# SPEECH RECOGNITION

A COURSE PROJECT REPORT

By

**ADITYA RAUNAK RAJ(RA2011003010575)**
**NILESH KUMAR(RA2011003010576)**
**SATWIK(RA2011003010580)**

Under the guidance of

**DR. G. ABHIRAMI**

*In partial fulfilment for the Course*

of

18CSC305J – ARTIFICIAL INTELLIGENCE

in SCHOOL OF COMPUTING


**FACULTY OF ENGINEERING AND**

**TECHNOLOGY SRM INSTITUTE OF**

**SCIENCE AND TECHNOLOGY**

**Kattankulathur, Chenpalpattu District**

APRIL 2023

TABLE OF CONTENTS

# ABSTRACT

As a cross-disciplinary, speech recognition is based on the voice as the research object.

Speech recognition allows the machine to turn the speech signal into text or commands through the process of identification and understanding, and also makes the function of natural voice communication.

Speech recognition involves many fields of physiology, psychology, linguistics, computer science and signal processing, and is even related to the person's body language, and its ultimate goal is to achieve natural language communication between man and machine.

The speech recognition technology is gradually becoming the key technology of the IT man-machine interface .

The paper describes the development of speech recognition technology and its basic principles, methods, reviewed the classification of speech recognition systems and voice recognition technology, analyzed the problems faced by the speech recognition.

# INTRODUCTION

Speech recognition is the process by which a computer (or other type of machine) identifies spoken words. Basically, it means talking to your computer, AND having it correctly recognize what you are saying.

The following definitions are the basics needed for understanding speech recognition technology.

**Utterance**

An utterance is the vocalization (speaking) of a word or words that represent a single meaning to the computer. Utterances can be a single word, a few words, a sentence, or even multiple sentences.

**Vocabularies**

Vocabularies (or dictionaries) are lists of words or utterances that can be recognized by the SR system. Generally, smaller vocabularies are easier for a computer to recognize, while larger vocabularies are more difficult.

**Accuracy**

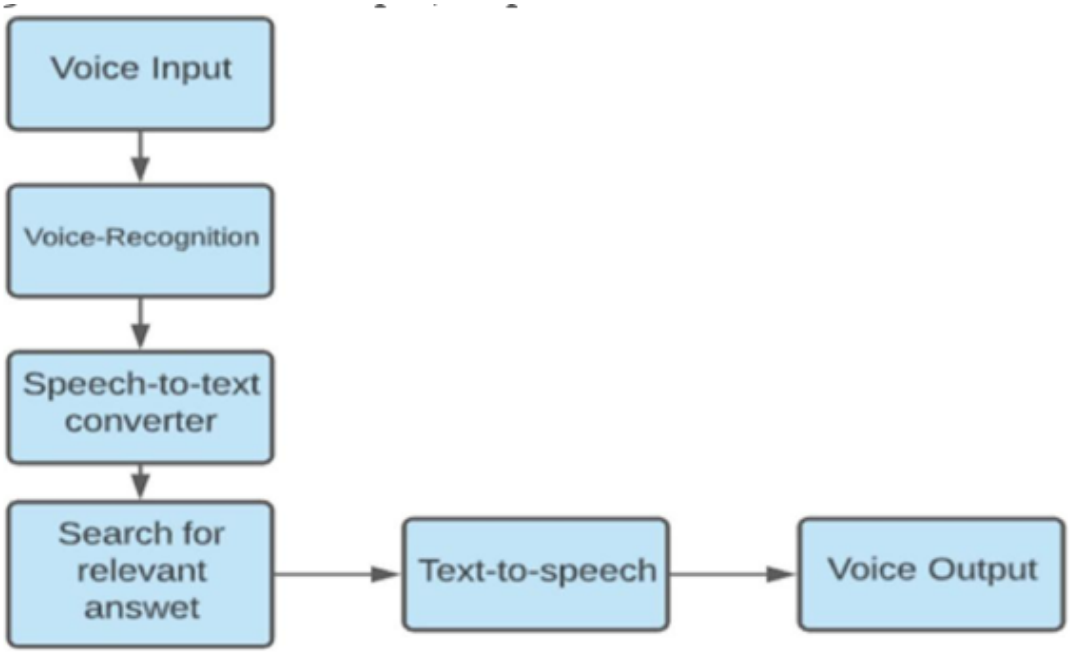
The ability of a recognizer can be examined by measuring its accuracy - or how well it recognizes utterances. This includes not only correctly identifying an utterance but also identifying if the spoken utterance is not in its vocabulary. Good ASR systems have an accuracy of 98% or more! The acceptable accuracy of a system really depends on the application.

**Training**

Some speech recognizers have the ability to adapt to a speaker. When the system has this ability, it may allow training to take place. An ASR system is trained by having the speaker repeat standard or common phrases and adjusting its comparison algorithms to match that particular speaker. Training a recognizer usually improves its accuracy.

# METHODOLOGY

The speech recognition system is essentially a pattern recognition system, including feature extraction, pattern matching, the reference model library. Its basic structure is shown in




- Import the libraries
- Train the model
- Let's predict

# IMPLEMENTATION

1. *IMPORT THE LIBABRIES*

```
import subprocess
import pyttsx3
import speech_recognition as sr
```

2. *CODE*

```
import subprocess
import pyttsx3

import speech_recognition as sr

#-----------------------------------------------------------------------------
# FUNCTION TO RECOGNIZE SPEECH AND RETURN RESPONSE

def recognize_speech_from_mic(recognizer, microphone):

    # check that recognizer and microphone arguments are appropriate type
    if not isinstance(recognizer, sr.Recognizer):
        raise TypeError("`recognizer` must be `Recognizer` instance")
    if not isinstance(microphone, sr.Microphone):
        raise TypeError("`microphone` must be `Microphone` instance")

    # adjust the recognizer sensitivity to ambient noise and record audio from the
microphone
    with microphone as source:
        recognizer.adjust_for_ambient_noise(source)
        audio = recognizer.listen(source)

    # set up the response object
    response = {
        "success": True,
        "error": None,
        "transcription": None
    }

    # try recognizing the speech in the recording
    # if a RequestError or UnknownValueError exception is caught,
    #     update the response object accordingly
```

```
    try:
        response["transcription"] = recognizer.recognize_google(audio)
    except sr.RequestError:
        # API was unreachable or unresponsive
        response["success"] = False
        response["error"] = "API unavailable"
    except sr.UnknownValueError:
        # speech was unintelligible
        response["error"] = "Unable to recognize speech"

    return response

#---------------------------------------------------------------------------
# FUNCTION TO CONVERT TEXT TO SPEECH

def texttospeech(app_string):
    engine = pyttsx3.init()
    engine.say(app_string)
    engine.runAndWait()

#---------------------------------------------------------------------------
# FUNCTION TO RUN THE SPECIFIC APPLICATION

def run_application(app_call):

    if "calculator" in app_call:
        try:
            texttospeech("Sure thing. Opening Calculator Now")
            subprocess.Popen('C:\\Windows\\System32\\calc.exe')
            app_open_reponse = "Success"
        except subprocess.CalledProcessError as e:
            app_open_reponse = e.output

    if "blender" in app_call:
        try:
            texttospeech("Sure. Opening Blender Now")
            subprocess.Popen('C:\\Program Files\\Blender Foundation\\Blender
3.2\\blender-launcher.exe')
            app_open_reponse = "Success"
        except subprocess.CalledProcessError as e:
            app_open_reponse = e.output

    if "audacity" in app_call:
```

```
        try:
            texttospeech("Sure. Opening Audacity Now")
            subprocess.Popen('C:\\Program Files\\Audacity\\Audacity.exe')
            app_open_reponse = "Success"
        except subprocess.CalledProcessError as e:
            app_open_reponse = e.output
    return app_open_reponse


#-------------------------------------------------------------------------
# MAIN : Where all functions are called

if _name_ == "_main_":

    APPLIST = ["calculator", "blender", "audacity"]

    # create recognizer and mic instances
    recognizer = sr.Recognizer()
    microphone = sr.Microphone()

    print("-- Speak Prompt --")
    texttospeech("Hello Nilesh. which app do you want to open?")
    #time.sleep(3)

    guess = recognize_speech_from_mic(recognizer, microphone)
    if guess["transcription"]:
        print("You said: {}".format(guess["transcription"]))
    if not guess["success"]:
        print("Error with API")
    if guess["error"]:
            print("ERROR: {}".format(guess["error"]))

    if any(w in guess["transcription"].lower().split(' ') for w in APPLIST):
        print("App Found")
        app_call = guess["transcription"]
        app_run_status = run_application(app_call)
    else:
        print("App Not Found")

#-------------------------------------------------------------------------
```

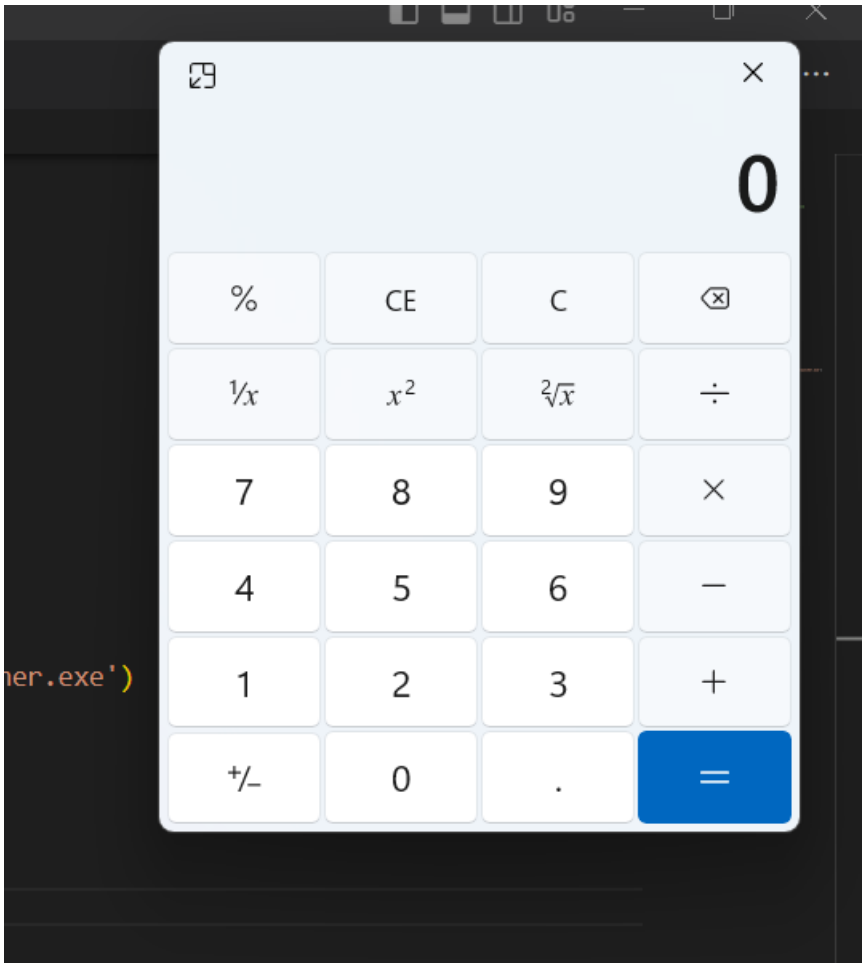
# RESULT

The result of speech recognition depends on the input audio and the speech recognition engine or API being used. In general, the result of speech recognition is the recognized text that corresponds to the spoken words in the input audio.

Here is an example of the result of speech recognition using the Google Speech Recognition API:

```
[Running] python -u "c:\Users\nk931\.vscode\speech.py"
-- Speak Prompt --
You said: open calculator
App Found

[Done] exited with code=0 in 18.785 seconds
```



# CONCLUSION

Through this project, we showed underlying emotion from speech audio data and some insights on the human expression of emotion through voice. This system can be employed in a variety of setups like Call Centre for complaints or marketing, in voice-based virtual assistants or chatbots, in linguistic research, etc. A few possible steps that can be implemented to make the models more robust and accurate are the following

An accurate implementation of the pace of the speaking can be explored to check if it can resolve some of the deficiencies of the model.

Figuring out a way to clear random silence from the audio clip.

Exploring other acoustic features of sound data to check their applicability in the domain of speech emotion recognition. These features could simply be some proposed extensions of MFCC like RAS-MFCC or they could be other features entirely like LPCC, PLP or Harmonic cepstrum.

Following lexical features based approach towards SER and using an ensemble of the lexical and acoustic models. This will improve the accuracy of the system because in some cases the expression of emotion is contextual rather than vocal.

Adding more data volume either by other augmentation techniques like time-shifting or speeding up/slowing down the audio or simply finding more annotated audio clips.

# REFERENCES


1. "Deep Speech: Scaling up end-to-end speech recognition" by A. Hannun et al. (2014)

   This paper presents a novel approach to end-to-end speech recognition using deep neural networks and demonstrates state-of-the-art performance on several speech recognition benchmarks.
2. "A comparative study of deep learning approaches for speech recognition" by H. Sak et al. (2014).

   This paper compares different deep learning approaches for acoustic modeling in speech recognition and provides insights into their strengths and weaknesses.
3. "Robust speech recognition in additive noise using adaptive microphone arrays" by M. Cooke et al. (2001).

   This paper presents a technique for improving speech recognition accuracy in noisy environments using adaptive microphone arrays.
4. "Speech recognition using neural networks: A survey" by T. K. Kim (1999)

   This paper provides a comprehensive survey of speech recognition using neural networks, including a historical overview and a discussion of the state of the art at the time of publication.
5. "Automatic speech recognition - A brief history of the technology development" by B. H. Juang (2005)

   This paper provides an overview of the history of automatic speech recognition technology and discusses the major advancements and challenges in the field.