COGNITIVE COMPUTING BASED MUSIC APPLICATION

Samundiswary Srinivasan
Department of Information Technology
SIESGST
samundiswarys@sies.edu.in

Gaurav Jaiswal
Department of Information Technology
SIESGST
gauravjit119@gst.sies.edu.in

Ajay Kannan
Department of Information Technology
SIESGST
ajaykit119@gst.sies.edu.in

Utkarsh Sharma
Department of Information Technology
SIESGST
utkarshsit119@gst.sies.edu.in

Jagan Nadar
Department of Information Technology
SIESGST
jagannit119@gst.sies.edu.in

Abstract - **The project is basically formulated on cognitive computing-based music player mobile application wherein it is an integration of voice recognition chatbot to play music on voice commands and a music recognition system that'll allow us to recognize any song in a given ambience despite its nature. So, often we seldom see both of these functionalities being utilized in a music application, as a matter of fact, most of the music applications don't have their own integrated voice bot or chatbot, so to say, that'll allow users to play songs on the basis of voice commands and also have the capability to recognize music within an environment. It is intended to use the algorithms that are used by ALAN bot and ACR cloud that are voice or speech processing algorithms and spectrogram respectively. The conversational bot which is utilized is known as ALAN bot, wherein all voice processing is performed in the Alan cloud. The Alan cloud is the AI-backend of the Alan Platform. This is where voice scripts are hosted and Spoken Language Understanding (SLU) and Natural Language Processing (NLP) tasks are accomplished. Furthermore, the music recognition software is known as ACR cloud, whose algorithm, i.e, Audio fingerprint algorithm is one the most robust and efficient and can perform the recognition quickly over a large database of music with nearly 11 Million tracks, and furthermore have a low number of false positives while having a high recognition rate.**

Index Terms - music player, alan, conversational bot, ACR cloud, cognitive computing.

I. INTRODUCTION

Android OS is used by the vast majority of the technology market, and a recent study concluded that the estimated magnitude of people using an android smartphone is over 2 to 3 billion worldwide. To better meet the needs of the smartphone community, novel and radical projects are incessantly being conceptualized. Music player happens to be one of the projects that possess the potential of impacting said community. People indulge in listening to music whilst they're engaged in the most frivolous of activities such as doing chores or running errands. Developers and software engineers worldwide are constantly on the job of contriving new functionalities and features that will ameliorate user experience. A futuristic idea happens to be a voice command based music application which will facilitate the music community immensely. Our project intends to conflate AlanAPI and Flutter SDK to create a music application.In doing so, we'll be formulating an application which will have an innovative touch to a music player application to play songs based on voice commands. The application is compatible on both Android and IOS operating systems.

Based on this we have devised and formulated a music player application namely "BlackDot" that basically provides us with all the rudimentary functionalities of a music player application such as having personalized playlists, latest songs that are trending and music charts with the most eminent playlists. We are vigorously utilizing the JioSaavan and Spotify API for getting the songs rendered onto our page with recent updates in the music industry.

We are also integrating the two functionalities that we discoursed earlier that are a conversational bot called ALAN and music recognition function using ACR cloud algorithm.

A.Conversational bot - ALAN

ALAN happens to be an infrastructure of voice pre-processing technology that has its backend hosted on a cloud to enable limitless scalability for voice scripts importing on virtual machines. It utilizes speech and voice based algorithms to instill the functionality of voice commands within systems irrespective of its domain. For more insight, here are the technologies used by ALAN.

Under the hood, the Alan cloud engages a combination of voice AI tools and technologies to simulate human-like dialog between the user and the app. Together, they allow Alan to interpret human speech, generate responses and perform the necessary actions in the app. The main voice technologies used by Alan are:
1. Natural language processing (NLP)
2. Spoken Language Understanding (SLU)
3. Automatic Speech Recognition (ASR)
4. Machine Learning (ML)
5. Speech-to-Text (STT) and Text-to-Speech (TTS)

Alan studio provides an integrated development environment to write the code that basically consists of intents on which ALAN will function and how it'll be activated. Going ahead, to allow users to interact with the application with speech and execute commands from voice scripts, you need to add the Alan button to your application which can be activated using a command such as "Hey Alan!" or "Hello, Alan!", so on and so forth. You can do it with Alan SDKs. Alan offers SDKs for different platforms with which it can be integrated such as:
1. Web frameworks: React, Angular, JavaScript, Electron
2. iOS: Swift and Objective-C
3. Android: Kotlin and Java
4. Cross-platform frameworks: Flutter, Ionic, React Native, Apache Cordova.

Once the integration has been done, we'll be able to flexibly and seamlessly use it in our environments based on our needs and requirements. Its versatile nature to be integrated allows developers to frictionlessly use it in their projects or software systems that they plan on devising.
With Alan, you do not need to plan for and deploy any components to run voice scripts. The solution architecture is serverless: voice scripts are run on VMs in the cloud.
Being backed by the cloud, Alan is highly scalable. It can scale up to millions of users as your requirements grow. All resources to support the necessary workloads are automatically provisioned by Alan.
The cloud is where voice scripts you create are executed and all voice processing takes place. Alan leverages advanced algorithms to handle voice commands and dialog flows of any complexity. Alan trains on intents using the terminology for your application and learns to understand the logic of your domain.
As we had discussed above, the backend being entirely hosted on the cloud allows it to be scalable, our voice scripts that are present in the virtual machines or voice user interface algorithms are all stored and managed by the cloud hosted backend of ALAN and it also provides a default hosting environment as a consequence.

B. ALAN Speech processing lifecycle

There are several steps which are entailed which dictates and determines how efficiently and proficiently the voice commands are to be read and how to effectively glean important adages or terminologies from within the commands.

1. The capturing of the voice command commences at the client side where the command, once captured, is sent to the ALAN cloud with pertinent voice data.
2. Afterwards, it capitalizes ASR(Automatic speech recognition) and STT(Speech to text) to convert speech data to text based user inputs.
3. With the help of SLU(Spoken language understanding) and NER(Named entity recognition) it parses the user input to extract meaningful phrases and significant terminologies. It creates a Domain language model for every application made in order to enable Alan's ASR(Automatic speech recognition) system to predict with high accuracy.
4. Furthermore, these extracted terms are then matched with the intents created by assigning a probability score of '1' for the most accurate match.
5. In case if we require ALAN to give responses it also utilizes text to speech to contrive voice based retortions.

C. ACR cloudAlgorithmic life cycle
ACR cloud is one of the most popular music recognition apps that has been around for almost 20 years now. Its algorithm is one the most robust and efficient and can perform the recognition quickly over a large database of music with nearly 11 million plus tracks, and furthermore have a low number of false positives while having a high recognition rate. Let's understand the basic pipeline of ACR cloud.

1. Understanding ACR cloud model or algorithm:
Before understanding the working of ACR cloud model lets understand more about sounds: Sound is a vibration that propagates as an acoustic wave, through a transmission medium such as a gas, liquid or solid. In human physiology, sound is the reception of such waves and their perception by the brain. Two main physical characteristics of sound are frequency and amplitude. Frequency is a number of oscillations per second, and it's measured in Hertz while amplitude, on the other hand, represents the size of each cycle.

2. Analog and Digital Sounds.
Every sound that exists in air or any other medium is analog. Analog waves are continuous and quite complex. They're so detailed that even the smallest fraction of a wave can still be divided into even smaller parts. Digital sound, on the other hand, is a representation of sound recorded in, or converted into, digital form. The process of transforming analog sound to digital is called **Sampling**. During this process certain information gets lost, and what we end up with is more of an approximate representation of a sound than the exact copy of it. It is important to know these things as the song that you hear is analog, and it gets transformed into a digital signal while being recorded. But it's still a wave, it has only been shrunk in size and stored. It needs to be transformed into some form that's easier to compare

and identify. For this transformation we make use of the Fourier transform.

3. Fourier transform

Fourier transform (FT) is a formula that transforms a sound wave into a graph of frequencies that the sound is made of, and their intensities. But this process leads to few problems:

i. FT takes a very long time to transform a song into a graph. ii. FT gives us only frequencies and their intensities (amplitude) without any information about timing. In other words, we don't know when in the song these frequencies occur.

The problem of speed can easily be solved by using **Fast Fourier Transform (FTT)**, which uses the process of downsampling to keep the frequency information at the expense of other data.

But FFT cannot solve the problem of incomplete visualization of a sound. In order to fix this ACR cloud uses a special type of graph called Spectrogram.

4. Spectrogram

Spectrogram is a visual representation of frequencies as they vary in time. In other words, it's a three dimensional graph.

In the below example we can see the axes are frequency and time and the amplitude is represented by the intensity of color of each point in the image.

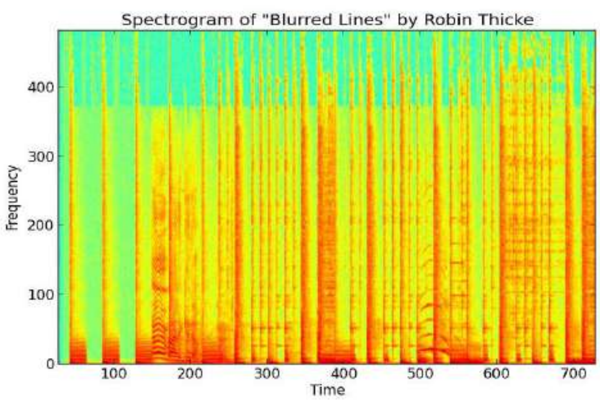


Fig. 1 Spectrogram heatmap for a particular song

The Spectrogram(confer to Fig. 1) is one of the core features behind this algorithm and is used to implement the Audio Fingerprint Algorithm.

5. Audio fingerprint Algorithm

Audio fingerprinting is the process of representing an audio signal in a compact way by extracting relevant features of the audio content. We can think of it as a condensed digital summary of a song. Just like human fingerprints, every song's acoustic fingerprint is unique, and can be easily identified even if there are small variations in data. This allows ACR cloud's algorithm to get rid of all unnecessary information about a certain song.

Once the metadata is collected for all the audio files in the database, the same process will be applied to the query and the metadata of the query is searched against the database.

After all these methodologies, around 68 Million songs in approximation have been fingerprinted successfully and stored in ACR cloud's global database readily available for songs to be recognized directly.

When a user is intending to scavenge for a song that they're arrantly unaware of, they can us the music recognition functionality offered by our application to recognize the song, which is performed by parsing the database that constitutes of over a multitude of songs that are unique because of the fingerprint attributed to them and uses the snippet to juxtapose and successfully find a match within a matter of seconds. With increasing and amplifying difficulty of song recognition, it doesn't disrupt the promptness with which it can recognize and recommend the song.

TABLE I. Research and their Surveys

II. Literature Survey

Sr. No	Author	Publications	Key findings/gaps
1	G. Kiruthikamani and E. Esakki Vigneswaran	Advanced music player with audio recognition and touch interface for visually impaired	This research paper basically deals with the implementation of an interface with the functionality of voice recognition in order to let visually impaired people to mitigate the problem to operate an interface, rather they can give voice commands to operate the application.
2	Avery Li-Chun Wang	An Industrial-Strength Audio Search Algorithm	This research paper extensively studies the industrial level applications that the ACR cloud music recognition application has to recognize music in any sort of ambience be it placid or hostile, regardless.
3	Nan Zhang	Mobile Music Recognition based on Deep Neural Network	This research paper entails the study of music recognition for piano and it's recognition can be

			improved using deep neural networks.

III. Objective

The objective of our project is mainly to provide two functionalities that aren't mostly available in any of the music applications that we usually come across or use on a daily basis, which are namely:

The chat bot or conversational bot that we have integrated is known as ALAN bot, which is a voice recognition chatbot to play music on voice commands and the music recognition software is known as ACR cloud, that helps in recognizing a song in any given ambience regardless of the ambience, which is apparently very synonymous to the liked of automatic content recognition apps such as SoundHound and Musixmatch.

Therefore, the main function of our project is the inculcation of the following functionalities:

1. Music Recognition
The basic function of music recognition will be to recognize the music from the surrounding or the user's environment and fetch the song in the application through which the user may play the song.

2. Speech Processing
This is particularly used by the user to search the songs through voice based commands. For instance, the user will command our ALAN bot to fetch the song by its following name. The ALAN bot then intakes the voice commands and processes through the system to find the recommended song.

3. Quick & Easy Search
Our music player application is a lightweight mobile application which enables the user to easily find their favorite music through the search bar. For example, the user types the name of the song and the application fetches the songs from the chart and shows other songs too, which have similar name or keywords as per the user's search.

4. Seamless user interface
A smoothly designed user interface that fulfills all of the user experience goals required for an application to gain the user appeal. Flutter allows us to create stateless components to have a modularized and more organized manner of designing the user interface.

5. Personalized playlists
The user has the freedom to explore songs and add it to their favorite playlist of all the songs that they've particularly have found an affinity for and have a personalized playlist that they have the full liberty to create and curate. This functionality allows them to formulate a playlist according to their preferences.

IV. Methodology (Flow Chart)

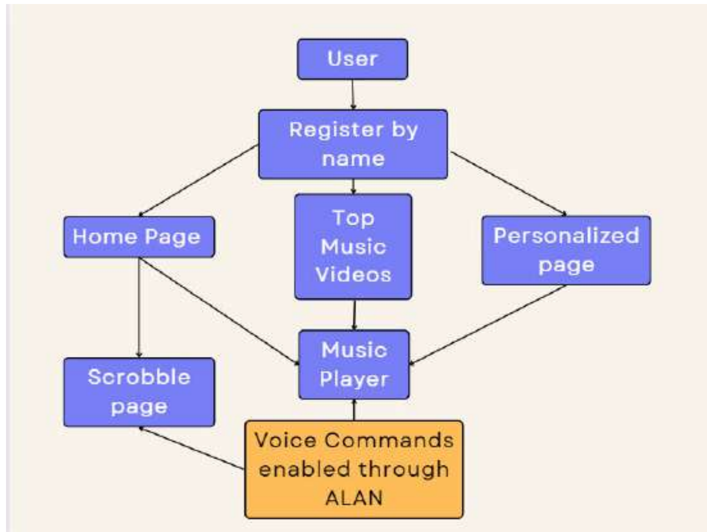


Fig. 2 Application workflow

As per the illustrated workflow(confer Fig 2) of the application displayed above, the user will instigate on operating with the application by just entering their name, as we're building an open source AI music player application, we won't be asking details that happen to be sensitive in nature.

Upon entering the application, the user will be able to see three interfaces that we've designed namely:

1. Home page: Which happens to be the main page of our application containing the personalized playlists of that particular user which will constitute of their favorite songs, a navigation bar to navigate through the entire application and the it's pages.

2. Top Music videos page: This page will contain all the music video snippets and video excerpts from the songs that are popularly and mostly listened.

3. Personalized page: This page basically consists of a personalized or specialized data of the user as which songs were liked and what kind of playlists has been formulated and so on and so forth. In contraction, a page to render the data pertaining a particular user according to their preferences.

4. Scrobble page: The term "scrobble" is actually a slang terminology for snatching, which in technical terms we'll be referring to as the recognition of songs from an excerpt played from any given song irrespective of the ambience it's played in.

Furthermore, the most essential part of our application lies in our homepage which is the instilling of speech processing based conversational bot named ALAN and also the automatic content recognition system of ACR cloud which is to be activated and triggered using a voice based command in the event of the need to recognize a particular song being played in a social setup, say, a party, or any given environment regardless of it's ambience.

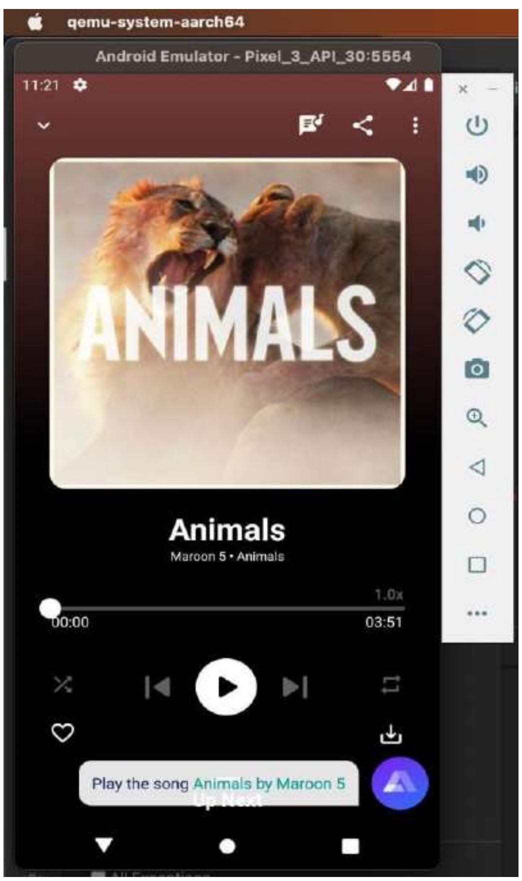
With that being said, this will be the overall flow or working of the mobile application that we've so devised.

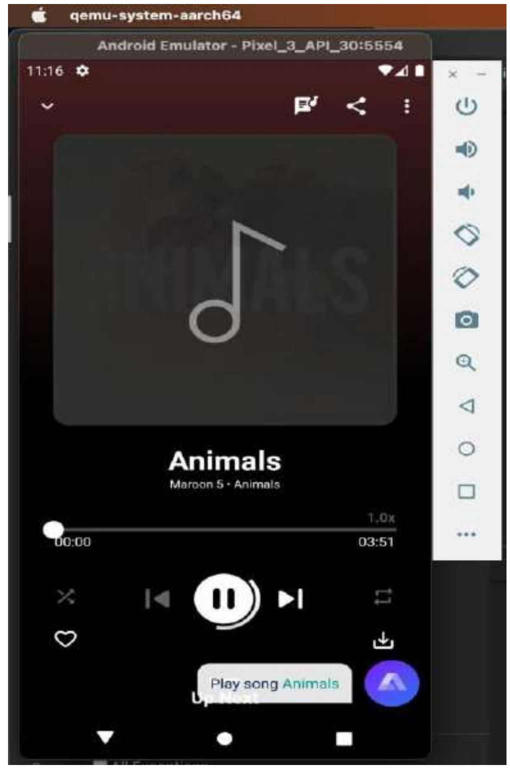
V. Experimental Work

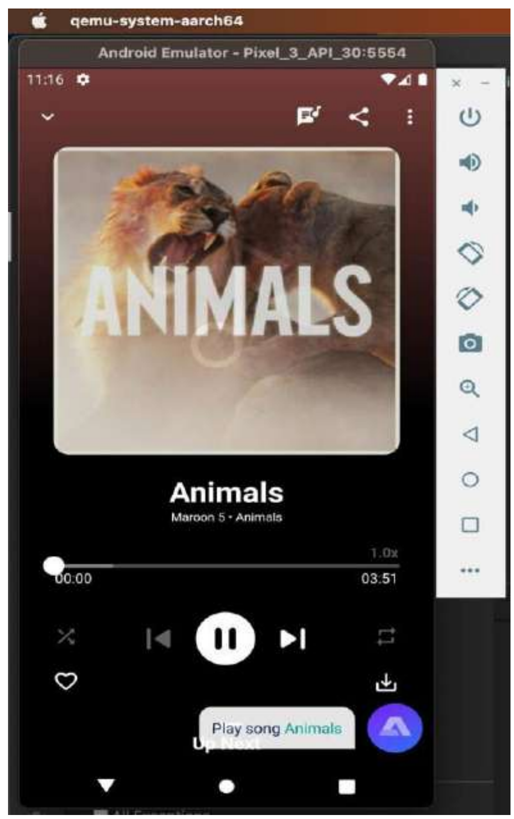
Conversational Bot - ALAN inputs and responses.

Here's a glimpse of how the conversational bot functions when it's been given a command:

On input - "Play song's <*name*>" it responds with "playing <*song name*>". In our circumstance, it'll be the following command, "Play song <Animals>".







VI. FUTURE SCOPE

- We intend to contrive a functionality of "humming" in our application which entails a user actually humming a song's melody or tune whose recognition also follows the same procedure as any song is recognized through a particular snippet or an excerpt from a song.

VIII. REFERENCES

[1] Kiruthikamani Govardhanaraj, Esakki Vigneswaran "Advanced music player with audio recognition and touch interface for visually impaired", January 2015.

[2] Avery Li-Chun Wang,Shazam Entertainment, Ltd.,"An Industrial-Strength Audio Search Algorithm".

[3] Nan Zhang, "Mobile Music Recognition based on Deep Neural Network", June 2022.

[4] C. Mahesh Reddy and Dr. Diana Moses, "AI based music application ".

[5] Shivam Sakore , Pratik Jagdale , Mansi Borawake , Ankita Khandalkar, "Music Recommender System Using ChatBot", Volume 9 Issue XII Dec 2021.

[6] Baptiste Caramiaux and Marco Donnarumma, "Artificial Intelligence in Music and Performance: A Subjective Art-Research Inquiry", 31 Jul 2020.

[7] Tiancheng Yang, Shah Nazir, "A Comprehensive Overview of AI-Enabled Music Classification and Its Influence in Games", 21 Jan, 2022'.

[8] Mandla Vamshi Krishna and Dasika Moukthika "Study on framework of audio fingerprinting and Shazam's working algorithm", December 2018.

[9] Devashish Ashok Pathrabe, Aboli Anil Gosavi, Yogesh Kumar, "Conversational Voice Controlled News Application", Jun 2022.

[10] Lifei Lu, Lida Xu,, Boyi Xu , Guoqiang Li, and Hongming Cai , Senior Member, IEEE, "Fog Computing Approach for Music Cognition System Based on Machine Learning Algorithm".

[11] Roberto De Prisco , Alfonso Guarino , Delfina Malandrino and Rocco Zaccagnino, "Induced Emotion-Based Music Recommendation through Reinforcement Learning", 4 November 2022.

[12] Azham Hussain, Emmanuel O.C. Mkpojiogu, Hassan Almazini, Hussein Almazini, "Assessing the Usability of Shazam Mobile App",