International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 15, No.1 (Feb-24)

http://dx.doi.org/10.12785/ijcds/150170

ArabAlg: A new Dataset for Arabic Speech Command Recognition for Machine Learning Applications

Nourredine OUKAS[1,2], Samia HABOUSSI[1], Chafik MAIZA[1] and Nassim BENSLIMANE[1]

[1]*LIM Laboratory, Department of Computer Sciences, Akli Mohand Oulhadj University of Bouira, Algeria*
[2]*LIMOSE Laboratory, M'Hamed Bougara University of Boumerdes, Algeria*

Received 17 Aug. 2023, Revised 31 Jan. 2024, Accepted 4 Feb. 2024, Published 5 Feb. 2024

Abstract: Automatic Speech Recognition (ASR) systems have witnessed significant advancements in recent years, thanks to the emergence of deep learning techniques and the availability of large speech datasets in various languages. With the increasing demand for Arabic voice-enabled technologies, the availability of a high-quality and representative dataset for the Arabic language becomes crucial. This paper presents the development of a new dataset called **ArabAlg**, specifically designed for Arabic Speech Command Recognition (ASCR), to support the integration of Arabic voice recognition systems into smart devices in the Internet of Things (IoT). This research focuses on collecting and annotating a diverse range of Arabic speech commands, encompassing various domains and applications. The dataset construction process involves recording and preprocessing several utterances from native Arabic speakers. To ensure precision and reliability, quality control measures are implemented during data collection and annotation. The resulting dataset provides a valuable resource for training and evaluating ASCR systems tailored for Arabic speakers using Machine Learning and Deep Learning.

Keywords: Automatic Speech Recognition, Machine learning, Dataset for limited vocabulary, Arabic Speech Command Recognition, Smart devices, Internet of Things.

1. Introduction

Automatic speech recognition is the technology that enables machines to understand human speech. It is a fast-growing field since humans are actively working to make human-machine interactions seamless, thereby improving our daily lives and making every action more efficient [1]. From controlling robots and machines with voice to making morning coffee, speech recognition is becoming indispensable, eliminating the need for pressing buttons [2], [3].

Moreover, limited vocabulary speech recognition and command recognition systems are technologies designed to understand and process spoken language input within a specific set of predefined words or commands [4]. These systems are often used in applications where the range of possible input commands is relatively small and well-defined, such as voice assistants, smart homes, or voice-controlled devices. These commands are typically associated with specific functionalities or operations that the system can perform. For example, in a voice-controlled smart home, the system might recognize commands like ”turn on the lights” or ”close the shades.” These systems are effective for applications with a limited vocabulary because they can achieve very high accuracy. However, they may struggle with out-of-vocabulary commands that are not part of their predefined set. However, Arabic command recognition is a complex task and requires a significant amount of computational and linguistic resources [5]. Recognition accuracy can vary depending on factors such as the quality of the audio input, the size and quality of the language model, and the complexity of the commands being recognized [6]. Ongoing research and advancements in natural language processing and speech recognition technologies are continually improving the accuracy and performance of Arabic command recognition systems.

Figure 1 illustrates the pipeline of an Arabic Automatic Speech Recognition (ASR) system. Machine learning and deep learning models require datasets for training and testing before deployment on smart devices. Broadly speaking, there is a significant lack of data, particularly in the Arabic language, to train models with high accuracy. In an effort to address this shortfall, this investigation

E-mail address: n.oukas@univ-bouira.dz; noureddine.mag@gmail.com

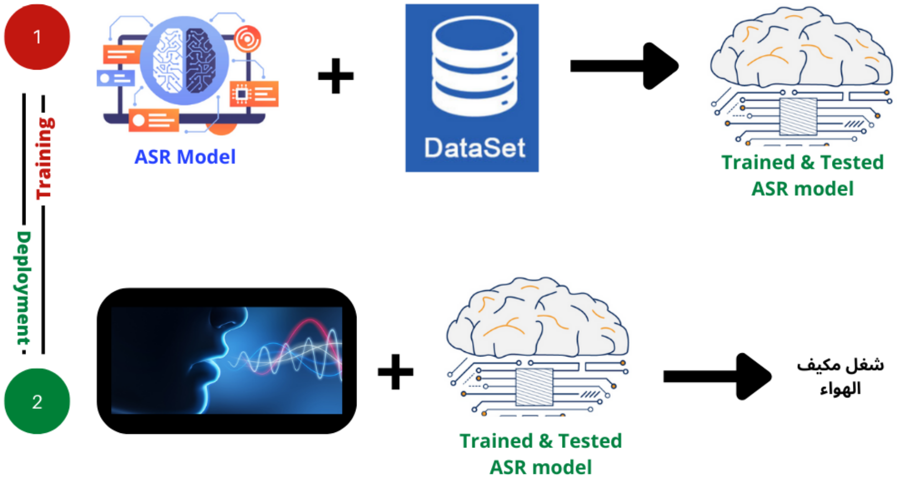


Figure 1. Arabic ASR pipeline

introduces the construction of an Arabic dataset specifically dedicated to command recognition systems. With such a dataset, the pipeline depicted in Figure 1 can be implemented to create Arabic Command Recognition systems.

Responding to the urgent need for advancements in Arabic speech recognition, this research aims to make a significant contribution by addressing the lack of large-scale Arabic voice command datasets. Our overarching goal is to construct a unique corpus expressly intended to enhance the performance of Arabic speech command recognition systems. Specifically, we present an extensive dataset comprising over 140 spoken voice commands, showcasing rich acoustic diversity. The utterances encompass a wide range of accents, intonations, speaking styles, and background noise conditions, capturing natural contextual variations in realistic user commands. Our dataset will empower Arabic speech recognition by facilitating robust training of deep neural models. Furthermore, it will drive innovations in Arabic-speaking intelligent assistants and voice control for native Arabic speakers interacting with IoT devices [7]. As a result, it serves as a crucial resource for researchers and facilitates exploration into technologies like Arabic-speaking intelligent assistants and IoT voice control for native users.

In summary, the primary objective of our project is to construct a dataset for Arabic command recognition. To achieve this, we undertake the following steps:

1) Designing a novel open-source mobile application to streamline the process of collecting speech data.
2) Addressing Arabic native speakers for recording purposes.
3) Collecting speech clips and organizing them in a TSV (Tab-Separated Values) file.
4) Verifying and validating the data with the assistance of specialized tools.
5) Testing the effectiveness of our dataset by training a deep learning model and presenting the obtained results.
6) Publishing new versions of our dataset along with the constructed recording tool.

Our focus will be primarily on the Arabic language, given the scarcity of expansive, publicly accessible speech datasets for Arabic. This deficiency poses a challenge for researchers aiming to advance automatic speech and command recognition systems in the field.

The structure of this paper is as follows: In Section 2, we will discuss the speech recognition process, examining command recognition systems, datasets, and the challenges in Arabic ASR. Section 3 presents related works. Moving on to Section 4, we will delve into the data, discussing the type of data we are handling, our collection and storage methods, and the structure of our dataset. Section 5 outlines the development of a tool for data collection and its structure. In Section 6, we present the conducted steps of data preprocessing and some statistics. Section 7 will cover experimental analysis and results. Finally, we conclude with a summary and offer some future recommendations.

2. Background

Automatic Speech Recognition (ASR) is the technology that lets computers understand speech, it transcribes audio into text, it is also called Speech to text (STT), which is used to help with the interaction of human-computer interface [8]. Therefore, ASR has been in constant development, especially in recent years, with numerous applications mainly in robotics, car systems, health care, education, and military use [9] [10].

Speech recognition is a very important research field as it is trying to improve the interface between humans and machines to increase productivity [11]. Users can now dictate documents, email responses, and other text without manually entering any information into a machine [12]. In addition, it reduces the risk of injury or accidents, for example in voice-controlled car media systems. It also provides the ability for people with limited range of motion or disabilities to use machines.

On the other hand, Arabic is a vastly propagating language with a variety of hundreds of dialects [13]. Our focus on this investigation is to provide a new dataset and an open-source application to help collect data for future works on Arabic ASR in order to improve the accuracy of speech recognition and also to serve and advance these systems that will hopefully help Arabic speakers.

A. ASR Architecture

ASR systems typically consist of several components, including an acoustic model, a language model, and a decoder [14]. The acoustic model is responsible for analyzing the acoustic characteristics of the input speech signal and generating a sequence of phonetic units that correspond to the input speech. The language model is responsible for predicting the likelihood of different words and phrases in a given language, based on the context of the speech. The decoder then combines the output of the acoustic and language models to produce a final transcription of the input speech [8].

The accuracy of ASR systems has improved significantly in recent years, due to advances in deep learning [15]. However, ASR systems still face many challenges, such as dealing with loud and accented speech, recognizing speech in different languages and dialects, and understanding spoken language in complex and dynamic environments [16]. To understand how Speech recognition works, we first need to understand human speech. Humans have been communicating since the dawn of time, with thousands of different languages and sounds all around the world. So it was inevitable to reach the point where we would communicate with machines using this age-old technique. Vocal cords vibrate to make sounds, the air coming out of the lungs is what makes them vibrate when the edges of the vocal cords come together. The propagation of waves through the air vibrates at a certain frequency to make a sound [17]. A speech recognition system's fundamental objective is to empower a computer or device that can hear and comprehend spoken or acoustic input to take the appropriate action [8].

ASR systems use a combination of acoustic and language modeling techniques to translate spoken language into text (see Figure 2). The process typically involves the following steps [18]:

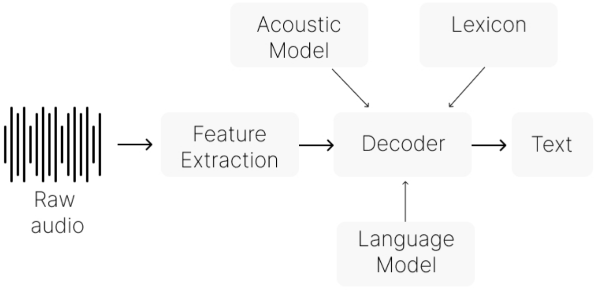


Figure 2. Overall Structure of ASRs [8]

- **Audio preprocessing:** The audio signal undergoes filtering and normalization to eliminate noise and other forms of distortion.
- **Extraction of features:** The audio signal is converted into a sequence of numerical features that represent the spectral characteristics of the sound. These characteristics are usually based on the short-term Fourier transform or other time-frequency analysis techniques [19].
- **Acoustic model:** The feature sequence is matched against a statistical model of speech sounds called a Hidden Markov Model (HMM), which represents the probability distribution of speech sounds in different contexts [20].
- **Lexicon:** The lexicon is a database or a list of words that the ASR system is trained to recognize. It contains information on how each word is pronounced, including its phonetic transcription, which is a representation of the sounds that make up the word [21].
- **Language model:** The transcribed text is then processed by a statistical language model, which uses probabilistic techniques to predict the likelihood of words and phrases in a given context [22].
- **Decoding:** The output of the acoustic and language models is combined to generate a final transcription of the spoken language.

B. Characteristics of ASRs

The characteristics of an ASR include [23]:

- **Vocabulary Size:** The size of the vocabulary that the system can recognize is an important factor. Larger vocabulary sizes require more processing power and memory to achieve good accuracy.
- **Robustness:** The ability of the system to handle variations in speech, such as accents, dialects, back-

ground noise, and different speaking rates is a critical characteristic.

- **Adaptability:** The ability of the system to adapt to new speakers or new languages is a desirable characteristic. This allows the system to learn from its mistakes and improve its accuracy over time.
- **Speed:** The speed at which the system can recognize speech and produce output is important for real-time applications such as voice-controlled assistants or automated transcription services.
- **User Interface:** The user interface of the system, including the ease of use, feedback provided, and the ability to interact with the system, is an important characteristic, especially for consumer-facing applications.
- **Scalability:** The ability of the system to scale and handle large volumes of speech data is important for applications such as call center automation or large-scale transcription services.
- **Performance metrics:** The accuracy of a system in recognizing spoken words is a pivotal factor that directly influences its utility for a specific application. In ASR field, various metrics are taken into consideration, including the Word Error Rate (WER) [24], the Character Error Rate (CER) [3], the Match Error Rate (MER) [25], and the Phoneme Error Rate (PER) [26], [25]. For instance, the MER reflects the percentage of inaccurately predicted and inserted words. A lower MER indicates superior performance, with a perfect score of 0 signifying flawless accuracy in the ASR system. The PER and WER encompass the total of phoneme and word errors, which include inserted, deleted, and changed phonemes or words [3]. This cumulative value is then divided by the total number of phonemes or words, respectively. To establish statistical significance in comparisons between models, the error rates from all tests across all examined speakers are averaged.

Overall, a good speech recognition system should have high accuracy, be robust and adaptable to different types of speech and users, have a fast response time, and be easy to use and scalable for various applications.

C. Command Recognition Systems

Command recognition systems can be implemented in various applications and devices, including virtual assistants (such as Siri, Alexa, or Google Assistant), smart home devices, automotive systems, and dictation software, among others [12]. These systems allow users to interact with devices and control them through voice commands, providing a more natural and intuitive user experience. In this paper, we will focus on helping people to create limited vocabulary systems for the Arabic language, which is a very diverse language with different dialects that have very limited data collected about them.

D. Challenges in Arabic ASR

ASR systems can be trained on large datasets of transcribed speech and text to improve their accuracy over time. However, ASR still faces challenges in accurately recognizing speech in noisy environments, with diverse accents and dialects, and in recognizing spontaneous speech with irregularities such as pauses, hesitations, and repetitions [27].

However, Arabic is a complex language with unique features that pose several challenges for ASR systems [16]. Some of the main challenges in Arabic ASR include:

- **Dialectal Variation:** Arabic has several dialects that differ in pronunciation, vocabulary, and grammar. ASR systems trained in one dialect may not be able to recognize speech from another dialect, which can limit their usefulness for multilingual or regional applications.
- **Phonemic Complexity:** Arabic has a complex phonemic system with many sounds that are not present in other languages. Some sounds are produced at the back of the throat, making them difficult to distinguish from other similar sounds.
- **Non-linear Script:** Arabic is written from right to left and uses a non-linear script in a cursive style, which can make it challenging for ASR systems to segment words and recognize them accurately.
- **Lack of Standardization:** Arabic lacks a standardized writing system, which can lead to inconsistencies in spelling and pronunciation. This can make it difficult for ASR systems to recognize speech accurately.
- **Limited Training Data:** There is a shortage of large, publicly available speech datasets for Arabic, which can limit the ability of ASR systems to learn and improve their performance over time.
- **High Error Rate:** Due to the above challenges, Arabic ASR systems often have a high error rate compared to ASR systems for other languages. This can make it difficult to use them for tasks such as transcription or voice-based applications.

Hence, addressing these challenges will require continued research and development in the field of Arabic ASR, including the creation of more comprehensive and standardized datasets which is our goal, and also improve algorithms for recognizing dialectal variation and non-linear scripts, and more robust modeling techniques that can handle the complexity of Arabic phonemes.

In order to contribute to the advancement and progress of ASR systems, particularly Arabic Command Recognition Systems (ACRS), we have developed a novel dataset dedicated to ACRS. The proposed dataset aims to address the shortage of available data and consists of a collection of Arabic commands designed for smart devices. This dataset facilitates the conversion of voice into command signals or codes, enabling smart devices to execute specific tasks based on the obtained code.

In summary, the proposed dataset tackles several challenges in Arabic ASR:

- Providing an open dataset for the development of Arabic Command Recognition Systems, contributing to the expansion of available data.
- The availability of Arabic datasets enhances the accuracy of the recognition process, leading to a decrease in error rates.
- Offering an open-source mobile application to assist individuals in collecting more data for ACRS or Arabic ASR in general.
- Providing an extensive list of Arabic commands to help interested individuals create their own datasets or participate in enhancing the ArabAlg dataset.
- ArabAlg boasts a considerable diversity of speakers, encompassing men, women, and teenagers, as well as linguistic variations. This diversity is instrumental in training models that demonstrate robustness to a range of speaker characteristics.
- ArabAlg dataset, recorded under varying noise conditions, contributes to the development of models that are more resilient to different acoustic environments.

To further enhance our dataset, we aim to collaborate with additional volunteers, especially those with distinct dialects such as Egyptian, Gulf, Levantine, and others. This expansion will enable the creation of multi-dialect models, improving the system's adaptability to diverse Arabic linguistic contexts.

3. Related Works

Data is very important to improve ASR systems, and when it comes to machine learning and deep learning it is required to have a large amount of data to obtain the most accurate results. For Speech-to-Text systems, there are some datasets such as Common voice [28], MGB-3 [29], QASR [30], and others [1]. In this investigation, the presented dataset is in Arabic to support future work in the field of Arabic command recognition systems. We have found little work in this field, some of the datasets only contain just a few commands with the numbers ranging from zero to nine. Here are some examples of the existing datasets on ACRSs:

- **Speech Commands: A Dataset for Speech Recognition with a Limited Vocabulary** Created by Pete Warden, it contains the digits zero to nine and about 25 words, some of them are common in Internet of Things: ”Up”, ”Down”, ”Left”, ”Right”, ”Backward”, ”Forward”, ”Follow” and ”Learn”. The final dataset contains 105829 utterances of thirty-five words that were recorded by 2618 users [31].
- **Database for Arabic Speech Commands Recognition:** A dataset that contains numbers from 0-9 and 6 commands: ”add”, ”back”, ”cancel”, ”delete”, ”confirm”, ”continue”, were obtained from forty different speakers [32].
- **Murtadha Yaseen Arabic speech commands dataset:** Available on Kaggle, it contains 40 keywords, recorded by 30 participants, and each of them recorded 10 utterances for each keyword [33].
- **General Conversation Speech Data in Arabic (Algeria):** A speech dataset that contains 20 hours of general conversation-type speech/audio data for the Arabic language speech recognition model. 30 people from different states/provinces of Algeria who are native, helped in collecting this data. The participants in the collection are males and females from the age group of 18 to 70 years. Each audio is a spontaneous and unscripted conversation between two people with an average duration of each audio file of 15 to 60 minutes [34].
- **Spoken command TV dataset:** This corpus was created by 100 Arabic native speakers using a speaker-independent methodology: including 68 adults (37 Males and 31 Females) and 32 children (13 Males and 19 Females) [35].
- **Arabic voice commands:** Comprises an individual spoken word command captured in the Arabic language through MATLAB recording:
  قف ، يسار، يمين ، اذهب
  from different humans (man and woman) ranging in age from 12 to 58 years [36].
- **Dataset of Arabic Voice Commands for Various Speech Processing Tasks:** developed by M.Lichouri et al [37] and it consists of 10 commands. Ten speakers repeated and recorded ten times each command.
- **Arabic Speech Commands Recognition Using PyTorch and GPU:** this dataset comprises two widely used classes, namely, ”NO” and ”YES” in English, equivalent to ”لا” and ”نعم” in Arabic. It consists of 600 one-second waveform audio recordings, each featuring a distinct word, collected from 30 different contributors [38].
- **Arabic voice command dataset** The researchers in

[39], designed a system for recognizing Arabic voice commands, utilizing a proprietary dataset comprising 1600 samples corresponding to 20 distinct commands. Each command was recorded in the (.wav) format at a frequency of 16000Hz.

- **Arabic voice commands dataset based on the user's preferences to control a home device** Salah et al [40], have created an Arabic voice commands dataset to interact with home appliances rather than just turning them on/off for elderly people who prefer using their mother-tongue. The dataset incorporates contributions from 12 speakers, including 6 males aged between 22 and 53 years, as well as 6 females aged from 12 to 75 years. This diverse age range was chosen to introduce a broad spectrum of voice variances in the dataset.

The robustness and superiority of our dataset over previously established datasets in the field are vividly demonstrated in Table I. This table provides a comprehensive comparison between our proposal and existing datasets, highlighting our dataset's exceptional performance across various dimensions. Notably, our dataset outshines others in terms of the sheer quantity of commands, diversity in participants, each distinguished by unique accents, and the broad spectrum of topics encompassed by the commands.

4. ArabAlg Architecture

Data is the most important component in deep and machine learning, we can't expect any improvement if we don't have a wide range of versatile and huge amounts of data. Our goal is to collect the most data possible from native Arabic speakers of different nationalities, ages, and from both genders. We developed an Android application to help us collect the data we need to improve Arabic speech recognition to help make it more accurate and effective.

A. Data Type

The type of data we're dealing with is human speech. A complex sound with a wide range of frequencies that varies from one human to another. A person's speech also changes depending on their mood, i.e., if they are tired or sad, or whether they are speaking softly or loudly. Another challenge is the dialects and accents; they can make some words sound completely different. By collecting massive amounts of voice data, the machine will be able to understand everyone, regardless of their race, culture, age, and other personal factors.

B. Collecting the Data

We developed a light android application that asks for essential but not sensitive information that we need to further increase the accuracy of the model like their Age, Nationality, and gender. From there, the users are asked to read from a list of 148 commands, record each and everyone, and send us the data so we can use it to create our dataset [1].

Even though the equipment is different for every user since it is on different phones, we expect a similar result since phone microphones have really advanced in recent years. The audio collected will be stored in WAV (Waveform Audio File Format) format, at 16-bit Bit depth and 16000hz sample rate, which is ideal for speech data collection [41].

C. Storing the Data

We store speech data using Google's Firebase[2], which is a NoSQL cloud storage that is a powerful, simple, and cost-effective object storage service, easy to setup and use for Android applications with a helpful analytics dashboard. The audio files will be stored as WAV files accompanied by the ID of the user, their Age, Nationality, gender, and a Boolean value for the audio file that refers to its validation. We have a mother collection that contains every word, and for each one, it has a collection that will store every recording from the users for that specific word.

D. Legal Requirements for Collecting Personal Data

The legal requirements for collecting personal data depend on the country or region where the data is being collected. However, some common legal requirements include [42]:

- **Consent:** Individuals must give their consent for their personal data to be collected. Consent must be informed, specific, and freely given.
- **Purpose limitation:** Personal data can only be collected for a specific and legitimate purpose. It cannot be used for any other purpose without obtaining further consent.
- **Data minimization:** Collect only the essential personal information needed to achieve the specific purpose for which it is obtained.
- **Storage limitation:** Personal data should not be maintained any longer than is required for the intended use.
- **Security:** Appropriate technical and organizational measures must be in place to guard against unauthorized access, disclosure, or loss of personal data.

It is important to note that there may be additional requirements depending on the nature of the data being collected and the specific legal framework in the country or region where the data are being collected. We avoided storing any personally identifiable information from volunteers like their first and last name or e-mail, since any such

[1] The ArabAlg dataset is available at the following URL: https://github.com/noukas/ArabAlg. For new versions, please contact the corresponding author.

[2] https://firebase.google.com/

TABLE I. Comparison between different previous datasets (Part.: Number of Participants, Lang.: Langage)

	Size	Topic	Commands count	Part.	Clips	Lang.	Tool
Speech commands data-set [31]	3.8 GB	IoT or robotics	35	2618	105,829	EN	Recorded By phone or laptop microphones
Database for ASC Recognition [32]	-	-	16	40	1600	AR	Recorded By Mobile phone
Arabic speech commands dataset	434.12 MB	IoT	40	30	12,000	AR	/
General Conversation Speech Data	-	-	-	30	-	AR	/
Spoken command TV dataset, [35]	-	TV	10	100	10000	Arabic	/
Arabic voice commands [36]	-	Human Computer Interaction System	4	-	4000	AR	Recorded by MATLAB
Dataset of Arabic Voice Commands for Various Speech Processing Tasks [37]	-	IoT	10	10	1000	AR	/
Arabic Speech Commands Recognition Using PyTorch and GPU [38]	-	IoT	2	30	600	AR	/
Arabic Voice command to help illiterate or blind for using computer [39]	-	Human Computer Interaction System	20	-	1600	AR	/
Arabic voice commands based on the user's preferences to control a home device [40]		IoT	36	12	432	AR	/
ArabAlg (our dataset)	150.67 MB	IoT and Robotics	148	100	2538	AR	Recorded by Android Application

data is required to be handled with extreme care for privacy reasons. We also made sure There is no sign-in using a user ID that could be connected to personal information.

E. Dataset Word Selection

We are employing a limited set of words to simplify the capturing process, while still maintaining sufficient variability for potential benefits in certain applications when creating models from the data. We opted to construct our vocabulary using 148 commands. This selection encompasses the numerical digits from 0 to 9 along with common Arabic commands that are suitable for the Internet of Things devices or robotics applications, such as words implying movement, setting timers or temperature, controlling volume, functions, and so on.

F. Data Structure

The audio data are stored in a collection, for every command there is a folder that contains all the paths to the audio files of that command with a single file. It is also linked to the user's id, nationality, age, gender, and a Boolean (true/false) for whether the audio is verified or not.

G. Data Fields

Let's begin by identifying and defining all the variables and fields present in the dataset:

- **User_id:** Is a String type Variable field that is unique to every user, it helps track recordings from a specific user.
- **User_age:** String type field that represents the user's age. It ranges from (7-85)
- **User_gender:** String type field that represents the user's gender. Either Male or Female.
- **Nationality:** String type field that represents the user's nationality.
- **code:** INT type field that contains an 8-bit unique code for every command that we will use as a target for the model.
- **transcription:** String type field that contains the

transcription of the recorded word, also used as a target for the model.

- **Audio_id:** String type Variable field that is unique to all the audio files present in the dataset.
- **Audio_url:** String type Variable field that is unique to the audio file and represents the exact URL link of that file in the database.
- **Verified:** Boolean type variable that is unique to all audio files: is true if that recording has been verified and false otherwise.

This is an example of data instance:

- audio_id: ”Talky.06_7j53hrF”
- audio_url:”firebasestorage.googleapis.com/v0/b /talky-76000.appspot.com/o/commands”
- nationality: ”Algerian”
- user_age: ”19”
- user_gender: ”Male”
- user_id: ”7j53hrFmwqbfxV9GXz0KAureoZD3”
- verified: false
- code: ”06”
- transcription: ”تحرك”

H. List of Commands

The first version of the proposed dataset contains around 148 commands. The dataset remains extensible in terms of commands and records. The initial list is as follows:

للأعلى ، للأسفل ، إلى الأعلى ، إلى الأسفل ، ثَبّت ،
تَحَرّك ، للأمام ، للخلف ، لليمين ، لليسار ، سلام ، مرحبا ،
شكرا ، لا ، نعم ، إلى ، من ، عند ، بين ، حول ، عبر ،
فوق ، تحت ، بعد ، قبل ، خلال ، بينما ، إذا ، عندما ، حتى
، أثناء ، ضع ، تأكيد ، تابع ، استمر ، صفر ، واحد ، اثنان ،
ثلاثة ، أربعة ، خمسة ، ستة ، سبعة ، ثمانية ، تسعة ، عشرة ،
عشرون ، ثلاثون ، أربعون ، خمسون ، ستون ، سبعون ،
ثمانون ، تسعون مئة ، إضافة ، إعادة ، إلغاء ، حذف ، إضافة
الى ، العودة ، العودة إلى الخلف ، كتم الصوت ، ارفع
الصوت ، خفض الصوت ، تغيير ، اتصل ، إنهاء المكالمة ،
الوضع ، الوظيفة ، الساعة ، مؤقت ، إفتح ، اغلق ، ابدء ،
تثبيت ، تشغيل ، إيقاف مؤقت ، إلغاء ، إعدادات ، المزيد ،
إضافة ، ضَبِطَ ، ضَبِطَ الحرارة ، أَحمَر ، أَخضَر ، أَزرَق ، تعيين
اللون ، أَسوَد ، أَبيَض ، بُرتُقَالِيّ ، أُرجُوَانِيّ ، تجنب ، اذهب ،
اتبع ، فحص ، كشف ، تحديد ، التقاط ، امسك ، أطلق ،
اضغط ، اسحب ، افتح ، اغلق ، أقفل ، تفعيل ، تشغيل ،
إيقاف مؤقت ، تخطي ، تسجيل ، حفظ ، تحميل ، حذف ،
مسح ، انسخ ، الصق ، تراجع ، بحث ، إعادة ، قراءة ،
تحدث ، ترجمة ، تحويل ، احسب ، عرض ، إخفاء ، تكبير ،
تصغير ، اكتب ، إعادة التشغيل ، نوم ، استيقظ ، تأجيل ،
الرد ، إرسال رسالة ، استلام الرسائل ، إعادة توجيه الرسائل ،
الرد على الرسائل ، إرسال بريد إلكتروني ، استلام البريد
الإلكتروني ، إعادة توجيه البريد الإلكتروني ، الرد على
البريد الإلكتروني ، كيف هو الطقس ، كم الساعة الآن ،
تاريخ اليوم ، تشغيل الإضاءة ، إيقاف الإضاءة

5. TALKY TOOL DESCRIPTION

For collecting the data we developed a lightweight Android application, called TALKY, using Flutter and Google’s Firebase that we deployed as a separate mobile application [3]. It has an intuitive UI/UX and depends on native Arabic speakers to volunteer and try to help us gather the audio necessary for us to make an impact on improving Arabic speech recognition.

Android applications, as a category, encompass a broad range of functionalities, and data collection applications are no exception. When comparing Android data collection applications with other types of data collection tools, such as web-based or desktop applications, several differences can impact the quality of data collection and the richness of vocabulary. For instance: 1) Android data collection applications are designed to work offline, which is particularly beneficial in areas with limited or no internet connectivity. 2) They can leverage native device features such as GPS and the camera. These features enhance the types of data that can be collected, allowing for more diverse and contextually rich information. For example, a data collection app on Android might capture not only audio but also images and location data.

The primary purpose of TALKY is to collect high-quality Arabic speech data through voluntary user participation. Upon launching the application, users are greeted with a user-friendly interface that guides them through the data collection process. The first step involves providing some basic information, including age, nationality, and sex, which helps to ensure a diverse and representative data set.

Once registered, users embark on a journey to record speech samples by pronouncing specific words provided by

[3] The source code is available here for free download: https://github.com/elvirtuozo/my_talky

TALKY. With a meticulously curated list of 148 words, the application ensures comprehensive coverage of the Arabic language, capturing a wide range of phonetic variations, dialects, and accents. Each word is presented on the screen, and users are prompted to record their pronunciation. TALKY goes the extra mile in creating a seamless user experience by allowing users to review and confirm their recordings. After each word is pronounced, users have the choice to either confirm or delete their recording. This feature ensures that only accurate and high-quality speech data is collected, maintaining the integrity of the dataset. The TALKY tool boasts an intuitive and visually appealing design, making the speech data collection process enjoyable and engaging. It leverages state-of-the-art technologies to enhance the clarity and quality of recordings, enabling users to capture their voices with remarkable precision.

The impact of TALKY goes beyond individual contributions. By aggregating speech data from a diverse range of users, the application facilitates the creation of a vast and representative Arabic dataset. This dataset serves as a valuable resource for researchers, linguists, and developers working on speech recognition, voice assistants, and machine learning applications in Arabic. Figure 3 shows the general design of TALKY.

A. *General use cases diagram*

In order to illustrate the variety of ways that a user might engage with a system, the use cases diagram is being introduced. Figure 4 shows the use case diagram of the TALKY application. The unregistered user is first welcomed with a splash screen, then asked to select his gender, age, and nationality from three drop-down lists. Then he becomes registered and assigned a user id, only then he can start recording the commands listed and gets the ability to listen back to the recording to confirm or delete it in case of maybe unwanted background noise. Every time he records and confirms a word the next one shows on screen until he finishes all of the commands.

B. *Class diagram*

Figure 5 Depicts the static structural diagram portraying the TALKY system's composition, which outlines its classes, attributes, functions, and interconnections between classes.

6. Data Preprocessing and Statistics

After the period of collecting and recording, the obtained data undergoes various treatments. Hereafter, we present the preprocessing steps for the data and also provide some statistics related to ArabAlg V 1.0.

A. *Data Preprocessing*

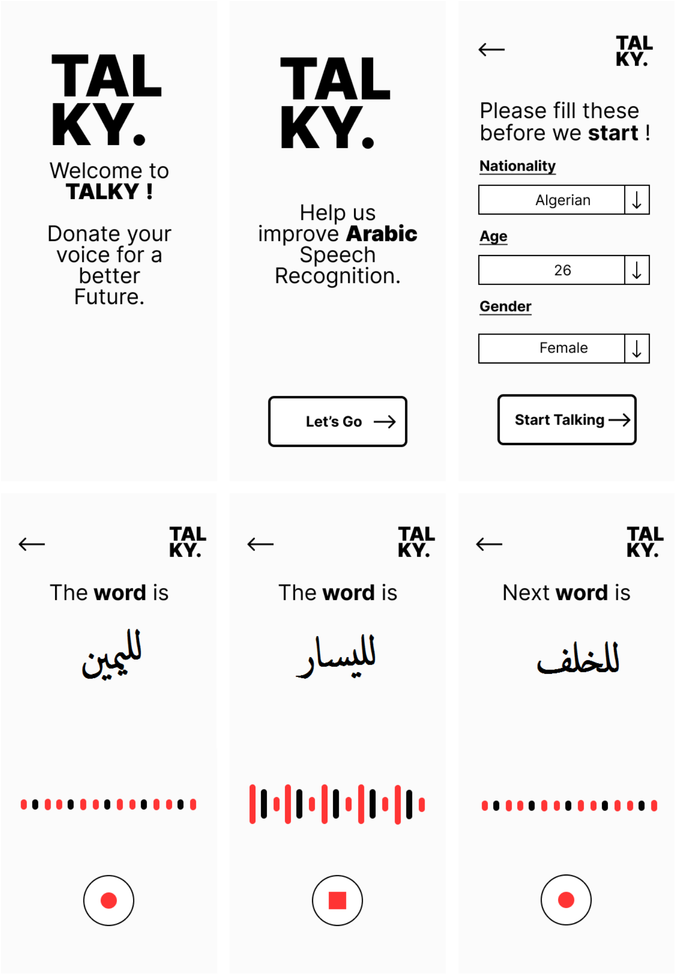
Speech data preprocessing refers to the techniques and methods used to prepare speech data for use in speech recognition or other natural language processing applications. We will employ some of the common techniques outlined in [43], depending on the model.

- **Data validation:** We have carefully verified each user record to ensure that it accurately matches the transcriptions from our list. To maintain data accuracy and reliability, we have implemented various quality control measures. For example, we have checked the metadata associated with the recordings to ensure that speaker IDs and demographics are correctly linked to the corresponding recordings and transcripts, thus preventing any data integrity issues. Additionally, we have conducted an acoustic analysis of the voice data to confirm that it aligns with the expected characteristics. We have also standardized punctuation, spelling, abbreviations, and other elements across the transcriptions through text normalization. Furthermore, we have ensured that factors such as gender and accents are well represented in the distribution of the data. These rigorous quality control protocols helped us to verify that our speech corpus adheres to the intended specifications and can be depended upon for research purposes.
- **Speech data augmentation:** Involves applying various transformations or modifications to existing speech recordings to create new samples. The goal is to simulate realistic variations that can occur in real-world scenarios. Some common data augmentation techniques in speech recognition include: **Pitch shifting** (i.e, Altering the pitch of the speech signal to simulate different voice characteristics or vocal conditions) and **Noise injection** (i.e, Adding background noise or environmental sounds to the speech signal to mimic different acoustic conditions. This helps the model to become more robust to noisy environments).
- **Normalization:** This involves scaling the features to a standard range, typically between 0 and 1. It ensures that the features have equal weight during the training process and improves the overall accuracy of the model.

Consequently, data management and preprocessing is the step that we have tried to focus on the most, by reason of their importance for achieving great results. The quality of the preprocessed data is directly linked to its impact on the performance of the speech recognition system, so it is really crucial to carefully optimize each step in the process.

B. *Statistics*

Herein, we conduct a comprehensive analysis of the demographic characteristics within our dataset, focusing on key factors such as age, region, and gender of the participants. The age distribution revealed a diverse range, with the majority of users falling within the age range of 22 to 27 years old as shown in Fig 6. Thus a diverse age group in the dataset provides a broader context for the model to understand how certain words or phrases are used differently in different age cohorts. Additionally, an examination of gender distribution uncovered that 59% of participants are



TALKY.

Welcome to **TALKY !**

Donate your voice for a better Future.

TALKY.

Help us improve **Arabic** Speech Recognition.

← TALKY.

Please fill these before we **start** !

Nationality Algerian ↓

Age 26 ↓

Gender Female ↓

← TALKY.

The **word** is

لليمين

← TALKY.

The **word** is

لليسار

← TALKY.

Next **word** is

للخلف

Figure 3. General Design of TALKY

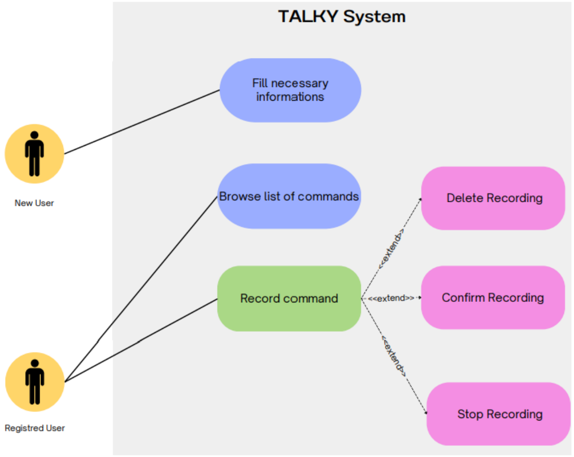


Figure 4. General use cases diagram of TALKY

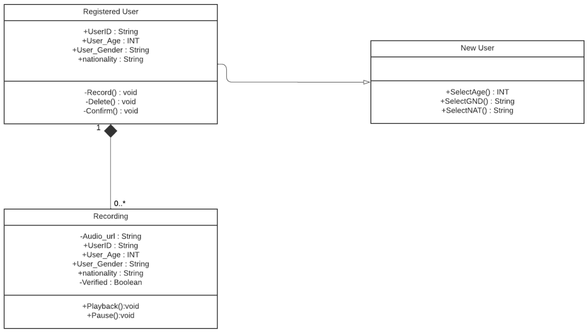


Figure 5. TALKY's class diagram

male and the rest are female (look at Fig 7). A diversity of participants within each gender category (i.e speaking styles, accents, dialects,... etc) underscores the importance of gender inclusivity in our study.

Geographically, our participants spanned across (”Algiers”, ”Bouira”, ”Boumerdes”, ”Béjaia”, ”M’sila”), with notable concentrations in ”Bouira” see Fig 8. This geographic dispersion helps in creating a more balanced and representative model, reducing the risk of unintentional biases associated with a particular region.

ArabAlg is a valuable resource for improving Natural Language Processing (NLP) models by extracting diverse and informative features from audio signals. One of the feature extraction algorithms used is Mel-Frequency Cepstral Coefficients (MFCCs), which captures the spectral characteristics of the audio signal, providing information about the energy distribution across different frequency bands. The inclusion of speaker-related features, such as speaker embeddings and voice quality, enhances the model’s ability to recognize and differentiate between speakers. Additionally, background noise and environmental factors, including noise level and identification of environmental sounds, provide insights into the recording environment, which can affect the robustness of NLP models. Language-related features, such as language identification and accents/dialect analysis, also contribute to tasks where linguistic variations are important.

7. Experimental Analysis

In this section, our primary goal is to illustrate the utilization of our dataset through the implementation of a deep learning model. Our proposed approach holds the potential to benefit other models as well. Specifically, our focus lies in advancing Arabic command recognition systems by formulating and training a model grounded in the amassed dataset.

To achieve this, we employ a straightforward Convolutional Neural Network (CNN) as outlined by O’Shea and Nashashibi [44]. The primary task of this CNN is to categorize spectrograms derived from the recorded audio into image representations. It is important to note that, while we acknowledge the constraint imposed by the limited quantity of data, our primary objective is to showcase the potential inherent in the gathered audio dataset. The overarching aim is to pave the way for future advancements in Arabic command recognition systems.

During the model training phase, we applied a series of preprocessing steps, including data cleaning, normalization, and augmentation techniques. The augmentation process comprised straightforward methods such as modifying pitch values within the range of -3 to +3 semitones, introducing noise, and combining pitch alterations with noise. As a result, three augmented audio files were produced for each corresponding original recording.

A. CNN Architecture

- **Input Layer:** The input layer is defined using the layers. Input function, taking the input_shape as the parameter. The input_shape corresponds to the shape of the input spectrograms.
- **Resizing Layer:** The input is resized to a smaller dimension of 32x32 using the layers. Resizing function. This latter samples the input spectrograms.
- **Normalization Layer:** The normalization layer is instantiated as norm_layer and is used to normalize the input data. The layer is fitted to the training spectrograms using Normalization.adapt.
- **Convolutional Layers:** The model consists of two convolutional layers. The first layers.Conv2D has 32

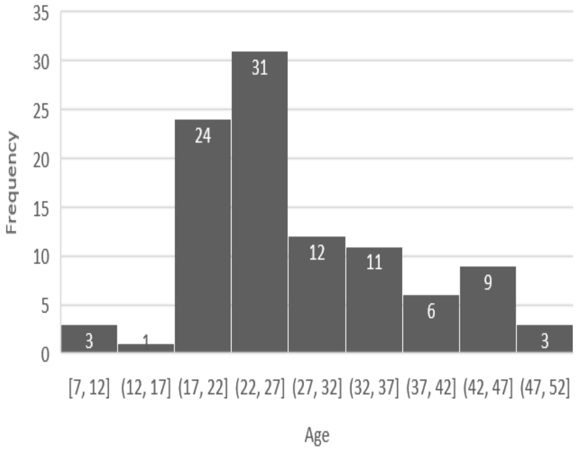


Figure 6. Age distribution in ArabAlg V 1.0

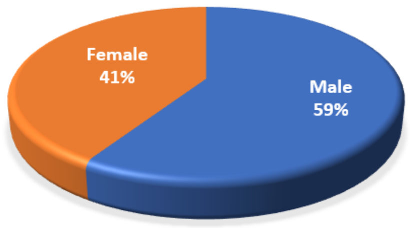


Figure 7. Gender distribution in ArabAlg V 1.0

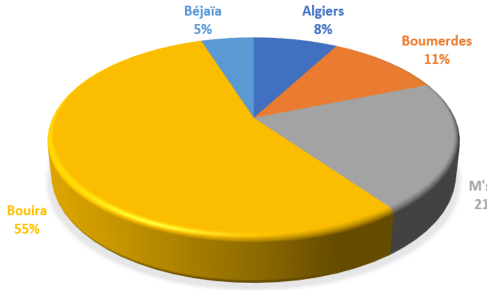


Figure 8. Region distribution in ArabAlg V 1.0

filters with a kernel size of 3x3 and applies the ReLU activation function. The second layers.Conv2D has 64 filters with a kernel size of 3x3 and also applies the ReLU activation function.

- **Max Pooling Layer:** After each convolutional layer, a layers.MaxPooling2D layer is added to downsample the feature maps by selecting the maximum value in a pooling window. This helps reduce the spatial dimensions.
- **Dropout Layer:** After the max pooling layer, a layers.A dropout layer with a dropout rate of 0.25 is added. In order to avoid overfitting, Dropout randomly sets a portion of the input units to 0 during training.
- **Flattening Layer:** A layers.Flatten layer is added to flatten the output of the previous layers into a 1D vector, preparing it for the fully connected layers.
- **Fully Connected Layers:** The flattened output is fed into layers. Dense layer with 128 units and applies the ReLU activation function. A dropout rate of 0.5 is added after the previous dense layer to further regularize the model. The final layers. The dense layer has num_labels units, which corresponds to the number of classes in your Arabic command recognition task. There is no activation specified for this layer, as it will be followed by a softmax activation in the final output layer.
- **Output Layer:** The last layers. Dense layer represents the output layer, which has num_labels units, representing the predicted probabilities for each class.

B. Datasets Description

For the Dataset we will use two fractions of the one we created, first we called it ”Mini-1” consists of 6 commands:

للأسفل، لليسار، للأعلى، لليمين، إلى الأسفل، إلى الأعلى

The second fraction, ”Mini-2”, consists of 35 command words:

تابع، بين، بعد، بينما، إلى الأسفل، إلى الأعلى، إلى، إعادة، إلغاء، أثناء، نعم، لليمين، لليسار، مرحبا، من، للأمام، للخلف، للأعلى، لا، قبل، فوق، للأسفل، عند، عندما، عبر، ضع، شكرا، سلام، حول، خلال، حتى، ثبت، تحرك، تأكيد، تحت

It is worth noting that these subsets encompass the commonly used commands in IoT smart devices. Furthermore, this partitioning enables us to rapidly attain convergence in the training process. On the other hand, we encompassed data augmentation for both of the model training that consists of pitch, noise, and pitch+noise.

C. Model Training using Mini-1 and Mini-2

For the Mini-1 dataset, we split the data 80% for training and 20% for validating, out of a whole 772 audio files belonging to 6 classes (commands) it was split as 695 files for training and 77 files for validation. After splitting the data it is Transformed from waveforms to spectrograms thanks to The Short-Time Fourier Transform (STFT) [45]. This technique converts a time-domain signal into the frequency domain by dividing it into overlapping frames. After computing the STFT a new dimension is added at the end of the tensor, effectively converting the spectrogram from a 2D tensor to a 3D tensor. This is done to match the expected input shape of convolutional layers in TensorFlow. The resulting tensor can be used as input for the convolutional neural networks to process the image data.

After preprocessing the audio files it is run through a sequential model with the following layers:

- Resizing Layer: This layer resizes the input to a shape. It is used to standardize the input shape to match the subsequent layers’ expectations.
- Normalization Layer: This layer performs normalization on the input data, bringing it to a standardized scale. It operates on the input shape.
- Conv2D Layer: This convolutional layer applies 32 filters of size 3x3 to the input. It produces an output shape of (30, 30, 32), where the last dimension represents the number of filters.
- Conv2D Layer: Similar to the previous layer, this convolutional layer applies 64 filters of size 3x3 to the previous layer’s output. Produce an output shape of (28, 28, 64).
- MaxPooling2D Layer: This layer performs max pooling over a 2x2 window, reducing the spatial dimensions by half. It operates on the (28, 28, 64) input and produces an output shape of (14, 14, 64).
- Dropout Layer: To avoid overfitting, this layer randomly sets a portion of the input units to 0 during training. It operates on the input (14, 14, 64).
- Flatten Layer: This layer flattens the input tensor to a 1D vector.
- Dense layer: This fully connected layer has 128 neurons and applies a linear transformation to the input. It operates on the previous output of the layer as an input shape and produces an output shape of (128).
- Dropout Layer: Similar to the previous dropout layer, this layer applies dropout regularization to the input (128).

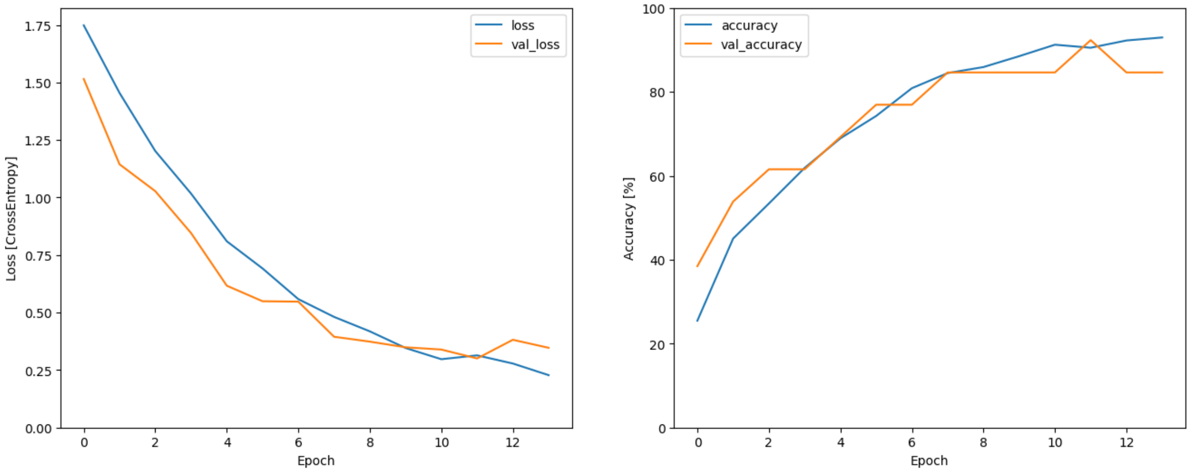


Figure 9. Training and validation graphs of Loss and accuracy using Mini-1

- Dense layer: The final dense layer has 6 neurons, corresponding to the number of output classes. It applies another linear transformation and produces an output shape of (6).

In summary, the sequential model takes an input with shape (batch, samples, channels) and passes it through a series of convolutional, pooling, dropout, and dense layers. The model has a total of 1,625,353 parameters, of which 1,625,350 are trainable. The model aims to classify the input into one of six possible classes corresponding to the commands of the Mini-1 dataset. As for the Mini-2, we ran the same process of data augmentation and trained a similar CNN model only modifying the parameters to suit the 35 classes (commands).

D. Results

1) Mini-1 Results

We ran the model for 25 epochs and got an early stoppage at 14 epochs since we had used the **EarlyStopping** callback. It monitors the validation loss and stops training early if this monitored metric stops improving. for a final result of val_loss at 0.3469 and a val_accuracy at 84%. Figure 9 and Table II illustrate the obtained result.

TABLE II. Test results of the model using Mini-1

Loss	Accuracy	Val_loss	Val_accuracy
0.2280	0.9295	0.3469	0.8462

The confusion matrix provides a summary of the predicted and actual classifications made by the model on the Mini-1 dataset. The matrix is organized into rows and columns, where each row represents the instances in an actual class, and each column represents the instances in a predicted class. Figure 10 represents the confusion matrix of the model using Mini-1.

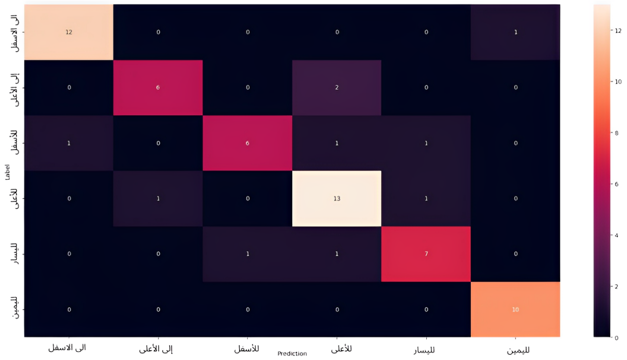


Figure 10. Confusion matrix of the model using Mini-1

2) Mini-2 Results

The Model that was trained on Mini-2, was run for 25 epochs and got an early stoppage at 19 epochs. The final result of val_loss was at 0.7279 and a val_accuracy of 83%. Table III summarises the obtained results. Both training and validation graphs of Loss and accuracy using Mini-2 are represented in Figure 11. The confusion matrix of the model using Mini-2 is shown in Figure 12.

TABLE III. Test results of the model using Mini-2

Loss	Accuracy	Val_loss	Val_accuracy
0.3932	0.8813	0.7279	0.8363

E. Analysis and Discussion

Both models were trained for 25 epochs with an early stoppage mechanism in place. The EarlyStopping callback was used, which monitors the validation loss and stops training early if there is no improvement in this metric.

For the Mini-1 Model, the training process stopped at 14 epochs due to the EarlyStopping mechanism. The final validation loss achieved was 0.3469, indicating that the

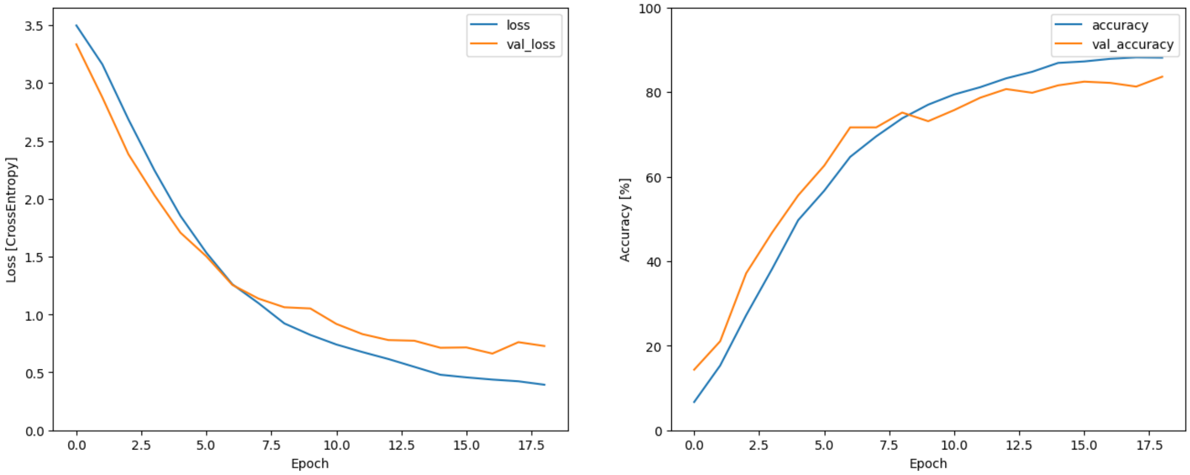


Figure 11. training and validation graphs of Loss and accuracy using Mini-2

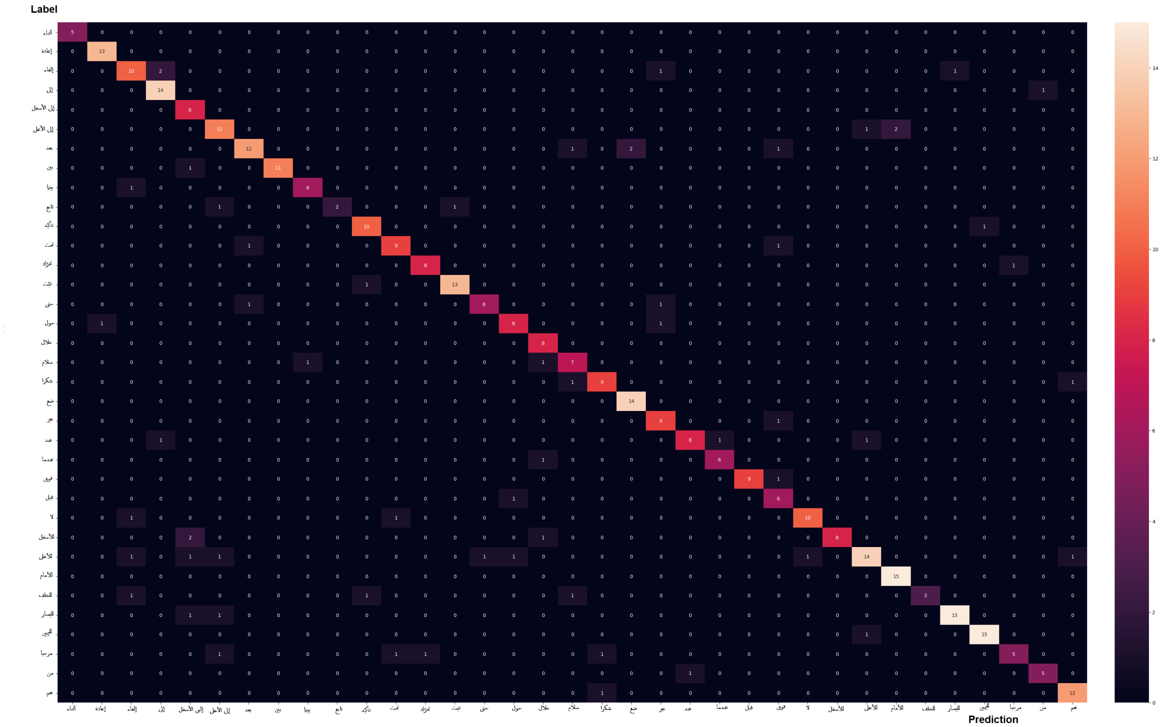


Figure 12. Confusion matrix using Mini-2

model was able to reduce the error during training. The validation accuracy obtained was 84%, suggesting that the model performed well in recognizing speech commands in the validation set. This is an encouraging result, indicating that the model has learned to generalize and make accurate predictions.

On the other hand, the Mini-2 Model, trained on a dataset containing 35 command words, stopped training at 19 epochs. The final validation loss for this model was 0.7279, which is higher compared to the Mini-1 model. However, the validation accuracy achieved was 83%, indicating that the model was still able to perform well in recognizing speech commands despite the higher loss value. This suggests that the model has learned to generalize effectively on a larger vocabulary of command words.

In addition, data augmentation techniques were applied to training data, including pitching from -3 to +3 and noise injection. These techniques helped to increase the variability of the training data, enabling the models to better handle variations in speech patterns and background noise.

It is important to note that the accuracy of the models is directly related to the amount of data collected. Since the precision of both models reached a plateau of 83-84%, it suggests that collecting more data could potentially lead to further improvements in accuracy. A larger and more diverse dataset would provide the models with a broader range of examples to learn from, enabling them to generalize better and make more accurate predictions.

In summary, both the Mini-1 and Mini-2 models achieved satisfactory results in terms of validation accuracy, demonstrating their ability to recognize speech commands. The early stopping mechanism ensured that the models were trained until no significant improvements were observed. The impact of data augmentation techniques was evident in the models' performance, enabling them to handle variations in speech patterns and noise. Collecting more data in the future could potentially enhance the accuracy of the models and further improve their performance in Arabic Speech Commands Recognition tasks.

8. Conclusion

In this manuscript, we have introduced the creation of a novel dataset named ”ArabAlg” designed for Arabic Speech Commands Recognition. The purpose of this dataset is to bolster the integration of Arabic voice recognition systems into smart devices and automated machinery, laying a foundational basis for future investigations in the realm of Arabic Speech Command Recognition. Our study sheds light on the challenges and opportunities inherent in the compilation and utilization of extensive datasets. To demonstrate the efficacy of the ArabAlg dataset, we trained a CNN-based model, achieving commendable results. Through the application of data augmentation techniques, we attained an accuracy of 84%.

However, we did not delve deeply into the optimization of specific speech recognition algorithms or explore the real-time performance of these systems in practical, dynamic settings. Future research can delve into refining and optimizing speech recognition algorithms specifically tailored for the nuances of Arabic speech. This involves exploring advanced machine learning techniques, deep neural networks, or hybrid models to further enhance accuracy and efficiency.

It is noteworthy that, with a larger and more diverse dataset, we anticipate a substantial improvement in the model's accuracy.

Looking ahead, we acknowledge the imperative to continue expanding our dataset to further enhance the performance of ASR systems. This entails ongoing efforts to engage users, refine data collection strategies, and explore additional avenues for data augmentation. By addressing these challenges and building upon our present endeavors, we aim to facilitate the development of more precise and reliable Arabic voice recognition systems in the Internet of Things landscape.

References

[1] S. Haboussi, N. Oukas, T. Zerrouki, and H. Djettou, “Arabic speech recognition using neural networks: An overview,” 2023.

[2] M. S. Haleem, “Voice controlled automation system,” in *2008 IEEE International Multitopic Conference*. IEEE, 2008, pp. 508–512.

[3] N. Oukas, T. Zerrouki, S. Haboussi, and H. Djettou, “Arabic speech recognition using deep learning and common voice dataset,” in *2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*. IEEE, 2022, pp. 642–647.

[4] J. L. K. E. Fendji, D. C. Tala, B. O. Yenke, and M. Atemkeng, “Automatic speech recognition using limited vocabulary: A survey,” *Applied Artificial Intelligence*, vol. 36, no. 1, p. 2095039, 2022.

[5] A. Ghandoura, F. Hjabo, and O. Al Dakkak, “Building and benchmarking an arabic speech commands dataset for small-footprint keyword spotting,” *Engineering Applications of Artificial Intelligence*, vol. 102, p. 104267, 2021.

[6] V. Lyashenko, F. Laariedh, S. Sotnik, and A. M. Ayaz, “Recognition of voice commands based on neural network,” 2021.

[7] N. Oukas, M. Boulif, and K. Arab, “A novel fluid-based modeling approach using extended hybrid petri nets for power consumption monitoring in wireless autonomous iot devices, with energy harvesting capability and triple sleeping strategy,” *Wireless Networks*, pp. 1–24, 2024.

[8] K. S and E. Chandra, “A review on automatic speech recognition architecture and approaches,” *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, pp. 393–404, 04 2016.

[9] J. Vajpai and A. Bora, “Industrial applications of automatic speech recognition systems,” *International Journal of Engineering Research and Applications*, vol. 6, no. 3, pp. 88–95, 2016.

[10] V. Bhardwaj, M. T. Ben Othman, V. Kukreja, Y. Belkhier, M. Bajaj, B. S. Goud, A. U. Rehman, M. Shafiq, and H. Hamam, “Automatic speech recognition (asr) systems for children: A systematic literature review,” *Applied Sciences*, vol. 12, no. 9, p. 4419, 2022.

[11] M. S. Alkatheiri, “Artificial intelligence assisted improved human-computer interactions for computer systems,” *Computers and Electrical Engineering*, vol. 101, p. 107950, 2022.

[12] J. Li, L. Deng, R. H.-Umbach, and Y. Gong, *Robust Automatic Speech Recognition: A Bridge to Practical Applications*. Academic Press, 2015.

[13] E. Al-Wer and R. Jong, “Dialects of arabic,” in *The Handbook of Dialectology*, C. Boberg, J. Nerbonne, and D. Watt, Eds. Wiley, 2017, p. 525.

[14] W. Ghai and N. Singh, “Literature review on automatic speech recognition,” *International Journal of Computer Applications*, vol. 41, no. 8, p. 42, March 2012.

[15] M. Alam, M. D. Samad, L. Vidyaratne, A. Glandon, and K. M. Iftekharuddin, “Survey on deep neural networks in speech and vision systems,” *Neurocomputing*, vol. 417, pp. 302–321, 2020.

[16] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, “Automatic speech recognition for under-resourced languages: A survey,” *Speech Communication*, vol. 56, 2014.

[17] I. R. Titze, *Principles of Voice Production*. Prentice Hall (currently published by NCVS.org), 1994.

[18] L. D. Xuedong Huang, “”an overview of modern speech recognition, in handbook of natural language processing”,” 2010.

[19] J. Savage, C. Rivera, and V. Aguilar, “Isolated word speech recognition using vector quantization techniques and artificial neural networks,” 1991.

[20] R. Aggarwal and M. Dave, “Acoustic modelling problem for automatic speech recognition system: Conventional methods (part i),” *International Journal of Speech Technology*, vol. 14, pp. 297–308, 2011.

[21] K. S. Manoj and K. Omendri, “Speech recognition: A review,” *Special Conference Issue: National Conference on Cloud Computing & Big Data*, 2015.

[22] J. Sorensen and C. Allauzen, “Unary data structures for language models,” in *INTERSPEECH 2011*, 2011.

[23] S. Benkerzaz, Y. Elmir, and A. Dennai, “A study on automatic speech recognition,” *Department of Exact Sciences, University of TAHRI Mohamed, Smart Grid & Renewable Energies Laboratory, Computer Science & Sciences Didactic Team*, 2021.

[24] E. Kumalija and Y. Nakamoto, “Performance evaluation of automatic speech recognition systems on integrated noise-network distorted speech,” *Frontiers in Signal Processing*, vol. 2, p. 999457, 2022.

[25] V. Silber-Varod, I. Siegert, O. Jokish, Y. Sinha, and N. Geri, “A cross-language study of speech recognition systems for english, german, and hebrew,” *Online Journal of Applied Knowledge Management*, vol. 9, no. 1, pp. 1–15, 2021.

[26] B. He and M. Radfar, “The performance evaluation of attention-based neural asr under mixed speech input,” *arXiv preprint arXiv:2108.01245*, 2021.

[27] S. Alharbi, M. Alrazgan, A. Alrashed, T. Alnomasi, R. Almojel, R. Alharbi, S. Alharbi, S. Alturki, F. Alshehri, and M. Almojil, “Automatic speech recognition: Systematic literature review,” *IEEE Access*, vol. 9, pp. 131 858–131 876, 2021.

[28] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, “Common voice: A massively-multilingual speech corpus,” *arXiv preprint arXiv:1912.06670*, 2019.

[29] A. Ali, S. Vogel, and S. Renals, “Speech recognition challenge in the wild: Arabic mgb-3,” in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 316–322.

[30] H. Mubarak, A. Hussein, S. A. Chowdhury, and A. Ali, “Qasr: Qcri aljazeera speech resource–a large scale annotated arabic speech corpus,” *arXiv preprint arXiv:2106.13000*, 2021.

[31] G. B. Pete Warden, “”speech commands: A dataset for limited-vocabulary speech recognition”,” April 2018.

[32] L. Benamer and O. Alkishriwo, “Database for arabic speech commands recognition,” 12 2020.

[33] M. ”yaseen, *“Arabic Speech Commands Dataset.”*, 14 May 2022, https://www.kaggle.com/datasets/murtadhayaseen/arabic-speech-commands-dataset.

[34] ”FutureBeeAI”, ““general conversation speech data in arabic (algeria).”,” https://www.futurebeeai.com/dataset/speech-dataset/general-conversation-arabic-algeria.

[35] N. Zerari, S. Abdelhamid, H. Bouzgou, and C. Raymond, “Bidirectional deep architecture for arabic speech recognition,” *Open Computer Science*, vol. 9, no. 1, pp. 92–102, 2019.

[36] H. S. Hassan, S. J. Harbi *et al.*, “Arabic command based human computer interaction,” in *Journal of Physics: Conference Series*, vol. 1530, no. 1. IOP Publishing, 2020, p. 012027.

[37] M. Lichouri, K. Lounnas, and A. Bakri, “Toward building another arabic voice command dataset for multiple speech processing tasks,” in *2023 International Conference on Advances in Electronics, Control and Communication Systems (ICAECCS)*. IEEE, 2023, pp. 1–5.

[38] O. Mahmoudi and M. F. Bouami, “Rnn and lstm models for arabic speech commands recognition using pytorch and gpu,” in *International Conference on Artificial Intelligence & Industrial Applications*. Springer, 2023, pp. 462–470.

[39] S. K. Ali and Z. M. Mahdi, “Arabic voice system to help illiterate or blind for using computer,” in *Journal of Physics: Conference Series*, vol. 1804, no. 1. IOP Publishing, 2021, p. 012137.

[40] A. Salah, G. Adel, H. Mohamed, Y. Baghdady, and S. M. Moussa, “Towards personalized control of things using arabic voice commands for elderly and with disabilities people,” *International Journal of Information Technology*, pp. 1–22, 2023.

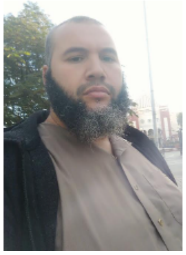
[41] N. Campbell, “Recording and storing of speech data,” *JST/CREST Expressive Speech Processing Project*, 2006.

[42] L. Hamilton. (2023) Legal requirements to collect personal data. TermsFeed. Online article. [Online]. Available: https://www.termsfeed.com/blog/legal-requirements-collect-personal-data/

[43] Y. A. Ibrahim, J. C. Odiketa, and T. S. Ibiyemi, “Preprocessing technique in automatic speech recognition for human computer interaction: An overview,” *Department of Computer Science, Bingham University, Karu, Nigeria*, 2017.

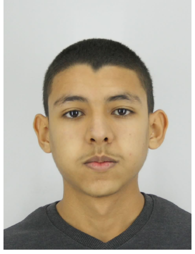
[44] K. O’Shea and R. Nash, “An introduction to convolutional neural networks,” *arXiv preprint arXiv:1511.08458*, 2015.

[45] D. Griffin and J. Lim, “Signal estimation from modified short-time fourier transform,” *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 2, pp. 236–243, 1984.



Nourredine OUKAS is a teacher researcher at Akli Mohand Oulhadj University of Bouira- Algeria. He is a member of LIM laboratory in the same university. He received his Engineering degree in Advanced information systems at the University of Boumerdes- Algeria, and his Magister degree in Mobile Informatics from Houari Boumediene University of Science and Technology (USTHB- Algeria). He received his Doctorate degree in computer sciences at the University of Boumerdes- Algeria. His area of research includes Wireless sensor networks, IoT, Systems Modeling, Systems Optimization, and Natural language processing.

Samia HABOUSSI is a PhD student in computer networks. She obtained her bachelor's degree in computer science at the University of Bouira-Algeria. She earned her master's degree from the same university. Her interests include Arabic Speech Recognition, Internet of Things and System Optimization. Today, she is a member of the LIM Laboratory at the University of Bouira-Algeria.



Chafik MAIZA is a Diplomate Master in Computer Systems Engineering. He obtained his bachelor's degree in computer science at the University of Bouira-Algeria. He obtained his Master's degree from the same university. His interests include natural language processing and software development.

Nassim BENSLIMANE is a Diplomate Master in Computer Systems Engineering. He obtained his bachelor's degree in computer science at the University of Bouira-Algeria. He obtained his Master's degree from the same university. His interests include Arabic speech recognition systems and mobile application development.