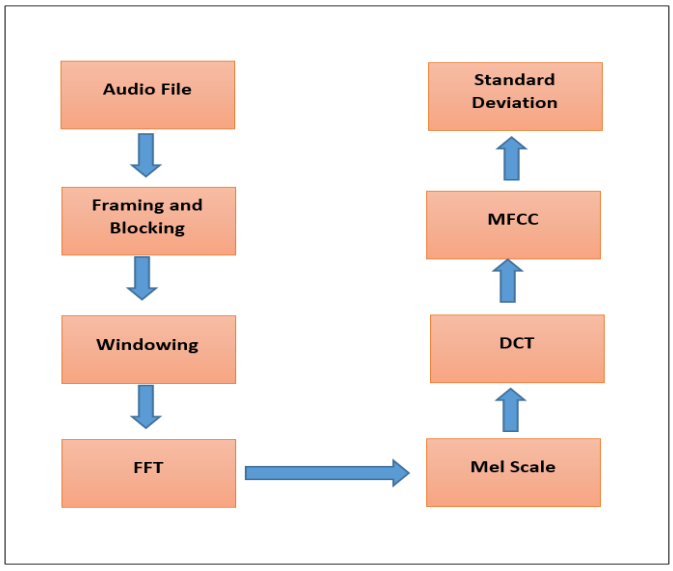


Figure 2. Proposed features extraction method architecture

a. **Framing and blocking:** In this step the continuous 1D signal are blocked into small frames of N samples, with next frames separated by M samples. The reason of dividing the given 1D signal into small frames having sufficient samples to get enough information. Because, if the frame size smaller than this size is taken then the number of samples in the frames will not be enough to get the reliable information and with large size frames it can cause frequent change in the information inside the frame. So, while working with MFCC these parameters are very common in practice. This process of breaking up the signals into frames will continue until the whole 1D signal is broken down into small frames Figure (3).

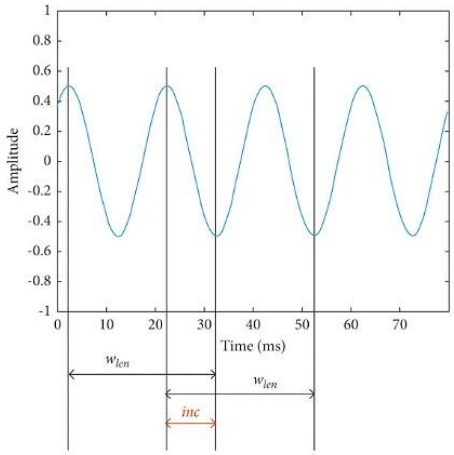


Figure 3. Audio signal framing

b. **Windowing:** is done for minimizing the disruptions at the starting and at the end of the frame, the frame and window function is being multiplied Figure (4). If the window being defined is $W_n\ (m), 0\ \leq\ m\ \leq\ N_m - 1$ where N_m stands for the quantity of samples within every frame, the output after windowing the signal will be presented as $Y\ (m)\ =\ X\ (m)\ W_n\ (m), 0\ \leq\ m\ \leq\ N_m - 1$ where Y (m) represents the output signal after multiplying the input signal represented as X(m) and Hamming window represented by $W_n(m)$. Basically, mainly hamming window is applied for carrying out windowing which usually represented as:

$$W_n(m)\ =\ 0.54 - 0.46\ Cos\ (2\pi m/\ (N_m\ - 1)), 0\ \leq\ m \leq\ N_m\ - 1$$

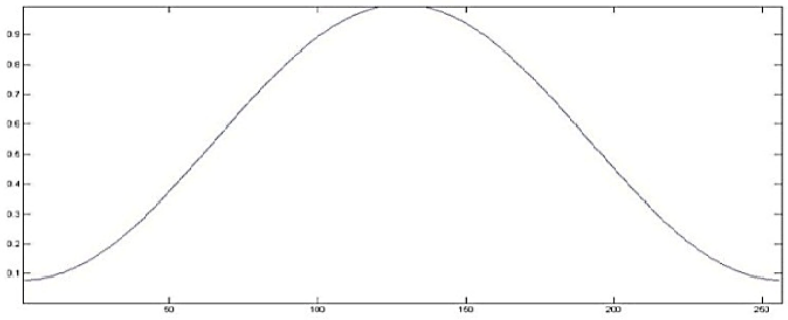


Figure 4. Humming window

c. **FFT (Fast Fourier Transform):** is used for doing conversion from the spatial domain to the frequency domain. Each frame having N_m samples are converted into frequency domain. Fourier transformation is a fast algorithm to apply Discrete Fourier Transform (DFT). Each frame with N-M samples directly will be used as a sequence for Fourier transformation.

d. **Mel scale:** In this step, the above calculated spectrums are mapped on Mel scale to know the approximation about the existing energy at each spot with the help of Triangular overlapping window also known as triangular filter bank. These filter bank is a set of band pass filters having spacing along with bandwidth decided by steady Mel frequency time. Thus, Mel scale helps how to space the given filter and to calculate how much wider it should be because, as the frequency gets higher these filters are also get wider. For Mel-scaling mapping is need to done among the given real frequency scales (Hz) and the perceived frequency scale (Mels) Figure (5).

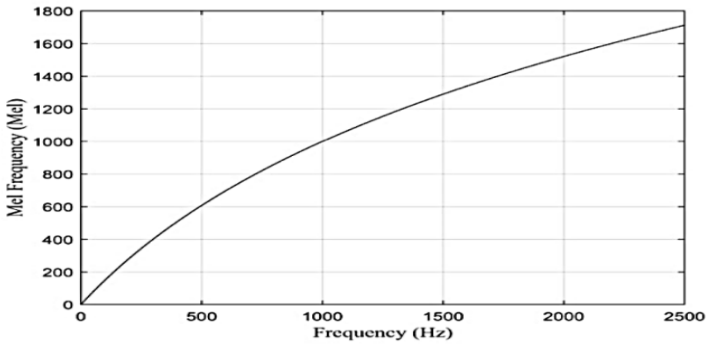


Figure 5. Mel scale

e. **Discrete cosine Transform (DCT):** This process of carrying out DCT is done in order to convert the log Mel spectrum back into the spatial domain. For this transformation either DFT or DCT both can be used for calculating Coefficients from the given log Mel spectrum as they divide a given sequence of finite length data into discrete vector. However, DFT is generally used for spectral analysis where as DCT used for data compression as DCT signals have more information concentrated in a small number of coefficients and hence, it is easy and requires less storage to represent Mel spectrum in a relative small number of coefficients. This instead of using DFT DCT is desirable for the coefficients calculation as DCT outputs can contain important amounts of energy. The output after applying DCT is known as MFCC (Mel Frequency Cepstrum Coefficient).

f. **Standard deviation:** is calculated by taking the square root of a value derived from comparing data points to a collective mean of a population the steps of compute standard deviation illustrate in Algorithm (3).

Algorithm (3): Standard Deviation
Input: MFCC Features
Output: Standard Deviation Features
Process: alculate the mean of all MFCC features. The mean is calculated by adding all the MFCC features and dividing them by the number of features. lculate the variance for each MFCC feature. The variance for each MFCC feature is calculated by subtracting the mean from the value of the MFCC features. **Step3:** Square the variance of each MFCC feature (from Step 2). **Step4:** Sum of squared variance values (from Step 3). ivide the sum of squared variance values (from Step 4) by the number of MFCC features in the data set less 1. **Step6:** Take the square root of the quotient (from Step 5). **End**

3.4. Proposed Audio Recognition Model

In the proposed word prediction model, a model was made by merging the CNN algorithm with the DNN algorithm. Deep learning (Convolution Neural Network CNN) and Deep Neural Network DNN algorithm was used for the purpose of classifying and retrieving words split from the image. The proposed model for classification contains seven main layers which are (three convolution layer, three pooling layers, and one dropout layer. After that add DNN model, Figure (6) illustrate proposed model.

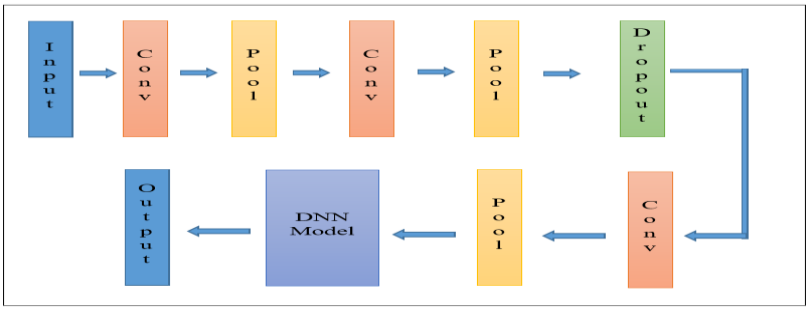


Figure 6. Proposed model

a. Convolution Layers

In the proposed system, three convolution layers were used where the aim was to increase the number of features extracted from detection word image. The three layers were generated in different sizes of kernels to extract features map. Table (1) illustrate convolution layers characteristics.

Table 1: Convolution layers' characteristics

Layer	Function of Activation	Kernel Size
Conv. Layer 1	Relu	128
Conv. Layer 2	Relu	64
Conv. Layer 3	Relu	32

b. Pooling Layers

In this layer only important and strong features are selected and three layers of pooling have been used for the purpose of choosing the features that most affect the decision to specifying word. Table (2) illustrate pooling layer characteristics.

Table 2: Pooling layers' characteristics

Layer	Kernel Size
Pool Layer 1	5
Pool Layer 2	2
Pool Layer 3	2

c. DNN Model

The DNA model consists of seven layers that use these layers to train features extracted from previous layers to be classified. Figure (7) illustrate layers of DNN model.

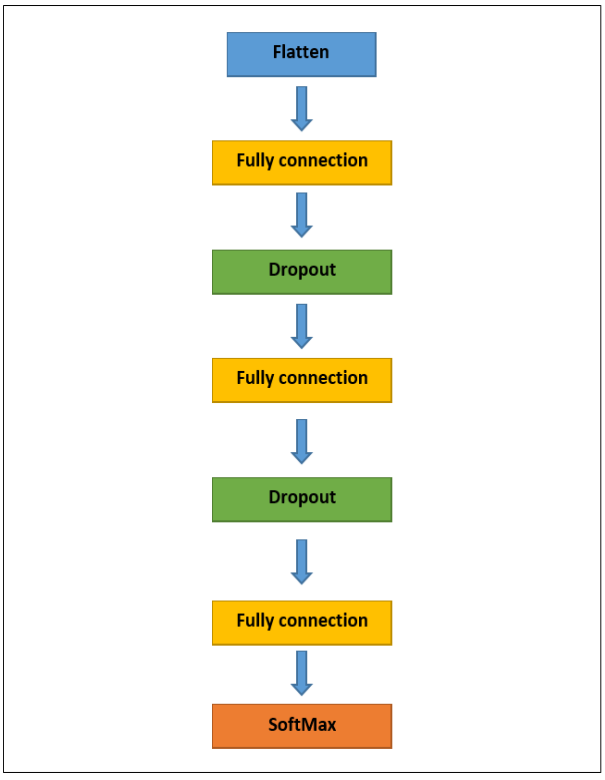


Figure 7. DNN Model

Converts features from 2D to 1D array, this process called Flatten. In the proposed DNN model there are three layers of fully connection layer and two dropout layers, in addition to SoftMax layer. Table (3) illustrate DNN layer characteristics.

Table 3: DNN layer characteristics

Layer	Size	Activation Function
Fully Connection 1	300	Relu
Dropout 1	0.5	-
Fully Connection 2	150	Relu
Dropout 2	0.5	-
Fully Connection 3	128	Relu
SoftMax	Number of classes	-

Through this layer the dataset is trained and a weight is determined for each category of images depending on the features extraction of the images of this category.

3.5. Prediction

In this step, the proposed model was trained on MFCC features images of words that were created in different person voice for the purpose of obtaining a trained model to be used in recognizing words in the prediction step. Figure (8) illustrate training and testing proposed model.

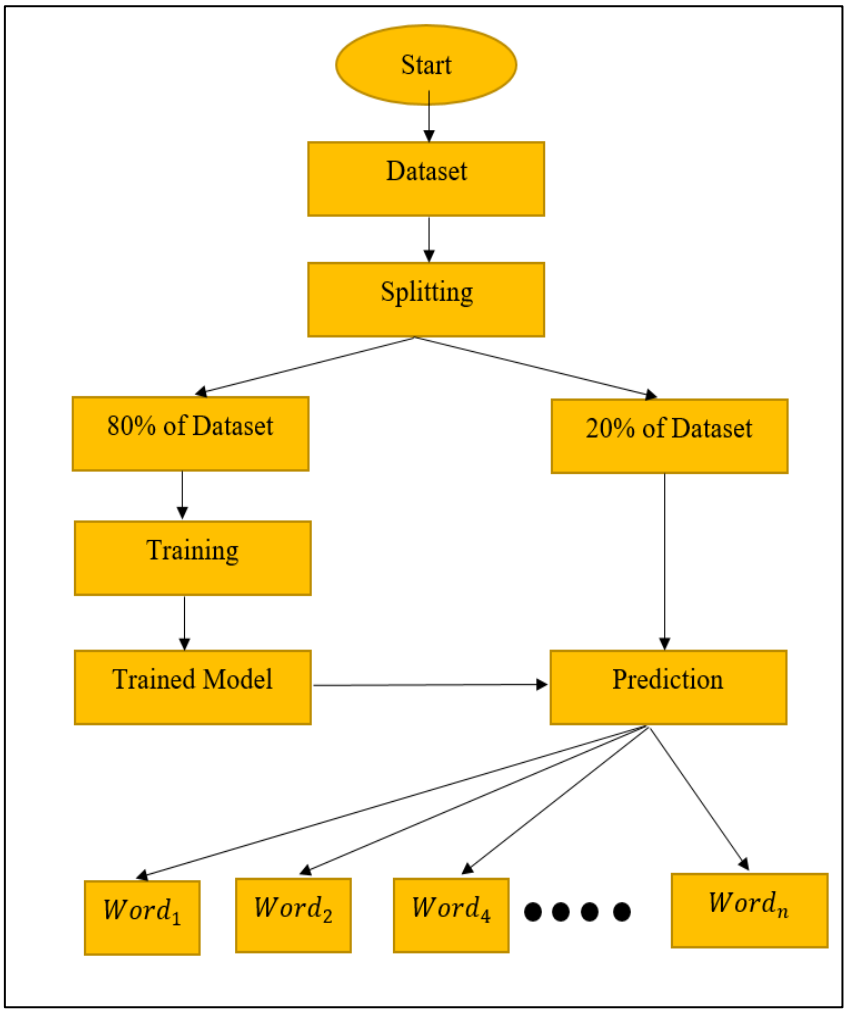
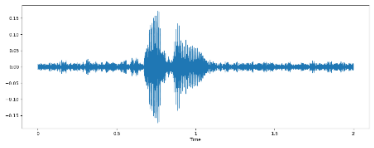
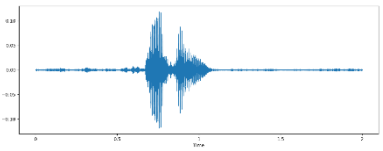
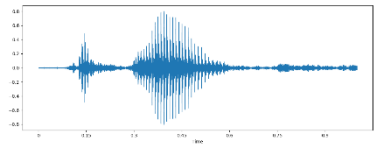
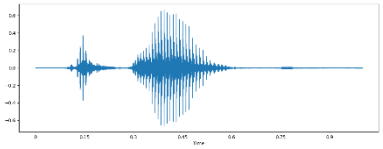
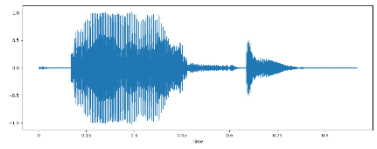
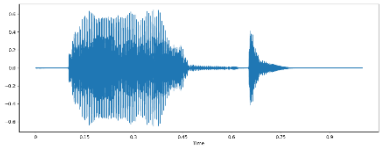
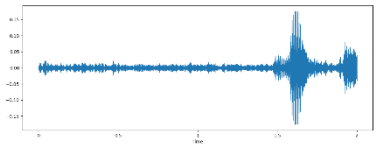
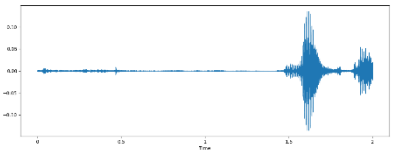


Figure 8. Training and testing proposed model

4. Results

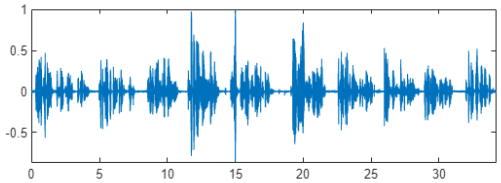
In this section, the experimental results obtained through the proposed system in converting audio files to written text will be illustrated through the use of voice processing techniques and the proposed deep learning algorithms in word prediction. The results of removing noise from the audio file will be illustrated by applying the noise removal algorithm (1) shown in Table (4) some of the experiments to remove noise.

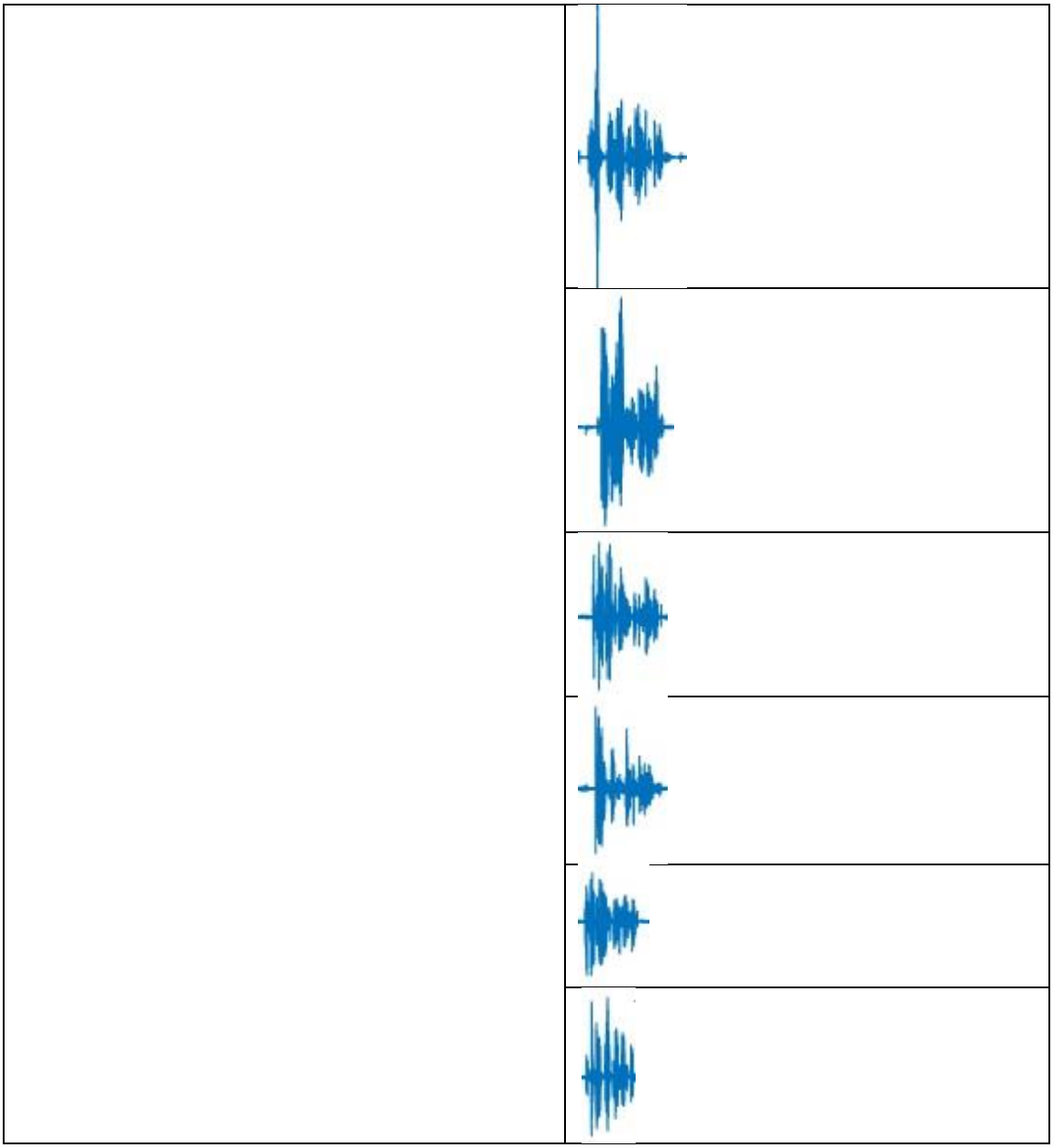
Table 4: Noise reduction results

Audio File No.	Audio File	Noise Reduction
1		
2		
3		
4		

The second step is used if the audio file contains a sentence consisting of several words. In this step, the audio file is divided into a group of files, using the proposed audio splitting Algorithm (2) and in Table (5) illustrates the results of splitting audio files that contain sentence of 10 words.

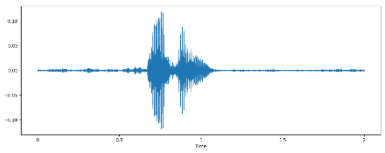
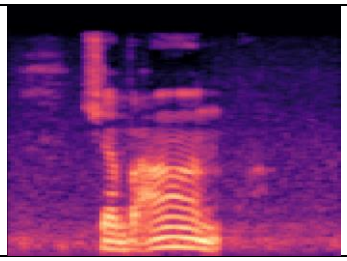
Table 5: Audio file splitting results

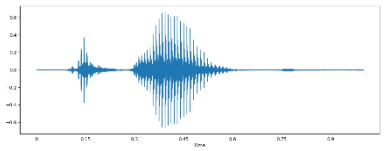
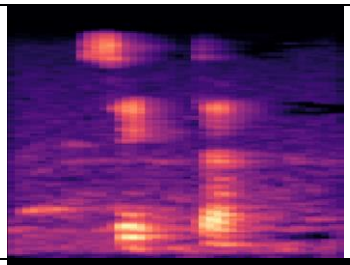
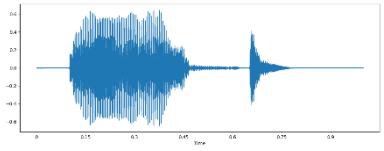
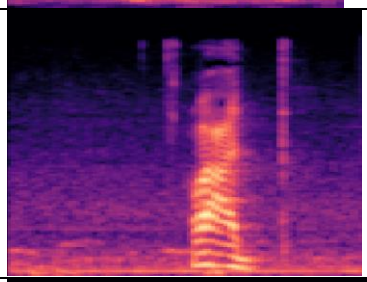
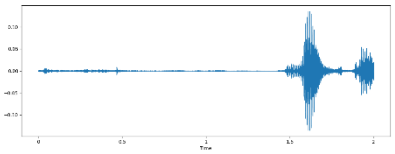
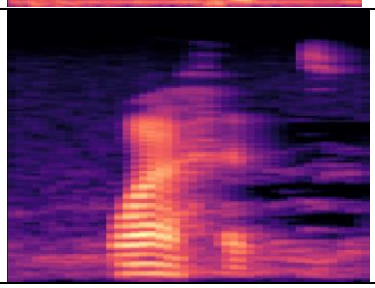
Sentence Audio File	Splitting results
	   



Now the results of extracting features from audio files and converting them into two-dimensional images for the purpose of training them will be illustrated in the next step using deep learning. The MFCC algorithm is considered one of the most accurate algorithms in extracting the features of the audio file, and an algorithm has been developed to extract the features of the audio file by adding standard deviation to the algorithm, which in turn increased the accuracy of the extracted features. Table (6) illustrate results of proposed features extraction method for some audio file.

Table 6: Features extraction results

Audio File No.	**Audio File**	**Features Extraction**
1		

2		
3		
4		

Finally, the practical experiments of training the proposed deep learning algorithm will be explained, where the algorithm has been tried by changing the splitting dataset percentage. The division of the three datasets was applied to 80 percent for training and 20 percent for testing, and the training results showed excellent accuracy in predicting words. By using the following equations can evaluate the system:

$$Accuracy = \frac{\sum_{i=1}^{N} 1(y_{Pred_i} = y_i)}{N}$$

Where ***N*** is the total number of samples, y_{Pred_i} is the predicted label for the $i\ th$ sample, y_i is the true label for the $i\ th$ sample.

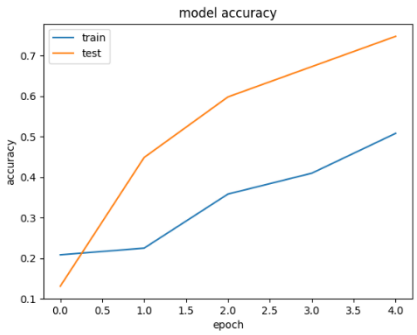
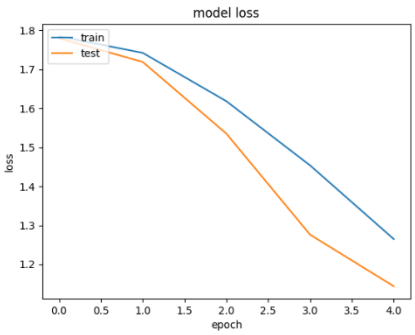
$$Loss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(y_{pred\ i,c})$$

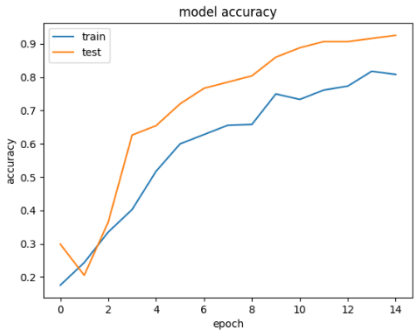
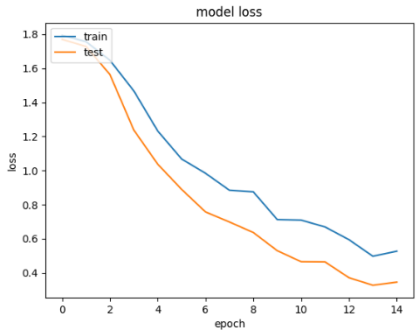
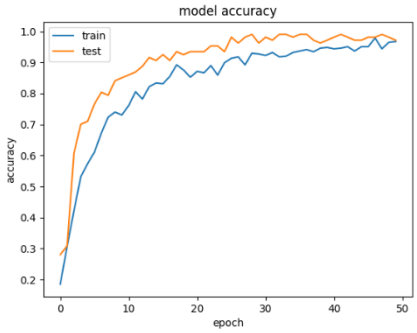
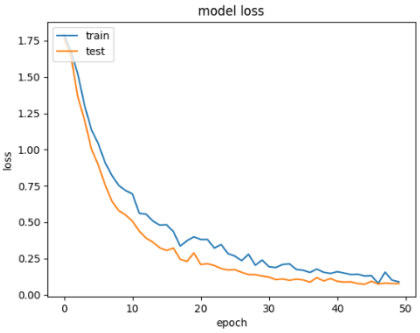
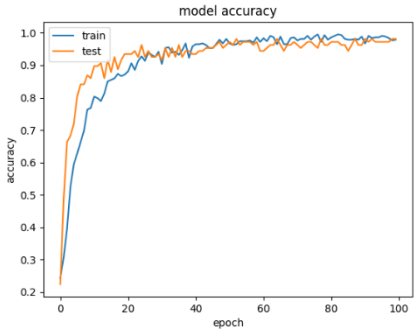
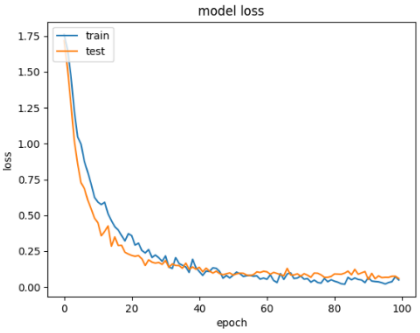
where C is number of classes, $y_{i,c}$ is 1 if sample i belongs to class c and $y_{pred\ i,c}$ is predicted probability of class c for sample i.

where experiments were conducted by changing the number of training sessions, as in the cases below:

a. **Arabic Dataset:** the training and testing results of the first dataset when using different number of epoch illustrate in Table (7).

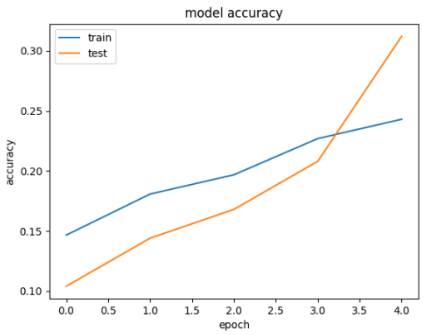
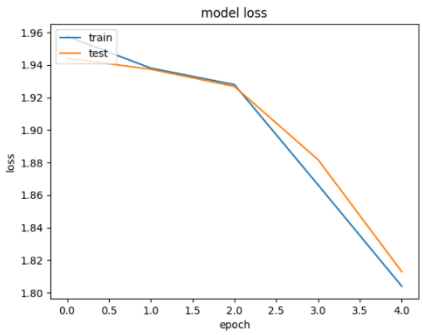
Table 7: Train-test Arabic dataset splitting 80-20 results

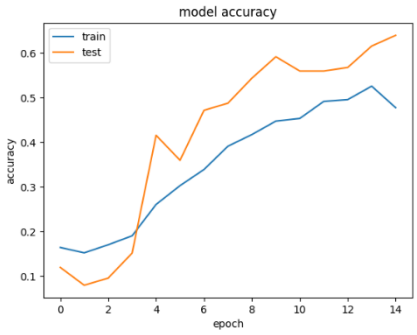
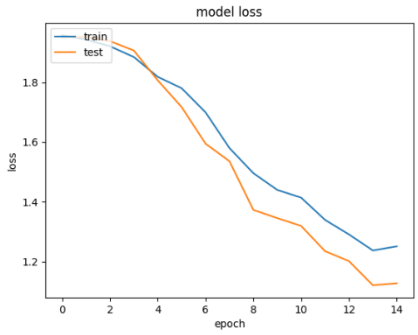
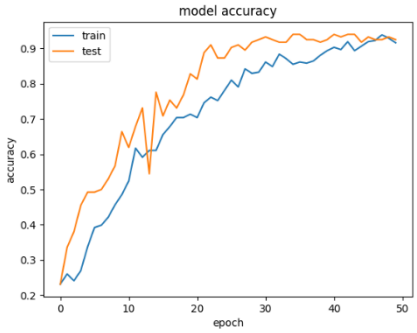
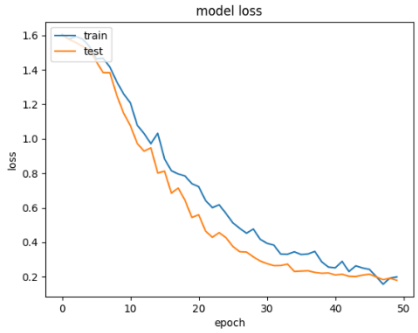
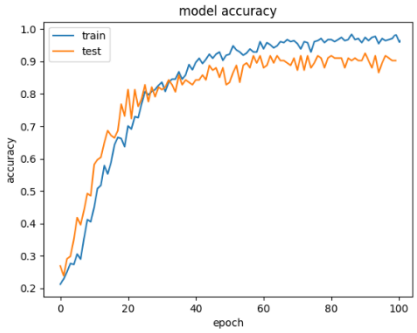
Epoch No.	ACC.	Loss
5		

15		
50		
100		

b. **English Dataset:** the training and testing results of the second dataset when using different number of epoch illustrate in Table (8).

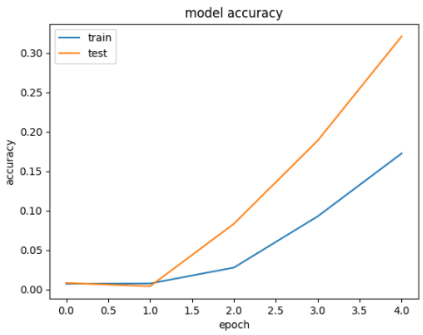
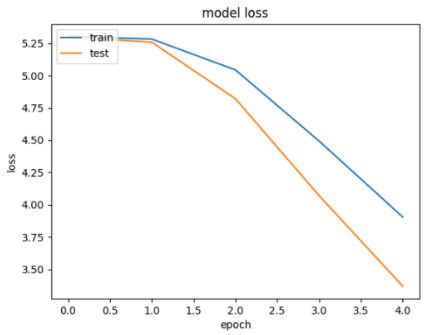
Table 8: Train-test English dataset splitting 80-20 results

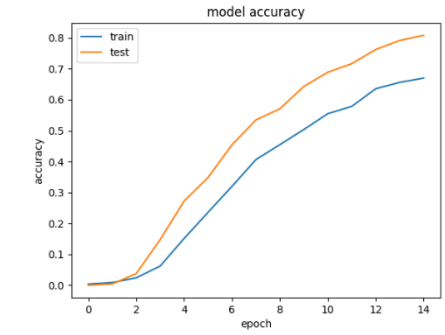
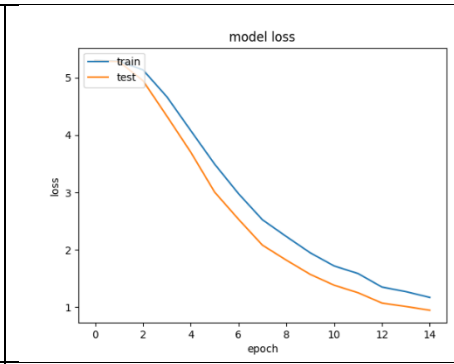
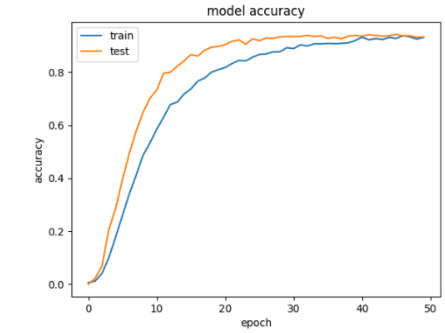
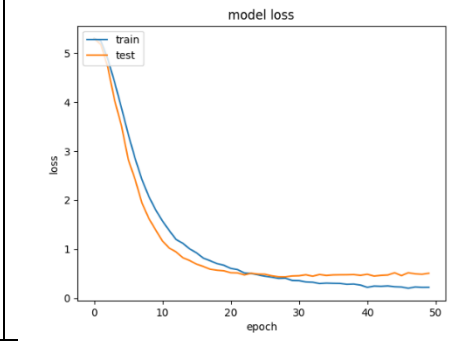
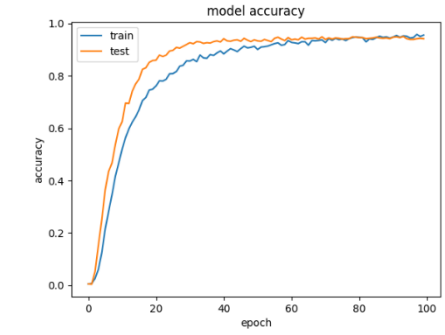
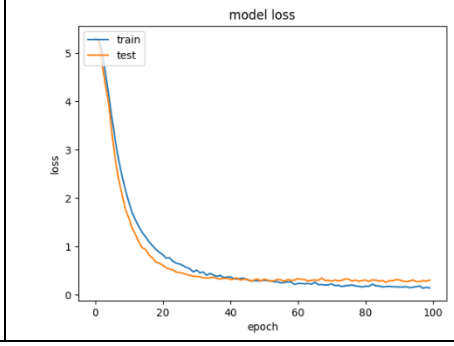
Epoch No.	ACC.	Loss
5		

15		
50		
100		

c. **Real Dataset:** the training and testing results of the third dataset when using different number of epoch illustrate in Table (9).

Table 9: Train-test Real dataset splitting 80-20 results

Epoch No.	ACC.	Loss
5		

15		
50		
100		

The previous results can be summarized in Table (10) where it can be concluded that the best number of training epoch reached is 50 in all previous experiments and on three datasets it was concluded that the best accuracy is at the number of epochs 50. The test accuracy for Arabic dataset, which contains 40 classes, reached 0.9820, while the test accuracy for English dataset, which contains 30 classes, reached 0.9454. In addition, the accuracy of the test for real dataset, which contains 200 classes, reached 0.9517. The best results for each dataset comparison can be illustrating in Figure (9).

Table 10: Summary of Train-Test Result

Split Size	Dataset	Epoch No.	Train ACC.	Train Loss	Test ACC.	Test Loss
80-20	**Arabic Dataset**	**5**	0.5082	1.2648	0.7477	1.1439
		15	0.8080	0.5266	0.9252	0.3450
		50	0.9772	0.0492	0.9820	0.0599
		100	0.9689	0.0891	0.9713	0.0772
	English Dataset	**5**	0.2430	1.8039	0.3120	1.8129
		15	0.4779	1.2511	0.6400	1.1277
		50	0.9428	0.0667	0.9454	0.1993
		100	0.9419	0.2006	0.9030	0.2818
	Real Dataset	**5**	0.1725	3.9045	0.3208	3.3687
		15	0.6700	1.1703	0.8083	0.9469
		50	0.9524	0.1416	0.9517	0.3036
		100	0.9415	0.2189	0.9333	0.5043

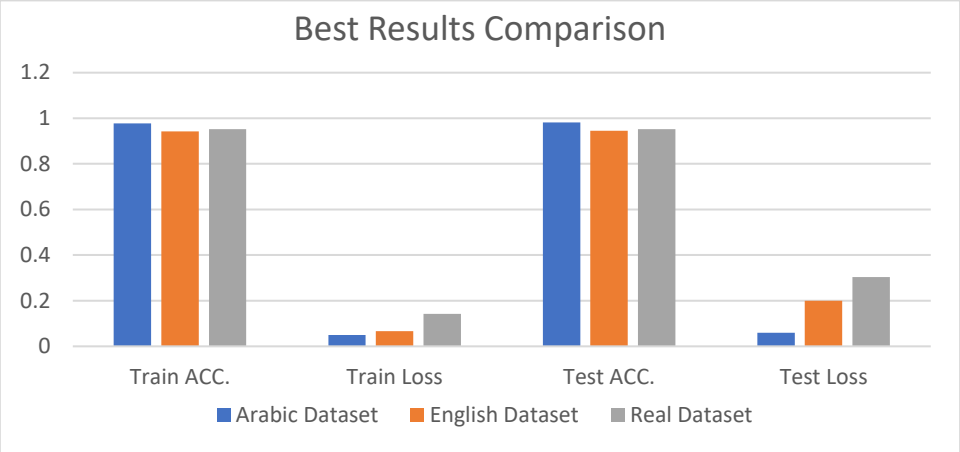


Figure 9. Best results for each dataset comparison

The results of the proposed system will be compared with previous works through the accuracy of word prediction and comparison between the number of words used and trained in addition to the algorithms used. In the Table (11) is a comparison between the results of the proposed model with previous works.

Table 11: Comparison between the results of the proposed model with previous works.

Ref.	**Published Year**	**Dataset Words**	**Features Extraction Algorithm**	**Speech Recognition Algorithm**	**Accuracy**
[6]	2022	64	MFCC	LSTM	71%
[7]	2021	-	MFCC	HMM+LSTM	91%
[8]	2021	6	Conv.	CNN	97.06%
[9]	2021	20	MFSC+GFCC	CNN	99.77%
[10]	2020	10	1D Conv.	1D CNN	81.49%
[11]	2023	10	MFCC	CNN+LSTM	98.6%
[12]	2023	20	MFCC	DTW	93%
Proposed Work		40	MFCC + Standard Deviation	CNN+DNN	98.20%
		30			94.54%
		200			95.17%

6. Conclusion

The results of the proposed system for converting audio files to text have led to several important conclusions. Firstly, the process of removing noise from audio files has a significant impact on the accuracy of speech prediction, as noise removal makes the audio clearer, leaving only the spoken words. Secondly, the proposed algorithm for splitting the audio file into smaller files, each containing a single word, has shown good accuracy by detecting decreases in sound between words to establish boundaries between them. Additionally, merging the MFCC (Mel-Frequency Cepstral Coefficients) algorithm with standard deviation for feature extraction has proven effective in identifying the best features, with notable differences observed between the features of each word. Finally, it was concluded that the optimal division of datasets is 80% for training and 20% for testing.

Funding: “This research received no external funding”

Conflicts of Interest: “The authors declare no conflict of interest.”

References

[1] H. A. Abdulmohsin, et al., "Automatic illness prediction system through speech," compute. *Electr. Eng*., vol. 102, p. 108224, 2022.

[2] H. A. Abdulmohsin, "A new proposed statistical feature extraction method in speech emotion recognition," Compute. *Electr. Eng*., vol. 93, p. 107172, 2021.

[3] Z. K. Mohammed and N. A. Z. Abdullah, "Survey for Arabic part of speech tagging based on machine learning," *Iraqi J. Sci*., vol. 63, no. 8, pp. 2676-2685, 2022.

[4] A. A. Hussien and N. A. Z. Abdullah, "A review for Arabic sentiment analysis using deep learning," *Iraqi J. Sci*., vol. 64, no. 12, 2023.

[5] A. R. Ali, "Multi-dialect Arabic speech recognition," in Proc. 2020 Int. Joint Conf. *Neural Networks (IJCNN)*, 2020.

[6] P. D. Reddy, C. Rudresh, and A. S. Adithya, "Multilingual speech recognition methods using deep learning and cosine similarity," *CS & IT Conf. Proc*., vol. 12, no. 7, pp. 1-7, 2022.

[7] H. P. Arun, et al., "Malayalam speech to text conversion using deep learning," *IOSR J. Eng*., vol. 11, no. 7, pp. 24-30, 2021.

[8] A. Alsobhani, H. M. A. ALabboodi, and H. Mahdi, "Speech recognition using convolution deep neural networks," *J. Phys.: Conf. Ser*., vol. 1973, no. 1, 2021.

[9] E. R. Abdelmaksoud, et al., "Convolutional neural network for Arabic speech recognition," *Egypt. J. Lang. Eng*., vol. 8, no. 1, pp. 27-38, 2021.

[10] A. Bhavani and N. R. Moparthi, "Speech recognition using the NN," *Int. J. Adv. Res. Eng. Technol*., vol. 11, no. 6, pp. 2663-2671, 2020.

[11] C. Sridhar and A. Kanhe, "Performance comparison of various neural networks for speech recognition," *J. Phys.: Conf. Ser*., vol. 2466, no. 1, 2023.

[12] K. Yalova, M. Babenko, and K. Yashyna, "Automatic speech recognition system with dynamic time warping and mel-frequency cepstral coefficients," *COLINS*, vol. 2, pp. 1-7, 2023.