People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
ChadliBendjedid University
Faculty of Science and Technology
IT department

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة الشاذلي بن جديد
كلية العلوم و التكنولوجيا
قسم الاعلام الالي

جامعة الشاذلي بن جديد
UNIVERSITE CHADLI BENDJEDID

Master Thesis in Computer Science

Presented by
BOUZATA Hadjer

Specialty: Intelligent Computer Systems

Theme

Development of a biometric authentication platform using voice recognition.

Defence on: 08/09/2022

Member of the jury:

Quality	First and second Name	Grade	University
President	Mme. **GASMI I**	**MCA**	ChadliBendjedid El-Tarf
Supervisor	Mr. **BENMACHICHE A**	**MCA**	**ChadliBendjedid El-Tarf**
Examiner	Mme. **MAKHLOUF A**	**MCB**	ChadliBendjedid El-Tarf

University year: 2021/2022

الملخص

يركز هذا العمل على التعرف على الكلام من كلمة واحدة، حيث يتمثل الهدف النهائي في التعرف بدقة على مجموعة من الكلمات المحددة مسبقًا من مقاطع صوتية قصيرة. يمكن استخدام التعرف على الكلام أحادي الكلمة في الواجهات الصوتية للتطبيقات مع اكتشاف الكلمات الرئيسية، والتي يمكن أن تكون مفيدة على الأجهزة المحمولة والأجهزة المدمجة. غالبًا ما يكون لهذه الأجهزة متطلبات صارمة من حيث قوة الحوسبة والذاكرة، وهو ما يتم التعرف عليه في تصميم نموذج التعرف على الكلام. لتصنيف العينات، نستخدم الشبكة العصبية التلافيفية(CNN) ذات التلافيف ثنائية الأبعاد على شكل الموجة الصوتية. على عكس الأساليب التقليدية حيث تعد هندسة الميزات أمرًا بالغ الأهمية، فإننا نستفيد من قوة التعلم العميق لتعلم تمثيل الميزة أثناء التدريب. يحقق النموذج معدل دقة يبلغ 0.9633 و0.9340 في مجموعة التحقق، وخطأ قدره 0.1274. تظهر النتائج أن النموذج يمكنه التنبؤ بعينات من الكلمات التي شاهدها أثناء التدريب بدقة عالية، لكنه يكافح إلى حد ما للتعميم على الكلمات خارج نطاق بيانات التدريب والعينات الصاخبة للغاية.

الكلمات الرئيسية: التعرف التلقائي على الكلام؛ استخراج الميزات؛ التعلم العميق؛ الشبكات العصبية؛ التصنيف؛ الشبكات العصبية التلافيفية.

Abstract

This work focuses on single-word speech recognition, where the end goal is to accurately recognize a set of predefined words from short audio clips. Single-word speech recognition can be used in voice interfaces for applications with key word detection, which can be useful on mobile and embedded devices. These devices often have strict requirements in terms of computing power and memory, which is recognized in the design of the speech recognition model. To classify samples, we use a Convolutional Neural Network (CNN) with two-dimensional convolutions on the audio waveform. As opposed to more traditional methods where feature-engineering is crucial, we leverage the power of deep learning to learn the feature representation during training. The model achieves an accuracy rate of 0.9633 and 0.9340 accuracy on the validation set, and an error of 0.1274. The results show that the model can predict samples of words it has seen during training with high accuracy, but it somewhat struggles to generalize to words outside of the scope of the training data and extremely noisy samples.

Keywords : Automatic speech recognition ; Features extraction ; Deep learning ; Artificial Neural networks ; Classification ; Convolutional neural networks.

Thanks

My thanks go first to God Almighty for the will, health and patience he gave me during all these years of study.

In particular, I would like to thank **Dr Abdelmadjid Benmachiche**, who supervised me throughout this thesis, for all his valuable advice. For his active listening, his availability.

Thank you for enriching our knowledge and guiding us all this time.

You are the teacher who managed to inspire me, and also who managed to make me want to know more.

Thanks for everything you've done!

I would also like to express my gratitude to **Dr Ali Abdelatif Betouil** for his wise advice. I thank him for having encouraged and helped me in difficult times during the realization of this work.

I sincerely thank:

Dr-Gasmi Ibtissem, for having honored me to chair the jury

Dr-Makhlouf Amina, for agreeing to be an examiner.

To the teachers of our university and especially those of the computer science department.

A big thank you to all my family for their concern and concern for me, their encouragement and followed them with patience during the course of my studies.

My thanks also go to my colleagues, for the good times we passed together. Finally I thank all those who helped me from afar or from the meadows.

Thank you all, from the bottom of the heart

Dedication

I dedicate this modest work

To the one who made great efforts for my happiness.

To the one who dreamed of seeing this day.

To the one who guided me and taught me the secrets of life

"my father".

To the one who represents my world and my happiness.

To the one who represents the day and the light of my life.

To the one who always wakes up to see me the best.

To the one who opened the portals for me and gave me tenderness and courage.

To the one who mourns to see me happy to the one who warmly awaits this day

"my dear mother".

To my brother **Abdeldjalil** and my sister **Sara**.

To my family in general.

To my best friend **Manel**.

To the entire 2022 class of the 2nd year in computer science.

To all those who have helped me from near or far**.**

Contents

Thanks **3**
Dedication **4**
Contents **5**
List of Figures **7**
List of Tables **9**
General Introduction **11**
Chapter 1: State of the Art 15
1. Introduction 15
2. Human speech 15
2.1 Speech production process 15
2.2 Speech perception 16
3. Automatic speech recognition 18
4. Acoustic speech processing techniques 19
4.1 Acoustic pretreatment 19
4.2. Speech signal feature extraction techniques 20
4.2.1 Analysis of mel scale cepstral coefficients 20
4.2.2Analysis by linear predictive coding 21
4.2.3Analysis by linear prediction cepstral coefficients 22
4.2.4Analysis by perceptual linear prediction 22
4.2.5Relative spectral analysis 23
4.3. Comparison 23
5. Methods used for ASR systems 24
5.1 Gaussian mixture model 24
5.2 Hidden markov model 25
5.3 Dynamic time wrapping 26
5.4 Support vector machines 26
5.5 Artificial neural network 27
A. Multi-layer perceptron 28
B. Deep learning 29
C. Recurrent neural networks 29
D. Long short-term memory neural networks (LSTM) 31
E. Convolutional neural networks 31
6. Advantages and disadvantages of the discussed models 35
7. Conclusion 37
Chapter 2: System Design 39

1. Introduction 39
2. UML designe 39
2.1 Use case diagram 39
2.1.1 Identification of the actors and their role 39
2.2 Sequence diagrams 40
2.2.2 Audio files recognition 41
3. Our system 41
3.1. Pre_processing and feature extraction 42
3.2. Convolution neural networks 44
3.2.1 Training the model 44
4. Conclusion 47
Chapter 3: Implementation and Realization 49
1. Introduction 49
2. Tools and working environments 49
2.1 Hardware configuration 49
2.1.1 Physical environment 49
2.1.2 Remote hardware configuration (google colab) 49
2.2 Software and libraries used in the implementation 50
2.2.1 Python 50
2.2.2 Tensorflow 51
2.2.3 Keras 51
3. The graphical interface of our project 51
4. Testing the model and recognition result 54
4.2 Comparison results between models 57
4.2.1 Testing Model 2 with Number of epochs 57
4.3 The results we obtained using MFCC in preprocessing 60
4.4 Comparing of our obtained results with RNN model 61
5. Conclusion 62
Conclusion and perspectives 63
References 65

List of Figures

Figure 1: Anatomical structure of the vocal tract. 16
Figure 2: Human auditory system. 16
Figure 3: Analogy between human and machine perception. 17
Figure 4: The different phases of the MFCC technique 21
Figure 5: Block diagram of the LPC technique. 21
Figure 6: block diagram of the extraction of LPCC. 22
Figure 7: HMM Model. 25
Figure 8: Classification of two linear separable classes. 27
Figure 9: Dense, acyclic and layered structured neural network commonly called (MLP). 29
Figure 10: RNN algorithm. 30
Figure 11: Architecture of CNN layers. 33
Figure 12: The common activation functions. 33
Figure 13: Underfitting, Optimal and Overfitting. 34
Figure 14: An example of dropconnect. The right network has had weights dropped with probability 0.5. 35
Figure 15:Speech recognition use case diagram. 40
Figure 16: Sequence diagram illustrating speech recognition. 40
Figure 17: Sequence diagram illustrating audio files recognition. 41
Figure 18: Our system architecture. 42
Figure 19: Conversion of audio signal from 1D to 2D (image format). 44
Figure 20: Configuration of the CNN model. 46
Figure 21: Hp personal computer. 49
Figure 22: Python program. 50
Figure 23: Interface of our project. 52
Figure 24: Output.wav file. 52
Figure 25: Window allows us to choose wav file. 53
Figure 26: File location. 53
Figure 27: An example of execution. 54
Figure 28: The architecture of model 1. 55
Figure 29: The result of model 1. 55
Figure 30: The architecture of model 2. 56
Figure 31: The result of model 2. 56
Figure 32: Accuracy and error of the CNN model with a number of epoch = 50. 57

Figure 33: Accuracy and loss of the CNN model with a number of epochs=90. 58
Figure 34: Accuracy and error of the CNN model with a number of epochs = 110 58
Figure 35: The result obtained using MFCC at number of epochs = 50 60
Figure 36: The result obtained using MFCC at number of epochs = 90 60
Figure 37: The result obtained using MFCC at number of epochs = 110 60

List of Tables

Table 1. Advantages and disadvantages of the discussed feature extraction methods 24
Table 2. Advantages and disadvantages of the discussed methods 36
Table 3. Identification of the actors and their role 39
Table 4. Comparison between the structure of Model 1 and Model 2 57
Table 5. Result of training and testing with epoch = 50 57
Table 6. Result of training and testing with epochs = 90.....58
Table 7. Result of training and testing with epoch = 110 59
Table 8. Comparaison results obtained by training our model with different numbers of epochs 59
Table 9. Comparaison of results obtained using MSLFB and MFCC……...61

List of Acronyms

ASR	Automatic Speech Recognition
ANN	Artificial Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
CNN	Convolutional Neural Networks
DBN	Dynamic Bayes Network
DNN	Deep Neural Networks
FFNN	Feed-Forward Neural Network
ReLU	Rectified Linear Unit
MLP	Multi-Layer Perceptron
TDNN	Time-Delay Neural Networks
DTW	Dynamic Time Warping
HMM	Hidden Morkov Model
GMM	Gaussian Mixture Model
SVM	Support Vector Machine
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
HCI	Human Computer interaction
LPC	Linear Predictive Coding
LPCC	Linear Prediction Cepstral Coefficients
MFCC	Mel-Frequency Cepstral Coefficients
MSLFB	Mel and Log Scaled Filter Bank
PLP	Perceptual Linear Prediction
RASTA	RelAtiveSpecTrAl

General Introduction

Artificial intelligence (AI) is a branch of computer science where machines are trained and designed to mimic decisive and responsive human functions without human intervention. AI consists of implementing a number of techniques aimed at enabling machines to imitate a form of real intelligence. AI is being implemented in a growing number of application areas.

The concept was born in the 1950s thanks to the mathematician Alan Turing. In his book Computing Machinery and Intelligence, the latter raises the question of providing machines with a form of intelligence. He then describes a test now known as the "Turing Test" in which a subject interacts blindly with another human, and then with a machine programmed to formulate sensible responses. If the subject is not able to tell the difference, then the machine has passed the test and, according to the author, can truly be considered "intelligent" [1].

From Google to Microsoft via Apple, IBM or Facebook, all the major companies in the IT world are now working on the issues of artificial intelligence by trying to apply it to a few specific areas. Each has thus set up artificial neural networks made up of servers and making it possible to process heavy calculations within gigantic databases [2].

With this advancement of computer science and technology, a wider area of research has opened in the area of Automatic speech recognition, Since speech is the most basic, common and efficient form of communication method for people to interact with each other, therefore persons would also like to interact with computers via speech. In this context, designing a machine that mimics human behavior, in particular the ability to use speech in a natural way and respond correctly to spoken language, has attracted the attention of engineers and scientists over the past century. Since the 1930s, when Homer Dudley of Bell Laboratories proposed a model of a system that analyzes and synthesizes speech, the need to use speech to design intelligent machines has been progressively studied, from one simple machine that responds to a small set of words, to a sophisticated machine that responds to commonly spoken natural language and takes into account the varying conditions under which speech is produced [3].

Automatic Speech Recognition (ASR) as a modeling tool, can meet this need, due to its massive applications that can be developed to help humans in their daily tasks. It can be considered as an emerging technology to enable and improve human-computer interactions. Based on major advances in statistical speech modeling in the 1980s, ASR systems today are found in tasks that require a human-machine interface, such as communications, assistance to the disabled, avionics, automotive,

domestic applications, voice control in robotics, office automation, voice response in remote query systems, etc... [4].

There are different model categorizations for ASR systems, which can be categorized according to utterances, vocabulary size, and speaker dependence. With regard to statements, the different classifications, like isolated words, connected words, continuous speech, and spontaneous speech.

As for the size of the vocabulary, the accuracy of the system depends on the complexity and the requirements of the processing. Some applications are designed to process a few words, others require an extensive vocabulary. They are with different categories such as small size vocabulary, vocabulary of medium size, large Vocabulary, very large vocabulary [5]. The last categorization concerns speaker dependence, such as a speaker dependent, speaker independent, and speaker Adaptation [6]. Depending on the recognition technique used, many modern ASR systems could be designed and created using several classification techniques, namely: Hidden Markov Models (HMM) [7], Dynamic Time Warping (DTW) [8], Dynamic Bayesian Networks (DBN) [9], Support Vector Machine (SVM) [8], K - nearest neighbors (KNN) [10]. The most widely used recognition technique is based on artificial neural networks (ANN) [11]. Recently, new families of approaches called deep neural networks have been applied successfully to different ASR problems [12].

In this thesis, our concern is to discover a method for automatic speech recognition. Where the point lies in discovering a way that enables us to process the signal, extract features, recognize the input of the human voice, and respond appropriately to the application. More explicitly, our work aims to answer the accompanying exploration question:

How to develop a biometric authentication platform by voice recognition where the human voice is converted into text with an error rate comparable to that of humans?

To answer the research question, we suggest using a modified Convolution Neural Network (CNN) method to solve the problem of speech recognition, which is an algorithm in the field of deep learning mostly used for image identification and classification. Due to the reason that CNN works better on a 2D matrix, we will use this idea to apply the CNN algorithm to the wave database.

This thesis is made up of three (03) Chapters, the first proposes the essential elements for understanding the ASR research field and the context associated with it and presents a brief summary of the state-of-the-art of the main works found in the literature based on the techniques used in the processing of acoustic speech and the classification methods used. The second chapter contains the modeling of the proposed system and explains the methodology proposed in this thesis. The last chapter contains the system implementation, the results obtained and the discussions.

Finally, a general conclusion reviews the main contributions of this thesis and offers guidelines for future work.

Chapter 1:

State of the art

Chapter 1: State of the Art

1. Introduction

Speech is one of the main means of communication between human beings, and its simplicity makes it the most popular means of communication in human society. To this end, speech can also play a key role in the development of modern man-machine interaction interfaces. Nevertheless, this simplicity (for the human being) contains a very complex processing done by our brain, from the production of speech to its perception and understanding, which makes speech difficult to automate for a machine [13].

In order to realize such interfaces, it is essential that the human communication process is imitated at the design level. As a result, the field of automatic speech recognition was born. This often refers to the sciences and technologies allowing the development and implementation of algorithms on machines whose purpose is to manipulate them vocally.

Research in this area has made commendable advances over the past few decades, driven by advances in signal processing, algorithms, computer architectures and hardware. For this purpose, this chapter essentially presents an inherent introduction to the field of automatic speech recognition and its main components. This chapter also emphasizes the methods used for speech recognition and Acoustic Signal Processing techniques.

2. Human speech

Speech is the most natural mode of communication in any human society because it is learned from childhood. It provides an easy way for humans to establish clear communication. Section 2.1 globally presents some principles related to the production of speech from the articulatory and acoustic point of view and in section 2.2 those related to its perception.

2.1 Speech production process

The speech production process takes place inside the vocal tract extending from the glottis to the lips. The process is energized from air-filled lungs. The vocal tract is a chamber of extremely complicated geometrical shape whose dimensions and configuration may vary continuously with time and whose walls are composed of tissues having widely ranging properties. Figure1 shows the anatomical structure of the vocal tract. The glottis is a slitlike orifice between the vocal cords (at the top of the trachea). The cartilages around the cords support them and facilitate adjustment of their

tension. The flexible structure of the vocal cords makes them oscillate easily. These oscillations are responsible for periodic excitation of vowels. The excitation of other sounds may be through a jet of air through a constriction within the vocal tract or a combination of the periodic excitation and what is produced by that jet of air. The nasal tract constitutes an ancillary path for sound transmission. It begins at the velum and terminates at the nostrils [14].

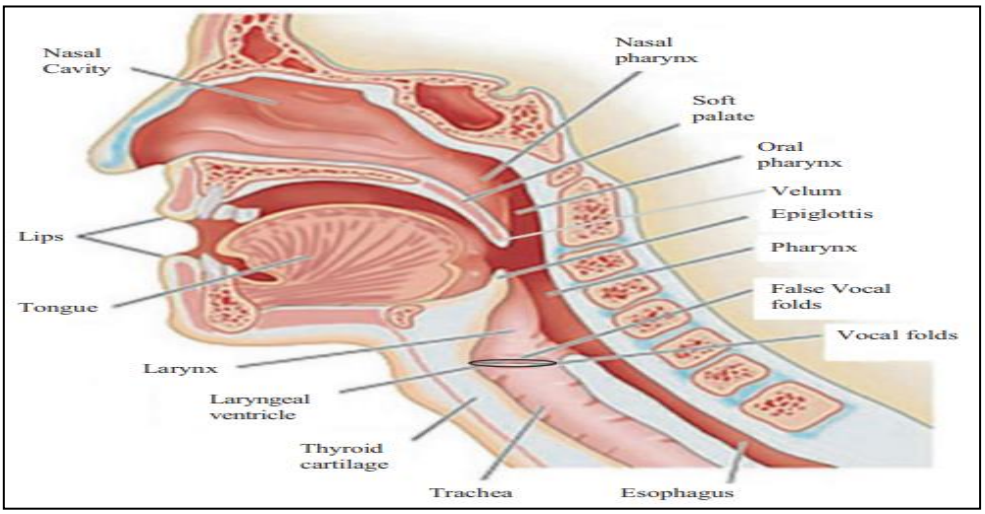


Figure 1: Anatomical structure of the vocal tract[14].

2.2 Speech perception

The ear is the main organ in the process of speech perception. It consists of an outer part, a middle part, and an inner part. Figure 2 shows the structure of the human auditory system. The main function of the outer ear is to catch sound waves which is done by the pinna. The pinna is pointed forward and has a number of curves to be able to catch the sound and determine its direction. After the sound reaches the pinna, it is guided to the middle ear using the external auditory canal until it reachesthe ear drum.

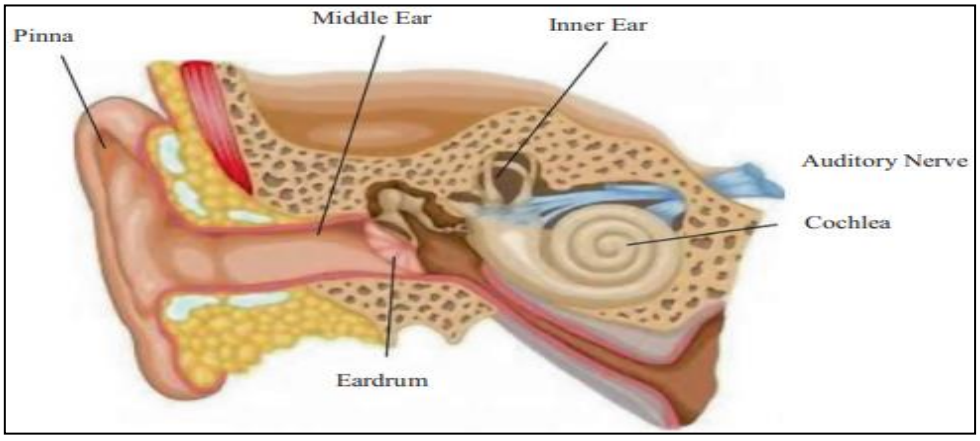


Figure 2: Human auditory system [14].

The main function of the middle ear is to magnify the sound pressure because the inner ear transfers sound through fluid not air as in the middle and outer ears. Thereafter, the inner ear starts with the cochlea, the most important organ in the human ear. The cochlea performs the spectral analysis of the speech signal through splitting it into several frequency bands which are called critical bands [9]. The ear averages the energies of the frequencies within each critical band and thus forms a compressed representation of the original stimulus. Studies have shown that human perception of the frequency content of sounds, either for pure tones or for speech signals, does not follow a linear scale. The majority of the speech and speaker recognition systems have used the feature vectors derived from a filter bank that has been designed according to the model of auditory system. There are a number of forms used for these filters, but all of them are based on a frequency scale that is approximately linear below 1 kHz and approximately logarithmic above this point. Wavelet multi resolution analysis can provide accurate localization in both time and frequency domains which can emulate the operation of the human auditory system [15]. Figure 3 describes the main steps of human perception (right part) and also illustrates the transcription of these steps in the field of signal processing (left part).

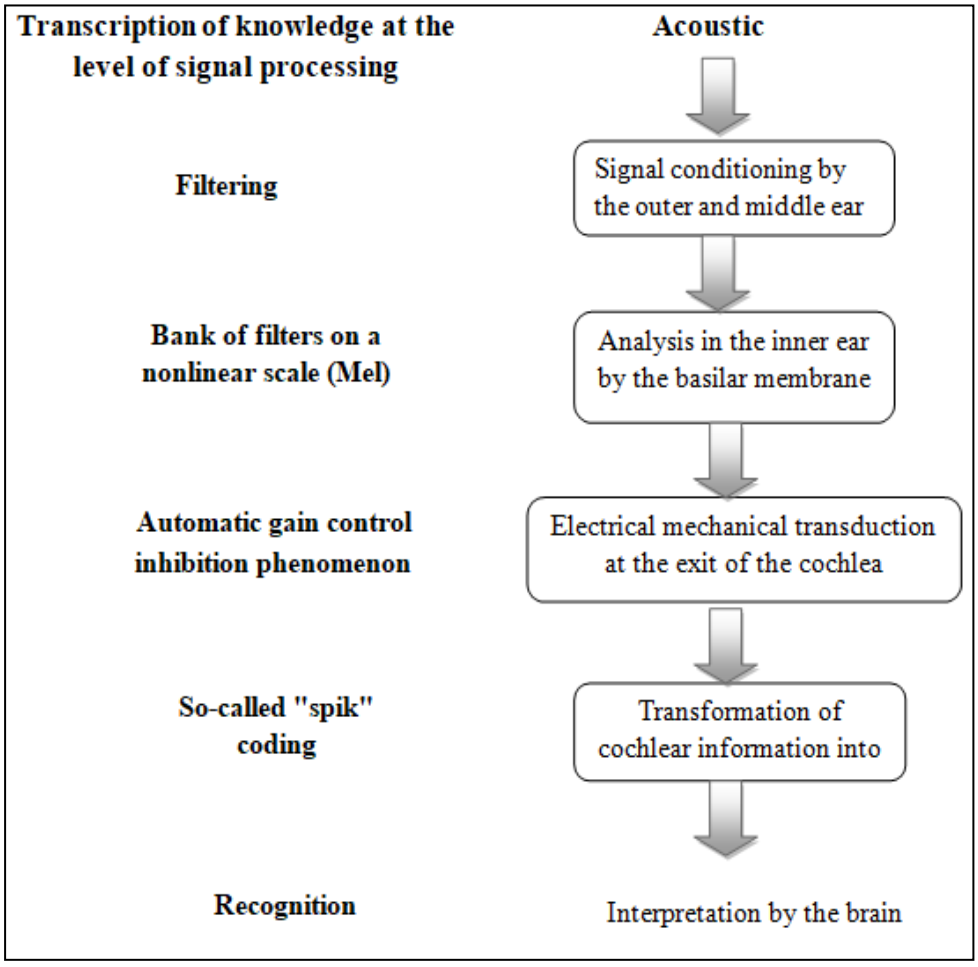


Figure 3: Analogy between human and machine perception [16].

3. Automatic speech recognition

Automatic speech recognition being a branch of artificial intelligence, mainly aims to automatically convert the speech signal into a sequence of words through an algorithm implemented as a software or hardware module [17].

ASR System is categorized into the following class depending upon the type of speaker, utterance, and vocabulary.

A. Types of speaker model

Each speaker possesses a distinctive voice that goes under the influence of different elements such as one's physical structure or even personality. The Speech Recognition system is classified into main categories based on speaker models, i.e., speaker dependent and speaker independent [18].

- Speaker-dependent models: Designed for a specific speaker, more accurate [18]. Speaker Independent Model: Designed for a variety of speakers. It recognizes the speech patterns of a large group of people [19].

B. Type of Utterance

The utterance is the speaking of a word. It could be a sub-word, word, or word sequence. It is classified into isolated words, connected words, continuous speech, and spontaneous speech as follows:

1. **Isolated word:** Words spoken with a halt prior to and after each word. It is relatively straightforward and easiest to implement because word limits are clear, which are the significant advantages of this type [20].
2. **Connected Words:** Words Expressed carefully but with no precise pause in between. Connected word systems have a resemblance to that of the isolated words but permit separate utterances to be run together [21].
3. **Continuous Speech:** Words Spoken fluently as in conversational Speech. Recognizers with continuous speech capacities are the most challenging to create because they use particular approaches to determine utterance boundaries. [20]
4. **Spontaneous Speech:** This sort of speech is natural and not rehearsed. For an ASR system to have spontaneous speech, it requires the ability to deal with an assortment of natural speech characteristics, an example words being run together, “ums” and “ahs” and even slight stutters [20].

C. Types of Vocabulary

The size of vocabulary affects the accuracy of the ASR System. Some ASR systems require a few words only (Digit) some require a large vocabulary (Spontaneous speech).

In ASR systems the types of vocabulary can be classified as follows.

1. Small vocabulary - ten words.
2. Medium vocabulary - hundreds of words.
3. Large vocabulary – thousands of words.
4. Very large vocabulary – tens of thousands of words.
5. Out-of-Vocabulary – Mapping a word from the vocabulary into an unknown word [5].

Apart from the above characteristics, the environment variability, channel variability, speaker style, sex, age, and speed of speech also make the ASR system more complex. But the efficient ASR systems must cope with the variability in the signal [5].

4. Acoustic speech processing techniques

Speech is a random, continuous, finite energy, non-stationary signal [23]. The essential goal of speech signal processing is to give a less redundant representation of speech, while allowing a fairly accurate extraction of the relevant parameters that characterize the speech signal [24]. In this part, we briefly recall the most used methods in speech analysis, for its automatic recognition. But before carrying out the acoustic analysis of the speech signal, it is necessary to carry out an acoustic preprocessing on this signal.

4.1 Acoustic pretreatment

The speech signal picked up by the microphone being analog, it is necessary to digitize it before any processing. An analog filtering operation on this signal is first performed by means of a low pass or band pass filter. The low cut off frequency must be chosen below 80 Hz, in order to avoid filtering the ″pitch ″ (fundamental frequency which prints the melody). The high cut-off frequency in the case of bandpass filtering is 11 kHz for good quality speech and can go up to 16 kHz for high quality speech. Digitizing the signal consists of sampling this signal and then quantifying each sample. According to Shannon's theorem, the loss of information between the analog signal and the corresponding discrete signal is zero if and only if we have [24]:

$$f_e \geq 2.f_{\max} \qquad (1)$$

f_e: sampling frequency.

$f_{\max}$: maximum frequency of the signal to be processed.

Bearing in mind that the frequency domain of the human voice, with the exception of opera singing, is limited to 10khz, the value of the sampling frequency must be chosen according to the type of analysis sought, but especially of the application: telephone channel, high voice quality system, system with memory size limitation, etc.

4.2 Speech signal feature extraction techniques

The redundancy and variability of the speech signal do not allow its direct use in an ASR system, hence the need for acoustic analysis. This analysis also called feature extraction is the set of methods used to extract information from this signal while maintaining the discriminating power of the signal and reducing its dimensionality. In what follows, the analyzes widely used for the creation of the vector containing the discriminative characteristics of the speech signal are detailed, while highlighting in particular the analysis of the Mel Frequency Cepstral Coefficients (MFCC) [25], given its wide use in the field of ASR. Among these different analyses, we present: – Analysis by Linear Predictive Coding (LPC) [26]; Analysis by Linear Prediction Cepstral Coefficients (LPCC) [27]; Analysis by Mel Frequency Cepstral Coefficients (MFCC); Analysis by Perceptual Linear Prediction (PLP) [28]; Relative spectral analysis (RelAtiveSpecTrAl : RASTA) [16].

4.2.1 Analysis of mel scale cepstral coefficients

The analysis of Mel scale cepstral coefficients (MFCC) was presented by Davis and Mermelstein in 1980[29]. Since then, it has been used successfully in the various ASR tasks. This analysis is based on a calculation of cepstral coefficients on a Mel scale which approximates the frequency perception of the human ear. The main idea is to average the spectrum in frequency bands corresponding to the filtering performed by the basilar membrane. In order to extract features from speech signals, MFCC analysis uses a number of Mel filter banks, from 15 to 24 linearly spaced triangle filters up to 1 kHz and logarithmically above 1 kHz, to smooth and capture the different linguistic features of the speech signal spectrum [16]. The different phases of the MFCC technique are illustrated in Figure 4.

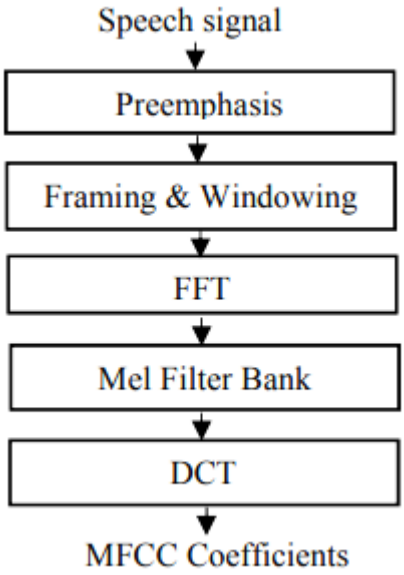


Figure 4: The different phases of the MFCC technique [25].

4.2.2 Analysis by linear predictive coding

Linear Predictive Coding (LPC) is a speech analysis technique and facilitating a features extraction. In 1978, LPC uses to make a speech synthesis. LPC doing an analysis with predicting a formant decided a formant from signal called inverse filtering, then estimated an intensity and frequency from residue speech signal. Because speech signal has many variations depending on a time, the estimation will do to cut a signal called frame [26]. The different steps that define the LPC technique are illustrated in Figure 5.

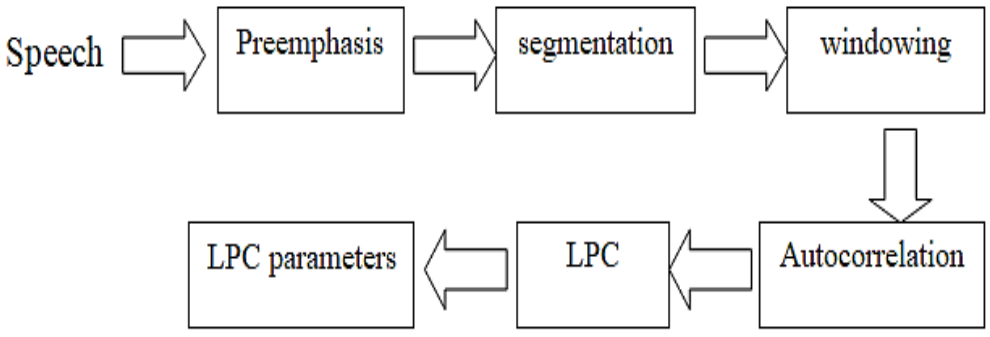


Figure 5: Block diagram of the LPC technique [26].

4.2.3 Analysis by linear prediction cepstral coefficients

The technique of linear prediction cepstral coefficients (LPCC) is mainly derived from linear predictive analysis, where the LPCCs parameters (the first p cepstral coefficients Cn) are calculated using the following formula (2):

$$c_1 = a_1$$
$$c_n = \sum_{k=1}^{n-1} \left(1 - \frac{k}{n}\right) a_k c_{n-k} + a_n \qquad (2)$$

Where:

c_i: the cepstrum coefficient of order i;

a_i: the linear predictor coefficient.

LPCC was created to address the limitations of LPC by delivering less correlated coefficients instead of the strongly correlated ones provided by LPC. Figure 6 illustrates the block diagram of the extraction of LPCCs. It should be noted that all the steps for calculating the LPCs are maintained in the calculation of the LPCC. The LPCC characteristics are calculated by introducing the cepstral coefficients in the LPC parameters [27].

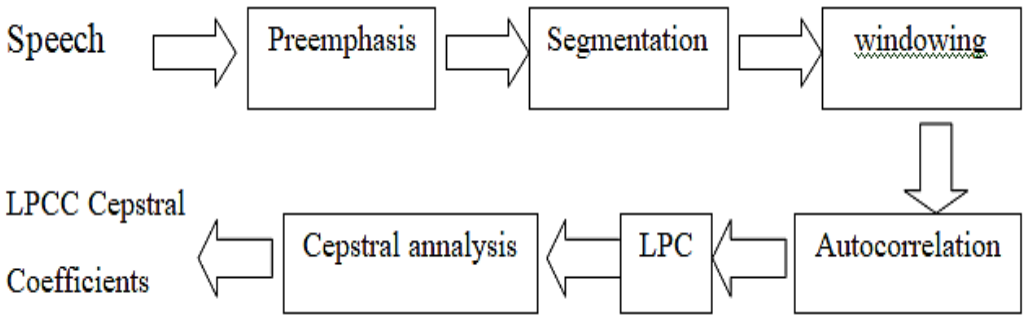


Figure 6: block diagram of the extraction of LPCC [27].

4.2.4 Analysis by perceptual linear prediction

Perceptual linear prediction (PLP) technique combines the critical bands, intensity-to-loudness compression and equal loudness pre-emphasis in the extraction of relevant information from speech. PLP gives a representation conforming to a smoothed short-term spectrum that has been equalized and compressed similar to the human hearing making it similar to the MFCC. In the PLP approach, several prominent features of hearing are replicated and the consequent auditory like spectrum of speech is approximated by an autoregressive all–pole model. It uses linear predictions

for spectral smoothing, hence, the name is perceptual linear prediction. PLP is a combination of both spectral analysis and linear prediction analysis [28].

4.2.5 Relative spectral analysis

Relative spectral analysis (RASTA) is derived from PLP analysis. The basic design is to remove the too slow or too fast variations by filtering on the amplitude spectrum, with the aim of retaining only the variations linked to the signal produced by the human being. Articulators can't move too quickly, so if features change too quickly, they may not be coming from speech. Also, if the characteristics change too slowly, this change will not be perceived. RASTA analysis is often combined with PLP analysis, giving RASTA_PLP, in order to increase the robustness of the parameters used by ASR systems [16].

4.3. Comparison

Feature Extraction Method	Advantages	Disadvantages
Linear Predictive Coding [30]	• It represents the vocal tract and is an accurate and reliable method of getting features. • It is very robust and can extract features even from speech signals that have a low bit rate [30].	• Poor speech quality and it gives residual error as output • Is not able to distinguish the words with similar vowel sounds. • Cannot represent speech because of the assumption that signals are stationary and hence it is not able to analyze the local events accurately [30]
Linear Predictive Cepstral Coefficients [31]	• The high correlation of LPC is removed by applying the cepstral analysis. • It is more robust than a simple LPC analysis [31].	• It might not be able to represent speech properly, as it assumes that the given signal is stationary and cannot analyze local events accurately. • It cannot retain prior information in the testing phase [31].
Perceptual Linear Prediction [30]	• The difference between voiced and unvoiced inputs is reduced. • It is independent of the length of the	• The feature vector being produced is highly dependent on the spectral balance of the formant amplitudes. • Channel, noise, and the equipment

	vocal tract. • The feature vector produced has relatively fewer dimensions.[30]	used to get the input signal can easily change the spectral balance.[32]
Relative Spectra–Perceptual Linear Prediction[33]	• Captures frequencies with low modulations that correspond to speech • Removes the slow varying environmental variations as well as the fast variations in artifacts [33].	• It doesn't perform well for speech signals without noise [33].
Mel-Frequency Cepstral Coefficients [35]	• MFCC provides good discrimination between phonemes. • It closely resembles the human auditory perception system because it is not linear. • It can capture important information present in the signal [32]. • The recognition accuracy is high. That means the performance rate is high • Low complexity [34].	•It is not robust to noise. • It may not be able to map continuous phonemes correctly [35].

Table 1. Advantages and disadvantages of the discussed feature extraction methods

5. Methods used for ASR systems

ASR systems use different algorithms for speech recognition. In this section, we will show some of the most popular methods:

5.1 Gaussian mixture model

In 1995 GMM was successfully applied to the speech recognition system. GMM has been widely used in statistical speaker recognition. GMM are interpreted to represent the broad acoustic classes. It Provide a smooth approximation to the underlying long term sample distribution of observations obtained from utterances by a given speaker. The Gaussian model is a probability density function. Parameters of the GMM are mean, standard deviation and component weights. GMM parameters are derived from training data using the iterative Expectation Maximization (EM) algorithm or Maximum A Posteriori (MAP) estimation from a well-trained prior model [36].

In 2010 Daniel Povey describe an acoustic modeling approach in which all phonetic states share a common Gaussian Mixture Model structure, and the means and mixture weights vary in a subspace of the total parameter space. We call this a Subspace Gaussian Mixture Model (SGMM) [37].

5.2 Hidden markov model

In the late 1960 and early 1970's Baum and his colleagues was implemented the HMM for speech recognition this models are very rich in mathematical structure and hence can form the theoretical basis for use in wide range of application [38].

An HMM [22] model is defined as a set of states, each of them associated with a probability distribution (usually multidimensional). Transitions between states are governed by a set of probabilities called transition probabilities [39]. In a particular state, an outcome or observation may be generated according to the associated probability distribution. As opposed to a classical Markov model where the state is directly observable by an external observer, in an HMM model the state is not directly observable and only variables influenced by the state are. The states are therefore hidden, hence the name hidden Markov model.

Four factors are associated with a HMM:

H- set of hidden states.

V -set of visible states.

a_{ij} – transition probabilities corresponding to the visible states.

b_{jk} – emission probability of the visible state from the hidden states. This can be understood easily with the help of the figure given below [40].

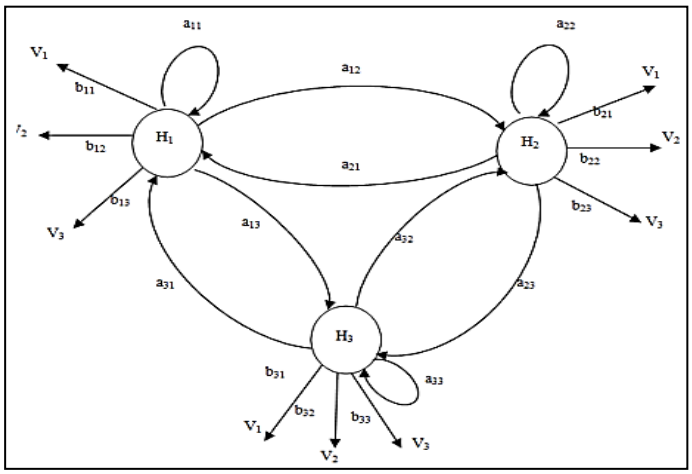


Figure 7: HMM Model [40].

In the figure, H_1, H_2 and H_3 are three hidden states and V_1, V_2 and V_3 are visible states. In any hidden state the machine may emit any one of the visible states. Here a12 is the state transition probability from hidden state H_1 to H_2. The machine may transit from one state to another state or may remain in the same state. For example a11 is the transition probability of H_1 remaining in the same state. In general the state transition is shown as:$H_{i(t-1)}\, H_{j(t)}$.

Again b_{11} is the emission probability of the hidden state H_1 if it emits visible state V_1. We can say that b_{jk} is the state emission probability of the visible state V_k if the machine is in state H_j.

5.3 Dynamic time wrapping

Dynamic time warping is a well-known algorithm in many areas. While first introduced in 1960s and extensively explored in 1970s by application to the speech recognition [41]. DTW is an algorithm that calculates the optimal warping path between two data from sound so that the output is the path warping values and the distance between the two data. Warping path is the distance between a comparison of two patterns, the smaller the warping path that is produced, the two patterns can be said to be the same.

Two words from the same word by the same user can have different times. For example, two can be pronounced with two or too. DTW solves this problem by aligning words correctly and calculating the minimum distance between two words [42].

5.4 Support vector machines

SVM (Yang 2007; Chen et al. 2014; Liu et al. 2017) can be implemented independently or as a hybrid model with HMM. SVMs solution relies on maximizing the distance between the samples and the classification border. This distance is known as the margin and, by maximizing it, they are able to generalize unseen patterns. These boundary points are referred as support vectors. SVMs use linear and nonlinear separating hyperplanes for data classification. Let's explain how SVM find an optimal hyperplane which categorizes new examples. Let's denote feature vector by$x_i \in R^n$, i=1....*M*, where *M* is a number of training samples, n-is number of features of speech signal [43]. Our goal is to classify speech unit into the two classes $y_i = -1$(out of vocabulary unit), or $y_i = +1$(in vocabulary unit) by hyper plane. This hyper plane is described as following form:

$$w * x + b = 0 \qquad (3)$$

Where *w* is a normal to the plane, *b* is a bias.

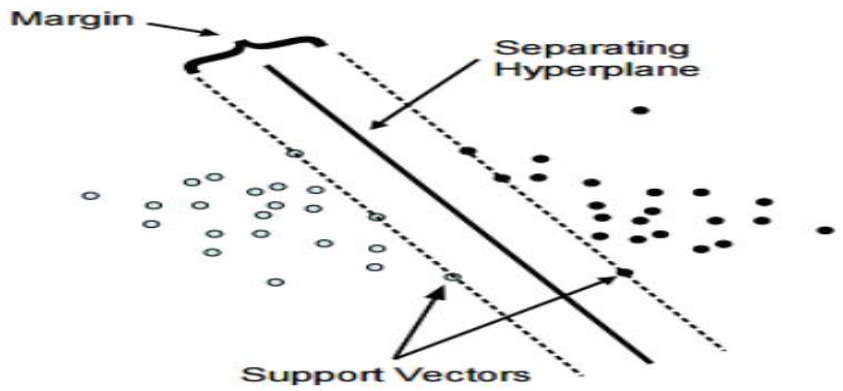


Figure 8: Classification of two linear separable classes [43].

5.5 Artificial neural network

The Artificial Intelligence approach is a hybrid of the acoustic phonetic approach (ANN) and pattern recognition approach (HMM).In particular recognition systems based on HMMs are effective under many circumstances, but do suffer from some major limitations that limit applicability of ASR technology in real word environments. Attempts were made to overcome the limitations with the adoption of ANNs as an alternative paradigm for ASR [44].

The success of artificial neural networks has undoubtedly sparked a revolution in the world of artificial intelligence in the last ten years. However, is that the basic idea of neural networks has been around since 1950s. In 1957, Frank Rosenblatt [45] invented the perceptron, a type of neural network where binary neurons units are connected via adjustable weights. Rosenblatt was inspired by the work of neuroscience in 1940s which led him to create a crude replication of the neurons in the brain. Although the idea was theoretically revolutionary and despite the countless efforts to find more efficient model layouts or better learning algorithms, the computational power of the computers of the time did not allow Rosenblatt's perceptron to obtain convincing results. Strongly convinced of his idea, Rosenblatt did not give up easily, and tried to build a real machine, with a similar functioning to that of a computer, built solely for the purpose of making the hypothesized perceptron work. The machine was made up of adjustable resistors controlled by small motors that turned the resistors off and on while the machine "learned". When completed, Rosenblatt's machine was able to classify images of simple shapes or letters.

At the time it was already clear that the conjunction of multiple levels to the Rosenblatt model would have allowed not only to significantly improve the reliability of the results of the system, but also to considerably increase the complexity of the tasks that the perceptron was able to perform. Still, at the time, there was no algorithm capable of training such a network, and the project was abandoned for several years. It took seventeen years until such an algorithm, now known as “back

propagation" was devised. The new algorithm, invented by Rumelhart, D., Hinton, G. and Williams[46] , which theoretically allowed neural networks to approximate any function, ensured that in the field of artificial intelligence, great things were about to happen [11]. The answer arrived several years later, the problem that prevented neural networks from functioning as expected seemed evident: the lack of data and computation power. For reference, at the beginning of 2002 the total amount of data produced worldwide per year was estimated to be approximately 5 Exabytes, while the data generated annually in the year 2019 is estimated to be 10 Zetabytes, an increase with the factor of 20006 [45].

In recent years, many types of neural networks with very different properties have been developed. More specifically, there are two types of networks: those whose connection graph has at least one cycle and those for which this is not the case. The former are called recurrent and the latter are called acyclic. Acyclic networks include perceptron networks and convolutes networks.

A. Multi-layer perceptron

The multi-layer perceptron which is a network of acyclic neurons structured in layers. A multi-layer perceptron is thus composed of an input layer, one or more hidden intermediate layers and an output layer. For each layer, all its nodes are connected to all the nodes of the previous layer. Figure 9 shows a representation of a multi-layered perceptron with two hidden layers, this algorithm is applied during the training phase; it is based on the backpropagation approach and the concepts of lateral inhibition. The generated output is based on the output neuron with the highest activation. One of the major drawbacks of this model is that they can only take input of fixed length, which makes them unable to handle the dynamicity of the input speech signal. Another problem is that this algorithm can only deal with small vocabularies efficiently, which makes them a good phoneme recognizer but not an efficient word recognizer [32].

Sha F, Saul LK (2007) used MLP to recognize digits of the Urdu language. The dataset used for the training purposes composed of speech signals of a single user, recorded in a clean environment. FFT and MFCC were used to extract the features from the speech signal. An accuracy of 94% was achieved in the testing phase [47]. In 2011 Sivaram GS used MLP to recognize Persian digits. An accuracy of 98% was achieved by first using MFCC to perform denoising on the dataset, and DWT was used to extract the features. The dataset used for training purposes consisted of the data of a single male speaker [48].

Another research [49] used a deep MLP network to perform speech emotion recognition. The research used the speech data present in the IEMOCAP database [50]. The network was composed

of an input layer, five hidden layers, and three output layers, one layer for each metric. The model achieved mean scores of 0.453 and 0.469 when testing with speaker-independent and speaker-dependent data, respectively.

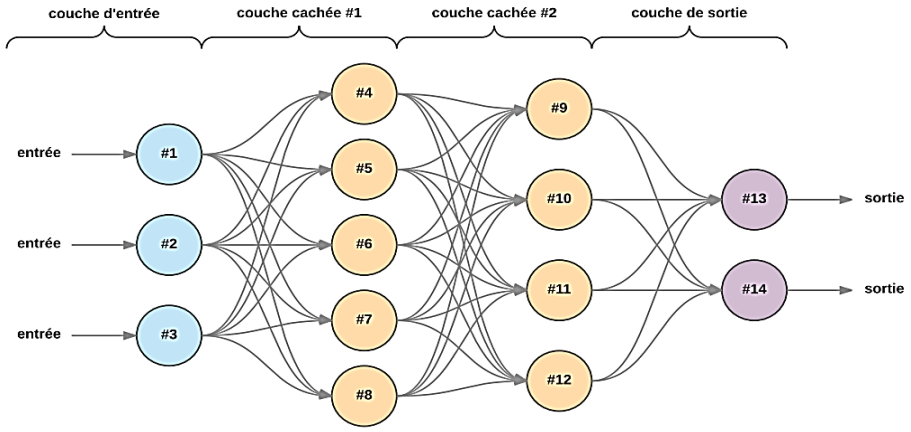


Figure 9: Dense, acyclic and layered structured neural network commonly called (MLP) [51].

B. Deep learning

Deep learning enables computational models composed of multiple levels to learn data representations with multiple levels of abstraction. These methods have improved the state of the art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning is capable of discovering complex patterns in large datasets, using the backpropagation algorithm [17], which enables the network to modify its internal parameters used to compute the representation in each level from the representation in the previous level. Recurrent nets have shone a light on sequential data such as text and speech, whereas deep convolutional networks have led to breakthroughs in image, video, speech, and audio processing [52].

C. Recurrent neural networks

A Recurrent Neural Network (RNN) is a neural network that has a memory that influences future predictions Sequential information which is stored in memory of RNNs is used for predictions.

Idea to use RNN instead of traditional neural network is in traditional neural network it is assumed that every input & every output are does not depends on each other [29]. Hence using traditional neural network is bad idea in speech processing. Prediction of any words in a sentence requires the information about the word which is utilized before i.e. past word which is processed. Having a memory is one of the specialty of RNN that makes it unique than other networks [53].

Algorithm steps involved in RNN algorithm [1] is:

X_t is input at time t.

X_{t-1} is past input, and X_{t+1} is the future input (sampled sound).

S_t is the hidden state. It is the hidden memory. St is calculated as: $S_t = f(U*X_t + W*X_{t-1})$.

Ot is output at the step t .For example if we want to predict the next word in a sentence it Would be a vector of probabilities across our vocabulary, $Ot = \text{softmax}(V*S_t)$ [54].

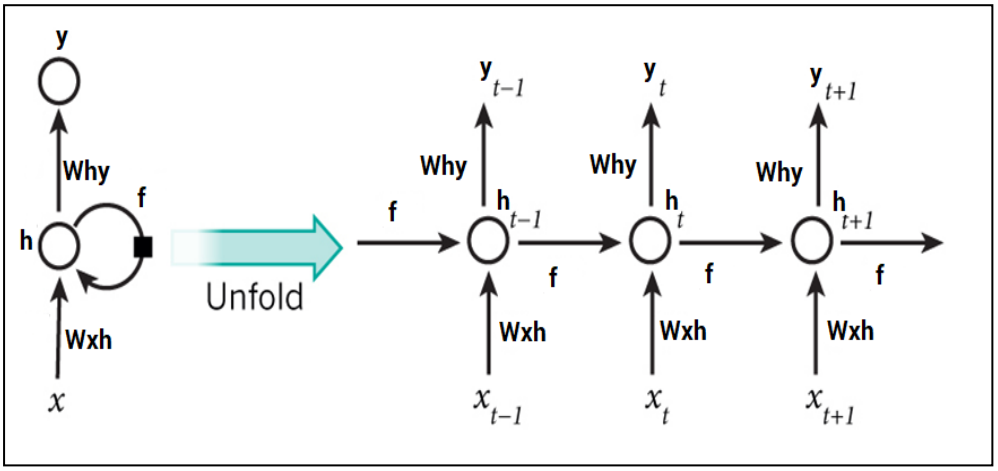


Figure 10: RNN algorithm [55].

Few things to note here are: State S_t is the memory of the recurrent neural network that can be hidden. St stores the data of what things took place in all the previous or past time steps. Output at step O_t is calculated exclusively based on the memory at time t". As mentioned above, it is a little more complicated in practice and practical implementation because St normally cannot capture data from too many time steps ago.

For implementation, traditional neural network, which are deep, uses various parameters at every layer while the same parameters are shared by RNN.

It uses (U, V, W) parameters as shown by all steps above. This shows us about that we are doing the same task at every single step, by passing various inputs at different steps. By this, there is a decrease in the number of parameters in that need to be learned.

The model proposed in [56] used RNN to recognize Bengali speech. The network consisted of three fully-connected layers, followed by a bidirectional RNN layer and then another fully connected layer with softmax as the activation function. The authors used 33 h of data from 508 speakers. The model achieved a WER of 34 with a dropout rate of 0.5, in combination with CTC and a language model.

The research proposed in [57] presented an audio-visual speech recognition system. The used an RNN-transducer, the encoder was composed of five bi-directional LSTM layers, the decoder consisted of two layers of uni-directional LSTM, and the joint space was 640 dimensional. A five-layer CNN model, called V2P, was used to extract features from the input video. The dataset used for testing and training purposes consisted of transcribed YouTube videos of 31,000 h. They received 21.5% WER on the audio-only system and 20.5 on the audio-visual system.

D. Long short-term memory neural networks (LSTM)

Due to the transformations that data undergoes when passing through an RNN, some information is lost at each time phase. After a while, in fact, the state of the RNN practically no longer contains any trace of the first inputs [45]. We can consider RNNs as short memory networks. To address this problem, various types of long-term memory cells have been introduced. Those new cells have been proved so effective that base cells are no longer used anymore. Long Short-Term Memory (LSTM) cell was proposed by Sepp Hochreiter and Jürgen Schmidhuber and gradually [58] improved over the years by several researchers. The key idea here is that the network can learn what to store in the long-term state, what to throw away, and what to read from it. [59] Compared the performance of commonly used types of GRU and LSTM, with a simple RNN. They used the TED-LIUM Corpus for testing and training. The model used consisted of one input and output layer with five hidden layers, where the fourth layer is bidirectional. LSTM performed the best with both 500-node architecture and 1000-node architecture, having WER% of 77.55, and 65.04 respectively. In terms of time, RNN was the fastest to train and LSTM took the longest.

E. Convolutional neural networks

A convolutional neural network is a subclass of neural networks that have at least one convolution layer. They are great for capturing local information (eg neighboring pixels in an image or surrounding words in text) as well as reducing model complexity (faster training, requires fewer samples, reduces the chance of overfitting). We can think of a Convolutional Neural Network (CNN) as an Artificial Neural Network (ANN) that has some kind of specialization for being able to pick-out or detect patterns from the input and make sense of them. This ability to pick up patterns is what makes CNN so useful for analyzing images or sounds, such as image classification or single word recognition [60].

A common CNN architecture is formed of alternative pooling and convolutional layers, with fully connected layers in the end, each one of them has the susceptibility to improve speech recognition performance [61].The architecture of CNN explained in figure 12.

A convolutional layer is composed of set neurons, where each neuron acts as a kernel. A convolutional kernel divides the input signal into smaller signals, called receptive fields. This particular feature allows CNN to capture most of the features present in a signal without using a large number of weights [62]. The focus of a convolution layer is to extract as many features as possible from a signal. The pooling layer performs the job of down-sampling, by retaining only the dominant value in each of the receptive fields, hence, further reducing the size of the input signal. By reducing the signal size, not only the network becomes less complex, but it also reduces the chances of over-fitting and increases generalization [63].

Vishal Passricha et al presented a compound approach with a non-homogeneous categorization of CNN and SVM where a layer was replaced softmax by SVM [61]. The research introduced in [64] applied the combination of CNN and Bidirectional LSTM (BLSTM) to identify Mandarin speech. The suggested model is of four CNN blocks, each followed by a layer of BLSTM, and then a full connected layer. Each CNN block has a convolutional layer, followed by a batch normalization layer, then a Rectified Linear Unit (ReLU) activation layer, and in the end a max pooling layer. The presented approach reached a WER% of 19.2.

Y. Yorozu and M. Hirano suggested an onward model of very deep convolutional neural networks devoid of corresponding layers and established that VDCNN performed better than CNN when experimented with MGB-3 [65].

Zied Elloumi & al [66] presented a multitasking system for performance prediction. This procedure is established on CNN. The outcomes acquired in the experimentation display that the prediction by CNN is more promising than the comparative strategy in terms of MAE (Mean Absolute Error) and Kendall scores. In addition, the combined intakes of texts and signals deliver positive results and more acceptable performance, also the CNN predicts rightly the distribution of word error rates on a set of records. Operated three various types of input, which contained MFCC, power spectrum, and raw wave format were linked with their model. The technique offered 12 convolutional layers. The stride of the convolutional model is altered according to the type of input. Improving the stride did not affect MFCC, whereas it was noticeable that the overall stride of the network played a vital role with the power spectrum and raw waveform. For training, testing, and validating purposes, the LibriSpeech dataset [32] was utilized. When employed with MFCC the model delivered 7.2% WER, with power spectrum as its input the model had 9.4% WER, and lastly, with raw waveform, it had 10.1% WER [67].

In 2020, Yang Xuebin et al, speech recognition system was designed for a group of words and used three methods of classification, including CNN, and obtained results of about 92.88% [68].

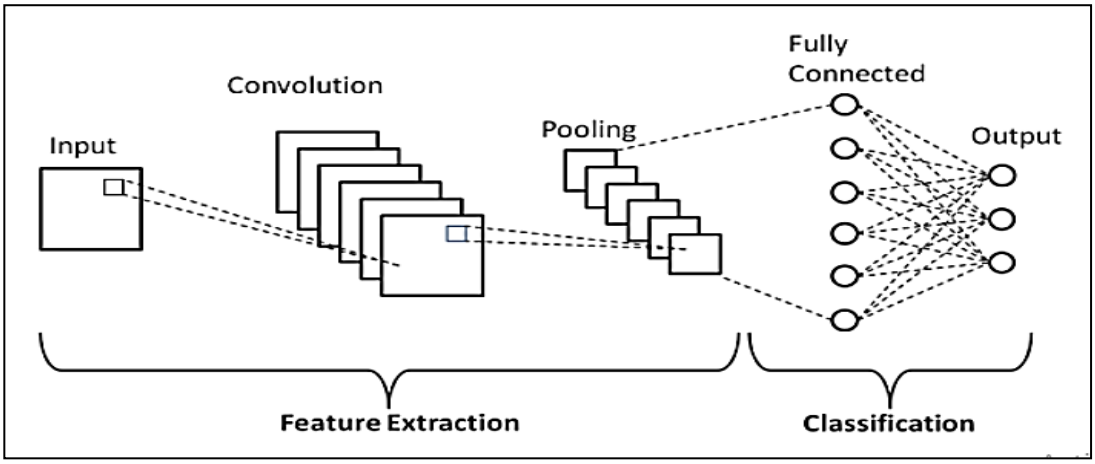


Figure 12: Architecture of CNN layers [62].

- **Activation functions**

Activation functions are non-linearities used between convolutional layers so that the neural network can model more complex than linear data. Fig 13 shows the common activation functions:

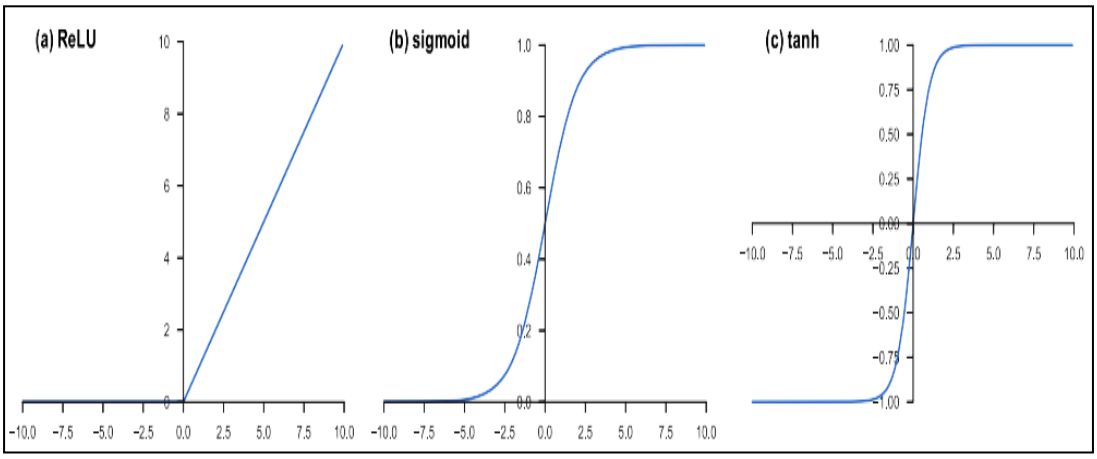


Figure 11: The common activation functions [72].

- ✓ **Sigmoid Activation Function:** The Sigmoid Function curve looks like a S-shape.

  The main reason why we use sigmoid function is because it exists between (0 to 1). Therefore, it is especially used for models where we have to predict the probability as an output. Since probability of anything exists only between the range of 0 and 1, sigmoid is the right choice.

- ✓ **Tanh Activation Function:** tanh is also like logistic sigmoid but the range of the tanh function is from (-1 to 1). tanh is also sigmoidal (s - shaped).

- ✓ **ReLU (Rectified Linear Unit) Activation Function:** The ReLU is the most used activation function in the world right now.Since, it is used in almost all the convolutional neural networks or deep learning [69].

- ➢ **Regularization**

When building neural networks or machine learning algorithms it is important to consider the problem of overfitting. Overfitting is when the model begins to learn features that are too specific to the training set. Basically, the model not only learns the general rules that lead from the input to the output, but also more rules, that perhaps describe the training set, but which are not necessarily valid at a general level [70].This process leads to a decrease in the training error but also to an increase in the evaluation error. As a result, our model will perform worse on unknown data due to these specific rules that the model has learned from the training set. If overfitting occurs when our model is too suitable for the training set, the opposite phenomenon is called underfitting, i.e. when the model learns too general rules [45]. We can find an illustration of the phenomena mentioned above in Figure 14. In the next subsection we will describe the dropout technique to overcome the problem of overfitting.

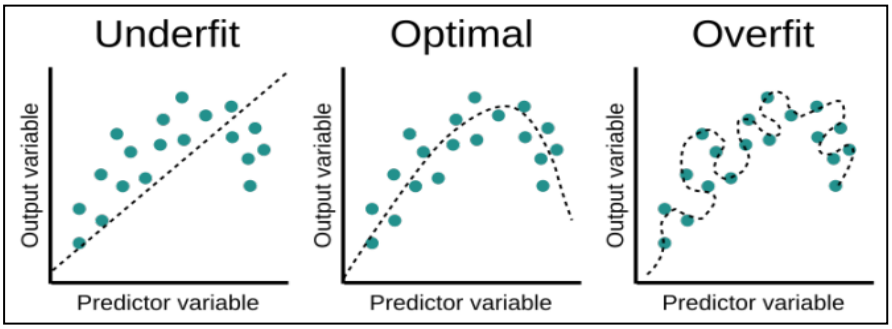


Figure 12: Underfitting, Optimal and Overfitting [45].

- ➢ **Dropout**

Dropout [71] is one of the most used and efficient regularization techniques to eliminate as much as possible the overfitting effect during the training of a neural network. During training a certain number of layer outputs are randomly ignored or "dropped out". This serves to reduce the number of weights in the network that focus on a few strong features of the training set, in order to generalize and prevent units from co adapting too much. On the basis of several benchmarks it has been shown that the introduction of dropout can give a great improvement on complex neural networks trained on a small training set17. When using dropout, it is necessary to decide the

dropout probability, which expresses the probability that each node will be excluded during training [70].

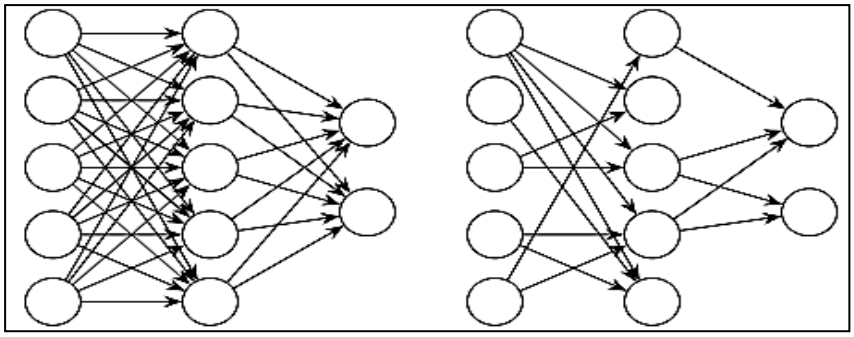


Figure 13: An example of dropconnect. The right network has had weights dropped with probability 0.5 [71].

6. Advantages and disadvantages of the discussed models

Classification Models	Advantages	Disadvantages
Hidden Markov Model [22]	• Extremely reduce the time and complexity of recognition process for training large vocabulary. • HMM can model the time distribution of a speech signal. • HMM can process variable-length inputs. • It can model both discrete and continuous sequences. [22]	• The long term dependencies are ignored as it is assumed that the current state is only dependent on the preceding state. • Must make a large priori modeling assumptions about the data, • The number of parameters that need to be set in an HMM is huge, • Amount of data that is required to train an HMM is very large, and • It does not minimize the probability of observation of instances from other classes. [77]
Gaussian mixture model[37]	•It can estimate probability perfectly and it can perform classification optimally [36]	•Time complexity [36].
Artificial Neural Network [46]	• It doesn't take time is solving difficult computational jobs efficiently • it can also take care of noisy and less quality data • It also demands slightest training	• The outcome for huge vocabulary is not proficient • it is costly require extra training time • During architecture of neural network additional fault difference transpires, entire nature of neural network is not completely comprehended till now

	data vocabulary • It can easily adapt to new environments. [41]	[47].
Multilayer Perceptrons [51]	• They are easy to implement. • MLP doesn't get stuck in local minima as it uses a random probability distribution. • It can learn according to discriminative criteria, [48]. • It can approximate any continuous function with a simple structure, • Do not require strong assumptions about the input data, • It produces reasonable outputs for inputs which have not been taught before how to deal with [51].	• They can only take fixed input lengths. • It works only for small vocabularies. • The temporal information present in the speech signal is ignored. • Inability to build speech model even though the recurrent structures are defined. [47].
Recurrent Neural Network [29]	• RNNs can process variable length inputs easily. • They retain temporal information. • Different time steps can share weights. [53]	• They are computationally expensive and require a lot of training time. • They are more suspectable to vanishing and exploding gradients. [29]
Support Vector Machines [43]	• SVM is very robust. • It does not face the problems of over-training and being stuck in local minima. • It can take high dimensional vectors as input. [43]	• It can only take fixed length inputs • The computational costs increase with the number of output classes. • Good kernel function is needed. [43]
Convolutional Neural Networkork [65]	• It can easily detect important features present in a signal. • It requires less training time [63].	• It cannot deal with variable size input [64].

Table 2: Advantages and disadvantages of the discussed methods

7. Conclusion

This chapter has attempted to introduce the field of automatic speech recognition by initially introducing human speech as the main actor and then explaining the main structure of ASR systems with a variety of applications for ASR systems. Next, a brief overview of the most widely used speech signal feature extraction techniques is discussed: LPC, LPCC, PLP and MFCC, which faithfully describe the most relevant features. At the end of the chapter, we presented some of the most common methods in ASR systems such as HMM, GMM, and ANN-based methods such as MLP, RNN and CNN. In the next chapter, we will present the modeling of our system and the proposed methodology.

Chapter 2:

System Design

Chapter 2: System Design

1. Introduction

In the previous chapter, we presented the state of the art of automatic speech recognition systems and mentioned the role of artificial neural networks in this field, which uses convolutional neural networks as a model. These networks have produced impressive and sometimes competitive results. The deep architecture of a CNN can be divided into two main parts. The first part, based on CNN convolutional layers, provides the ability to extract features, while the second is a fully connected neural network classifier whose role is to generate a prediction model for the classification task. A CNN model is described by many hyper-parameters, including the number of convolutional layers, the number of filters and their respective sizes, etc.

In this chapter, we will propose the design of our system starting with the description of different stages of automatic speech recognition, then we specify the part concerned by our study which is the recognition phase using convolutional neural networks (CNNs).

2. UML designe

2.1 Use case diagram

The purpose of the use case diagram is to give an overall view of the interfaces of future applications. It is the first UML diagram made up of a set of actors who act on use cases and who describe in the form of actions and reactions, the behavior of a system from the user's point of view. Actor: an actor is a user who communicates and interacts with the use cases of the system. It is an entity having a behavior like a person, system System: this element sets the limits of the system in relation to the actors who use it (outside the system) and the functions it must provide (inside the system).

2.1.1 Identification of the actors and their role

An actor is a person who has a well-defined role in our application:

Actor	Role
User	- Speak into a microphone. - Uploaded a file. - Check the result (play wave file).

Table 3: Identification of the actors and their role

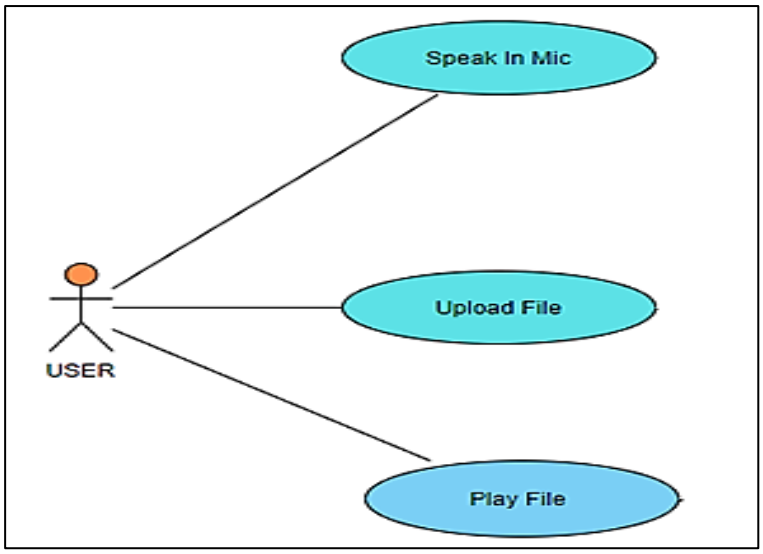


Figure 14: Speech recognition use case diagram.

2.2 Sequence diagrams

Sequence diagrams are used to describe HOW the elements of the system interact with each other and with the actors:

- The objects at the heart of a system interact by exchanging messages.
- The actors interact with the system by means of HMI (Human-Machine Interfaces).

2.2.1 Speech recognition from the microphone

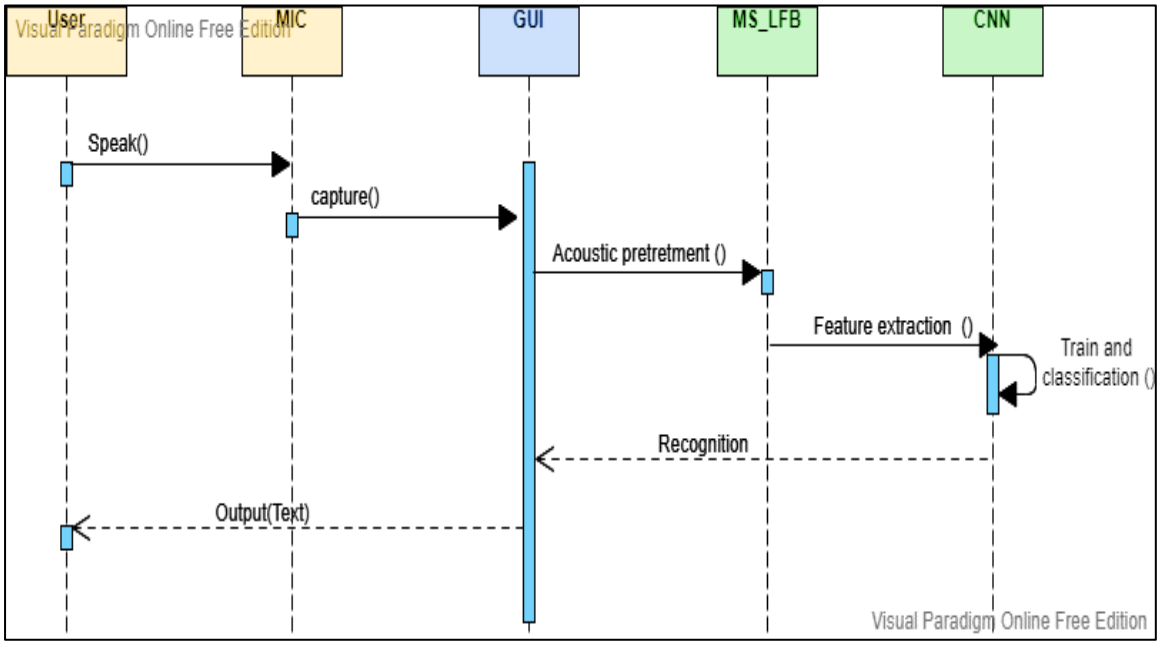


Figure 15: Sequence diagram illustrating speech recognition.

2.2.2 Audio files recognition

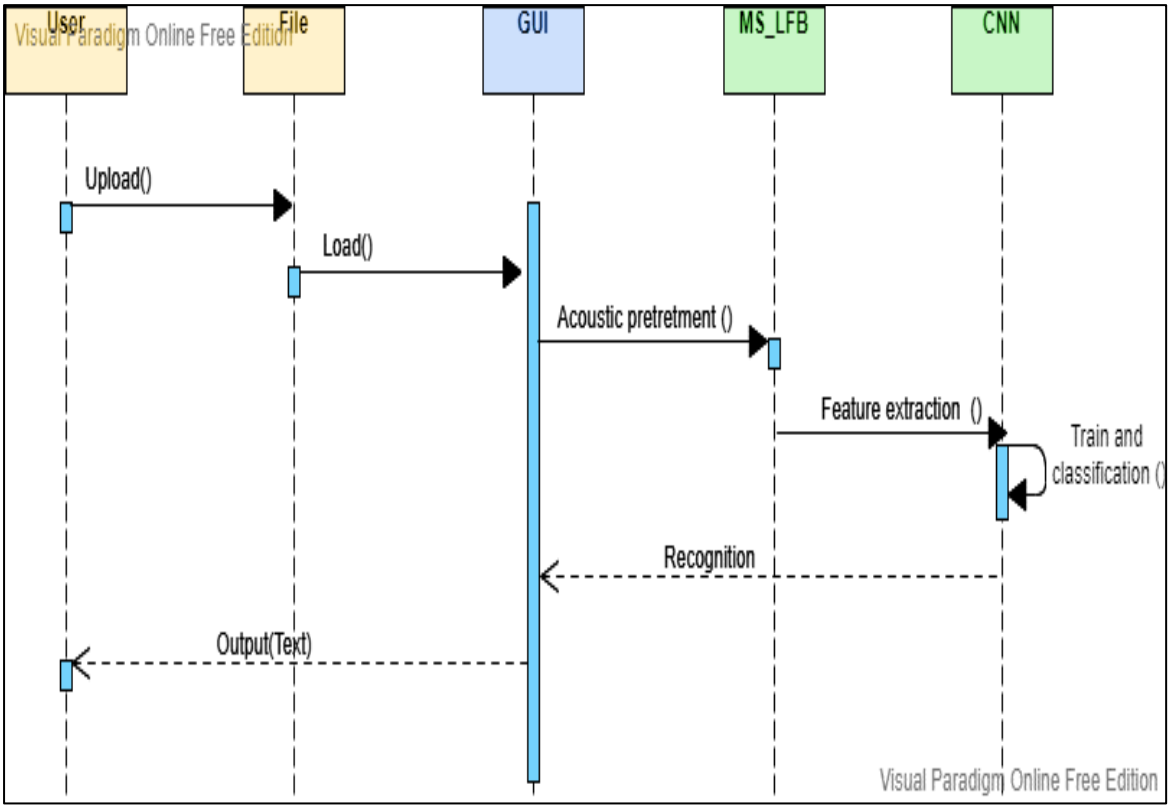


Figure 16: Sequence diagram illustrating audio files recognition.

3. Our system

The system starts to get the speech signal, then it does a preprocessing of the signal and then extracts the features from each word. The acoustic result is passed to (CNN), for training and testing and then the result of recognition, and our work aims to develop an application capable of recognizing spoken English words.

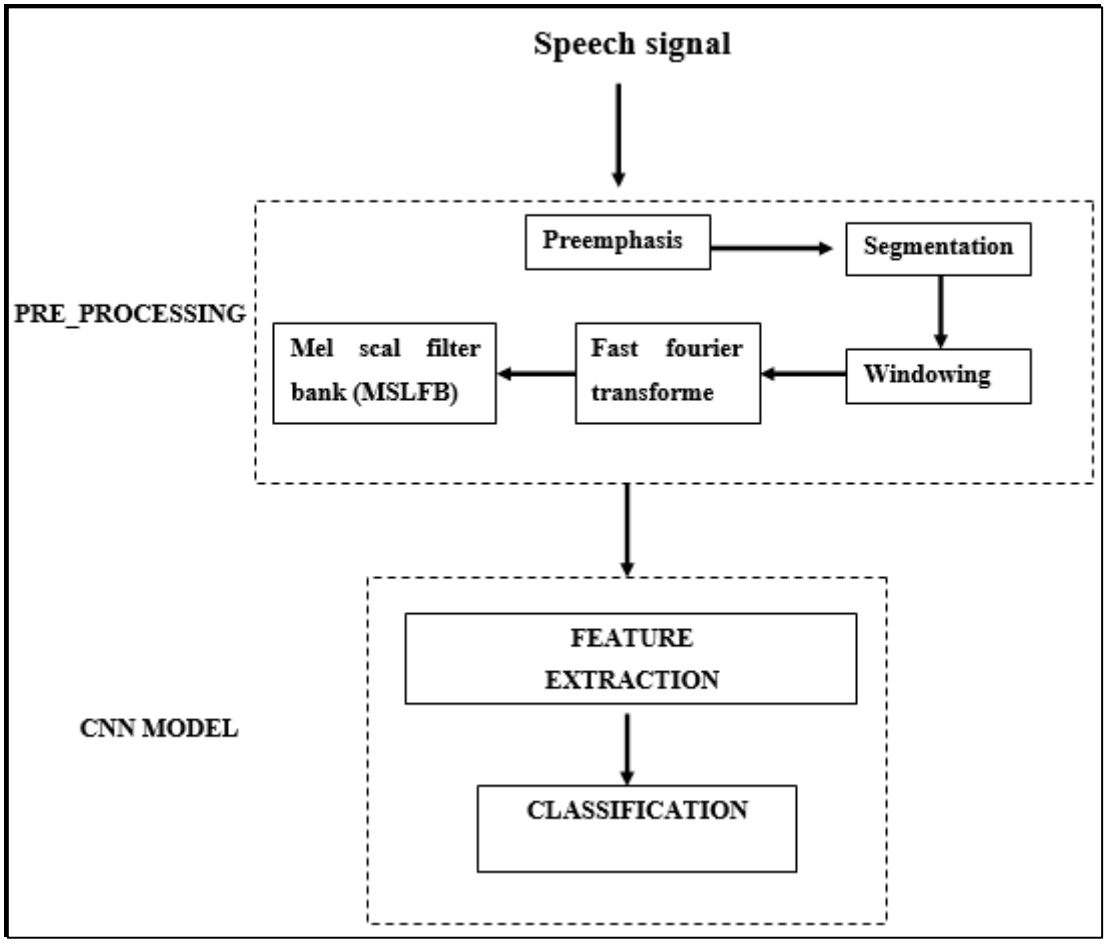


Figure 17: Our system architecture.

The main processes that our design includes:

3.1. Pre_processing

Methods based on DNNs like CNN in our case, are discriminative models and do not have similar constraints. MFCC and perceptual linear predictive (PLP) features are two most often used raw features. However, both these two types of features are derived from the Mel-scaled log filter-bank features (MS-LFB). Although these features are more invariant than the Mel-scaled log filter-bank features, some information useful for classification may be lost during the manual feature transformation process.

That's why we chose to use the MS-LFB features as the input to the CNN For this work. To calculate the MS-LFB of a signal we go through these steps:

- **Pre-emphasis**

In the field of ASR the vocal segments such as vowels have more energy at low frequencies than at high frequencies; this is caused by the nature of the glottal pulse. Pre -emphasis amplifies the

energy of high frequencies to make them more suitable for the recognition pattern. Pre-emphasis factor taking a value included in [0.9, 1.0].

- **Segmentation into frames**

The speech signal represents a long-term non-stationary random process, however it is considered stationary in analysis time windows of the order of 20 to 30 ms. For this purpose, after the pre-emphasis phase, to have stable acoustic characteristics, the speech signal must then be divided into a certain number of frames and examined on each of them where the property of short-term stationarity is verified. With, generally, a window overlap of 10 ms. The interest of this overlap is to obtain a temporal continuity of the characteristics. Hence, from each frame, a set of parameters is derived to form the feature vector.

- **Windowing**

Frame windowing is used to minimize signal discontinuities at the start and end of each frame, each window can be described by three parameters: its width called frame size, the offset between successive windows called frame shift or overlap and the shape of the window.

- **Discrete fourier transform**

After performing a windowing to attenuate the discontinuity of the signal at the start and end of the frame, the next phase consists in applying the Discrete Fourier Transform (DFT) which is calculated using the Fast Fourier Transform algorithm (FFT). This algorithm used for evaluating the frequency of the signal spectrum converts each windowed frame of N samples from the time domain to the frequency domain. DFT is used to extract spectral information from each window of the input signal.

- **Mel and log scaled filter bank**

The final step in filter bank calculation is to apply a ternary filter (usually 40 filters, nfilt = 40 at Mel level) to the power spectrum to extract the frequency band.

The purpose of the Mel scale is to imitate the human ear's perception of sound at lower frequencies by being more discriminating at lower frequencies and less discriminating at higher frequencies.

3.2 Convolution neural networks

One of the main aims of our project is to attempt to apply convolutional neural networks for speech recognition. Typical and well-studied CNN architectures are optimized for two-dimensional image processing tasks, an application where inputs have a very different shape than in audio processing: audio files are extremely long one-dimensional vectors. Our inputs, were single-second audio files at a sample rate of 16 kHz, we opted to modify the input data by translating the input audio data into spectrograms. A spectrogram is a method of using discrete Fourier transforms to translate a one-dimensional time sequence, into a two-dimensional image containing the same information, but organized in a way that makes locality meaningful in two dimensions rather than just one. Each column of the image represents the magnitude of the Fourier transform of the time sequence taken in a different contiguous window of time; the image as a whole represents a two-dimensional function from time and frequency to intensity rather than a one-dimensional function from time to absolute amplitude. Thus, in the resulting image, it is possible to directly see which frequency components change and how the frequency spectrum of the audio changes with time. The spectrogram in figure 19 shows the pitch contour of the word.

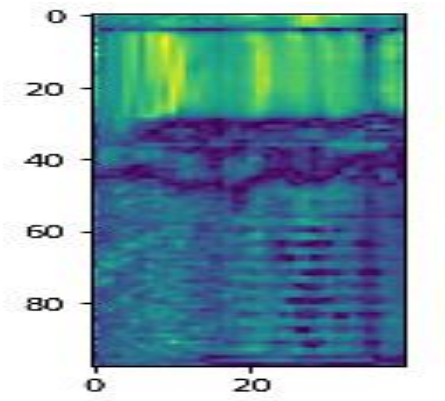


Figure 18: Conversion of audio signal from 1D to 2D (image format).

3.2.1 Training the model

Training is the most important process because we will create our model thanks to precise configurations: The dataset and CNN configuration.

- **Dataset**

In our project. The dataset (1.4 GB) has 65,000 one-second long utterances of 30 short words (eight, sheila, nine, yes, one, no, left, tree, bed, bird, go, wow, seven, marvin, dog, three, two, house, down, six, five, off, right, cat, zero, four, stop, up, on, happy) by thousands of different people, contributed

by public members through the AIY website. This is a set of one-second .wav audio files, each containing a single spoken English word. The recorded words are different in length of rely on the word itself. Also, some words differ in length from one person to another depend on the speaker himself and his pronounce ways. These conditions made the work more complicate especially for the training and classification process [50]. We divide the data set into three different types of files, respectively: training files, test files, and finally files to be used in the evaluation.

- **CNN configuration**

A CNN is a modified version of DNN where the input data undergoes several processes like convolution, max pooling, flattening and full connection in ordered fashion before being fed into the deep neural network.The network we created consists as follows: layer of Conv2D followed by A layer of Max Pooling process, we repeat this construction 3 times after that a layer of dropout followed by con2d and other layer of dropout, Dropout is a regularization technique, which aims to reduce the complexity of the model with the goal to prevent overfitting. After that we need to flatten the network output to a one dimension array in order to it to be passed to the dense layer, in these steps Each one of the layers is created by specifying its number of filters, its kernel (the size of each filter), its input shape (just for the first layer), its activation function and the kernel regularizer that is used to avoid the model to overfit the training set. For each convolutional layer the activation function that is utilized is “relu”, which has become the default activation function for many types of neural networks for faster learning a model that uses it is easier to train. Finally, for the model to make a decision, we use the sigmoid activation function. Figure 20 shows the structure of our CNN.

- **Convolution**

A convolution is a combined integration of two functions; it tells us how one function modifies the shape of the other function.

In the case of 2D array, the first function is the array itself, and the second function is the feature detector/kernel. The feature detector convolves over the whole array stride by stride and outputs the convolved feature map containing the most relevant/dominant information, thus reducing the dimension of the input image as well.

This convolution process creates feature maps where each individual feature map is a convolved result of different individual feature detector. Rectified linear unit (ReLu) activation function is then applied to increase the nonlinearity in the resulting feature maps in order to distinguish adjacent pixels of the maps more accurately.

➢ **Pooling**

The main purpose of pooling layer is to progressively reduce the spatial size of the input image, so that number of computations in the network are reduced. Pooling performs down sampling by reducing the size and sends only the important data to next layers in CNN.

➢ **Flattening**

The pooled feature map needs to go as an input in dense layer for further processing. For this, the 2D feature map is flattened, i.e., converted into 1D column by taking row by row values of the feature map. This results in dimensional column of feature map which acts as an input to the dense layer.

➢ **Full Connection**

The flattened layer, is fully connected with the subsequent layer 'dense ' as the combination of input features in the input layer and its attributes in the next layer results in much better accuracy.

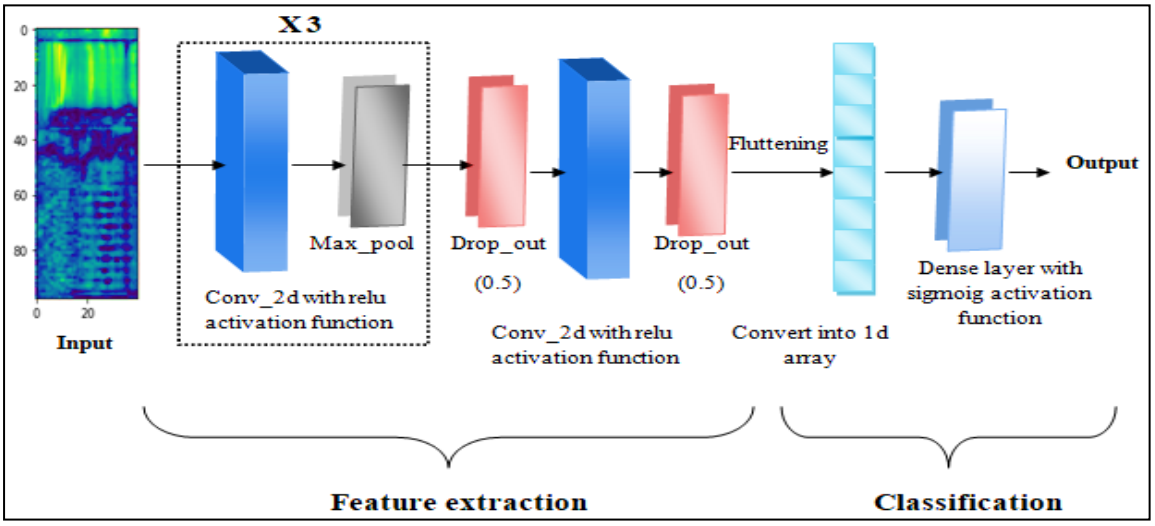


Figure 19: Configuration of the CNN model.

Our CNN model consists of the following components:

- ✓ Convolution Layer: In this layer, the filter size is initially fixed at 4 * 4, after that it takes the size of 3 * 3 and at the last is 2 * 2.
- ✓ The ReLU activation function: forces the network to return positive values. Any number less than 0 is converted to 0, this technique is applied for all layers except the last layer which uses the sigmoid function.
- ✓ The Pooling layer: we used a max pooling of size 2x2 to reduce the size of the parameters.

- ✓ Dropout layers: this is important in our training CNNs because they prevent overfitting on the training data. If they aren't present, the first batch of training samples influences the learning in a disproportionately high manner. This, in turn, would prevent the learning of features that appear only in later samples or batches.
- ✓ All convolutional layers use the ReLu function except the output layer uses the Sigmoid function.
- ✓ Flattening layer: After finishing the previous steps, we're supposed to have a pooled feature map by now. As the name of this step implies, we are literally going to flatten our pooled feature map into a column. The reason we do this is that we're going to need to insert this data into an artificial neural network (dense layer) later on.

4. Conclusion

We have presented in this chapter a general structure for our system then we went to the details of each step we specify the part concerned the classification approach based on networks convolutional neurons, encoding techniques in which we explained how the data is kept and selected to display the results, and parameters necessary for the successful results. For further details, we built a conceptual model of the model to illustrate the knowledge and information for better comprehension.

In the next chapter, it is realization and implementation. We will display the results of our work. We tested our algorithms, and it is essential to present the result. We will offer an illustrated representation of our outcome. After, we show the tools, programming languages, and libraries we used to achieve these results.

Chapter 3: implementation and Realization

Chapter 3: Implementation and Realization

1. Introduction

In the previous chapter, we presented the general architecture proposed as a solution to develop this system. In this chapter, we will present the implementation of our application based on a convolutional neural network using the Python language. We first start with a presentation of the chosen programming language. Then, we will show screenshots of our application running.

2. Tools and working environments

2.1 Hardware configuration

2.1.1 Physical environment

In order to realize our system, we used this hardware:

- A laptop PC: *hp* Intel(R) Celeron(R) CPU N3050 @ 1.60GHz 1.60 GHz.
- RAM size 4 GB.
- Operating system: 64-bit Windows 10.

Figure 20: Hp personal computer.

2.1.2 Remote hardware configuration (google colab)

- **Google Collaboratory:**

Google Collaboratory or Colab: a simple and free Google tool to introduce us to deep networks or collaborate with our colleagues on data science projects. Colab allows to improve our coding skills

in Python programming language, to develop applications in deep networks, using popular libraries such as Keras, TensorFlow without installation, as well as the use of a development environment (Jupyter Notebook) which requires no configuration. However, every 12 hours, the virtual machine provided by Google is reset, requiring an ongoing data backup mechanism. In addition, Colab (Jupyter Notebook) documents are saved directly to your Google Drive account [73].

- **Google Drive:**

The Google Drive service launched by Google in 2012 allows you to store 15 GB for free on a storage space accessible from any computer, tablet or smartphone (Android and iOS). It is possible to increase this storage capacity by opting for a paid version. But Google's offer does not stop there since the application communicates with all Google services, such as Google Docs for example. It is therefore possible to access and modify the files stored in the "cloud" wherever you are [76].

2.2 Software and libraries used in the implementation

2.2.1 Python

Python is a high-level, interpreted, object-oriented programming language. It is in high demand by a large community of developers and programmers. Python is a simple and easy to learn language. It is often compared (favorably of course) to Lisp, Tcl, Perl, Ruby, C #, Visual Basic, Visual Fox Pro, Scheme or Java ... etc. Python combines remarkable power with very clear syntax. It has modules, classes, exceptions, very high level dynamic data types and dynamic typing. There are interfaces to many system calls and libraries, as well as to various windowing systems. The new built-in modules are easy to write in C or C ++ (or other languages, depending on the implementation chosen). Python can also be used as an extension language for applications written in other languages requiring script interfaces. [75]

Figure 21: Python program [75].

2.2.2 Tensorflow

TensorFlow is an open-source software library released in 2015, by Google to make it easier for developers to design, build, and train deep learning models. It was originally planned as an internal library that Google developers would use to build models internally. At a high level, TensorFlow is a Python library that allows users to express arbitrary computation as a data flow graph. The nodes in this graph represent mathematical operations, while the edges represent data communicated from one node to another. TensorFlow data is represented as tensors, which are multi-dimensional arrays. While this framework for thinking about computation is valuable in many different fields, TensorFlow is primarily used for deep learning in practice and research [74].

2.2.3 Keras

Keras is a high-level neural network API, written in Python and capable of run on TensorFlow, CNTK or Theano. It was developed with the aim of allowing a rapid experimentation, be able to go from idea to result as quickly as possible, this is the key to doing research [15]:

- ✓ Allows easy and rapid prototyping (thanks to user-friendliness, modularity and extensibility).
- ✓ Supports convolutional networks and recurrent networks as well as combinations of the two.
- ✓ Works transparently on CPU and GPU.

3. The graphical interface of our project

It is an interface that allows us to recognize the specific word whether from recording speech or loading a wav file. here is the general interface :

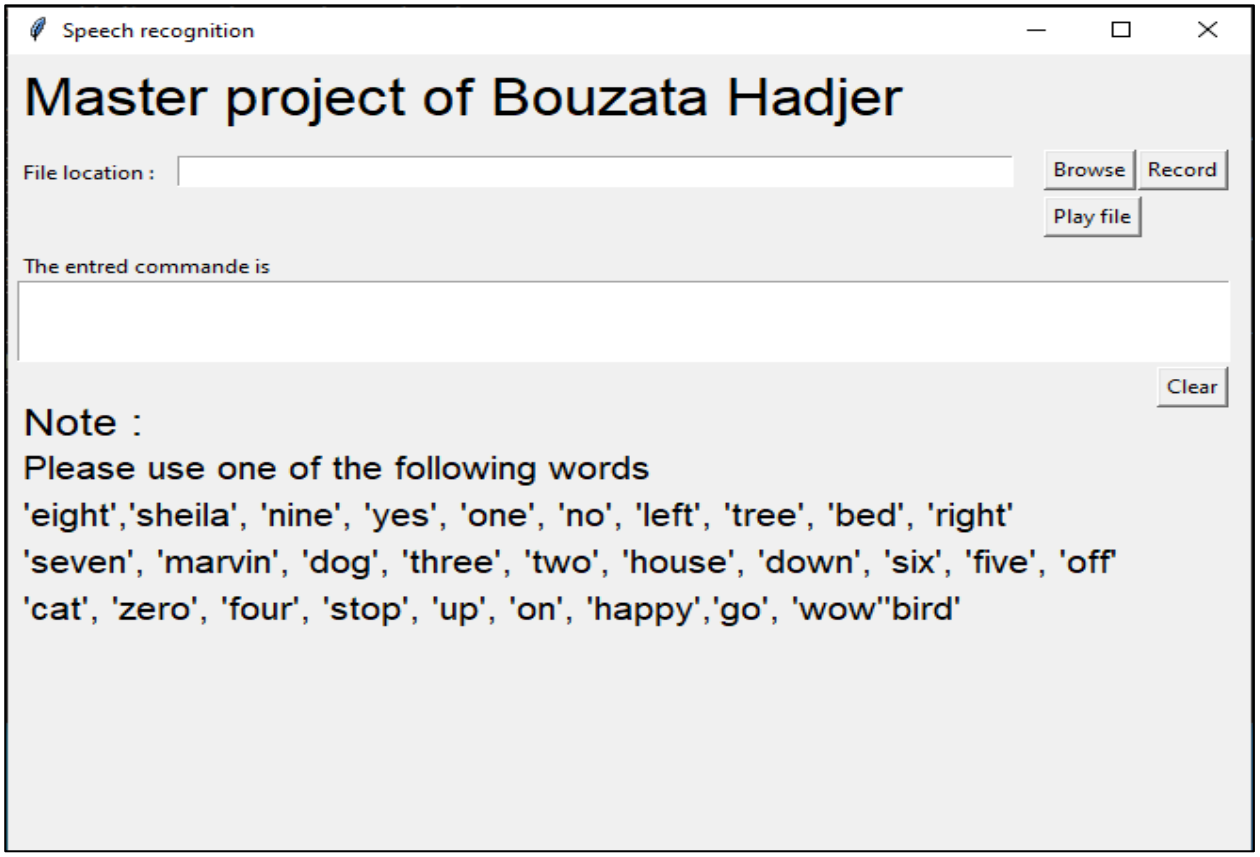


Figure 22: Interface of our project.

Our system allows to predict spoken or selected word using CNN, and for fulfilled this task we need:

- ✓ Contains a RECORD button to record your pronunciation of any of the 30 selected words, identify and convert them to text.
- ✓ File location: When we record a word, it is saved in the "Outpu.wavt" file.

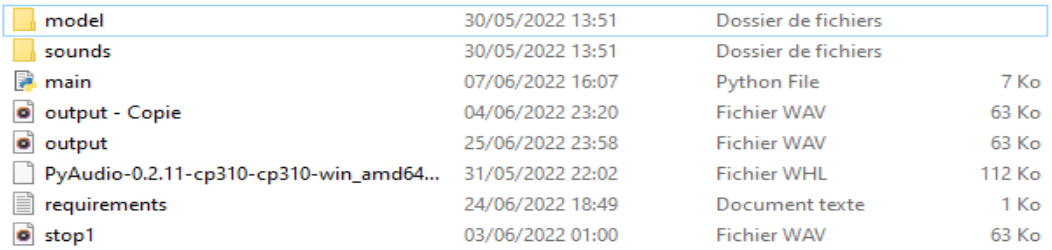


Figure 23: Output.wav file.

✓ BROWS button: if you click on this button a window is opened, this window allows us to choose wav files from our dataset and transcribe them to a text.

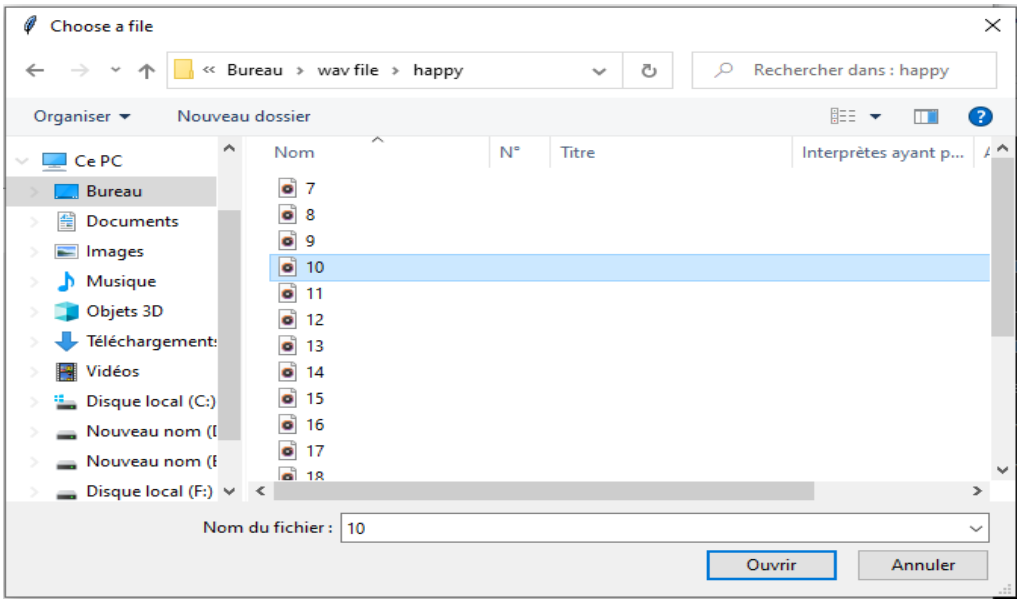


Figure 24: Window allows us to choose wav file.

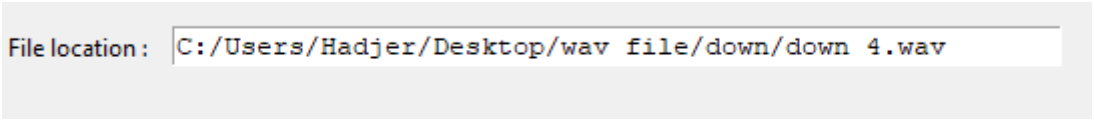


Figure 25: File location.

✓ And finally, the figure down below shows and example for the result of the predicted word.

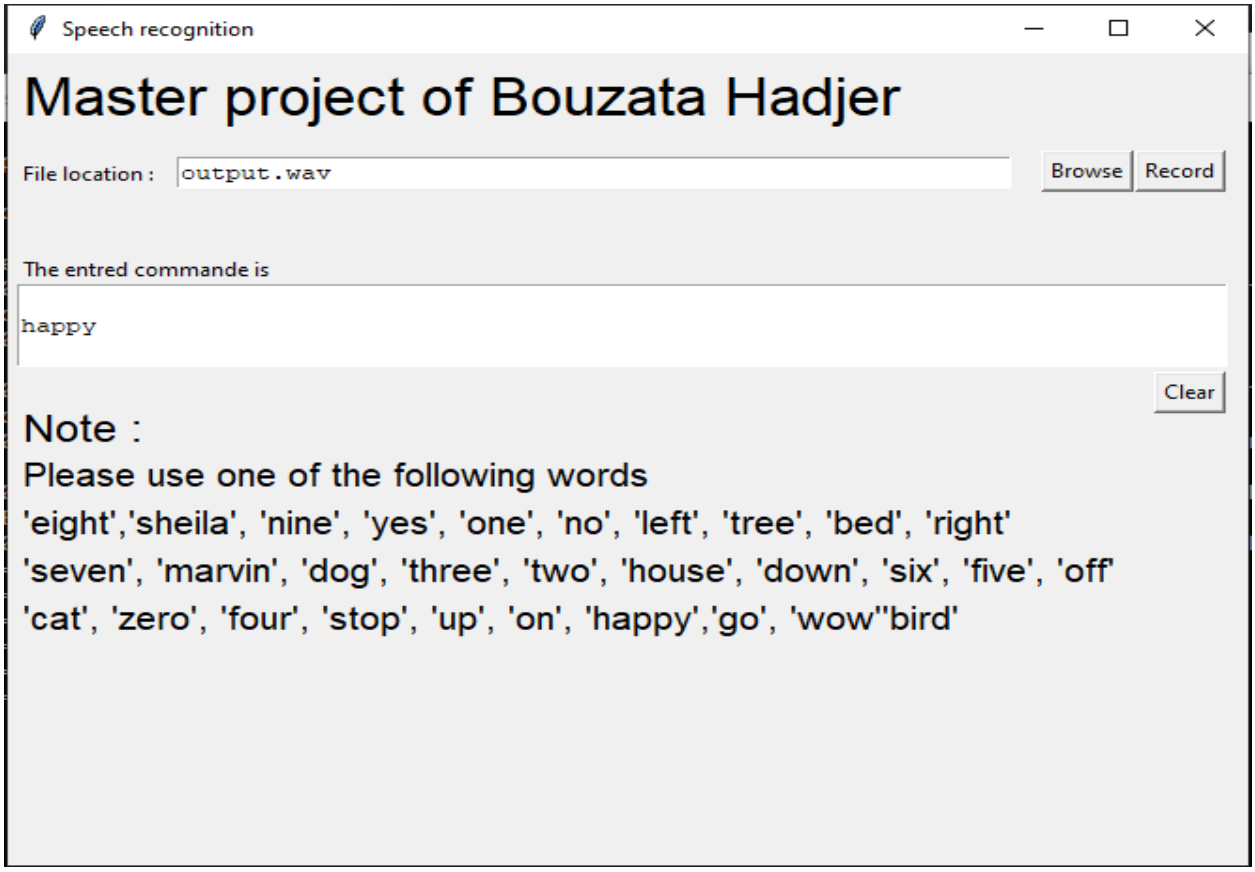


Figure 26: An example of execution.

4. Testing the model and recognition result

In order to show the obtained results for the model, Several models have been created by varying the hyperparameters in order to optimize the model in the most appropriate way, we illustrate in what follows the results in terms of precision and error compared to architecture of the models and the number of epochs (An epoch describes the number of times that the algorithm passes over the dataset).

4.1 Testing the model 1 and model 2

- **Model 1:**

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 conv2d_3 (Conv2D)            (None, 97, 37, 32)        544

 max_pooling2d_2 (MaxPooling  (None, 48, 18, 32)        0
 2D)

 conv2d_4 (Conv2D)            (None, 45, 15, 64)        32832

 max_pooling2d_3 (MaxPooling  (None, 22, 7, 64)         0
 2D)

 dropout_2 (Dropout)          (None, 22, 7, 64)         0

 conv2d_5 (Conv2D)            (None, 22, 7, 128)        32896

 dropout_3 (Dropout)          (None, 22, 7, 128)        0

 flatten_1 (Flatten)          (None, 19712)             0

 dense_1 (Dense)              (None, 30)                591390
```

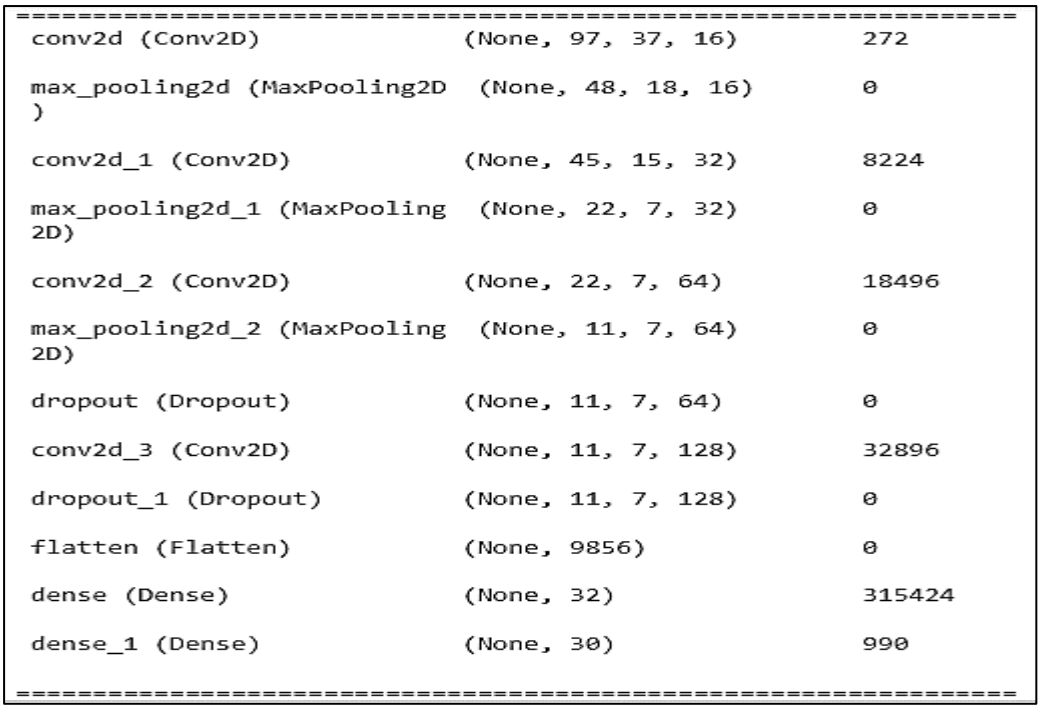
Figure 27: The architecture of model 1.

- **Result of Model 1:**

```
1012/1012 [==============================] - 125s 124ms/step - loss: 0.1838 - accuracy: 0.9394 - val_loss: 0.4297 - val_accuracy: 0.8
Epoch 18/30
1012/1012 [==============================] - 125s 123ms/step - loss: 0.1850 - accuracy: 0.9388 - val_loss: 0.4193 - val_accuracy: 0.8913
Epoch 19/30
1012/1012 [==============================] - 125s 124ms/step - loss: 0.1948 - accuracy: 0.9373 - val_loss: 0.3874 - val_accuracy: 0.8953
Epoch 20/30
1012/1012 [==============================] - 128s 127ms/step - loss: 0.1874 - accuracy: 0.9396 - val_loss: 0.3885 - val_accuracy: 0.8955
Epoch 21/30
1012/1012 [==============================] - 126s 125ms/step - loss: 0.1944 - accuracy: 0.9383 - val_loss: 0.3859 - val_accuracy: 0.8980
Epoch 22/30
1012/1012 [==============================] - 128s 126ms/step - loss: 0.1916 - accuracy: 0.9379 - val_loss: 0.4196 - val_accuracy: 0.8922
Epoch 23/30
1012/1012 [==============================] - 127s 125ms/step - loss: 0.1873 - accuracy: 0.9386 - val_loss: 0.3808 - val_accuracy: 0.9020
Epoch 24/30
1012/1012 [==============================] - 128s 127ms/step - loss: 0.1866 - accuracy: 0.9391 - val_loss: 0.4160 - val_accuracy: 0.8975
Epoch 25/30
1012/1012 [==============================] - 128s 127ms/step - loss: 0.1953 - accuracy: 0.9379 - val_loss: 0.4280 - val_accuracy: 0.8847
Epoch 26/30
1012/1012 [==============================] - 129s 128ms/step - loss: 0.1881 - accuracy: 0.9401 - val_loss: 0.4070 - val_accuracy: 0.8985
Epoch 27/30
1012/1012 [==============================] - 128s 126ms/step - loss: 0.1864 - accuracy: 0.9403 - val_loss: 0.4119 - val_accuracy: 0.8942
Epoch 28/30
1012/1012 [==============================] - 129s 127ms/step - loss: 0.1836 - accuracy: 0.9428 - val_loss: 0.3815 - val_accuracy: 0.8968
Epoch 29/30
1012/1012 [==============================] - 129s 128ms/step - loss: 0.1893 - accuracy: 0.9396 - val_loss: 0.3838 - val_accuracy: 0.8990
Epoch 30/30
 423/1012 [===========>..................] - ETA: 1:10 - loss: 0.1795 - accuracy: 0.9422
```

Figure 28: The result of model 1.

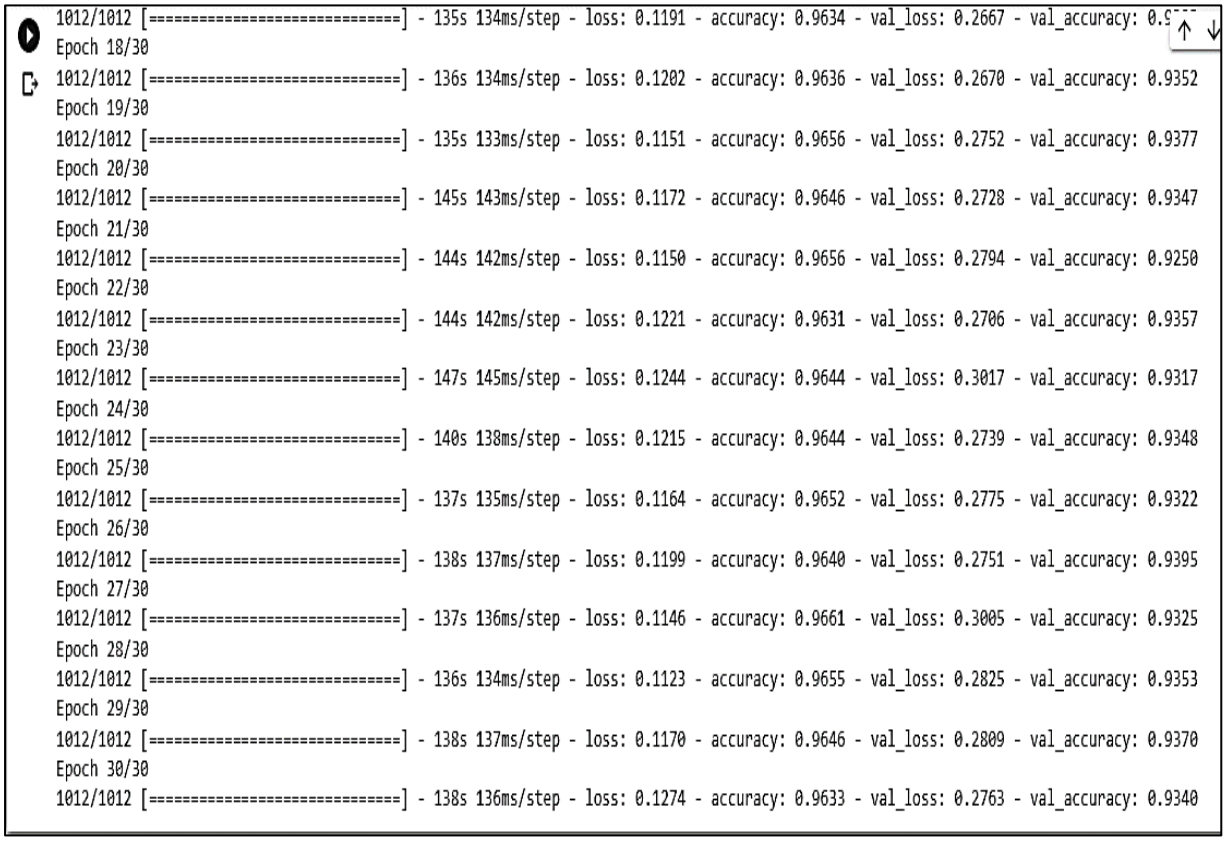
- **Model 2:** It is the model that we mentioned in the CNN configuration part, which we adopted in our application.



conv2d (Conv2D)	(None, 97, 37, 16)	272
max_pooling2d (MaxPooling2D)	(None, 48, 18, 16)	0
conv2d_1 (Conv2D)	(None, 45, 15, 32)	8224
max_pooling2d_1 (MaxPooling2D)	(None, 22, 7, 32)	0
conv2d_2 (Conv2D)	(None, 22, 7, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 11, 7, 64)	0
dropout (Dropout)	(None, 11, 7, 64)	0
conv2d_3 (Conv2D)	(None, 11, 7, 128)	32896
dropout_1 (Dropout)	(None, 11, 7, 128)	0
flatten (Flatten)	(None, 9856)	0
dense (Dense)	(None, 32)	315424
dense_1 (Dense)	(None, 30)	990

Figure 29: The architecture of model 2.

❑ Result of Model 2:



```
1012/1012 [==============================] - 135s 134ms/step - loss: 0.1191 - accuracy: 0.9634 - val_loss: 0.2667 - val_accuracy: 0.9
Epoch 18/30
1012/1012 [==============================] - 136s 134ms/step - loss: 0.1202 - accuracy: 0.9636 - val_loss: 0.2670 - val_accuracy: 0.9352
Epoch 19/30
1012/1012 [==============================] - 135s 133ms/step - loss: 0.1151 - accuracy: 0.9656 - val_loss: 0.2752 - val_accuracy: 0.9377
Epoch 20/30
1012/1012 [==============================] - 145s 143ms/step - loss: 0.1172 - accuracy: 0.9646 - val_loss: 0.2728 - val_accuracy: 0.9347
Epoch 21/30
1012/1012 [==============================] - 144s 142ms/step - loss: 0.1150 - accuracy: 0.9656 - val_loss: 0.2794 - val_accuracy: 0.9250
Epoch 22/30
1012/1012 [==============================] - 144s 142ms/step - loss: 0.1221 - accuracy: 0.9631 - val_loss: 0.2706 - val_accuracy: 0.9357
Epoch 23/30
1012/1012 [==============================] - 147s 145ms/step - loss: 0.1244 - accuracy: 0.9644 - val_loss: 0.3017 - val_accuracy: 0.9317
Epoch 24/30
1012/1012 [==============================] - 140s 138ms/step - loss: 0.1215 - accuracy: 0.9644 - val_loss: 0.2739 - val_accuracy: 0.9348
Epoch 25/30
1012/1012 [==============================] - 137s 135ms/step - loss: 0.1164 - accuracy: 0.9652 - val_loss: 0.2775 - val_accuracy: 0.9322
Epoch 26/30
1012/1012 [==============================] - 138s 137ms/step - loss: 0.1199 - accuracy: 0.9640 - val_loss: 0.2751 - val_accuracy: 0.9395
Epoch 27/30
1012/1012 [==============================] - 137s 136ms/step - loss: 0.1146 - accuracy: 0.9661 - val_loss: 0.3005 - val_accuracy: 0.9325
Epoch 28/30
1012/1012 [==============================] - 136s 134ms/step - loss: 0.1123 - accuracy: 0.9655 - val_loss: 0.2825 - val_accuracy: 0.9353
Epoch 29/30
1012/1012 [==============================] - 138s 137ms/step - loss: 0.1170 - accuracy: 0.9646 - val_loss: 0.2809 - val_accuracy: 0.9370
Epoch 30/30
1012/1012 [==============================] - 138s 136ms/step - loss: 0.1274 - accuracy: 0.9633 - val_loss: 0.2763 - val_accuracy: 0.9340
```

Figure 30: The result of model 2.

4.2 Comparison results between models

Comparison between the structure of Model 1 and Model 2 and the results obtained in each one

CNN model	Conv	Pooloing	Dropout	Flatten	Dense	Number of epoche	Accuracy	Loss
Model 1	2	2	1	1	1	30	0.9396	0.1893
Model 2	2	2	2	1	1	30	0.9633	0.1274

Table 4. Comparison between the structure of Model 1 and Model 2

The table shows the architecture as well as the number of layers used in in both the first and second experimental model. We chose the second model because it yielded better results in terms of accuracy and loss.

4.2.1 Testing Model 2 with Number of epochs

- ❑ Testing with number of epochs = 50

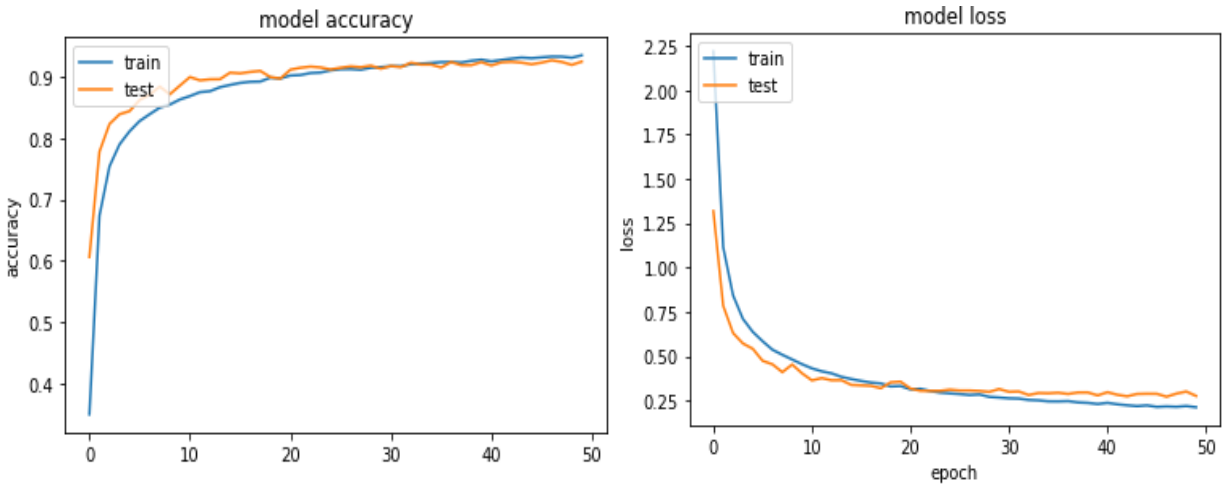


Figure 31: Accuracy and error of the CNN model with a number of epoch = 50.

Epochs number	Training		Validation	
	loss	accuracy	val_loss	val_accuracy
50	**0.2117**	**0.9346**	**0.2751**	**0.9243**

Table 5: Result of training and testing with epoch = 50

❑ Testing with Number of epochs = 90

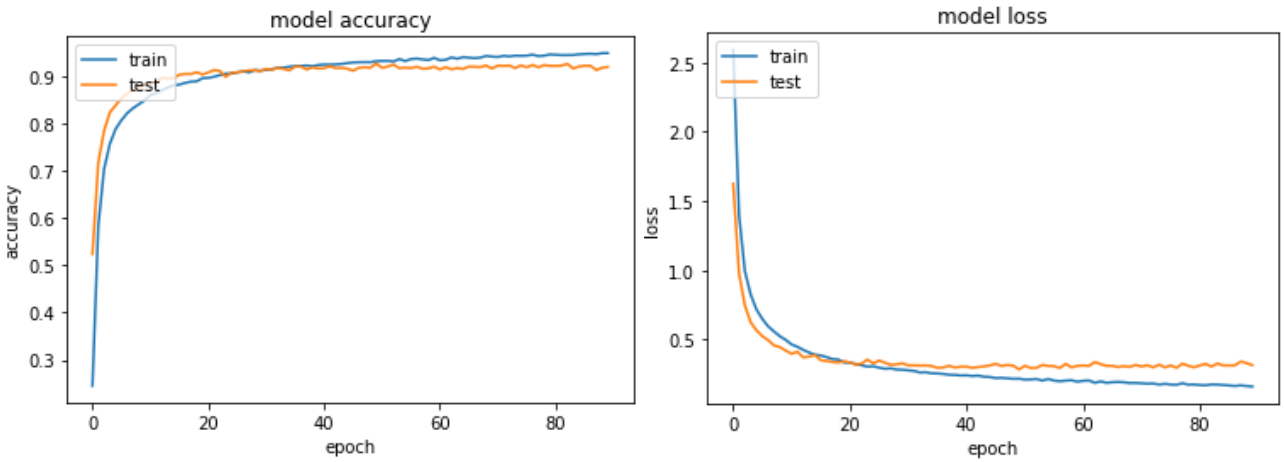


Figure 32: Accuracy and loss of the CNN model with a number of epochs=90.

Epochs number	Training		Validation	
	loss	accuracy	val_loss	val_accuracy
90	**0.1599**	**0.9485**	**0.3142**	**0.9193**

Table 6: Result of training and testing with epochs = 90

❑ Testing with number of epochs = 110

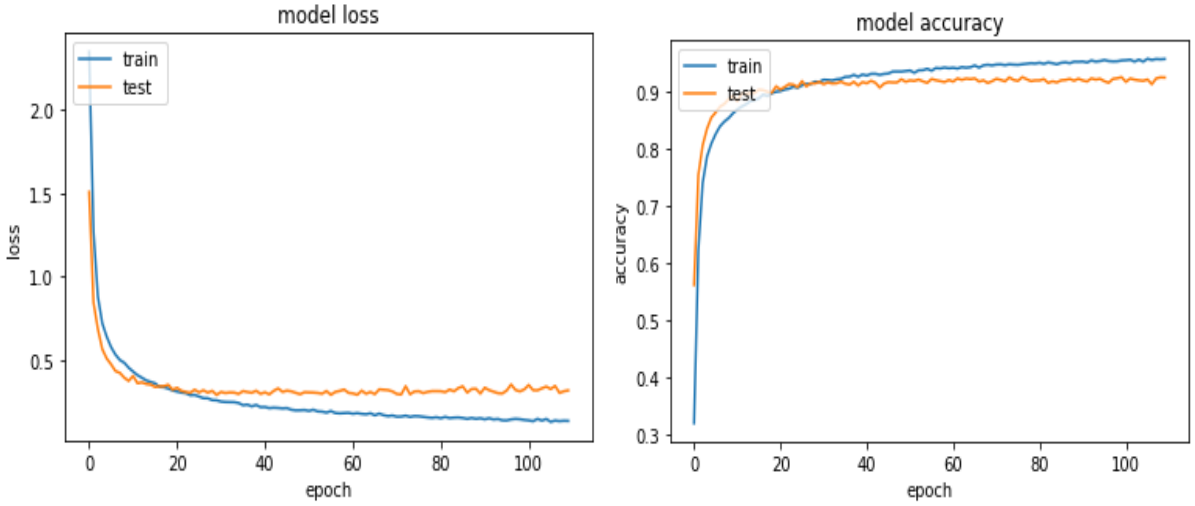


Figure 33: Accuracy and error of the CNN model with a number of epochs = 110

Epochs number	Training		Validation	
	loss	accuracy	val_loss	val_accuracy
110	**0.1343**	**0.9587**	**0.3172**	**0.9260**

Table 7: Result of training and testing with epoch = 110

From the results obtained and explained above, we note that the accuracy of learning and val_accuracy increases with the number of epochs. We observe the same for the loss and val_loss decreasing with the number of epochs.

- **Comparison of the obtained results:**

CNN model	Epochs number	Training		Validation	
		loss	accuracy	val_loss	val_accuracy
Model 2	50	0.2117	0.9346	0.2751	0.9243
	90	0.1599	0.9485	0.3142	0.9193
	110	0.1343	0.9587	0.3172	0.9260

Table 8: Comparison results obtained by training our model with different numbers of epochs

The table shows the architecture as well as the number of layers used in the model. The results obtained are expressed in terms of precision and error. We notice, that each time we increase the number of epochs, the accuracy rate increases and the error rate decreases, we notice also that this is not proportional because arrived at a certain threshold, it begins to stabilize and the increase is not as large as at the beginning. In general, convolutional neural network is important and deep, gives good results and the performance of our network degrade if we choose the misfitted parameters.

4.3 The results we obtained using MFCC in preprocessing

- ❑ Result when number of epochs = 50

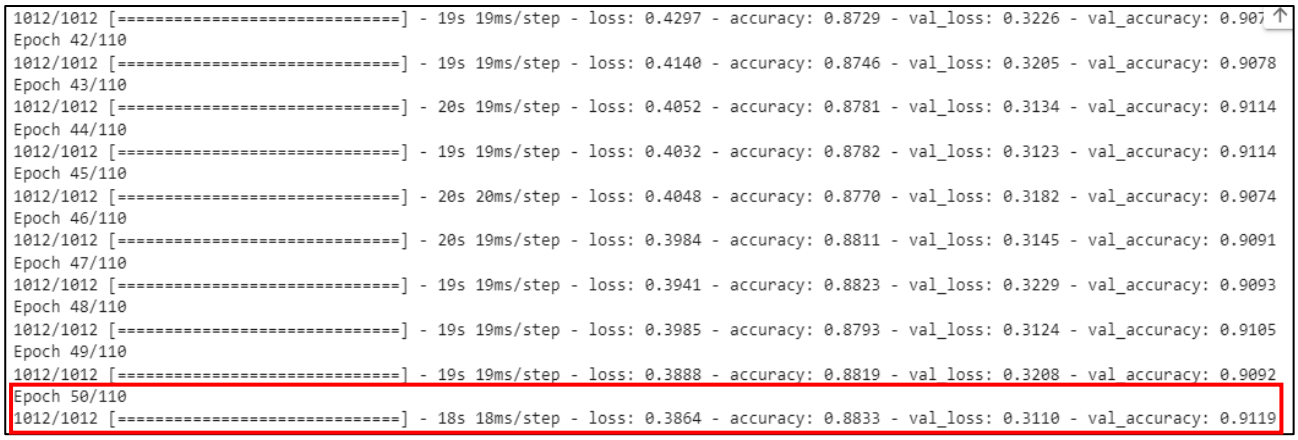


Figure 34: The result obtained using MFCC at number of epochs = 50

- ❑ Result when number of epochs = 90

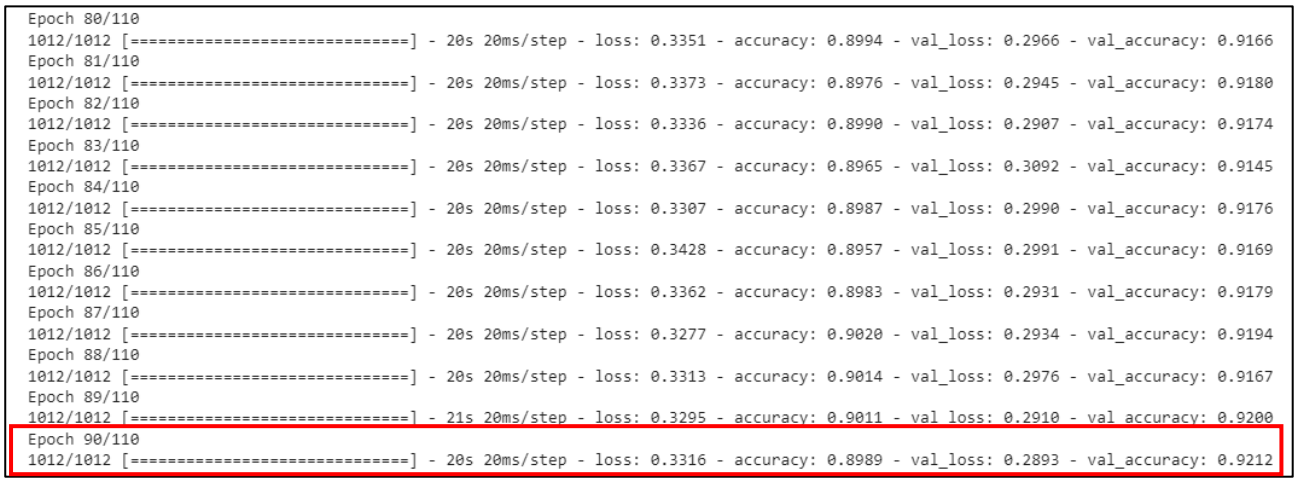


Figure 35: The result obtained using MFCC at number of epochs = 90

- ❑ Result when number of epochs = 110

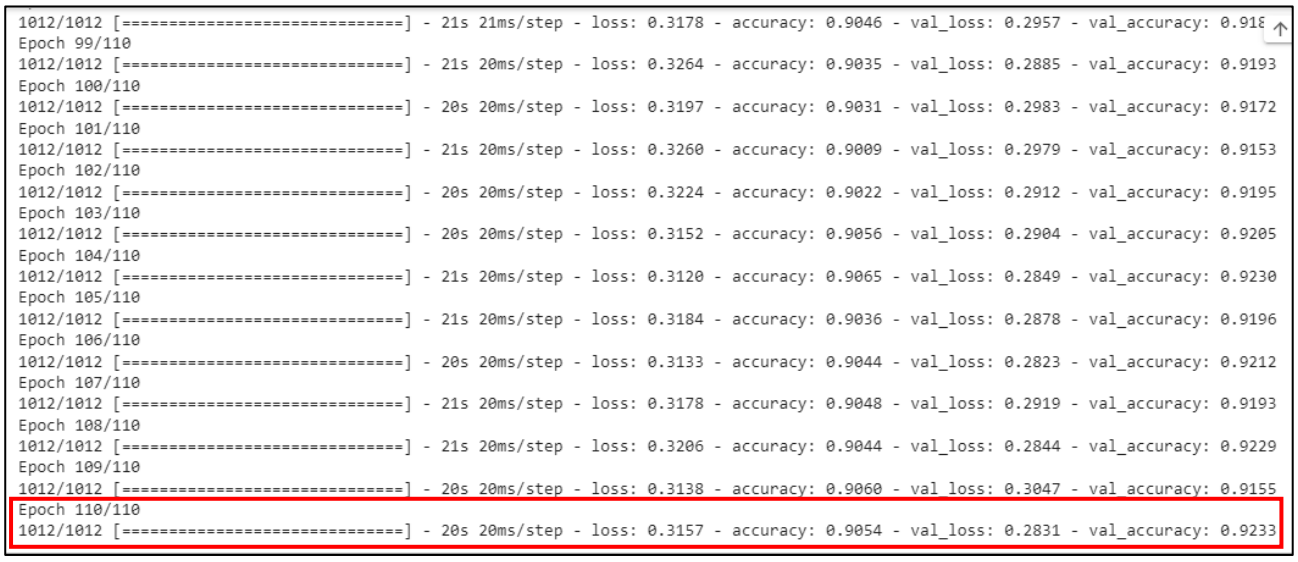


Figure 36: The result obtained using MFCC at number of epochs = 110

- **Comparison of results obtained using MSLFB and MFCC:**

Preprocessing method	Epochs number	Training		Validation	
		loss	accuracy	val_loss	val_accuracy
MSLFB	50	0.2117	0.9346	0.2751	0.9243
	90	0.1599	0.9485	0.3142	0.9193
	110	0.1343	0.9587	0.3172	0.9260
MFCC	50	0.3864	0.8833	0.3110	0.9119
	90	0.3316	0.8989	0.2893	0.9212
	110	0.3157	0.9054	0.2831	0.9233

Table 9: Comparison of results obtained using MSLFB and MFCC.

When comparing the results obtained using MSLFB and MFCC, we noticed that MSLFB gives better results in terms of accuracy and error compared to MFCC with longer execution time, while MSLFB gives acceptable results in terms of accuracy and error and takes less time to execute.

4.4 Comparing of our obtained results with RNN model

An ASR system using MFCC method as audio features, and a CNN for the speech modeling, was implemented as described in the previous sections. The whole programming was implemented in python.

In order to evaluate the performance of the proposed system, it has been tested with different types of settings. At first, we have fixed the number of epochs for training, and we have tried several CNN models, the second model gave the best results. After that, we tried the second model of CNN with a different number of epochs, to tell us that the accuracy rate increases with time until it starts to be stable when a certain limit is reached. The same thing for the error rate decreases with the increase in the number of epochs.

In the following, we present our experimental results on our ASR system with CNN compared to those obtained using RNN:

We discuss the work of Aditya Amerkar [54] on “Speech Recognition Using Recurrent Neural Networks”. I found this paper interesting because it deals with the same task I did and described in the previous part of this thesis, that is speech recognition, but using recurrent neural networks instead of a CNN. Thanks to this work, we can understand the differences of the two types of

networks and their performance. The main difference between the two models is that a convolutional neural network is not capable of processing sequential information. This is because the network processes every input taking into account only the present state of each neuron, which is not modified by previous inputs. A recurrent neural network instead has this capability. Each neuron has a sort of “memory”, that allows them to analyze and process each input taking into account some of the information contained in the previous processed input. This mechanism required a very large dataset and enormous computational power, three characteristics that make it difficult to achieve and pursue academically. While a convolutional neural network does a remarkable job for processes such as speech recognition for single and few words, for more complex speech recognition projects, which want to process more words or even complex sentences, the use of recurrent neural networks is essential.

5. Conclusion

In this chapter, we presented the system realized in our study using python language, and it is based on a convolutional neural network representing a type of deep learning, for this, we compared the structure of two different models and noted the difference in terms of accuracy and loss results. We also used a model with a different number of periods and then we interpreted the different results obtained in terms of accuracy and loss. Comparison of the obtained results showed that the number of epochs and the structure of the model are important factors for obtaining better results. We have seen the criteria on which we choose the programming language, the general interface and the various functionalities of the system, as well as the obtained results, the results are satisfactory and allow to deduce the advantages of the techniques used in this system. In the following we will present a general conclusion and perspectives.

Conclusion and perspectives

Over the past few decades, research in automatic speech recognition has been actively conducted around the world, spurred by advances in signal processing, algorithms, architectures, and hardware. ASR systems have been developed for a wide range of applications, including communications, disability assistance, avionics, automotive, home applications, voice control in robotics, office automation, voice response in remote interrogation systems, etc.

This thesis explored the problem of designing and implementing an automatic speech recognition system. As part of this work, we first presented an in-depth introduction to the field of automatic speech recognition and its main components, as well as a historical perspective on the main inventions that have enabled advances in speech recognition. Next, we discussed the most important methods used for speech recognition and audio signal processing techniques.

As conclusions, it can be said that the world of neural networks in the field of voice modeling and analysis is constantly expanding and producing new and interesting results every year. For a ASR or Speech Recognition model built using a neural network, the ideal would be the use of a recurrent neural network, a very large dataset and enormous computational power, three characteristics that make it a difficult goal to pursue in academic terms.

A simple Speech Recognition program, on the other hand, which deals with the understanding of a few simple words, can be implemented instead through a simpler network, obtaining very good result swhich is the case for our project, we applied convolutional neural networks to a data set of 65,000 words in one second from 30 short words by thousands of different people, after the speech signal preprocessing, sounds file has been entered directly to the designed network structure that contained multilevel learning procedure to achieve the model multi-classification task. From the experimentation and research carried out during the drafting of this thesis, it is however clear the enormous potential of these network structures, which over time, with new algorithms and new increasingly powerful processors, is becoming one of the most varied and practical practices in the world of information technology. The revolution brought about by neural networks radically changes our conception of programming, from something that, based on pre-established rules, is able to produce results, to something that can compose these rules autonomously by observing the results.

This type of application greatly expands the horizons of what can be achieved through a computer, making us touch a future in which artificial intelligence could be the master. As in the context of speech recognition, the use of neural networks can be applied to almost all practices that somehow

have an IT process behind them, greatly enhancing all programming areas that would require extreme complexity in their design.

The study of these fascinating models has further increased my interest in this subject, which I will certainly carry on as my studies progress.

References

[1] Quarteroni, A. (2022). “Artificial Intelligence, Learning Computers, Artificial Neural Networks”. In: Algorithms for a New World. Springer, Cham.

[2] Anjali, J. (2022). J. Phys.: Conf. Ser. 2273 012003.

[3] Ravindra, P. (2019). “Automatic Speech Recognition Systems for Regional Languages in India”. IJRTE.

[4] Saba, D. (2021). “Towards Artificial Intelligence: Concepts, Applications, and Innovations”. In: Hassanien, AE., Taha, M.H.N., Khalifa, N.E.M. (eds) Enabling AI Applications in Data Science. Studies in Computational Intelligence, vol 911. Springer, Cham.

[5] Jean, K. (2022). “Automatic Speech Recognition And Limited Vocabulary: A Survey”. Researchgate.

[6] Benkerzaz, S. (2019). “A Study On Automatic Speech Recognition”. Journal Of Information Technology Review Volume 10 Number 3.

[7] Ghosh, A. (2021). “Normalizing Flow Based Hidden Markov Models For Classification Of Speech Phones With Explainability”. Arxiv:2107.00730v1 [Cs.Lg] .

[8] Ismail, A. (2020). “Development Of Smart Healthcare System Based On Speech Recognition Using Support Vector Machine And Dynamic Time Warping”.

[9] Permanasari, Y. (2019). “Speech Recognition Using Dynamic Time Warping (Dtw)”. Icoaims 2019.

[10] Muktafin, E. Pramono, K. (2021). “Sentiments Analysis Of Customer Satisfaction In Public Services Using K-Nearest Neighbors Algorithm And Natural Language Processing Approach”. Telkomnika Telecommunication, Computing, Electronics And Control.

[11] Karmakar, P. Teng, S. (2021). “Thank You For Attention: A Survey On Attention-Based Artificial Neural Networks For Automatic Speech Recognition”. Arxiv:2102.07259v1 [Cs.Sd].

[12] Nassif, A. Shahin, I. Attili, I. (2019). “Speech Recognition Using Deep Neural Networks: A Systematic Review”. 2169-3536 2019 Ieee.

[13] Bendahmane, A. 2014, «Cours de Traitement Automatique de la Parole », Polycopié de l’USTO, Oran, Algérie,

[14] Farouk, M. (2018). “Application of Wavelets in Speech Processing || Speech Production and Perception”. [SpringerBriefs in Electrical and Computer Engineering]. 10.1007/978-3-319-69002-5(Chapter 2), 5–10. doi:10.1007/978-3-319-69002-5_2.

[15] Nagisetty, A. (2019). “Framework for Detection of Malicious Activities in IoT Networks

using Keras Deep Learning Library".

[16] Zerari, N. (2021). « Intégration d'un module de reconnaissance de la parole au niveau d'un système audiovisuel - application téléviseur ».

[17] Prakash, S. Ravindra, N. 2018. "A survey : Speech recognition approaches and techniques". In 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON).

[18] Singh, A. (2018). "A Survey: Speech Recognition Approaches And Techniques". 2018 5th Ieee Uttar Pradesh Section International Conference On Electrical, Electronics And Computer Engineering (Upcon).

[19] Sundarapandiyan, S. (2017). "A Survey On Automatic Speech Recognition System". International Journal Of Current Advanced Research.

[20] Algihab, W. (2019). "Arabic Speech Recognition With Deep Learning: A Review". Springer Nature Switzerland Ag 2019.

[21] Passricha V, A. (2019). "Convolutional support vector machines for speech". International Journal of Speech Technology , 22(3), 601-609.

[22] Benmachiche, A. Makhlouf, A. (n.d.). Optimization learning of hidden Markov model using the bacterial foraging optimization algorithm for speech recognition.

[23] Chenni, A. 2020. « Analyse et synthèse d'un signal de parole par la Matrice de Pencil en vue d'une discrimination de locuteurs ».

[24] Attari, M. 2007. « Reconnaissance Automatique de la Parole par les Modeles Connexionnistes ». Diplôme de Doctorat.

[25] Anggun, W. 2018 "Improvement of MFCC Feature Extraction Accuracy Using PCA in Indonesian Speech Recognition". International Conference on Information and Communications Technology (ICOIACT).

[26] Sanjaya, M. 2018 "Speech Recognition using Linear Predictive Coding (LPC) and Adaptive Neuro-Fuzzy". (ANFIS) to Control 5 DoF Arm Robot International Conference on Computation in Science and Engineering.

[27] Gaurav, A. Latika, S. (2018) "Classification of intellectual disability using LPC, LPCC, and WLPCC parameterization techniques". International Journal of Computers and Applications, DOI: 10.1080/1206212X.2018.1475330.

[28] Sabur, A. 2018 "Some Commonly Used Speech Feature Extraction Algorithms". DOI:

10.5772/intechopen.80419.

[29] Jiankun, H. 2021 “A High-Performance Mel-scale Frequency Cepstral Coefficients Digital Circuit Used on Keyword-Spotting”. ChipIEEE.

[30] Divya G. 2018 “The State of the Art of Feature Extraction Techniques in Speech Recognition”. Springer Nature Singapore.

[31] Maria L. 2021. “Automatic Speech Recognition Features Extraction Techniques: A Multi-criteria Comparison”. IJACSA. Vol. 12, No. 8.

[32] Mishaim, M. (2021). “Automatic speech recognition: a survey”.

[33] Prabhakar, O. Sahu, K.N. “A survey on: voice command recognition technique”. Int. J. Adv. Res. Comput. Sci. Softw. Eng. 3(5) (2013).

[34] Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N. “Deep neural networks for acoustic modeling in speech recognition”. IEEE Sig. Process. Mag. 29(6), 82–97 (2012).

[35] Shweta, K. 2018. “A Comparative Study of the Techniques for Feature Extraction and Classification in Stuttering”. IEEE Xplore Compliant - Part Number: CFP18BAC-ART; ISBN:978-1-5386-1974-2.

[36] Sundarapandiyan, S. 2017 “A Survey on automatic speech recognition system”. International Journal of Current Advanced Research Volume 6 No. 6287-6297

[37] Daniel, P. 2007. “MULTILINGUAL ACOUSTIC MODELING FOR SPEECH RECOGNITION BASED ON SUBSPACE GAUSSIAN MIXTURE MODELS”.

[38] Zhicheng, L. 2018. “Using hidden markov model to predict human actions with hybrid improved gravitational swarm intelligence search algorithm”. ICIC Express Letters. Volume 12, Number 6.

[39] AMINA, M. (2016). « Reconnaissance automatique de la parole en milieu réel ».

[40] Saikia, P. (2017). “HMM-DNN SPEECH RECOGNITION TECHNIQUES: A REVIEW”. IJDR International Journal of Development Research.

[41] Gao, J., Wan, G. "Review of the application of intelligent speech technology in education" Journal of China Computer-Assisted Language Learning, vol. 2, no. 1, 2022, pp. 165-178.

[42] Yurika, P. 2019 J. Phys.: Conf. Ser. 1366 012091.

[43] Deepam, G. 2019. “Support vector machines based non-contact fault diagnosis system for

bearings". Journal of Intelligent Manufacturing.

[44] Gelly, G. (2017). « Reseaux de neurones recurrents pour le traitement automatique de la parole ». Thèse de doctorat de l'Université Paris-Saclay préparée à l'Université Paris-Sud.

[45] Jacopo G, P. (2021). "Deep Learning Methods for Speech-to-Text Systems". Business and Management department Management and Computer Science.

[46] Xiaojuan Z. 2020. "Application of Deep Learning" in Ocean Big Data Mining ournal of Coastal Research.

[47] Sha F, Saul LK (2007) "Large margin hidden Markov models for automatic speech recognition". In advances in neural information processing systems (pp. 1249-1256)

[48] Sivaram G., Hermansky H. (2011) "Multilayer perceptron with sparse hidden outputs for phoneme recognition". In 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 5336-5339). IEEE.

[49] Mohamadpour M., Farokhi F. (2009) "A new approach for Persian speech recognition". In 2009 IEEE international advance computing conference (pp. 153-158). IEEE.

[50] Warden, P. 2017, Launching the Speech Commands Dataset. https://ai.googleblog.com/2017/08/launching-speech-commands-dataset.html?

[51] Gelly, G. 2017. « Reseaux De Neurones Recurrents Pour Le Traitement Automatique De La Parole ».

[52] Bounassif, A. 2019. « Speech Recognition Using Deep Neural Networks" A Systematic Review. IEEE.

[53] Jingjie G. 2021. « MA-LSTM: A Multi-Attention Based LSTM for Complex Pattern Extraction". IEEE.

[54] Aditya A. 2018. "Speech Recognition using Recurrent Neural Networks". IEEE

[55] Yann, L. 2015. "Deep learning".

[56] Jahirul I. 2019. "A Speech Recognition System for Bengali Language using Recurrent Neural Network". IEEE.

[57] Wang B., Yin Y., Lin H. (2020) "Attention-based transducer for online speech recognition". arXiv preprint arXiv:2005.08497.

[58] Qian F., X. Chen, "Stock Prediction Based on LSTM under Different Stability," 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), 2019, pp. 483-486, doi: 10.1109/ICCCBDA.2019.8725709.

[59] Islam J, Mubassira M, Islam MR, Das AK (2019) “A speech recognition system for Bengali language using recurrent neural network”. In 2019 IEEE 4th international conference on computer and communication systems (ICCCS) (pp. 73-76). IEEE.

[60] Muhammadjon. M. 2019. “Image Approach to Speech Recognition on CNN”.

[61] Passricha V. 2020 "A Hybrid of Deep CNN and Bidirectional LSTM for Automatic Speech Recognition" Journal of Intelligent Systems, vol. 29, no. 1, 2020, pp. 1261-1274.

[62] Alsobhani A. 2021 J. Phys.: Conf. Ser. 1973 012166.

[63] Sanjay K. 2020. “Speech Recognition: Key Word Spotting through Image Recognition”. arXiv:1803.03759v2.

[64] S. Benkerzaz (2019). “A Study On Automatic Speech Recognition”. Journal Of Information Technology Review Volume 10 Number 3.

[65] Poudel S., Anuradha, R. (2020) “Speech Command Recognition Using Artificial Neural Networks”. Joiv: International Journal On Informatics Visualization, 4(2), 73-75.

[66] Elloumi Z. (2019). « Prediction De Performance Des Systemes De Reconnaissance Automatique De La Parole A L’aide De Reseaux De Neurones Convolutifs ». Hal Id: Hal-01976284. Tal. Volume 59 - N◦ 2/2018, Pages X A X.

[67] Warden, P. 2017, « Speech Commands: A public dataset for single-word speech recognition”.

[68] Yang X., Yu H., Jia L. (2020) “Speech Recognition Of Command Words Based On Convolutional Neural Network”. In: 2020 International Conference On Computer Information And Big Data Applications (Cibda) (Pp. 465-469).

[69] Wei H. 2020. “ContextNet: Improving Convolutional Neural Networks for Automatic Speech Recognition with Global Context”. arXiv:2005.03191v3.

[70] Alexander H. Liu. 2020. “Sequence-to-sequence automatic speech recognition with word embedding regularization and fused decoding”. arXiv:1910.12740v2

[71] Alex L. 2019. “Survey of Dropout Methods for Deep Neural Networks”. arXiv:1904.13310v2

[72] Rikiya Y. 2018. “Convolutional neural networks: an overview and application in radiology”.

[73] TIAGO C, R. (2018). “Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications”.

[74] Yu, T. (2019). “Getting Started with TensorFlow Deep Learning”.

[75] Jakobowicz, E. (2018). « Python pour le data scientist : Des bases du langage au machine learning ».

[76] Jessica, C. 2020. “Dear UCLA Information Technology Planning Board”. IS 270.

[77] Vimala, C. 2015. “Isolated speech recognition system for tamil language using statistical pattern matching and machine learning techniques”. Journal of engineering science and technology vol. 10, no. 5.

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
CHADLI BENDJEDID EL-TARF UNIVERSITY
SCIENCES AND TECHNOLOGY FACULTY
COMPUTER SCIENCE DEPARTMENT

CERTIFICATE OF PARTICIPATION

Awarded to:

Abir Nouari

For presenting the paper entitled:

Fuzzy controller for driving a car using genetic algorithm.

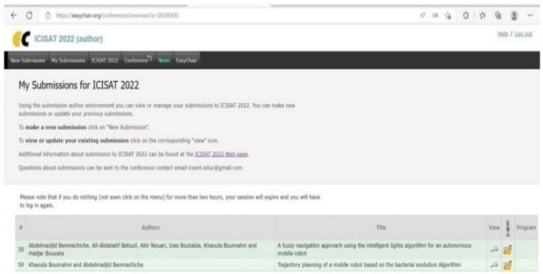
At the 1st International Conference on Autonomous Systems and their Applications (ICASA'22).

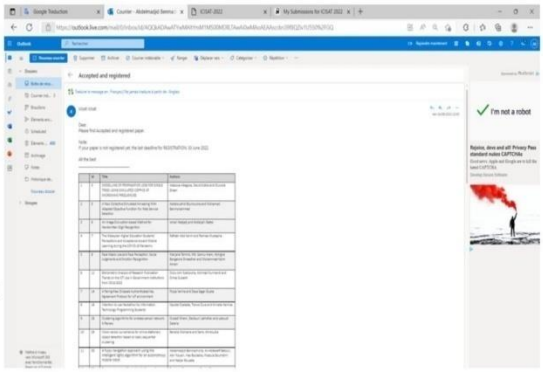
Author(s): Abir Nouari, Ahlem Mellouk and Hadjer Bouzata.

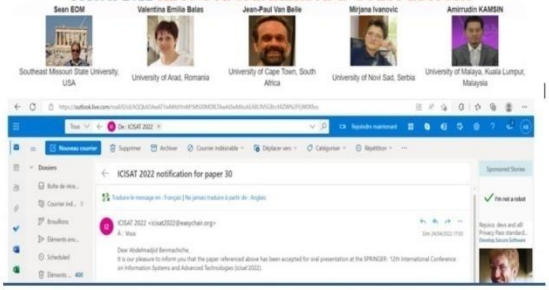
Dr Abdelmadjid Benmachiche

ICASA'2022 General Chair

UCBET / FST / INF - P.O. Box 73, El-Tarf 36000, Algeria - : http://univ-eltarf.dz - : cnia8.info@gmail.com







11	30	A fuzzy navigation approach using the intelligent lights algorithm for an autonomous mobile robot	Abdelmadjid Benmachiche, Ali-Abdelatif Betouil, Abir Nouari, Ines Boutabia, Khaoula Boumahni and Hadjar Bouzata