République Algérienne Démocratique et Populaire
Ministère De l'Enseignement Supérieur et De La
Recherche Scientifique Université 8Mai 1945 – Guelma

Faculté Des Sciences et de la Technologie

Département D'Electronique et Télécommunications

Thesis of End of Study

To obtain the Academic Master's Degree

Domain: **Sciences and Technologie**

Sector: **Telecommunications**

Speciality: **Network et Telecommunications**

Development of Arabic speech-to-text system for smart device control

presented by: **Redjaimia Zakarya** **Mouad Abu ayech**

under the direction of: **Dr. Abainia Kheireddine**

academic year: 2023/2024

Aknowledgement

We would like to thank God Almighty first and foremost for giving us the patience and energy to finish this work. We sincerely thank Mr. Abainia, who, after agreeing to take on the thesis's direction, frequently let me handle my work while providing the necessary feedback and guidance during its development. We sincerely appreciate his interest in this work, as well as his helpful advice, availability, and empathy. We would also want to thank the jury members for their interest in this study and their willingness to weigh in on its judgement. We would want to express my gratitude to all of my teachers for their support and encouragement, who have supported us and shown us their respects.

Finally, we would want to express our gratitude to everyone who helped make this thesis a reality by lending their support and advice.

Dédicas

To whom I prefer it to myself, and why not She sacrificed for me and spared no effort to make me happy always (my beloved mother). We walk the paths of life, and the one who controls our minds in every path we take remains the one with a good face and good deeds. He (my dear father) did not despise me throughout his life. To my friends, and all those who stood by me and helped me with everything they had, and I especially mention my friends from Ain Saleh, and in many aspects I present to you this research, and I hope that it will satisfy you

Mouad Abu ayech

Dédicas

بسم اللّٰه الرحمان الرحيم

I dedicate this small work

to my mother, who is my inspiration

and my source of strength. to my father,

to whom I owe everything,

discover in me the source of their pride,

to my sisters and brothers,

to my entire family, to my close friends,

to everyone who loves me, to everyone who I hold dear.

Redjaimia Zakaraya

ملخص المذكرة

إن تطوير نظام الكلام إلى النص باللغة العربية للتحكم في الأجهزة الذكية لديه إمكانات هائلة لإحداث ثورة في طريقة تفاعلنا مع التكنولوجيا وتعزيز إمكانية وصول الأفراد. يعالج هذا المشروع الحاجة الأساسية للتواصل السلس بين البشر والأجهزة الذكية، خاصة للمتحدثين باللغة العربية. من خلال دمج المكونات سهلة الاستخدام مثل الميكروفون و Raspberry Pi، يمكن للمستخدمين التفاعل مع النظام دون عناء. يتحدثون ببساطة إلى الجهاز باستخدام ميكروفون لتحويل الكلام إلى نص، وبعد ذلك، يتم استخدام النص للتحكم في الأجهزة الذكية أو أجهزة الكمبيوتر، بالإضافة إلى إمكانية استخدام النص للتنقل على الإنترنت (خاصة في أجهزة التلفزيون الذكية). يعتمد نظامنا على الكمبيوتر المدمج (Raspberry Pi)، ونموذج التعلم العميق المدعوم من TensorFlow لنسخ خطاب الإدخال في الوقت الفعلي.

من خلال الاستفادة من خوارزميات التعرف على الكلام المتقدمة، يقوم النظام بنسخ الكلام العربي إلى نص دقيق. يتغلب هذا النهج المبتكر على الحاجة إلى التفاعلات اليدوية أو الأجهزة عالية التكلفة (مثل التلفزيون الذكي عالي السعة). نركز بشكل خاص على أجهزة الكمبيوتر منخفضة السعة تحت نظام التشغيل Windows، بالإضافة إلى الأجهزة المضمنة مع نظام التشغيل Android (عادةً الهواتف الذكية).

Summary

The development of an Arabic-language speech-to-text system to control smart devices has enormous potential to revolutionise the way we interact with technology and enhance individuals' accessibility. This project addresses the basic need for smooth communication between humans and smart devices, especially for Arabic speakers. By integrating easy-to-use components such as the microphone and the Raspberry Pi, users can interact with the system effortlessly. They simply speak to the device using a microphone to convert the speech to a text, and subsequently, the text is used to control the smart devices or computers, as well as the text may be used to navigate on the internet (especially in smart TVs). Our system is based on the embedded computer (Raspberry Pi), and a deep learning model powered by TensorFlow to transcript the input speech in real time.

By leveraging advanced speech recognition algorithms, the system transcripts Arabic speech into accurate text. This innovative approach overcomes the need for manual interactions or high cost devices (e.g. high capacity smart TV). We particularly focus on low capacity computers under Windows OS, as well as embedded devices with Android OS (typically Smartphones).

Table of contents

Chapter 1. Embedded systems

Chapter 2. Arabic Speech-to-Text (STT)

Chapter 3. Deployment of speech-to-text on an embedded system

Chapter 4. Voice control application

List Of Figures

Chapter 1. Embedded systems

Chapter 2. Arabic Speech-to-Text (STT)

Chapter 3. Deployment of speech-to-text on an embedded system

Chapter 4. Voice control application

List Of Table

General introduction

The demand for competent transcription services at competitive **pricing** from big media powerhouses has led to a thriving **market** in today's **media transcription**. Making the most of media **assets worldwide** has made **audiovisual media** content **transcription an essential** component of the **media industry.** In order to meet their increasing demand for consistently **accurate** and efficient media transcription, an increasing number of media organizations are outsourcing this work.

Services for media transcription provide unparalleled **advantages** in meeting the **transcribing needs of clients.** Since the beginning of the revolution in media transcription outsourcing, numerous businesses have been offering accurate and high-quality transcription services to individual journalists as well as **media organizations**. The Transcription Hub, a cutting-edge provider of transcription services, is committed to the highest standards of quality and has the capacity to produce substantial amounts of precise media transcription work in the appropriate amount of time.

Simply put, media transcription is the process of taking **audiovisual** content and turning spoken words onto paper or a computer screen into **text**. It can also be used to create subtitles for movies or television shows, provide complete transcripts of broadcast or short-form YouTube videos, and produce webinars. **Experts in media transcription services** are aware of the amount of labour required for various facets of **television production**, and we collaborate to guarantee that our **media transcription services** preserve the authenticity of each audible moment.

The television industry is still quite competitive today, and TV management are under increased pressure to provide viewers with high-quality material that is easily accessible to them anywhere in the world due to the competition from digital streaming channels alone. **Personalized post-production transcripts**, time-coded transcripts, **captions**, multi-language translations, and audio transmission are **the components of TV transcription systems**. By customizing their advanced market strategies, their products have assisted their clients in the regional television business in regaining their competitive advantage. Having top-notch TV transcripts will help you succeed in **media production** and outcompete players on the internet. This is due to the fact that TV transcripts offer an easy-to-use point of reference for tasks like video clip editing, sound byte extraction, broadcasting authority clearance submission, etc.

Most of the **information** that may be accessible **online** is available as **podcasts** or **videos**. The **Internet** is evolving away from **text-heavy** content and towards more diverse forms of **communication**, much like the early advancements that **moved** away from **newspapers** and magazines and towards radio and television. Nonetheless, a large portion of the web's search engine **infrastructures** are built around **text** and do not account for content found in

audio or **video files**. Because of this, your content will appear completely bogus in Google or Yahoo's search results, making it difficult for prospective visitors to find. **Subtitles** and **transcripts** can double the rate at which videos are completed. They also typically improve **interactivity**. Given that Google now gives videos and webpages longer viewing times, this suggests the user is engaged and enjoying their experience.

Transcribing media also has **SEO** benefits in addition to being helpful for editorial and transparency. Precise transcription of audio or video enables visitors to locate and explore your digital media content more quickly using search engines by using the verbal content as a guide. This leads to a rise in the amount of people who view your media content.

Initially, closed captions were created to provide individuals who are hard of hearing or deaf with an equal viewing experience. For **the 48 million** Americans who have hearing loss and the **360 million** people worldwide who have hearing loss with impairments, the time-synchronized transcribed texts provide an essential answer. With closed captions, you can give these people access to your video content, let them watch it, and grow your audience at the same time. **Sixty percent** of people who have hearing loss are in the workforce or have a degree. Various anti-discrimination legislation have been put in place to protect the rights of those with disabilities to use the same resources as the general public, irrespective of their circumstances. A handful of those regulations require videos to have closed captions in order for them to be equally accessible. The **FCC** tightly controls closed-captioning requirements for US media and television broadcasts.

The **media** has exploded globally and grown to be a significant aspect of contemporary life everywhere. You may assume that, thanks to **television**, **newspapers**, **magazines**, **radio**, and the **internet**, even the most isolated villages and tiny towns are now connected to the rest of the world. **Films** are a popular kind of entertainment in every home and are undoubtedly a part of daily life. The dialogue in your video needs to be videotaped if you want to reach a larger, more relevant audience. Luckily, media transcriptions are frequently **less expensive** than closed captioning or subtitles, providing a practical choice for **businesses** looking to boost their **online visibility** without breaking the bank.

In order to effectively communicate the material to the audience, longer video segments need to be properly transcribed. Providing a **textual version** of **information** is especially important for short-form **interviews**, **seminars**, **product evaluations**, **keynote speeches**, and **video recordings** since it enables viewers to navigate quickly and easily to the desired place in the recording. **Google** and other search **engines** can more effectively translate and classify the information by adding written language to your photographs and using media **transcription services**. This will assist your **customers** and clients find and compare the information.

1.1 Project Overview:

The project aims to develop a voice-controlled smart TV system using a Raspberry Pi, a microphone, and Arabic speech recognition technology. The system allows users to control their smart TV using voice commands in the Arabic language.

1.2 Objectives:

The main objectives of the project include:

Implementing speech-to-text (STT) transcription functionality to convert Arabic voice commands into text.

Integrating the STT functionality with a Raspberry Pi and a microphone for audio input.

Designing and implementing TV control logic to interpret the transcribed commands and control the smart TV accordingly.

Providing an intuitive and user-friendly interface for users to interact with the system.

1.3 Scope and Limitations:

- **The scope of the project includes:**

Developing the software components required for audio recording, STT transcription, and TV control.

Supporting Arabic language recognition for voice commands.

Implementing basic TV control functions such as volume adjustment, channel selection, and media playback.

The limitations of the project may include:

Limited accuracy of the STT transcription, as speech recognition systems may not always accurately transcribe speech, especially in noisy environments or with certain accents.

The project may not support advanced smart TV features or specific functionalities unique to certain TV models.

The system may not provide support for other languages apart from Arabic.

1.4 Technology Stack Overview:

- **The technology stack for this project may include:**

Raspberry Pi The hardware platform for running the system.

Microphone An external or built-in microphone for audio input.

Python The programming language for implementing the software components.

TensorFlow An open-source machine learning framework that provides tools and libraries for building and training neural networks.

Keras A high-level neural network API that runs on top of TensorFlow and simplifies the process of building and training neural networks.

SpeechRecognition library A Python library for handling speech recognition functionality.

STT Service An STT service such as Google Cloud Speech-to-Text, Microsoft Azure Speech to Text, or Mozilla DeepSpeech for converting speech to text.

Smart TV APIs APIs or libraries provided by the smart TV manufacturer to control the TV functions.

User Interface A graphical user interface (GUI) using frameworks like PyQt or tkinter to provide a user-friendly interaction with the system.

It's important to note that the specific technology stack and tools used may vary depending on the preferences and requirements of the project.

Chapter 1. Embedded systems

1. Introduction

The number of linked devices globally has surpassed 15 billion this year, demonstrating the explosive growth of the Internet of Things in recent years. The core of these electronic gadgets are embedded system boards, which offer the processing capacity, communication, and functionality needed for things like smart security systems and thermometers.

Put another way, these boards make sure that our wearables, smart appliances, and a plethora of other Internet of Things devices work flawlessly, heeding our commands and improving the convenience of our lives.

However, these boards can also cause a great deal of confusion because hardware engineers have access to so many possibilities and a broad range of classifications and components. This is a guide to help you select the appropriate embedded system boards for your Internet of Things project[1].

2.Definition of an Embedded System

Specialized computer hardware, called embedded development boards, are made to do one or more specific functions inside of a larger system. These boards serve as the brains of all Internet of Things (IoT) devices, ranging from sophisticated manufacturing robots to comparatively basic smart doorbells[1].

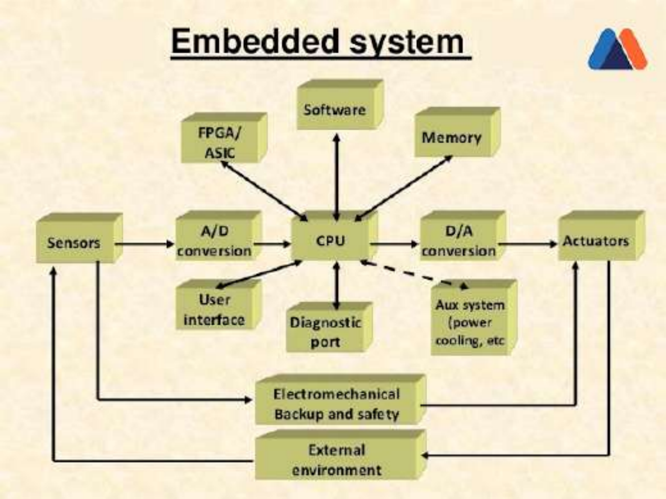


Figure 1.1. diagram of the basic structure and flow of information in embedded systems.

One of these boards resembles a typical circuit board, as you will note if you see one. Despite their unassuming appearance, embedded development boards have a plethora of uses in the Internet of Things due to their compact size, low power consumption, and relatively high processing capability.

3.History

The Apollo Guidance Computer, created in the **1960s** by Dr. Charles Stark Draper at the Massachusetts Institute of Technology for the Apollo Program, was the first contemporary embedded computer system to operate in real time. The purpose of the Apollo Guidance Computer was to automatically gather data and perform computations that were essential to the Apollo Command Module and Lunar Module's missions.

In order to improve embedded system design, the National Engineering Manufacturers Association released a standard for programmable microcontrollers in **1978**. By the early **1980s**, memory, input, and output system components had been integrated into the same chip as the processor, forming a microcontroller. Intel released the Intel **4004**, the first commercially available microprocessor unit, in **1971**.

Every element of customers' daily life, from traffic lights and thermostats to credit card readers and cell phones, would eventually have an embedded microcontroller system[1].

4.Embedded system technologies

Embedded system technologies refer to the combination of hardware and software components specifically designed for dedicated functions within a larger system. These technologies are typically used in various applications and industries where real-time control, reliability, and efficiency are crucial[2].

4.1. Different electronic boards

In embedded systems, a variety of electronic boards are frequently utilised.
Here are a few instances:

Raspberry Pi

A well-liked option for embedded systems is the Raspberry Pi. It is a single-board computer that comes in several varieties, each with unique features. Numerous applications, such as robots, IoT projects, and home automation, use Raspberry Pi boards.

Arduino

When developing embedded systems, Arduino boards are frequently used. They provide a straightforward and adaptable framework for developing and constructing a variety of projects, and they are based on microcontrollers. Arduino boards are widely used in automation, control systems, and sensor interface applications.

BeagleBone

BeagleBone boards combine an embedded Linux system with a microprocessor, making them strong embedded platforms. They are appropriate for applications demanding more processing power, like robotics and industrial control systems, and they provide a wide range of networking possibilities.

Intel Edison

Designed for Internet of Things applications, the Intel Edison is a small and potent development board. It has many I/O interfaces, Bluetooth and Wi-Fi connectivity, and a dual-core Intel Atom processor. Intel Edison boards are frequently used for small-space projects and Internet of Things prototyping.

STM32 Discovery

STM32 microcontrollers from STMicroelectronics serve as the foundation for STM32 Discovery boards. They are frequently employed in the development of embedded systems, particularly in industrial settings. These boards are an affordable option for low-power and high-performance projects, and they come with a large variety of peripherals.

Nvidia Jetson

Boards made especially for AI and computer vision applications are called Jetson boards. They have strong GPUs and can execute sophisticated machine learning algorithms. Smart surveillance systems, robots, and self-driving cars all use jetson boards.
Texas Instruments LaunchPad: Texas Instruments' LaunchPad development boards are well-liked for their ability to work with a variety of microcontrollers,
such as the MSP430 and Tiva C series. They are frequently used for development and prototyping in consumer electronics, industrial automation, and Internet of Things projects and provide a variety of functionalities.

These are but a few illustrations of the typical circuit boards seen in embedded systems. The project's particular needs, including those related to processing power, connectivity, I/O interfaces, and cost, will determine which board is best.

4.2. Different operating systems

An operating system serves as a conduit between the hardware and the user. It is sometimes referred to as a resource manager because it is a program that aids in the use of system hardware. Based on various system designs, there are various kinds of operating systems, each with unique characteristics and applications. with its intuitive interface, Windows is a widely used operating system for personal computers. Mac OS, which runs on Apple computers, is renowned for its powerful performance and elegant appearance. Open-source and preferred by developers for its security and flexibility is Linux. Most mobile devices run Android or iOS, which are designed with touchscreens and mobile apps in mind. Because each OS has its own advantages, it can be used for a variety of purposes and tastes. Let us talk about each in turn.

1. **Microsoft Windows**

Microsoft Disc Operating System, or MS-DOS for short, is a non-graphical command-line operating system designed for IBM-compatible computers that use x86 microprocessors. The user could navigate, open, and modify files on their computer by entering commands through the operating system's command line interface.

2. Windows OS

Microsoft created Windows as an operating system to run on common x86 Intel and AMD CPUs. It has a graphical user interface (GUI), which allows users to browse menus, dialogue boxes, buttons, tabs, and icons using a mouse, doing away with the need to learn command line syntax. Because the program are presented in a square shape, the operating system was given the name Windows. Both professionals who work in development and inexperienced users who use it at home will find this Windows operating system to be well-suited.

3. The Linux operating system

A freely available, cross-platform operating system built on the UNIX foundation, the Linux OS is an open source project. Linus Torvalds is the developer of this operating system. The Linux kernel is where the name Linux originates. In essence, a computer's system software is what enables users and apps to carry out particular tasks on the device. The Linux operating system, which became a symbol of software collaboration, led the way in the development of open source software.

4. The operating system Solaris

The Unix derivative operating system from Sun, also known as SunOS, was first created for Intel-based CPUs and the company's series of Scalable Processor Architecture-based processors (SPARC). This operating system had, at the time, mainly controlled the UNIX workstation market. Sun's Solaris systems became the most extensively used servers for websites as the Internet evolved. After acquiring Sun, Oracle changed its name to Oracle Solaris.

5. Android operating system

Android is an operating system developed by Google that runs on Linux and is mostly intended for touch-screen mobile devices, like smartphones and tablets. The three architectures on which the hardware supporting Android is based—ARM, Intel, and MIPS—allow users to manipulate mobile devices in an intuitive way by mimicking everyday gestures like pinching, swiping, and tapping, which makes using these applications comfortable for them.

6. IOS Mobile Operating System

The mobile operating system known as iOS, or iPhone OS, was designed and created by Apple Inc. specifically for its hardware, such as the A12 Bionic chip, which currently powers a number of its mobile devices, including the iPhone, iPad, and iPod. Use multi-touch movements like swipe, tap, pinch, and reverse pinch is the foundation of the iOS user interface. These finger movements are intended to enable the user quick, responsive inputs to the multi-touch capacitive screen display from several fingers.

7. Mac OS

Apple Inc. created the proprietary macOS operating system, which is based on Unix. It serves as the main operating system for Mac laptops and PCs made by Apple. Originally released as Mac OS X in 2001, the name was changed to macOS in 2016.

4.3. Different sensors used in embedded systems

Our world is filled with sensors. Many kinds of sensors are used in our homes, workplaces, cars, and other places to help us live easier. Examples of these tasks include opening garage doors when our car is close to the door, detecting smoke or fire, detecting our presence and turning on the lights, among many other things [2].

Sensors

Sensors enable all of these automated activities as well as many more. We will first look at a basic example of an automated system, which is made possible by sensors (as well as many other components), before delving into the specifics of what a sensor is, what kinds of sensors there are, and the applications of these many sorts of sensors.

Although the term "sensor" has many definitions, I would want to define it as an input device that produces an output (signal) in relation to a certain physical quantity (input).

In the definition of a sensor, the term "input device" refers to a component of a larger system that supplies input to the main control system (such as a processor or microcontroller).

The following is an additional, original definition of a sensor: It is a gadget that changes the energy domain of signals into the electrical domain. By considering an example, we may gain a better understanding of the definition of the Sensor.

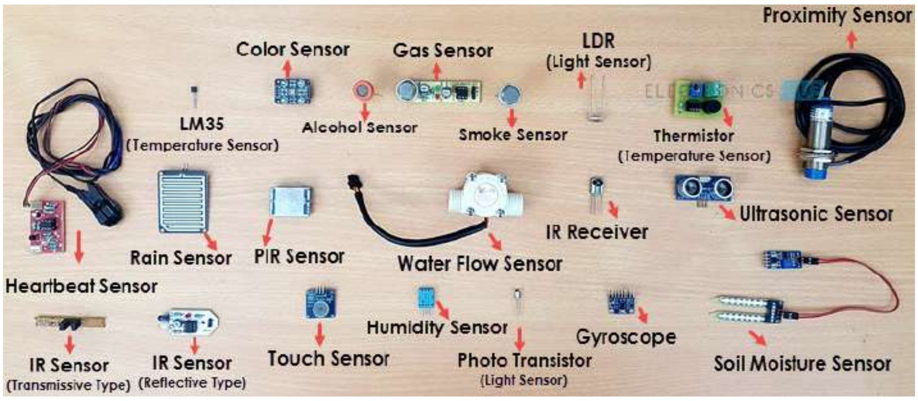


Figure 1.2. kind of Sensors

An LDR, or light dependent resistor, is the most basic type of sensor. This apparatus exhibits variability in resistance as a function of the light intensity it is exposed to. An LDR's resistance decreases dramatically with increasing light intensity, and increases dramatically with decreasing light intensity.

Along with another resistor, we can connect this LDR to a voltage divider to measure the voltage drop across the LDR. It is possible to adjust this voltage based on how much light hits the LDR. Thus, the Light Sensor.

Having established our understanding of what a sensor is, we can move forward with the classification of sensors.

Classification of Sensors

Various authors and professionals have classified sensors into multiple categories. While some are extremely intricate, others are rather simple. An expert in the field may already be using the following classification of sensors, however it is a fairly basic one.

The sensors are separated into two categories in the first classification: active and passive. Active sensors are ones that need a power signal or an external stimulation signal to function.

Conversely, passive sensors produce output response directly and do not need an additional power source.

The sensor's method of detection provides the basis for the other kind of classification. Biological, chemical, radioactive, and electric are a few types of detection methods.

The input and output, or conversion phenomena, form the basis of the following classification. thermoelectric, photoelectric, electrochemical, electromagnetic, thermoptic, and other conversion phenomena are a few of the frequently occurring ones.

Analogue and digital sensors make up the last grouping of the sensors. In relation to the quantity being measured, analogue sensors generate an analogue output, which is a continuous output signal (often voltage, but occasionally other values like resistance, etc.).

Digital sensors use discrete or digital data, as opposed to analogue sensors. Digital sensors use data that is digital in nature for conversion and transmission.

Different Types of Sensors

The numerous kinds of sensors that are frequently used in diverse applications are listed below, along with some examples. Any of the physical attributes, such as temperature, resistance, capacitance, conduction, heat transfer, etc., can be measured using any of these sensors.

Type of Sensor	**Used For**
Temperature Sensor	Controlling HVAC systems in homes and offices
Proximity Sensor	Detecting objects in automatic doors
Accelerometer Sensor	Screen orientation in smartphones
IR Sensor (Infrared Sensor)	Remote controls for TVs and other devices

Pressure Sensor	Monitoring tire pressure in vehicles
Light Sensor	Adjusting screen brightness on smartphones
Ultrasonic Sensor	Parking assistance in cars
Flow and Level Sensor	Managing water levels in tanks
Smoke, Gas and Alcohol Sensor	Detecting smoke and gas leaks in homes
Microphone (Sound Sensor)	Voice recognition in smart speakers
Touch Sensor	Touchscreens on smartphones and tablets
Color Sensor	Color detection in industrial sorting machines
Humidity Sensor	Controlling humidity levels in greenhouses
Magnetic Sensor (Hall Effect Sensor)	Detecting the position of a rotating object
Position Sensor	Tracking the position of machine parts
Tilt Sensor	Detecting the tilt of gaming controllers
PIR Sensor	Motion detection in security systems

Strain and Weight Sensor	Weighing items on digital scales
Gyroscope Sensor	Stabilizing drones during flight
Optical Sensor	Adjusting lighting in smart home systems
Capacitive Sensor	Touchpads on laptops
Piezoelectric Sensor	Detecting vibrations in musical instruments
Thermal Sensor	Temperature control in ovens
RFID Sensor	Tracking inventory in warehouses
Chemical Sensor	Monitoring air quality

Table 1.1. 25 Different Types Of Sensors And Their Uses

5. Embedded system applications

A number of technologies, such as machine-to-machine (M2M) devices and the internet of things (IoT), depend heavily on embedded systems. These days, almost all smart devices make use of this adaptable technology in one way or another [4].

Examples of embedded system uses in the real world:

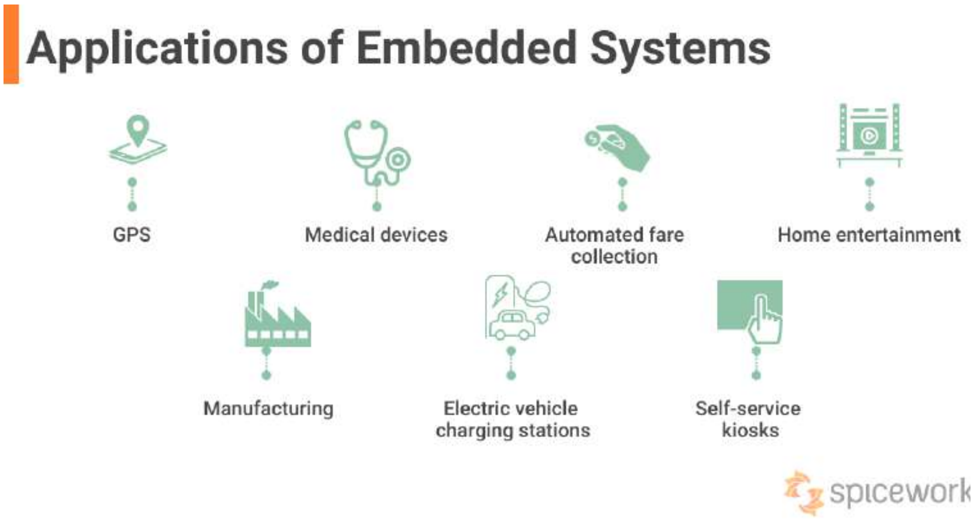

Applications of Embedded Systems
GPS
Medical devices
Automated fare collection
Home entertainment
Manufacturing
Electric vehicle charging stations
Self-service kiosks
spicework

Figure 1.3. Application of Embedded Systems

5.1. Automative applications

The purpose of automotive embedded systems is to improve vehicle safety through design and installation. The number of traffic fatalities has drastically decreased in recent years due to safety features installed in cars. The automotive industry is going above and beyond to equip cars with cutting-edge sensors and systems, which would not be feasible without embedded systems.

Figure 1.4. Automative applications

Adaptive speed control, auto breakdown warning, pedestrian recognition, merging assistance, airbags, and more are a few prominent instances of active safety systems. These are some of the qualities that are expected to reduce the likelihood of accidents and increase demand for embedded systems worldwide[4].

A few instances of embedded automotive systems

- Navigation system for cars
- Vehicle entertainment system
- anti-lock braking system

5.2. Industrial applications

The **GPS** is a navigation system that synchronize the location, time, and velocity data using satellites and receivers. To make the use of a global positioning system easier, the receiver or device that receives the data has an integrated embedded system. People may simply find their destinations and present positions thanks to the incorporated GPS gadgets. As a result, they are quickly gathering steam and overtaking other navigation devices in cars in terms of usage.

Figure I.5. GPS Devices

These days, GPS devices are typically used in

- Automobiles
- Mobile gadgets
- Handheld

Factory robots

made to carry out precise duties in hazardous environments. To link various subsystems, they have an integrated embedded system. Robots use actuators, sensors, and software in a mechanical work to sense its surroundings and safely produce the desired result [4].

Figure I.6. Factory robots

Robots would need to rely on external control or computer systems if they did not have embedded systems. Consequently, there may be increased safety risks as a result of a lag or broken connection between the manufacturing robot and its external computer system. In order to make machinery smarter, safer, and more efficient, these systems are incorporating artificial intelligence and machine learning as Industry 4.0 takes shape. For instance, this allows robots to recognize flaws that a human eye would not notice and take them out of production.

Factory robots are used in many different contexts

- assembly line
- quality assurance
- Painting, welding, and panelling

Automated teller machine

ATM is a computerized device used for banking that uses a network to connect to a host bank computer. While the embedded system in the ATM displays the transaction data and processes inputs from the ATM keypad, the bank computer checks all the data submitted by users and saves all transactions.

Figure I.7. Automated teller machine

- Take out cash
- verify account balance and transaction history
- Transfer funds to a different account.

Through ticket vending machines or online services, users can pay for their tickets using the Automated Fare Collection (AFC) ticketing system. Originally using money and tokens, these systems are now using smart cards or magnetic stripe cards in their place. A ticket vending machine, an automatic gate machine, and a ticket checking machine make up an AFC, a basic station equipment. These parts are embedded systems that guarantee smoother operations, quicker transactions, and more effective revenue collection.

Urban transport systems have embraced AFC with smart cards, which are low-cost technologies that offer additional protection along with data collection opportunities, although city transit buses and commuter trains still use paper tickets and passes.

Typically, automated fare collection systems may be found at

- metro areas
- stations for buses
- stations for trains

Charging points or units that provide electric power to charge linked automobiles are a feature of electric vehicle charging stations. The charging station has an inbuilt system that processes visual displays, reports device problems, and notifies technicians when maintenance is needed. This embedded solution offers a simple and economical method for keeping an eye on and repairing the infrastructure related to charging. Several **Digi** clients, including **AddEnergie**, are creating products to cater to this expanding market.

The following are a few typical applications for electric vehicle charging stations

Figure 1.8. electric vehicle charging stations

- ➢ automobile charging
- ➢ changing the batteries
- ➢ Parking automobiles

Telecommunications

- ➢ Network infrastructure equipment (e.g., routers, switches, base stations)
- ➢ Wireless sensor networks for monitoring and control
- ➢ Telecom service management and provisioning systems

5.3. Domestic applications

Home Appliances

Refrigerators (temperature and energy management, diagnostics)
Washing machines (cycle control, water and energy optimization)
Ovens and microwaves (cooking time and temperature control)
Air conditioners and HVAC systems (temperature and humidity control)

Home Entertainment

Smart TVs (user interface, content streaming, internet connectivity)
Gaming consoles (game processing, graphics rendering, user input)
Home theater systems (audio/video processing, remote control)
Home Automation and Security:

- ➢ Smart home controllers (lighting, shades, door locks, alarms)
- ➢ Security systems (surveillance cameras, motion sensors, access control)
- ➢ Home energy management (smart meters, thermostat control, load balancing)

Personal Electronics

Smartphones and tablets (mobile processors, sensors, connectivity)
Wearable devices (fitness trackers, smartwatches, AR/VR headsets)
Digital cameras and camcorders (image processing, storage, connectivity)

Home Assistance and Robotics

Voice assistants (speech recognition, natural language processing)
Robotic vacuums and lawn mowers (navigation, obstacle avoidance)
Smart home hubs (device integration, voice control, automation)

Household Appliances

Dishwashers (cycle control, water usage optimization)
Coffee makers (temperature regulation, automated brewing)
Blenders and food processors (speed control, safety features)

Personal Care

Electric toothbrushes (brushing time and pressure control)

Hair dryers and straighteners (temperature regulation, safety features)
Electric shavers and trimmers (motor control, battery management)

Central Heating Systems

In a furnace room, central heating systems transform chemical energy into thermal energy. This energy is then transferred into heat and distributed throughout a structure. In order to regulate the temperature, these systems must have thermostat controls, which are accomplished by an embedded system [4].

Figure I.9. Central Heating Systems

Without temperature controls, a central heating system may cause one room to get overheated while another remains cool. You may save a significant amount of energy and customize the temperature to your liking with the correct thermostat controls.

A variety of buildings that need temperature control for comfort and the management of items that are sensitive to temperature can be found that include embedded systems for central heating.

These domestic applications of embedded systems focus on improving convenience, efficiency, and user experience in the home environment. As smart home technologies continue to evolve, the integration of embedded systems in domestic appliances and devices is becoming increasingly prevalent.

5.4. E-care and well-being applications

Medical Devices

Figure I.10. Medical Devices

Healthcare facilities have been using embedded systems in their medical gadgets for a while now. Embedded systems are being used in a new class of medical devices to help treat patients who require continuous care and regular monitoring at home. These systems include sensors built in to collect health-related data from patients, such as heart rate, pulse rate, or implant readings. The data is then transferred to a cloud so that a physician can examine patient information wirelessly on their smartphone.
Many medical gadgets have been used to effectively diagnose and treat patients.

- Ultrasound scanners
- defibrillators
- Implantable medical devices (e.g., pacemakers, insulin pumps)
- Diagnostic and monitoring equipment (e.g., MRI, CT scanners, ECG)

- Rehabilitation and assistive technologies

Fitness trackers are wearable gadgets that keep tabs on your physical activity, including walking, jogging, and sleeping. These gadgets collect information about your body temperature, heart rate, and steps taken using embedded systems. This data is then transmitted to servers via WAN technologies like GPRS or LTE.

Typically, fitness trackers are used for:

- keeping an eye on one's own activities
- medical observation
- Exercise for sports

Figure I.11. fitness trackers

6. Conclusion

Embedded systems are becoming prevalent across the world, They are small, fast, and powerful computers used in many devices and equipment we use daily.
They guarantee the performance of real-time applications.
They are responsible for the completion of a task within a specified time limit, such as rapid graphics processing and artificial intelligence processing.
Additionally, embedded modules are becoming more sophisticated and powerful all the time, and are increasing in graphics performance and edge compute capabilities, giving embedded developers the tools to bring high-performance market-driven products to market.

Chapter 2. Arabic Speech-to-Text (STT)

1. Introduction

Arabic Speech-to-Text (STT) technology transforms spoken Arabic into written text, hence revolutionizing our interaction with the language. Arabic STT systems analyse and transcribe spoken words and phrases using automatic speech recognition (ASR) algorithms, facilitating accurate and efficient communication.

Arabic STT systems use cutting-edge acoustic and language modelling techniques to decode the phonetic units and linguistic patterns found in spoken Arabic. To produce the most likely text transcription, this multi-step technique preprocesses the audio input, applies acoustic modelling appropriate to a given language, and uses statistical language models.

Technological developments in machine learning and natural language processing have been extremely beneficial to the development of Arabic STT systems. These technological advancements improve accuracy and performance by allowing the system to adjust to different Arabic dialects, accents, and speech circumstances.

Arabic STT is used in many different fields, including voice assistants, contact centre, language instruction, transcription services, and accessibility services for those with hearing loss. It increases output, makes communication easier, and creates new opportunities for using spoken Arabic in digital contexts.

We may anticipate more improvements in accuracy, real-time transcription capabilities, and support for many Arabic language variations as research and development in Arabic STT continues. This will facilitate smooth communication in both personal and professional contexts by helping to close the gap between spoken and written Arabic [5].

2. Speech recognition systems

Voice technology is becoming a part of our daily life. We receive information, navigate, convert our voice to text, and even give voice assistants and our cars useful commands by using speech recognition and voice technologies.

Voice and speech recognition technology are being integrated by businesses into their marketing, workplace, and consumer-facing products.

As a result of this expansion, proponents of voice and speech technology, marketers, and end users have combined terms to refer to these innovations under one umbrella. The two technologies, however, produce different results and employ distinct procedures.

The simplest description of how voice and speech recognition differs is as follows:

Speech recognition software interprets any voice
Voice recognition recognises the voice of a particular user.
As more companies explore for methods to use voice and speech recognition technology to enhance operations, collaboration, and growth, it is critical to comprehend these technologies.

2.1. Definitions

Speech recognition is the ability of a computer or machine to interpret spoken words and translate them into text. It is also known as automatic speech recognition (ASR), computer speech recognition, or voice-to-text. Speech recognition is a type of artificial intelligence. Speech recognition software converts spoken words into written language or computer commands. It is sometimes mistaken for voice recognition, which recognizes the speaker rather than what they are saying.

2.2. General process

Every gadget, including computers and phones, includes an integrated microphone that records audio signals and speech samples. Next, the speech-to-text technology decodes the audio, eliminates any unwanted noise, and modifies the speech's pitch, loudness, and cadence. After that, it breaks down the digital data into frequencies and examines individual content segments.

Software for speech recognition begins to analyse human speech after it has processed the recording. The program generates mathematical representations of various phonemes, or basic units of sound, that distinguish one word from another and make assumptions about the speaker's meaning based on speech context. Acoustic modelling is a key component of contemporary speech recognition systems.

The recording is then written out in legible type by the software, which then creates word sequences that best fit the input speech signal. After the transcription has been identified, the user can review it again, fix any errors, and improve its correctness.

Even while the speech recognition technique seems straightforward, the software is quite intricate, incorporating machine learning, signal processing, and natural language processing. Furthermore, the technology processes information far more quickly than a human can. Nevertheless, the system application, language complexity, and original recording quality could all affect how accurate the output is [5].

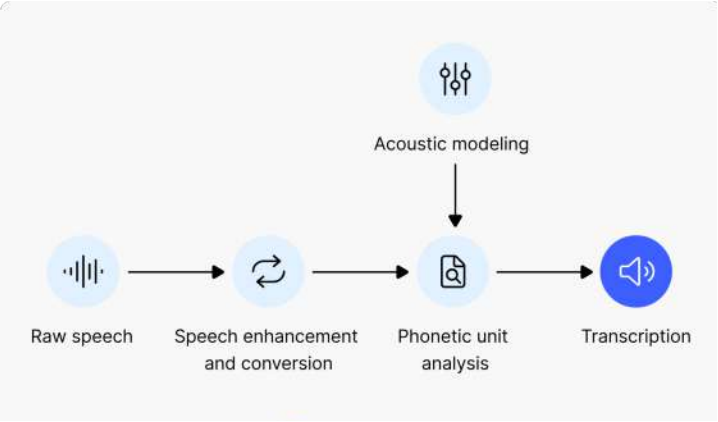


Figure 2.1. Automate Speech Recognition

Speech recognition algorithms explained

In a hybrid method, various speech recognition algorithms and computation approaches help translate spoken words into text and guarantee output correctness. The three primary algorithms that guarantee the transcript's accuracy are as follows:

Markov hidden model (HMM). An algorithm called HMM manages speech variation, including accent, speed, and pronunciation. It offers a straightforward and efficient framework for simulating the temporal structure of voice and audio signals as well as the phoneme sequence that constitutes a word. This is why an HMM is the foundation of the majority of speech recognition systems in use today.
Warping time dynamically (DTW). When comparing two distinct speech sequences with varying speeds, DTW is used. Take two audio recordings of someone saying "good morning"—one recorded slowly, the other quickly. In this instance, despite the two recordings' differences in speed and duration, the DTW algorithm is nevertheless able to sync them.
Artificial Neural Networks (ANN). ANN is a computational model that aids computers in comprehending spoken human language and is utilised in voice recognition applications. The machine is able to make decisions that resemble those of a human because it mimics the

patterns of how neural networks function in the human brain through the use of deep learning techniques.

Use cases of speech recognition

Speech recognition is a quickly developing technology that is used in many different industries. It enhances automated operations, which saves time and makes life more convenient for people. The following are a few typical applications for speech recognition:

- **Navigation systems**

With the aid of speech recognition software, which is frequently included in navigation systems, drivers can voice commands to automobile electronics like car radios while maintaining their hands and eyes on the wheel.

- **Virtual assistants**

Voice-activated personal assistants are becoming more and more indispensable in our day-to-day existence. Personal assistants on mobile devices, such as Siri or Google Assistant, can aid you in finding the information you need or carrying out certain tasks on your phone thanks to the speech-to-text functionality. The way your Microsoft Cortana or Amazon Alexa functions is the same; it understands your request, responds to your inquiries, or plays your preferred music.

- **Healthcare**

Accuracy and speed are crucial in the medical industry, where automatic voice recognition is employed. Using this technology, medical professionals can translate spoken words into text for use in clinical notes, medical reports, and electronic health record updates. Additionally, speech recognition software enhances clinical documentation, including treatment plans and diagnosis accuracy.

- **Call centers**

Client assistance Speech recognition software is frequently used by contact centre to automate client interactions. By analysing speech input and responding to client demands, the algorithms free up human agents' time to handle more complex problems.

- **Accessibility**

People with disabilities may find it easier to use technology and the internet using speech-to-text processing. Voice search can be used by people with restricted movement to use their gadgets, including taking phone calls and surfing the web.

- **Language translation**

Speech recognition software is also used by machine translation software to translate human speech between languages.

- **Voice search**

Search engines also include speech recognition software, which enables voice commands for web browsing.

Artificial intelligence in the form of speech recognition helps automate procedures and enhance accuracy and efficiency in a variety of vocations as well as daily life. In the meanwhile, it keeps developing, and this technology will probably be used even more widely.

2.3. Preprocessing

The processing of a speech signal involves several steps, the most significant of which is speech preprocessing. This phase includes filtering, amplification, noise suppression, speech activity detection, normalisation, and speech enhancement. Speech signal processing is a field of active research nowadays. Studies in the field of voice recognition have demonstrated that the effectiveness of the speech signal preprocessing has a significant impact on the quality of speech recognition[6]. In order to provide well-prepared data for completing additional speech recognition tasks, we have gathered the various speech signal preprocessing approaches in this work. These processing techniques include Voice Activity Detection, Noise Reduction, Pre-emphasis, Framing, Windowing, and Normalisation.

The primary goal of the speech signal preprocessing is to prepare the signal for the feature extraction speech recognition stage. These sequential stages are used to preprocess the voice signal. The VAD is used in the first stage to retain only the speech signal's voice components. Subsequently, the voice signal's pre-emphasis is applied to decrease noise and enable the signal's frequency to be balanced. Following this, the entire signal is framed.

The speech stream's signal spectrum and signal-to-noise ratio (SNR) are then enhanced by using a windowing function and normalising the vocal tract. The sections following go into detail about each preprocessing stage.

A. Voice activity detection (VAD)

VAD is a crucial method for identifying the voiced portions of speech during speech signal preprocessing[8]. Speech activity detection is used in ASR to identify the beginning and end of speech utterances, as well as to maximise CPU utilisation and fulfil hardware limitations.

VAD is applied by first splitting the voice signal into small frames and subsequently identifying whether or not a frame contains speech. The foundation of the VAD algorithms is the feature selection process, which aims to capture the distinguishing characteristics between noise and speech. It is possible to separate two types of features: time-domain features, which include Zero-Crossing Rate (ZCR), Short-Term Energy (STE), and Short-Time Average Magnitude (STAM), are among the most popular features because of their ease of usage. However, surrounding noise can diminish these features.

Spectral Flatness (SF) and Spectral Power (SP), two frequency-domain properties, make up the second category. Other elements, including the Spectro-Temporal Modulation (STM) and Amplitude Modulation Spectrogram (AMS), mimic human perception to identify speech.

B. Noise removal

Noise in the surrounding environment is a significant factor in speech preprocessing denoising a voice signal increases speech quality and strengthens the resilience of speech recognition systems. The method of eliminating noise from a mixed sound of speech and noise so that only clear speech remains is known as noise removal[7].

C. Pri-emphasis

Pre-emphasis is used to balance the high and low frequency components of a voice signal and flatten the magnitude spectrum since the frequency components of a speech signal are high frequency. Conversely, pre-emphasis filtering lowers the speech waveform's high dynamic range while increasing the signal-to-noise ratio.

In voice recognition processing, the pre-emphasis preprocessing approach is less frequently utilised. Nevertheless, this preprocessing method raises the speech signal's energy at high frequencies, which has an impact on the final spectrum's consistency between frames[8].

D. Framing

The speech signal's characteristics are not static throughout time due to its non-stationary nature, however during brief periods of time, it is regarded as a stationary signal that is simple to process[13]. In order to enable block processing of the speech signal, the Framing stage closes this gap by dividing the continuous speech signal into a sequence of blocks called frames that are 20–40 ms long[10][9]. A key method in speech signal processing is framing;

the resulting frames are of uniform length and remain fixed across time, making it much simpler to extract the speech signal's valuable features.

E. Windowing

The analysis of speech signals in each frame is known as windowing, and it involves multiplying the waveform of the speech signal by a temporal window function in order to highlight certain characteristics of the speech signal[10].[9][11].

By setting the signal at the start and finish of each frame to zero, the windowing concept helps to smooth the signal and prevent signal discontinuity caused by the spectral distortion during the framing stage.

Nevertheless, there are situations when applying windowing techniques affects how the signal is shown at the start and finish of the frame. There are two primary effects of multiplying the voice wave by the window function. First, the amplitude is gradually decreased at both ends of the time period to prevent a sudden change at the frame's endpoints. The voice spectrum and the window function's Fourier transform are convolved as the second effect[11]. High-frequency resolution and little spectral leakage are hence requirements that must be met when selecting the window function to be used to lessen spectral distortion. It is advised to employ a brief window for ASR analysis, with a frame length of 20 to 30 ms and an overlap of 5 to 10 ms [10].

F. Normalisation

The last stage of preprocessing a speech signal is called normalisation, and it entails balancing the signal spectrum and converting the signal data into a normalised form based on a threshold [16]. The primary purposes of normalisation are to lessen the effects of noise, channel distortion, and voice signal distortion [12]. SNR is increased and signal spectrum variance is either normalised or eliminated during the normalisation phase. There are several different normalisation techniques [13][12], and in the sections that follow, we go over each one's characteristics.

These preprocessing techniques are typically applied before the audio is fed into the acoustic modelling stage of the Arabic STT system. By improving the quality of the audio signal, preprocessing can enhance the accuracy and robustness of the subsequent speech recognition process, resulting in more accurate transcriptions

2.4. Different technologies

Transcription, the process of converting spoken language into written text, involves a combination of technologies and algorithms.

- **Automatic Speech Recognition (ASR)**

ASR systems use algorithms to convert spoken language into written text. They analyse audio signals and apply techniques such as acoustic modelling, language modelling, and signal processing to recognize and transcribe the speech accurately.

Hidden Markov Models (HMM): HMMs are statistical models widely employed in speech recognition. They represent the probabilistic relationships between speech sounds and help in decoding and aligning phonemes to generate word-level transcriptions [5].

- **Deep Neural Networks (DNN)**

DNNs have been instrumental in improving speech recognition accuracy. They are used for various tasks, including acoustic modelling, language modelling, and end-to-end speech recognition systems. DNN architectures like convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been applied to improve transcription accuracy.

- **Long Short-Term Memory (LSTM)**

LSTMs are a type of RNN architecture commonly used in speech recognition. They are designed to capture long-term dependencies in sequential data, making them suitable for modelling speech and language patterns.

- **Connectionist Temporal Classification (CTC)**

CTC is a technique used in sequence learning tasks like speech recognition. It allows the model to learn an alignment between input audio frames and output transcriptions without requiring explicit alignment information.

- **Transformer Models**

Transformer models, such as the popular "Attention Is All You Need" architecture, have been successful in various natural language processing tasks, including transcription. They employ self-attention mechanisms to capture global dependencies and have achieved state-of-the-art performance in speech recognition.

- **Language Modeling**

Language models, including n-gram models and neural language models, play a crucial role in transcription. They help in predicting and correcting errors, improving the fluency and accuracy of transcriptions.

- **Post-processing Techniques**

Transcription systems often employ post-processing techniques to refine the output. These can include language-specific rules, grammar models, punctuation and capitalization correction, contextual analysis, and spell checking.

The field of transcription is continually evolving, and new technologies and algorithms are being developed to enhance accuracy and efficiency.

3. Speech transcription

Spoken words, numbers, or acronyms can be swiftly converted to text using speech recognition and transcription. While there are various uses for speech recognition and transcription, the most common ones are in the fields of healthcare documentation, legal document preparation or logs of court proceedings, and video transcript production for educational and entertainment purposes[14].

Both voice recognition and transcription have advantages and disadvantages, even though they may be utilised to get comparable outcomes in any of these applications. Another option is a hybrid one in which text identified by speech recognition is further edited by a human.

Pros and Cons of Transcription

When considering transcription versus speech recognition, it is essential to start by examining the benefits and disadvantages of transcription. Transcriptionists listen to speeches, then write or record what they hear in written or printed format. Before the appearance of speech recognition, transcriptionists or scribes were experts in voice transcription. Now that speech recognition can rival (and, in many cases, surpass) transcription, it is gradually being discarded. Many industries and organisations prefer or continue to use transcription because of its benefits, while others have evolved because of its disadvantages.

The "human" aspect of transcription can be both an advantage and a disadvantage. A transcriptionist's job is to translate human language into text. An individual can easily recognise words through different accents and refer to the context of a sentence to fill inaudible voids, or simply mark a totally inaudible speech as "inaudible.".

Although AI can perform some of these actions through speech recognition, humans can still surpass automatic speech recognition systems in terms of accuracy. The disadvantage of the "human" aspect of transcription is speed. The job requires fast typists, but even the fastest typists in the industry would not have the ability to compete with speech recognition automation, which can recognise and develop complete sentences in one step rather than having to write all the words.

One drawback for transcribers is that their line of work necessitates a somewhat specialised skill set. A transcriptionist needs to be proficient in any subject they are transcribing in addition to having rapid typing skills in order to properly convert speech to text. For example, a transcriptionist lacking medical expertise will find it impossible to keep up with a doctor as they dictate because medical language is typically difficult for a non-medical person to spell or understand.

This holds true for any field: even those with the necessary typing abilities, which are already rare, would find it difficult to transition to a new transcription industry without sufficient training.
Software that recognises speech can transcribing far more quickly than a person can. Speech recognition only recognises words and produces them quickly, whereas transcriptionists must manually transcribe words after they are recognised. As a result, each word recognition is faster[14].

Accessible to a large audience, speech-to-text technology is complemented by speech recognition programmes tailored to particular industries, such legal or medical speech recognition. There is also free voice recognition software available. Free automatic voice recognition software, however, is more constrained.

In addition to accessibility, voice recognition has many other benefits, such as increased productivity and cost savings, but it also has drawbacks. Transcription and speech recognition are rivals in terms of speed and accuracy. The foundation of every voice recognition system is accuracy.

Though accuracy in speech recognition has improved significantly, consider speech recognition as a competition between computers and humans to identify human speech more accurately. Although neither group can achieve 100% accuracy consistently, transcriptionists can still exceed voice recognition in terms of accuracy. When transcription and voice recognition are directly compared, speech recognition is nearly always quicker than human transcription.

The programme must identify and evaluate the context of each word in a sentence, as well as the topic matter for more specialised speech-to-text, in order to increase the accuracy of speech recognition. Machine learning and artificial intelligence have the potential to improve accuracy even more.

The usability of speech recognition software may also be a constraint. Speech recognition should never get in the way of a more productive workflow; rather, it should be utilised to transcribe speech more quickly and affordably.

3.1. Sound denoising

is the practice of eliminating sounds from a speech without compromising its quality.

Any undesired audio segments for human hearing, such as wind noise, car horn sounds, or even static noise, are considered noises in this context [5].

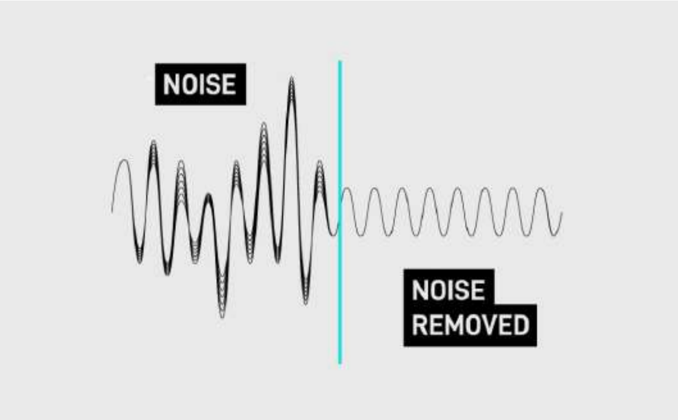


Figure 2.2. Remove noise

Since it improves voice quality, it is also known as speech enhancement. Enhancing speech is a crucial activity that is utilised as a preprocessing step in many applications, including speaker detection, automatic speech recognition (ASR), audio/video calls, and hearing aids. In the remaining sections of this post, we will look at how to eliminate noise from an audio stream[15].

3.2. Sound enhancement

The technique of enhancing audio signals' quality, clarity, and comprehensibility is known as audio augmentation. For audio recordings to be free of undesired noise, distortions, or artefacts, denoising a critical component of audio processing is required. It contributes to the improvement of audio content's fidelity and audibility, making it more enjoyable to listen to and enhancing the precision of any further analysis or applications.

3.3. Transcription process

Modern ASR systems use a range of models and algorithms in order to generate fast and precise results[16].

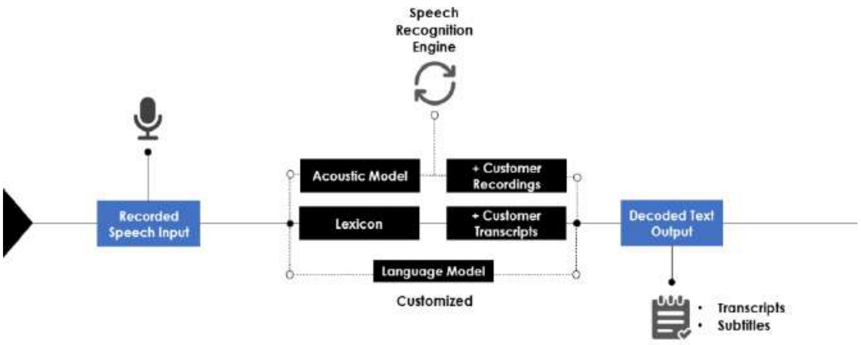


Figure 2.3. Speech to text process generic model

The process of **transcription** is intricate, involving several steps and collaborative AI models. The essential speech-to-text steps:

- **preliminary processing**

The input audio frequently goes through a few pre-processing stages before it can be transcribed. This can involve methods to improve the audio signal's quality, such as echo cancellation and noise reduction.

- **Feature extraction**

After that, the audio waveform is transformed into a format that is better suited for analysis. Typically, this entails taking specific elements of the audio signal—like frequency, loudness, and duration—and extracting them in order to capture significant aspects of the sound. In speech processing, mel-frequency cepstral coefficients, or MFCCs, are often utilised characteristics.

- **Acoustic modelling**

entails building a statistical model through training that associates the variables that have been retrieved with phonemes—the smallest units of sound in a language.

- **Language modelling**

The linguistic component of speech is the main focus of language modelling. It entails building a probabilistic model of the likelihood that words and phrases will occur in a given language. In light of the words that have come before them in the sentence, this aids the system in determining which words are most likely to occur.

- **Decoding**

During the decoding stage, the system converts the audio into a string of words or tokens by using the linguistic and acoustic models. Finding the most likely word order to match the provided audio attributes is the process at hand.

- **Post-processing**

Errors like misidentifications and homophones—words that sound the same but have distinct meanings—may still be present in the decoded transcription. Before generating the final output, post-processing techniques such as contextual analysis, grammatical rules, and language constraints are used to enhance the accuracy and coherence of the transcription.

- **Output**:

The final output of the speech-to-text process is a textual representation of the spoken content. This can be in the form of plain text, formatted text, or structured data, depending on the requirements of the application.

4. Conclusion

The accuracy of speech recognition systems can vary based on factors such as the quality of the audio, language complexity, speaker variability, and the specific algorithms and models used. Ongoing research and advances in machine learning, deep learning, and natural language processing continue to improve the performance of speech recognition systems.

Chapter 3. Deployment of speech-to-text on an embedded system

1. Introduction

Implementing a speech recognition system that can translate spoken language into written text while working within the constraints of the embedded hardware is necessary to deploy speech-to-text on an embedded system. The system requirements, model optimisation, method selection, and real-time processing all need to be carefully taken into account during this procedure. Real-time and accurate speech recognition on resource-constrained embedded systems can be achieved by incorporating power-saving strategies, monitoring memory utilisation, and optimising algorithms for efficiency. Data gathering, training, feature extraction, integration, and iterative optimisation are frequently included in the deployment process. All things considered, implementing speech-to-text on an embedded system allows devices and apps to offer speech-based input and interaction while running small and effectively.

2. Target embedded board (Raspberry Pi)

Deploying speech-to-text on a Raspberry Pi refers to the process of implementing a speech recognition system on a Raspberry Pi embedded board that can convert spoken language into written text. It involves configuring the Raspberry Pi hardware, setting up the necessary software environment, selecting suitable speech recognition algorithms, optimizing models for efficient performance, and enabling real-time processing of audio input. The goal is to leverage the capabilities of the Raspberry Pi to enable voice-based input and interaction on resource-constrained embedded systems.

2.1. Brief history

Origins and the history of the company

After noticing a decrease in the quantity and calibre of young students applying to computer science degrees, staff at the University of Cambridge Computer Laboratory founded the Raspberry Pi Foundation in 2008 as a charity[8] and a private business limited by guarantee [9].[10]

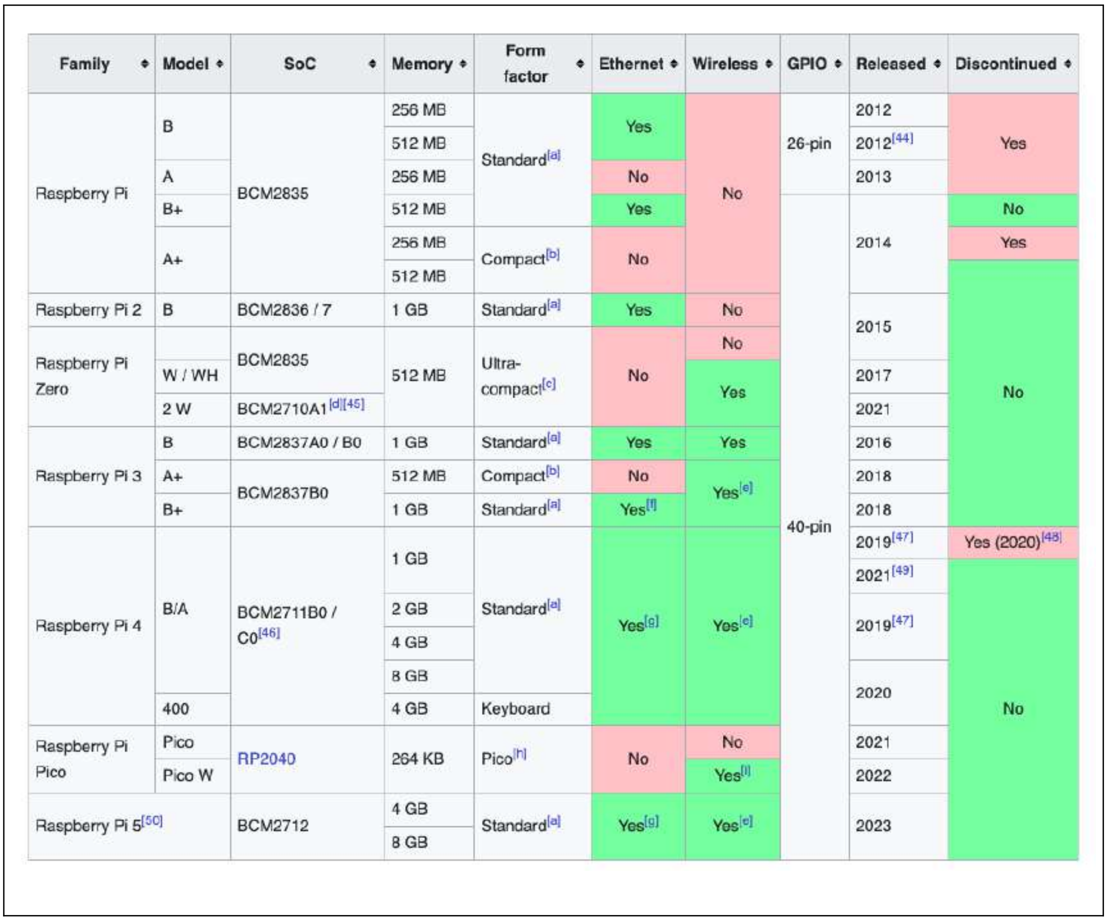
Following the launch of the second board type in 2012, the Raspberry Pi Foundation established Raspberry Pi (Trading) Ltd.,[11] a new company tasked with developing their computers, and appointed 2008 group member Eben Upton as CEO.[12] The Foundation was reestablished with the goal of advancing the teaching of fundamental computer science in both developed and developing nations.

Raspberry Pi (Trading) Ltd became Raspberry Pi Ltd in 2021.[11][13] In June 2024, it went public and debuted on the London Stock Exchange, where it is currently traded under the ticker code RPI.[14][15][16][17]

While some Raspberry Pis are created in China and Japan, the majority are produced in a Sony plant in Pencoed, Wales [18].[19][20]

The Raspberry Pi comes in three series, with multiple generations of each available. While Raspberry Pi Pico has an RP2040 system on chip with an integrated ARM-compatible central processing unit (CPU), Raspberry Pi SBCs have a Broadcom system on a chip (SoC) with an integrated ARM-compatible CPU and on-board graphics processing unit (GPU).

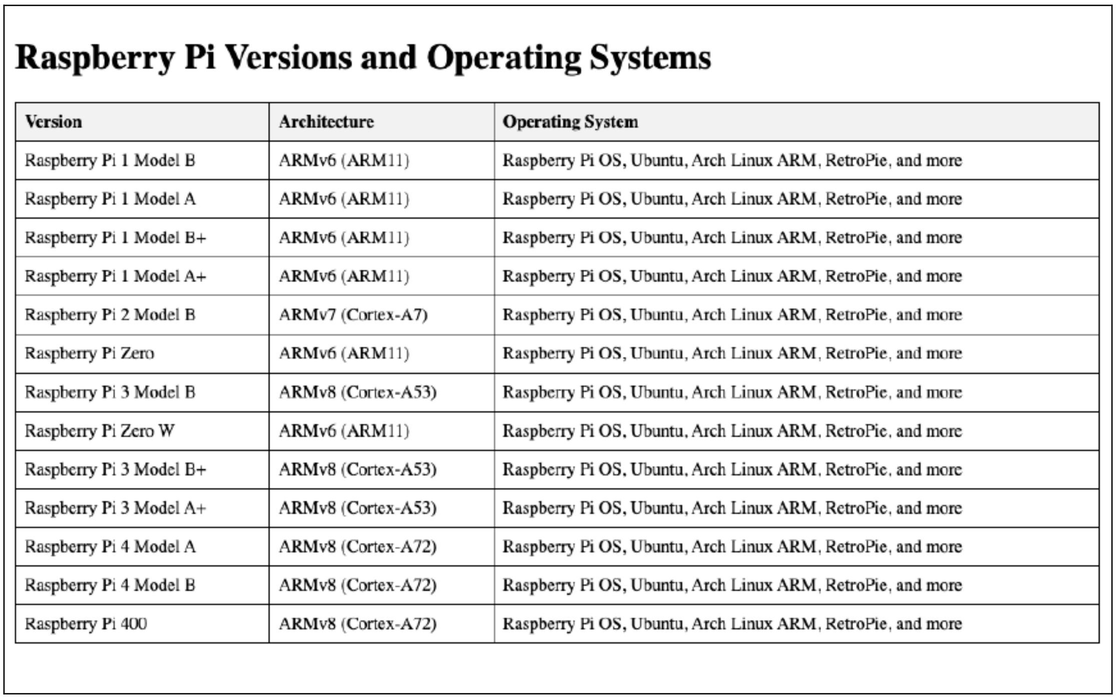
2.2. Different versions



Family	Model	SoC	Memory	Form factor	Ethernet	Wireless	GPIO	Released	Discontinued
Raspberry Pi	B	BCM2835	256 MB	Standard[a]	Yes	No	26-pin	2012	Yes
			512 MB					2012[44]	
	A		256 MB		No			2013	
	B+		512 MB		Yes		40-pin	2014	No
	A+		256 MB	Compact[b]	No				Yes
			512 MB						No
Raspberry Pi 2	B	BCM2836 / 7	1 GB	Standard[a]	Yes	No		2015	
Raspberry Pi Zero		BCM2835	512 MB	Ultra-compact[c]	No	No			
	W / WH					Yes		2017	
	2 W	BCM2710A1[d][45]						2021	
Raspberry Pi 3	B	BCM2837A0 / B0	1 GB	Standard[a]	Yes	Yes		2016	
	A+	BCM2837B0	512 MB	Compact[b]	No	Yes[e]		2018	
	B+		1 GB	Standard[a]	Yes[f]			2018	
Raspberry Pi 4	B/A	BCM2711B0 / C0[46]	1 GB	Standard[a]	Yes[g]	Yes[e]		2019[47]	Yes (2020)[48]
								2021[49]	No
			2 GB					2019[47]	
			4 GB						
			8 GB					2020	
	400		4 GB	Keyboard					
Raspberry Pi Pico	Pico	RP2040	264 KB	Pico[h]	No	No		2021	
	Pico W					Yes[i]		2022	
Raspberry Pi 5[50]		BCM2712	4 GB	Standard[a]	Yes[g]	Yes[e]		2023	
			8 GB						

Table 1.1. Different versions Of Raspberry-Pi

2.3. Target version architecture and operating system



Raspberry Pi Versions and Operating Systems

Version	Architecture	Operating System
Raspberry Pi 1 Model B	ARMv6 (ARM11)	Raspberry Pi OS, Ubuntu, Arch Linux ARM, RetroPie, and more
Raspberry Pi 1 Model A	ARMv6 (ARM11)	Raspberry Pi OS, Ubuntu, Arch Linux ARM, RetroPie, and more
Raspberry Pi 1 Model B+	ARMv6 (ARM11)	Raspberry Pi OS, Ubuntu, Arch Linux ARM, RetroPie, and more
Raspberry Pi 1 Model A+	ARMv6 (ARM11)	Raspberry Pi OS, Ubuntu, Arch Linux ARM, RetroPie, and more
Raspberry Pi 2 Model B	ARMv7 (Cortex-A7)	Raspberry Pi OS, Ubuntu, Arch Linux ARM, RetroPie, and more
Raspberry Pi Zero	ARMv6 (ARM11)	Raspberry Pi OS, Ubuntu, Arch Linux ARM, RetroPie, and more
Raspberry Pi 3 Model B	ARMv8 (Cortex-A53)	Raspberry Pi OS, Ubuntu, Arch Linux ARM, RetroPie, and more
Raspberry Pi Zero W	ARMv6 (ARM11)	Raspberry Pi OS, Ubuntu, Arch Linux ARM, RetroPie, and more
Raspberry Pi 3 Model B+	ARMv8 (Cortex-A53)	Raspberry Pi OS, Ubuntu, Arch Linux ARM, RetroPie, and more
Raspberry Pi 3 Model A+	ARMv8 (Cortex-A53)	Raspberry Pi OS, Ubuntu, Arch Linux ARM, RetroPie, and more
Raspberry Pi 4 Model A	ARMv8 (Cortex-A72)	Raspberry Pi OS, Ubuntu, Arch Linux ARM, RetroPie, and more
Raspberry Pi 4 Model B	ARMv8 (Cortex-A72)	Raspberry Pi OS, Ubuntu, Arch Linux ARM, RetroPie, and more
Raspberry Pi 400	ARMv8 (Cortex-A72)	Raspberry Pi OS, Ubuntu, Arch Linux ARM, RetroPie, and more

Table 1.2. Version architecture and operating system

- ➢ We use as target version Raspberry-Pi **4 Model B**

2.4. External sensors and configuration

Using a Raspberry Pi and external sensors:

A wide range of external sensors can be interfaced with by the Raspberry Pi, a multipurpose single-board computer. The following are a few typical sensors used with the Raspberry Pi:

GPIO Sensors: Basic analogue sensors, switches, buttons, LEDs, and other types of sensors may all be connected to and read data from using the Raspberry Pi's General Purpose Input/Output (GPIO) ports.

I2C Sensors: The Raspberry Pi may be equipped with a variety of sensors, including accelerometers, gyroscopes, temperature and humidity sensors, and more, thanks to the widely used Inter-Integrated Circuit (I2C) protocol.

SPI Sensors: The Raspberry Pi can be connected to sensors, displays, ADCs, and other digital peripherals via the Serial Peripheral Interface (SPI) communication protocol.

USB Sensors: A lot of sensors include a USB interface that can be attached straight to the USB ports on the Raspberry Pi.

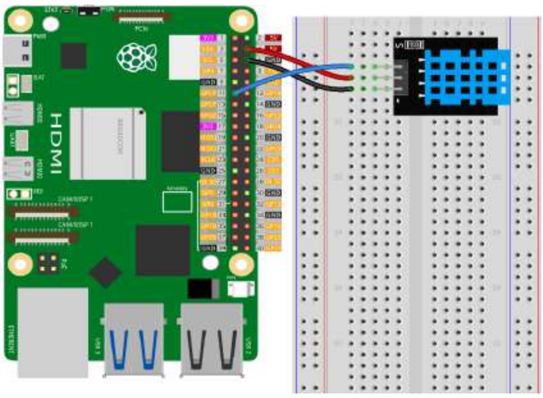


Figure 3.1. GPIO used by sensors

Generally, to use external sensors with a Raspberry Pi:

- **Attach the Sensor**: Attach the sensor, if necessary, to the GPIO, I2C, SPI, or USB ports on the Raspberry Pi.

- **Put Required Libraries in Place**: Installing particular Python libraries or device drivers may be necessary in order to interface with the sensor. A few well-known libraries are spidev, smbus, and Rpi, GPIO.

- **Compose the Control Code.** Write code in Python (or another programming language) that reads sensor data and carries out any necessary actions or processing.

3. Speech transcription

The process of translating audio into writing is called transcription. Although voicemails are the primary application for it these days, call recordings and meetings are also using it[17].

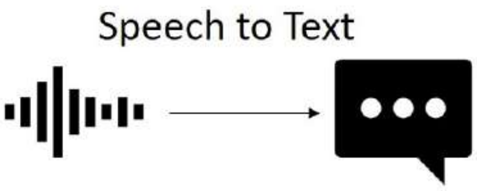


Figure 3.2. Transcription

In actuality, the process of turning speech to text is rather intricate. Consider the variety of languages, accents, and dialects spoken throughout the world. WAV, MPEG3, and OPUS are just a few of the several audio formats available for digitally encoded and compressed audio. This voice cannot be successfully decoded into text without complex algorithms. This is the reason speech to text transcription is frequently delegated to other servers that are mostly used for voicemail and call recording transcription, rather than being done in real-time. But for longer voicemails, the delay is usually measured in seconds, and in no case more than a minute[17].

Thanks to technological advancements, transcribing is now highly dependable. Artificial Intelligence (AI) is widely used by algorithms to learn and then gradually improve the algorithms over time.

3.1. Implemented pipeline

Identify the project/application: Determine the purpose of your project. For example, let's consider a project that monitors environmental conditions in a greenhouse.

Select the sensors: Choose the appropriate sensors based on the requirements of your project. In this case, we might select a temperature and humidity sensor (DHT11/DHT22) and a light sensor (LDR).

Connect the sensors: Follow the wiring instructions provided by the sensor manufacturer. Connect the temperature and humidity sensor to GPIO pins for digital communication, and connect the light sensor to an ADC module for analog-to-digital conversion.

Set up the Raspberry Pi: Install the necessary operating system (e.g., Raspberry Pi OS), libraries, and dependencies on the Raspberry Pi.

Install sensor-specific libraries: Install the required libraries for each sensor. For example, install the Adafruit DHT library for the temperature and humidity sensor and the MCP3008 library for the ADC module.

Write the code: Use a programming language like Python to write the code that will read data from the sensors.

3.1.1. Preprocessing

- **The initial method takes in the following parameters**

sample_rate: The sample rate of the audio file.

frame_length: The length of the frame used for the short-time Fourier transform (STFT).

frame_step: The step size between consecutive frames in the STFT.

num_mels: The number of Mel-frequency bins to use.

The preprocess method takes an audio file path as input and returns

the log-scaled Mel-frequency spectrogram.

It uses the librosa.load function to load the audio file.

It calculates the Mel-frequency spectrogram using the librosa.feature.melspectrogram function.

It converts the spectrogram to log-scale using the librosa.power to db function.

This preprocessor can be used as part of a speech transcription model, where the spectrogram is used as the input to the model. The model can then be trained to predict the transcription of the audio input.

3.1.2. Transcription process

- ***The speech transcription process works***

The speech transcription Model class inherits from the **OnnxInferenceModel** class, which provides a standard way to load and use ONNX models.

The preprocess method takes the input spectrogram and pads it to match the model's expected input shape.

The predict method runs the ONNX model on the preprocessed spectrogram and returns the predicted transcription using the ctc_decoder function from mltu.utils.text_utils.

The evaluate_transcription function takes the input audio file path, the reference text, and the speech transcription model, and calculates the Character Error Rate (CER) and Word Error Rate (WER) between the predicted transcription and the reference text.

In the example usage, the model is loaded from the saved ONNX model file, and the transcription is evaluated on a test audio file.

The CER and WER metrics provide a way to quantify the performance of the speech transcription model, which can be used to fine-tune the model or compare it to other models.

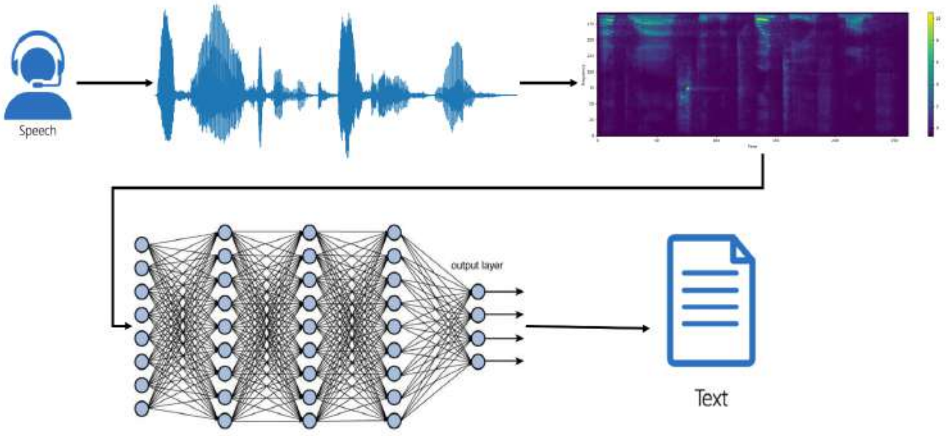


Figure 3.3. General transcription process

3.1.3. Involved libraries

TensorFlow and Keras are popular libraries in the field of deep learning and neural networks.

- **Keras**

A high-level neural networks API, written in Python and capable of running on top of TensorFlow, designed to enable fast experimentation with deep neural networks.

- **Librosa**

A Python library for audio and music analysis, providing tools for loading, processing, and analyzing audio data, as well as extracting various audio features.

- **Matplotlib**

A comprehensive library for creating static, animated, and interactive visualizations in Python, used for data analysis, scientific computing, and machine learning projects.

- **NumPy**

A Python library that provides support for large, multi-dimensional arrays and matrices, as well as a collection of high-level mathematical functions to operate on these arrays.

- **ONNX**

An open standard format for representing machine learning models, allowing models trained with different frameworks to be converted and deployed on a variety of platforms, devices, and cloud runtimes.

- **Pandas**

A Python library that provides high-performance, easy-to-use data structures and data analysis tools, especially for tabular data.

- **TensorFlow**

An open-source machine learning library for numerical computation and large-scale data processing, widely used for building and deploying deep learning models.

3.2. Embedding the program within the Raspberry

The Raspberry Pi-based smart TV control system is a compact, cost-effective, and versatile solution that enables Arabic-speaking users to control their smart TVs using natural language voice commands. By embedding the system directly on the Raspberry Pi hardware platform, it provides a seamlessly integrated and compatible control unit that can be easily deployed across a range of smart TV brands and configurations. The system leverages the Raspberry Pi's computing power and the availability of a robust ecosystem of add-on components to deliver an intuitive and accessible user experience, addressing the challenges faced by certain user groups, such as the elderly or individuals with physical disabilities, in operating complex smart TV interfaces.

3.2.1. Training the model

Voice recognition is the process of converting spoken language into written text. The goal is to train a CNN model to identify and distinguish different voices, enabling the system to recognize and understand human speech accurately [6]. The application of CNN on audio data offers promising results, as it can learn and extract meaningful features from spectrograms and other audio representations.

- **Data Preprocessing:**

  Start by collecting a labeled dataset of audio samples containing various voices. Convert the audio files into suitable formats and preprocess the data to ensure uniformity in audio length and sampling rates.

- **Audio Representation:**

  CNN operates on visual data, so we need to convert audio into a visual representation that CNN can process. Commonly used representations include Mel-frequency cepstral coefficients (MFCCs) and spectrograms. These representations capture the frequency and time-domain information, crucial for voice recognition.

- **Data Splitting:**

Split the preprocessed data into training, validation, and testing sets. This ensures that the model learns from one set, generalizes on another, and evaluates its performance on a separate set.

- **CNN Model Architecture:**

Design the CNN model architecture for audio processing. The model should consist of convolutional layers, pooling layers, and fully connected layers. Experiment with different configurations to optimize performance.

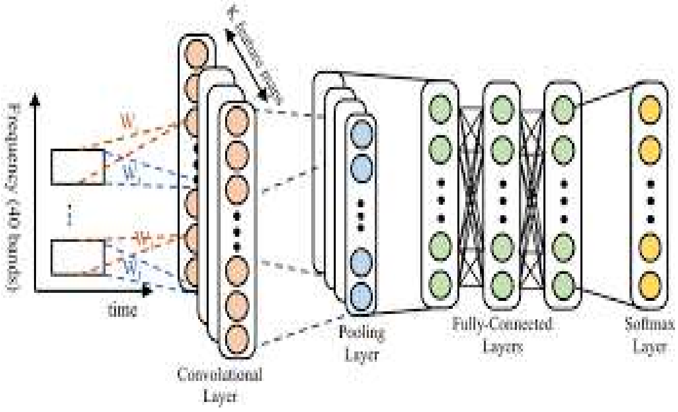


Figure 3.4. CNN architecture for speech recognition [19]

- **Training the Model:**

Train the CNN model using the training dataset and tune hyperparameters such as learning rate, batch size, and number of epochs. Monitor the model's performance on the validation set and make adjustments as needed.

- **Evaluation:**

Once the model is trained, evaluate its performance on the testing set. Analyze metrics like accuracy, precision, and recall to assess how well the model recognizes different voices.

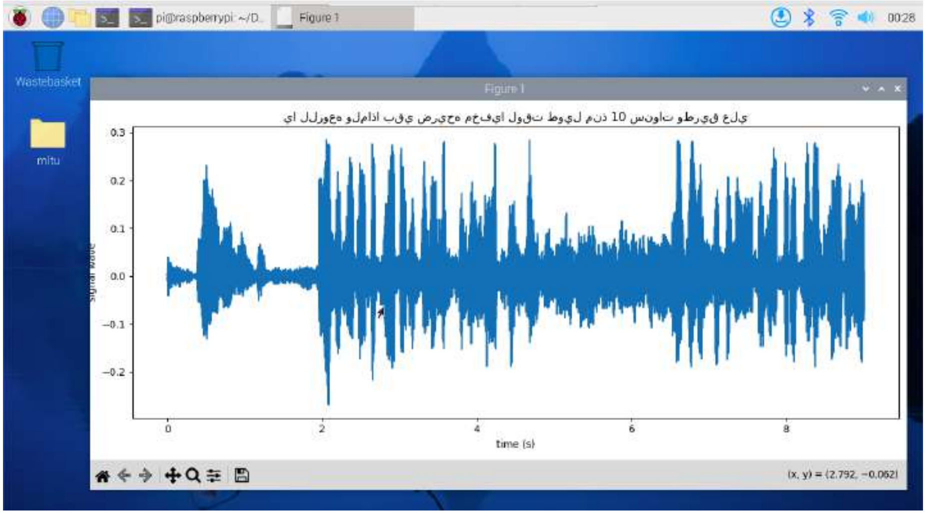
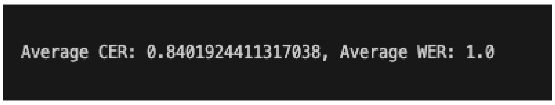


Figure 3.5. iterating through dataset metadata and preprocessing the 'wav' audio data with actual transcription



```
Average CER: 0.8401924411317038, Average WER: 1.0
```

1.00 means that there is no correct word, but it does go down while training

There is no such thing as ideal size. It depends on your dataset quality and complexity. If it stays at 1.00 try to expand dataset or improve model architecture.

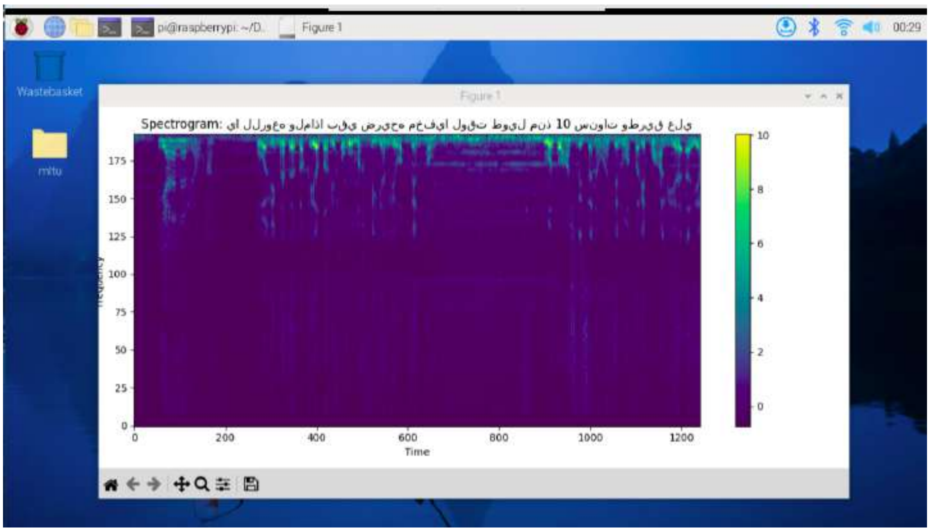


Figure 3.6. Example of audio spectrogram analysis and processing

3.2.2. Porting the model to Raspberry

The model contain this important file that use to predict speech after transform into text that what calls “Transcription”

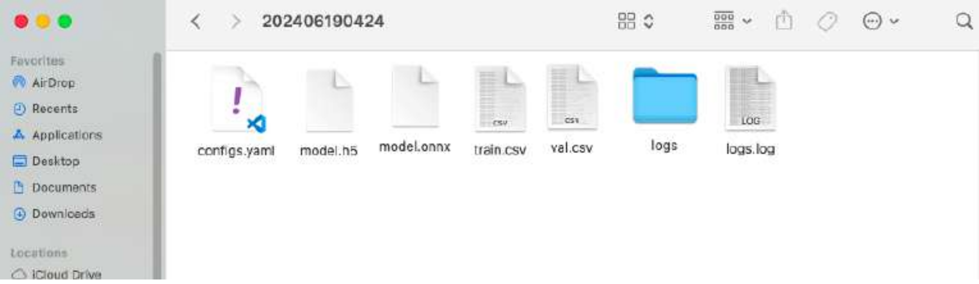


Figure 3.7. Trained Model

- **Using Python programming language predicting voice to text**
- **Using Librosa Numpy, Pandas “mltu”, ctc_decoder, get_cer, get_wer**

3.2.3. Configuring the raspberry for audio streaming

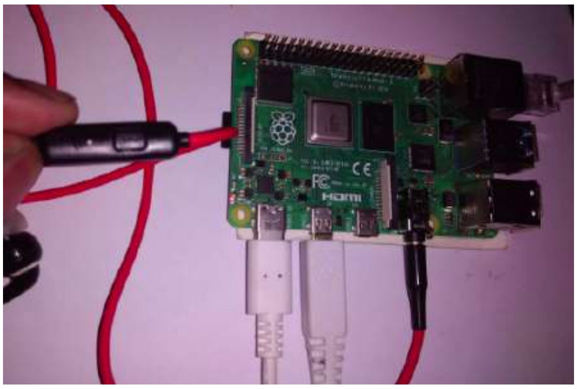
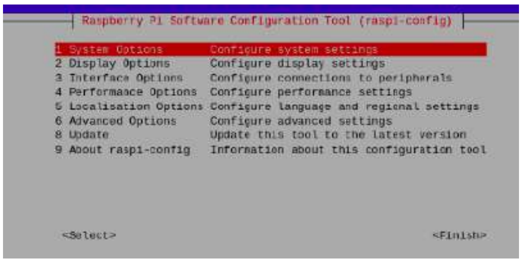


Figure 3.8. Using mic with raspberr

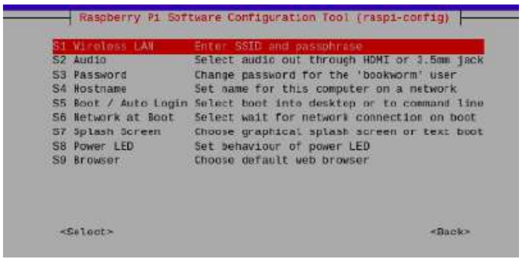
- *Use Command* $ **sudo raspi-config**



```
        | Raspberry Pi Software Configuration Tool (raspi-config) |

1 System Options        Configure system settings
2 Display Options       Configure display settings
3 Interface Options     Configure connections to peripherals
4 Performace Options    Configure performance settings
5 Localisation Options  Configure language and regional settings
6 Advanced Options      Configure advanced settings
8 Update                Update this tool to the latest version
9 About raspi-config    Information about this configuration tool


 <Select>                                                  <Finish>
```



```
Raspberry Pi Software Configuration Tool (raspi-config)

S1 Wireless LAN        Enter SSID and passphrase
S2 Audio               Select audio out through HDMI or 3.5mm jack
S3 Password            Change password for the 'bookworm' user
S4 Hostname            Set name for this computer on a network
S5 Boot / Auto Login   Select boot into desktop or to command line
S6 Network at Boot     Select wait for network connection on boot
S7 Splash Screen       Choose graphical splash screen or text boot
S8 Power LED           Set behaviour of power LED
S9 Browser             Choose default web browser

<Select>                                                <Back>
```

Figure 3.9. configure Raspberry-Pi

- ▰ ***Audio*** Specify the audio output destination.
- ■ ***Install Audio Software*** $ sudo apt-get install pulseaudio
- ▰ ***Config PulseAudio***

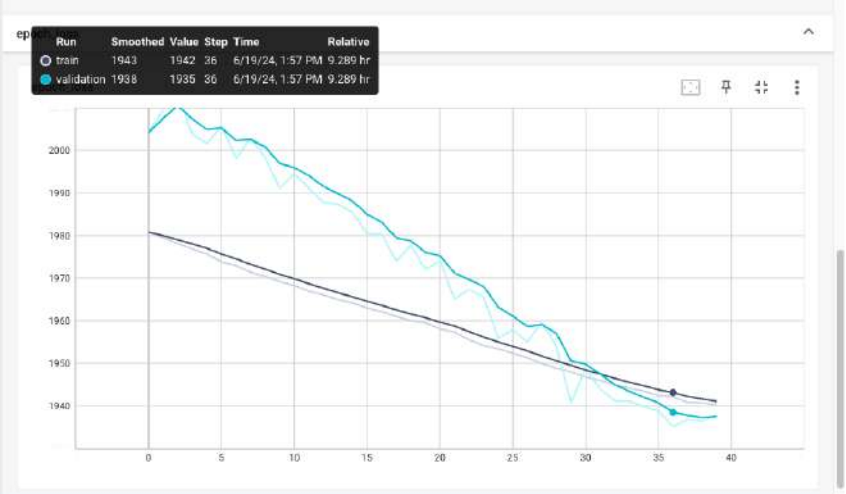
Once PulseAudio is installed, you'll need to configure it for your specific use case. This may involve setting up input and output devices, adjusting volume levels, and configuring network streaming.

- ▰ ***Test Audio Streaming*** *$arecord /path/to/audio/file.wav*

3.3. Evaluation metrics

Training process

To track the training process, we added the TensorBoard metric, there we can check what our



	Run	Smoothed	Value	Step	Time	Relative
○	train	1943	1942	36	6/19/24, 1:57 PM	9.289 hr
●	validation	1938	1935	36	6/19/24, 1:57 PM	9.289 hr

Figure 3.10. curves of Loss

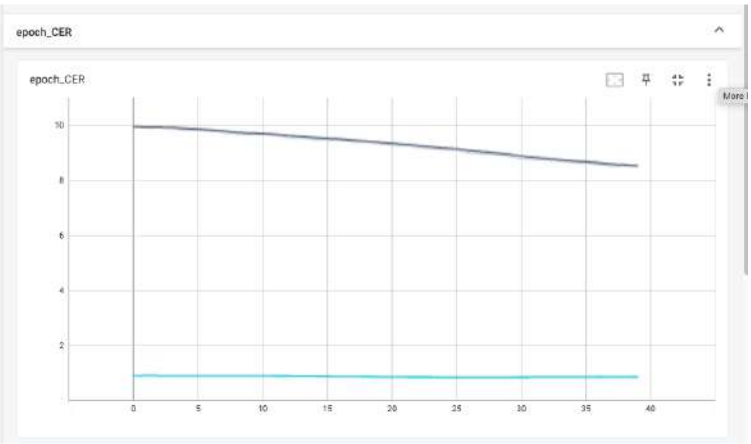
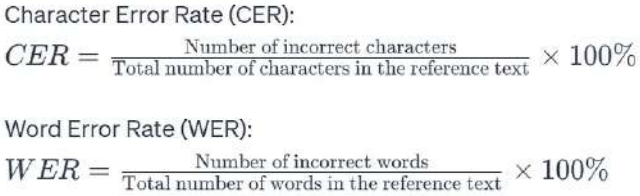


Figure 3.11. curves of CER, and WER metrics.



Character Error Rate (CER):

$$CER = \frac{\text{Number of incorrect characters}}{\text{Total number of characters in the reference text}} \times 100\%$$

Word Error Rate (WER):

$$WER = \frac{\text{Number of incorrect words}}{\text{Total number of words in the reference text}} \times 100\%$$

Figure 3.12. CER and WER

4.Conclusion

The field of artificial intelligence (AI) speech recognition has come a long way since the 1940s. Speech recognition has improved in accuracy and efficiency with the merging of deep learning and natural language processing. The following are some of the primary obstacles in speech recognition:

- **Handling the diversity of human speech**
- **Identifying words that are similar**
- **The quality of the audio signal**

Speech recognition employs a number of methods, such as phonetic-based methods, dynamic time warping, hidden markov models, and deep learning. Furthermore, the accuracy of speech recognition models can be raised by applying beamforming and noise cancellation techniques.

voice recognition has been greatly impacted by the introduction of transformers, which have made it possible to create more accurate models for applications like virtual assistants, natural language processing, and voice recognition.

This article showed how to use TensorFlow to combine a 2D CNN, RNN, and CTC loss to create a rudimentary voice recognition model. Speech recognition has the potential to be an effective tool for numerous sectors when the proper methods and data are used.

Chapter 4. Voice control application

1.Introduction

In today's digital age, technology is constantly evolving to make our lives easier and more convenient, one of the latest innovations that is revolutionizing user experience is voice control [20]. From smartphones to smart home devices, voice control technology is changing the way we interact with our devices by allowing us to simply speak commands instead of having to type or tap.

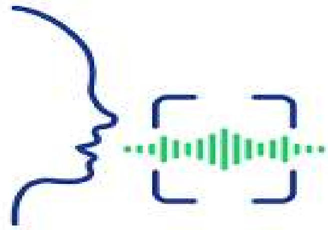


Figure 4.1. voice control technology

Voice control technology has come a long way in recent years, thanks to advancements in artificial intelligence and machine learning. Companies like Amazon, Google, and Apple have invested heavily in voice recognition technology, leading to the development of virtual assistants like Alexa, Google Assistant, and Siri. These virtual assistants are able to understand natural language commands and carry out tasks such as setting reminders, playing music, and even controlling smart home devices.

Voice Control applications allows users with mobility and motor impairments to control their device using speech. Common functions include navigating through apps, writing and editing text, and making calls [21]. Some applications allow users to customise the voice controls to suit their needs.

Key Features of Voice Control Technology:

- ➢ Hands-free operation
- ➢ Different languages processing
- ➢ Integration with smart home devices
- ➢ Personalized user experience

- Controlling low capacity devices

The Benefits of Voice Control Technology:

- Increased convenience
- Improved accessibility
- Enhanced user engagement
- Greater efficiency

Raspberry PI and voice controlling:

Voice-activated devices such as the Amazon Echo are becoming ever popular, and you can make your own using a Raspberry Pi [22], an inexpensive USB microphone and some suitable software. You too can have your Raspberry Pi search YouTube, open web pages, launch applications and even respond to questions, simply by speaking.

The Raspberry Pi has no built-in soundcard or audio jack, so you need a USB microphone or a webcam with built-in microphone.

2. Target platforms and devices

When creating a voice control application, it is essential to identify the platforms and devices it will target to ensure compatibility and functionality across different environments. Moreover, when targeting platforms and devices for a voice control application using a Raspberry Pi, both the hardware capabilities of the Raspberry Pi and the potential use cases must be taken on consider.

Tablets and Smartphones:

- IOS: Use Apple's Speech framework for voice recognition when developing in Swift or Objective-C.

- Android: Utilize the built-in speech recognition capabilities of Android or Google's Speech API while developing in Java or Kotlin.

Computers and laptops:

- Windows: Make use of the Windows Speech Recognition API or the Microsoft Speech SDK.

Utilize Apple's Speech framework on macOS.

- Linux: Make use of web resources like Google Speech-to-Text or open-source libraries like Pocket Sphinx.

2.1. Low capacity computers and their limits

In comparison to mid-range or high-end systems, low-capacity computers [23], often known as low-end or budget computers, usually have fewer hardware resources. They are distinguished by:

- **Processor:** Clock speeds and number of cores are generally lower in slower CPUs. Low-capacity PCs are commonly equipped with CPUs from AMD, Intel, Intel Pentium, Intel Celeron, and lower-end AMD Ryzen or Intel Core i3 models.
- **Memory (RAM):** RAM in smaller increments, usually 2 GB to 8 GB. 4GB of RAM is a common feature of low-end computers, and it is seen to be the bare minimum for smoothly functioning modern operating systems.
- **Storage:** 128GB to 500GB of conventional Hard Disk Drives (HDDs) are frequently used due to their limited storage capacity. In order to boost performance, some low-capacity laptops might feature smaller Solid State Drives (SSDs), but these are typically between 64 and 256GB in size.
- **Graphics**: Integrated graphics rather than dedicated graphics cards. This is usually sufficient for basic tasks like web browsing and office work.
- **Build Quality:** To reduce expenses, use simpler construction materials and designs. Although it lowers the overall cost, this may have an impact on durability and appearance.
- **Battery Life:** Due to smaller, less effective batteries, laptop battery life varies but may be shorter.

Figure 4.2. Low capabilities computer

2.1.1. Computer limits

- Capacity PCs are appropriate for the following:

Word processing, spreadsheets, presentations, and other office programs are examples of basic office work.

Web browsing includes using social media, email, and internet surfing.

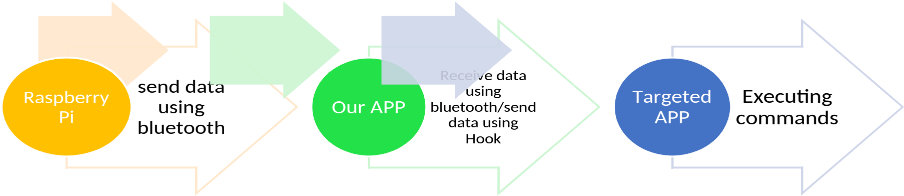
Media consumption includes music listening, movie watching, and modest photo editing.

For educational purposes: ideal for research, coursework, and online learning for students.

Usage at Home: Standard computer activities at home, such as handling personal finances, arranging pictures, and light gaming.

2.1.2. Application architecture

The working mechanism and construction of our application is as follows:



Raspberry Pi
send data using bluetooth
Our APP
Receive data using bluetooth/send data using Hook
Targeted APP
Executing commands

Figure 4.3. Application architecture

2.1.3. Hook application

A software technique known as a "hook application," or simply "hook," [24] is used to intercept system messages or events and enable customized treatment or processing of certain messages or occurrences. In software development, hooks are frequently used to expand or change an operating system or program's behavior without changing its source code.

Hook Types

- Framework Hooks: These are used to intercept system-wide events like file system changes, window messages, and keyboard and mouse input.
- Keyboard Hooks: Record and handle keystrokes.
- Mouse Hooks: Record and handle mouse movements.
- File System Hooks: Keep an eye on and eavesdrop on system activity.
- Application Hooks: These are used to intercept events—like function calls, messages, or events—that are unique to a particular application.

2.2. Smart devices under Android system

Android smart devices [25] refer to a wide range of gadgets powered by the Android operating system, developed by Google. These devices span multiple categories, including smartphones, tablets, smartwatches, smart TVs, and more. Here are some key types of Android smart devices:

1. Smartphones

Features: High-resolution displays, powerful processors, advanced cameras, extensive app ecosystems via Google Play Store, and frequent updates.

2. Tablets

Features: Larger screens compared to smartphones, often used for media consumption, productivity, and gaming.

3. Smartwatches

Features: Fitness tracking, notifications, health monitoring (heart rate, sleep, etc.), integration with smartphones.

4. Smart TVs

Features: Access to streaming services, apps, voice control, and integration with other smart home devices.

5. Smart Home Devices

Features: Home automation, voice control via Google Assistant, integration with other smart home ecosystems.

6. Android Auto

Description: An interface that brings the Android experience to car dashboards.

Features: Navigation, hands-free calling, music streaming, voice commands, and access to various apps.

7. Wearable Tech

Features: Health and fitness monitoring, augmented reality capabilities, notifications, and interaction with other devices.

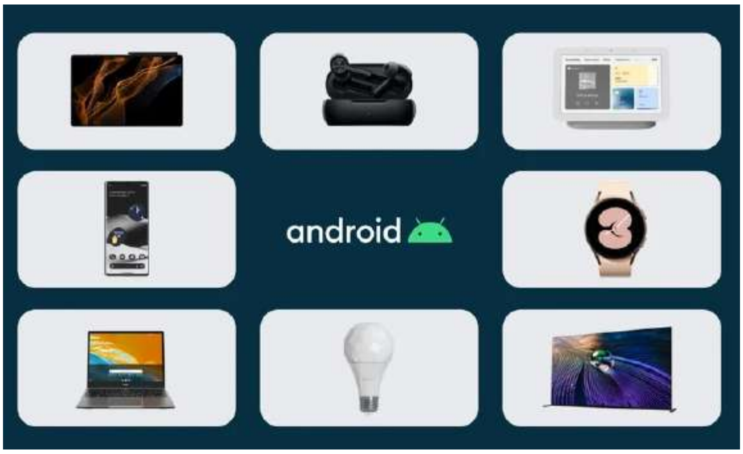


Figure 4.4. Smart devices under Android system

Benefits of Android Smart Devices:

- Customization: Users can personalize their devices with apps, widgets, and settings.
- Variety: A wide range of devices to suit different preferences and budgets.
- Integration: Seamless integration with Google services and other smart devices.
- Updates and Security: Regular updates to improve performance, add features, and enhance security.

2.2.1. Smartphones limits

Mobile phones come with lots of limitations; these limitations [26] play out in good mobile user experiences, there are many examples of this:

Small Screen

Screen size is a big limitation for mobile devices. The content displayed above the fold on a 30-inch monitor requires 5 screenfuls on a small 4-inch screen. Thus, mobile users must incur a higher interaction cost in order to access the same amount of information, also rely on their short-term memory to refer to information that is not visible on the screen. It is thus not surprising that mobile content is twice as difficult.

Touchscreen

It is hard to type proficiently on a tiny virtual keyboard and it is easy to accidentally touch the wrong target. In addition, Touch-typing is impossible in the absence of haptic feedback; plus, keypads themselves are small and keys are crowded. Because on a touchscreen there can be many target areas, it is easy to make accidental touches.

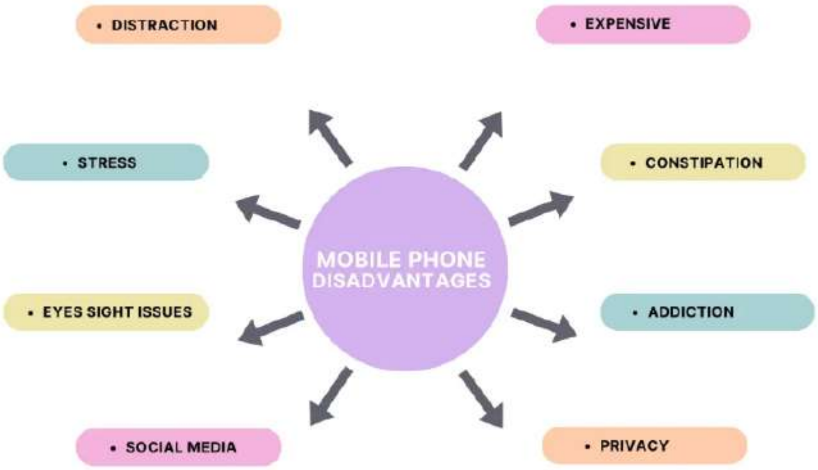
The leak of Security and Privacy

Our daily lives now can't function without our smartphones. It is a terrific method to stay in touch with friends and family and convenient. Sadly, this dependence on technology has resulted in various security problems that might endanger you. Photos, movies, and music are continually captured on smartphones. However, the smartphone is readily hackable, which poses a danger of your private information leaking (videos, photos). Many programs like this steal all of your data.

Health problem

By the way, the claim that smartphone towers harm health is not entirely supported. But according to some investigations, mobile phone towers can harm your health in various ways, including! Abnormal cell development, brain tumors, weakened immunity, sleep deprivation, anxiety, children's blood cancer, infertility, miscarriage, and other health issues.

Smartphones include attractive games and social media platforms that can lead to addiction, especially for young children developing before age 10. As a result, several schools have prohibited cell phone use within their buildings.



4.5. Smartphones limits

2.2.2. Smart TV limits

Complexity and Learning Curve

With their advanced features and capabilities, smart TVs [27] can be more complex than traditional televisions. Navigating through various apps, settings, and customization options may require some initial learning and adjustment. While most smart TVs strive to provide user-friendly interfaces, less tech-savvy individuals may find it challenging to fully utilize all the available features. It is important to invest some time in understanding the functions and settings of your smart TV to make the most of its capabilities. However, with a little patience and familiarity, the learning curve can be overcome, and you can fully enjoy the features and benefits that smart TVs offer. Find them here on rent!

Security and Privacy Concerns

As smart TVs connect to the internet and gather user data, security, and privacy concerns come into play. There have been instances of smart TVs being vulnerable to hacking attempts or unauthorized access. To mitigate these risks, it’s crucial to ensure that your smart TV has robust security features and that you follow best practices, such as keeping the software up to date and securing your home network. It is also essential to be mindful of the permissions and data-sharing practices of the apps and services you use on your smart TV. Reading privacy policies, disabling unnecessary data sharing, and exercising caution when granting permissions can help protect your privacy and ensure a secure viewing experience.

Dependence on Internet Connection

Since smart TVs heavily rely on internet connectivity, a stable and reliable internet connection is essential for optimal performance. If you experience internet outages or have a slow internet connection, it can hinder your ability to stream content smoothly or access online features. Additionally, some smart TV functions, such as software updates or app installations, may require a constant internet connection, which could be a limitation in areas with limited or unreliable internet access. It is important to consider the quality of your internet connection and the potential limitations it may impose on your smart TV experience.

Cost and Price Range

Compared to traditional televisions, smart TVs tend to be pricier due to their advanced features and technology. While the cost varies depending on the brand, size, and specific features, smart TVs generally come at a higher price point. However, it is important to consider the long-term benefits and convenience that smart TVs offer when assessing their value for your entertainment needs. Investing in a smart TV can provide you with a wide range of features and capabilities that enhance your viewing experience and provide access to a vast selection of content. If you are low on budget, you can take a TV on rent and avail of its benefits.

3. Wireless communication

Wireless communications is the transmission of voice and data without cable or wires. In place of a physical connection, data travels through electromagnetic signals broadcast from sending facilities to intermediate and end-user devices. The first wireless transmitters went on the air in the early 20th century using radiotelegraphy, which is radio communication using Morse code or other coded signals. Later, as modulation made it possible to transmit voice and music wirelessly, the medium became known as radio. Wireless transmitters use electromagnetic waves to carry voice, data, video or signals over a communication path.

The groundwork for modern wireless networking was laid in the early 1970s with the launch of ALOHA net in Hawaii. The network, technically a wide area network (WAN), relied on ultra-high frequency signals to broadcast data among the islands. The technology underpinning ALOHA net helped fuel the creation of Ethernet in 1973 and played an important role in the development of 802.11, the first wireless standard.

The evolution of wireless features

As a medium, wireless communications has been around for more than a century. However, it has only been in the past 15 years particularly after the ratification of the 802.11ac and 4G standards that the technology evolved enough to permit the development of applications and services comprehensive enough for widespread enterprise and consumer adoption. To that end, wireless features have evolved from simple data transfers at rates of only 54 Mbps to operations that require gigabits of data to complete.

Each new generation of wireless communications creates capabilities that are more sophisticated, giving users more flexibility in how they access the information and services they need. As a result, people can now connect to resources from almost anywhere. At the same time, mobile devices have become more powerful and versatile, giving users the opportunity to complete complex tasks. Advances in performance, capacity and coverage will continue.

3.1. Emitter application protocol (Raspberry)

Steps to send text via the protocol:

1. Set up Bluetooth on Raspberry Pi: Make sure your Raspberry Pi has Bluetooth enabled and is discoverable.

2. Pair the Raspberry Pi with the Android device: Ensure that the devices are paired.
3. Install necessary software on Raspberry Pi: Use a library like `pybluez` for Bluetooth communication in Python.

4. Develop a Python script on the Raspberry Pi: Write a script to divide the text into sections and send them via Bluetooth.

5. Develop an Android app: Write an Android application to receive the text sections and reassemble them.

3.2. Receiver application protocol

3.2.1. Required libraries for Windows

1. DevicesزWindowsزBluetooth:
- This library, which is a component of the Universal Windows Platform (UWP), offers a number of APIs for using Bluetooth devices.

- Makes it possible for you to find, connect, and communicate with Bluetooth LE devices.
- Well-documented and Microsoft-backed.
- Docus.microsoft.com/en-us/uwp/api/windows.devices.bluetooth (Microsoft Documentation)

2. 32feet.NET:
- A Bluetooth.NET library.
- Accommodates Bluetooth Low Energy (BLE) as well as Bluetooth Classic.
- It has been around for a while and is simple to use for.NET developers.
The GitHub repository for 32feet.NET(https://inthehand.github.com/32feet)

3. InTheHand.Net.Bluetooth:

- An additional.NET library that offers Bluetooth capabilities.
- Bluetooth Classic is supported.
- Applies to both.NET Framework and.NET Core programmes.
- The GitHub repository for InTheHand.Net(https://inthehand.github.com/32feet)

3.2.2. Required libraries for Android

1. Android Bluetooth API:
- Everything you need to integrate Bluetooth communication into your apps is provided by Android's native Bluetooth API. Classes like `BluetoothAdapter`, `BluetoothDevice`, `BluetoothSocket`, and so forth are included in it.

- https://developer.android.com/guide/topics/connectivity/bluetooth is the URL for the Android Bluetooth documentation.

2. Android-BluetoothSPPLibrary:
- A straightforward and user-friendly library for Bluetooth Serial Port Protocol (SPP) communication.

- The repository on GitHub(Android-BluetoothSPPLibrary: https://github.com/akexorcist/)

3. BluetoothKit:
- An intuitive and basic library for Android that supports Bluetooth Classic and Bluetooth Low Energy (BLE).
- [GitHub Repository](bluetooth-ktx.github.com/louiscad)

4. RxAndroidBle:
- An integrated RxJava library that is robust, user-friendly, and adaptable for Bluetooth Low Energy (BLE) for Android.
- The repository on GitHub(Polidea/RxAndroidBle/github.com)

5. BLESSED-Android:
- An Android library for Bluetooth Low Energy (BLE) based on Kotlin and Coroutines.
- The repository on GitHub(blessed-android.github.com/weliem)

6. TinyB:
- An Android-compatible Java and C++ library for Bluetooth Low Energy (BLE) on Linux-based operating systems.
- https://github.com/intel-iot-devkit/tinyb is the TinyB GitHub page.

4. Performance evaluation

4.1. Response time

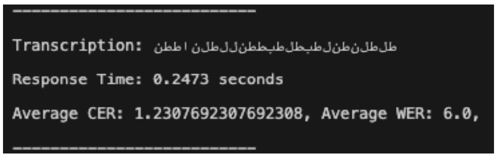
- Response Time: 0.4987 seconds

4.2. Voice control

- **librosa version:** 0.10.2.post1
- Average CER: **1.2307692307692308**
- Average WER: **6.0**

4.3. Speech transcription

Transcription: نططانلطاللنططبطلطبطلنطلنطلط



```
---------------------------
Transcription: طلطلنطنلطبطلطبططنللطلن اططن
Response Time: 0.2473 seconds
Average CER: 1.2307692307692308, Average WER: 6.0,
---------------------------
```

Figure 4.6. Evaluation

5. Limits and difficulties

5.1. Limits of the current project

Accuracy of Speech Recognition: The dependability of the speech-to-text (STT) function can be impacted by variations in dialects and accents, making Arabic speech recognition an extremely challenging task. This can reduce the smart TV control system's responsiveness and overall user experience.

Limitations on Processing and Memory: As a single-board computer, the Raspberry Pi lacks the processing power and memory of more potent smart TV controllers or specialised voice control platforms. This might affect the system's responsiveness and real-time performance, particularly when handling several inputs at once or when executing intricate voice instructions.

Compatibility and Integration Challenges: Because different smart TV models and brands may have different control protocols and interfaces, it can be difficult to integrate the Raspberry Pi-based solution with them. It could be necessary to do a lot of testing and customising in order to guarantee a smooth integration and uniform user experience across various TV models.

Dependency on Internet connectivity: The STT function of the system depends on a dependable internet connection, which might be a drawback because it creates a point of failure and may not be appropriate in all usage scenarios, such as places with patchy or unreliable internet access.

5.2. Future work

- **Enhancing Arabic Speech Recognition:** Work together with Arabic STT providers and researchers to improve the precision and resilience of the speech recognition capabilities, paying particular attention to the issues raised by accent and dialect changes.
Improving Processing Capabilities: To increase the system's total processing power and memory, consider using more potent single-board computers or integrating it with other hardware. This will allow the system to handle complex voice commands and sensor data more effectively.

- **Creating a Complete Integration for Smart Homes:** Include connection with other smart home systems and appliances, like lighting, HVAC, and home security, in addition to smart TV management. A more streamlined and centralised voice-controlled smart home experience would result from this.

- **Offline Capabilities and Local Processing:** Look into ways to make the system less dependent on internet access by utilising the Raspberry Pi's local processing or adding offline speech recognition features. This will increase the system's overall responsiveness and dependability.

- **Standardised Customisation and Integration:** Possibly by developing standardised plug-in modules or APIs, establish a system architecture that is flexible and modular so that it may be readily integrated with a range of smart TV models and brands. This would make it easy to customise for various user requirements and encourage wider adoption.

6. Conclusion

There are benefits and drawbacks of integrating Arabic audio texts with a Raspberry Pi to manage smart TVs. By utilising the Arabic STT resources and the Raspberry Pi's adaptability, it is possible to develop a smart TV control solution that is easier to use and more approachable. However, further development and integration work is required to overcome the limitations in voice recognition accuracy, processing resource constraints, and compatibility issues.

The suggested system can advance to give Arabic-speaking users a more dependable, responsive, and feature-rich voice-controlled smart TV experience by concentrating on boosting processing power, developing thorough smart home integration, and improving Arabic speech recognition capabilities. In order to fully realise the promise of this Raspberry Pi-based solution and facilitate the seamless integration of smart TVs into a larger smart home ecosystem, it will be necessary to address these constraints and investigate potential future work areas.

General conclusion

General conclusion

Arabic voice commands combined with a Raspberry Pi-based smart TV management system offer a viable way to improve the accessibility and user experience of smart TVs for Arabic-speaking consumers. Through the utilisation of the Raspberry Pi's adaptability, affordability, and Arabic speech recognition resources, this project showcases the possibility of expanding the reach of voice-activated smart home features.

The capacity of the suggested approach to operate smart TVs more naturally and intuitively, minimising the need for conventional remote controls and on-screen menus, is one of its main advantages. This can promote a more smooth integration of smart TVs into the daily life of Arabic-speaking households and greatly increase accessibility, particularly for users with physical or visual disabilities.

The project must, however, overcome a number of obstacles, including boosting the Raspberry Pi's processing power, guaranteeing thorough integration with a variety of smart TV models, and upgrading the accuracy of Arabic speech recognition. These constraints can be overcome, and the Raspberry Pi-based smart TV control solution can develop into a more dependable, responsive, feature-rich platform for voice-activated smart home experiences by iteratively improving the system's architecture and functionality.

It will be more and more crucial to be able to easily integrate and control multiple devices through natural language interfaces as smart home and Internet of Things (IoT) technologies develop. The effective creation and implementation of this Raspberry Pi-based system can help voice-activated smart home solutions become more widely used, enabling Arabic-speaking users and opening the door for more inclusive and accessible smart home technology.

Bibliography

[1] https://www.nabto.com/embedded-system-boards/

[2] https://www.geeksforgeeks.org/different-operating-systems/

[3] https://www.digi.com/blog/post/examples-of-embedded-systems

[4] https://www.electronicshub.org/different-types-sensors/

[5] https://nordvpn.com/blog/what-is-speech-recognition/

https://www.researchgate.net/publication/360336790_An_overview_of_Automatic_Speech_Recognition_Preprocessing_Techniques

[6] N. Mamatov, N. Niyozmatova, and A. Samijonov, “Software for preprocessing voice signals,” Int. J. Appl. Sci. Eng., vol. 18, no. 1, pp. 1–8, Mar. 2021, doi: 10.6703/IJASE.202103_18(1).006.

[7] S. Lee and H. Kwon, “A Preprocessing Strategy for Denoising of Speech Data Based on Speech Segment Detection,” pp. 1–24, 2020, doi: 10.3390/app10207385.

[8] Bharath Y.K, Veena S, Nagalakshmi K.V, M. Darshan, and R. Nagapadma, “Development of robust VAD schemes for Voice Operated Switch application in aircrafts: Comparison of real-time VAD schemes which are based on Linear Energy-based Detector, Fuzzy Logic and Artificial Neural Networks,” in 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), 2016, no. 1, pp. 191–195, doi: 10.1109/ICATCCT.2016.7911990.

[9] O. K. Hamid, “Frame Blocking and Windowing Speech Signal,” J. Information, Commun. Intell. Syst., vol. 4, no. 5, pp. 87–94, 2018.

[10] Y. A. Ibrahim, J. C. Odiketa, and T. S. Ibiyemi, “Preprocessing technique in automatic speech recognition for human computer interaction: an overview,” Ann. Comput. Sci. Ser., vol. 15, no. 1, pp. 186–191, 2017.

[11] S. Furui, Digital Speech Processing, Synthesis and Recognition, vol. 148. 2014.

[12] R. Singh, U. Bhattacharjee, and A. K. Singh, “Performance Evaluation of Normalization Techniques in Adverse Conditions,” Procedia Comput. Sci., vol. 171, no. 2019, pp. 1581–1590, 2020, doi: 10.1016/j.procs.2020.04.169.

[13] O. Kalinli, G. Bhattacharya, and C. Weng, “Parametric Cepstral Mean Normalization for Robust Speech Recognition,” ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., vol. 2019-May, pp. 6735–6739, 2019, doi: 10.1109/ICASSP.2019.8683674.

[14] https://www.dolbeyspeech.com/blog/transcription-vs-speech-recognition/

[15] https://www.analyticsvidhya.com/blog/2022/03/audio-denoiser-a-speech-enhancement-deep-learning-model/

[16] https://www.gladia.io/blog/introduction-to-speech-to-text-ai

[17] https://www.whichvoip.com/articles/speech-to-text-transcription.html

[19] https://www.researchgate.net/figure/Convolutional-Neural-Network-Architecture-for-Speech-Recognition_fig1_336819865

[20] Dustin A. Coates, (2019),” Voice Applications for Alexa and Google Assistant”, Manning CO, (ISBN: 97816172953171617295310).

[21] Malathi Subramanian، Fadi Al Turjman، Ramakrishnan Malaichamy, (2023),” Proceedings of the 6th International Conference on Intelligent Computing”, Atlantis Press International BV, (ISBN: 97894646325079464463250).

[22] Tanay Pant, (2016),” Building a Virtual Assistant for Raspberry Pi”, Apress CO, (ISBN: 97814842216791484221672).

[23] Laurence T. Yang, (2005),” Embedded and Ubiquitous Computing - EUC 2005”, De-Springer.ogg, (ISBN: 97835403080723540308075).

[24] John Larsen, (2021),” React Hooks in Action with Suspense and Concurrent Mode”, Manning Publications, (ISBN: 97816172976321617297631).

[25] Dan Gookin, (2020),” Android for Dummies”, Wiley-VCH, (ISBN: 97811197113531119711355).

[26] Ann McMurray, (2016),” Ten Tips for Parenting the Smartphone Generation”, Rose Publishing, (ISBN: 97816286237031628623370).

[27] Benjamin Michéle, (2015),” Smart TV Security Media Playback and Digital Video Broadcast”, Springer International Publishing, (ISBN: 97833192099443319209949).

[28] https://pylessons.com/speech-recognition