iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** V **Month of publication:** May 2023

DOI: https://doi.org/10.22214/ijraset.2023.51825

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

Advanced Audio Signal Processing for Speaker Recognition and Sentiment Analysis

Dr. Nisha Auti[1], Atharva Pujari[2], Anagha Desai[3], Shreya Patil[4], Sanika Kshirsagar5[6], Rutika Rindhe[6]
[1]*Dr,* [2]*Students, Department of Computer Engineering, Jspm Narhe Technical Campus, Maharashtra, India*

Abstract: ***Automatic Speech Recognition (ASR) technology has revolutionized human-computer interaction by allowing users to communicate with computer interfaces using their voice in a natural way. Speaker recognition is a biometric recognition method that identifies individuals based on their unique speech signal, with potential applications in security, communication, and personalization. Sentiment analysis is a statistical method that analyzes unique acoustic properties of the speaker's voice to identify emotions or sentiments in speech. This allows for automated speech recognition systems to accurately categorize speech as Positive, Neutral, or Negative. While sentiment analysis has been developed for various languages, further research is required for regional languages. This project aims to improve the accuracy of automatic speech recognition systems by implementing advanced audio signal processing and sentiment analysis detection. The proposed system will identify the speaker's voice and analyze the audio signal to detect the context of speech, including the identification of foul language and aggressive speech. The system will be developed for the Marathi Language dataset, with potential for further development in other languages.***

Keywords: Automatic Speech Recognition, Speaker Recognition, Sentiment Analysis, Applications, Personalization, Accuracy.

I. INTRODUCTION

Speaker recognition, also known as voice recognition, is a technology that enables automatic identification of an individual based on their unique voice characteristics. It is commonly used in various applications, such as security systems, virtual assistants, and biometric authentication. The system architecture of speaker recognition comprises of various components, including an audio input device, a feature extraction module, a machine learning algorithm, and a speaker model database.

The first step in speaker recognition involves capturing the audio input and converting it into digital form through pre-processing techniques. The feature extraction module then extracts relevant features from the signal that can be used to identify the speaker, such as pitch, spectral content, and other voice characteristics. These extracted features are then fed into a 1-Dimensional Convolutional Neural Network (CNN) that compares them to a database of speaker models to identify the most probable speaker. Techniques like clustering or classification can be used by the CNN to improve the accuracy of the identification process.

Once the speaker has been identified, the system can proceed to perform sentiment analysis on that piece of audio. Sentiment analysis is the process of identifying and extracting the sentiment behind a given piece of audio file in .wav format. This process classifies speech into different categories like Neutral, Aggressive, Negative, or Abusive to generate a Graded Sentiment Analysis report. The ability to analyze sentiment can be beneficial in applications such as call center quality monitoring, customer feedback analysis, and market research.

Overall, speaker recognition is an essential technology that has numerous applications, including speech recognition in virtual assistants, secure access to highly secure areas, voice dialing, banking, and computers. The system architecture of speaker recognition is a combination of audio input and signal processing, machine learning, and a database of speaker models, making it a powerful tool for automatic identification of individuals by their unique voice characteristics and performing sentiment analysis on the audio file.

II. BACKGROUND

Speaker recognition is the process of identifying a person by their voice, in this case, a speaker. In audio signal processing, we can use various techniques and algorithms to extract speech from an audio signal. Sentiment analysis of audio signals is a method for determining whether an audio message has positive or negative sentiment. This can be used to improve speech recognition systems by providing a more accurate transcription of the original audio messages.

Speaker recognition algorithm is one of several techniques used in our work which includes feature extraction methods like Mel-frequency cepstrum coefficients (MFCC) and Mel-frequency cepstrum transform (MFCT).Another method for recognizing speakers is by using neural networks. This technique has been shown to outperform acoustic modeling for most applications because it allows for more complex models than acoustic models without sacrificing recognition accuracy.

Speaker recognition methods have two important modules which are Enrolment and Verification. At the time of enrolment, the subject's audio signals are captured to extract important features from them and finally, a model is created which is also termed as voice print. For the next phase that is verification , a voice sample or commonly termed as utterance is balanced against the previous model. The important difference between identification methods and verification methods is that , for verification only one voice print is considered to be balanced against a single voice pattern whereas identification methods tend to balance the utterance against multiple models.

In order to perform speaker recognition and sentiment analysis over audio signals, there are several technologies that are used such as speech recognition systems, text-to-speech synthesis engines and acoustic models.These technologies can be used for different purposes such as voice command systems where it speaks commands or reminders to users or for entertainment purposes where it reads books or plays music.

Speaker recognition is a very powerful tool for identifying speakers in any given environment, but it does not just provide accurate results; it also has an impact on the outcome of an application's performance. That's because speaker recognition depends on many factors: environment (temperature, lighting, background noise), acoustics (room size, acoustic properties), and even age (acoustics change as we get older). All of these variables can have an impact on how well your system performs when it comes time to identify speakers.

A. *Limitations of Existing System*

The major limitation is that the system can only differentiate between the voices as male or female, but it cannot distinguish the identity of the person who the voice belongs to. The tonal frequencies of marathi Language are studied at a very small scale which constitutes a major limitation that we are trying to overcome. Another Limitation is that the system cannot implement recognition as well as sentiment analysis of the speech at the same time and there is no mechanism of deployment for the existing models.

Features of Neural Network based Speaker Recognition models-

1) *Improved security:* Speaker recognition can be used as a form of authentication to grant or deny access to secure systems or locations. This can help prevent unauthorized access and increase the overall security of a system.
2) *Increased convenience:* Speaker recognition can be used to automate tasks such as logging into accounts or making phone calls, which can save time and make everyday tasks more convenient for users.
3) *Better Accuracy:* Recognition technology can be more accurate than other forms of authentication, such as passwords or PIN numbers.
4) *Enhanced Customer Experience:* Speaker recognition can be used in customer service environments to improve the user experience by allowing customers to quickly and easily identify themselves and access their account information.
5) *Cost Savings:* In some cases, speaker recognition technology can help organizations reduce their operational costs by automating tasks and reducing the need for manual labor.

III. METHODOLOGY

A. *System Architecture*

1) *User Audio Input:* Audio files in MP3 format have been compressed to reduce their size while maintaining good audio quality. MP3 is a popular format for storing and sharing audio files because of its high compression rate. However, it's important to consider the original audio file's quality when processing MP3 files, as some audio data may be lost during the compression process. Different MP3 encoders may also result in varying levels of audio quality, so using a high-quality encoder is recommended for the best results.

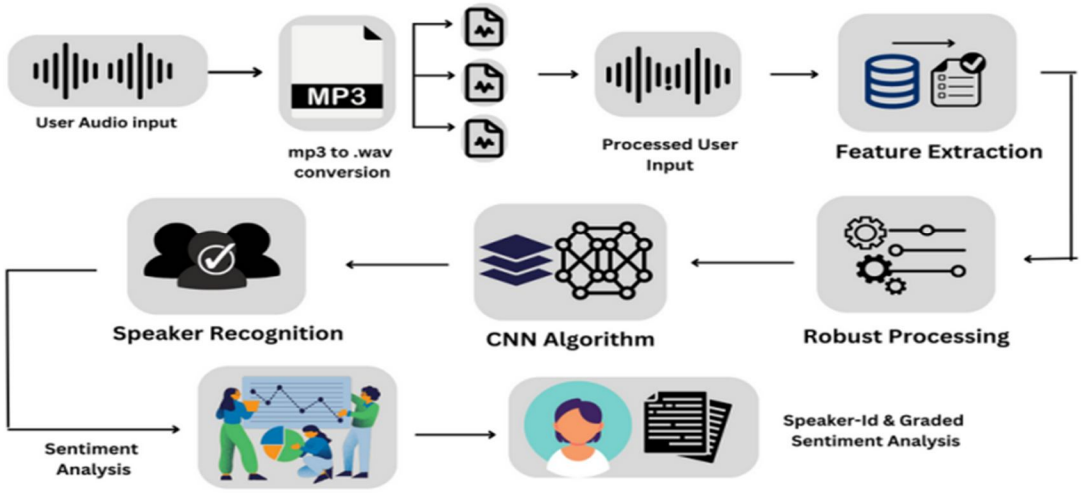


Fig. 1.System Architecture

2) *MP3 to Wave Conversion:* MP3 files undergo compression, resulting in the loss of some original audio data during encoding. To enhance audio quality, conversion of MP3 files to WAV is preferred, as WAV files are uncompressed and hold all the initial audio data. WAV files are popular in professional audio editing software, allowing more in-depth adjustments to levels and effects. These files preserve all original audio data and are perfect for archiving and maintaining high-quality audio recordings.
3) *Processed User Input:* Converting an MP3 file to Wave format involves decompressing the original audio data that was lost during MP3 compression, resulting in a higher quality and more accurate representation of the original audio. The Wave format is ideal for professional audio editing and production due to its greater accuracy and precision. The converted Wave file can be used for mixing, mastering, or as an archive copy of the original recording. However, it's important to note that Wave files have a larger file size than MP3 files due to their uncompressed nature.
4) *Feature Extraction:* To perform feature extraction on audio signals in Wave format, the raw audio data is analyzed to extract relevant characteristics that represent the audio signal. These features can include frequency content, amplitude, duration, and temporal structure. Feature extraction methods include time-domain, frequency-domain, and cepstral analysis. Extracted features include spectral, temporal, and statistical features, such as spectral centroid, zero-crossing rate, and mean signal energy. These features are useful for audio classification and signal processing applications.
5) *Robust Processing:* To enhance the quality and accuracy of audio data, robust processing of wave format audio signals involves applying techniques that can mitigate various types of distortions and noise that can occur during the recording, transmission, or storage of audio data. Techniques for robust processing include noise reduction, error correction, dynamic range compression, equalization, and source separation. These techniques can improve the signal-to-noise ratio, reduce errors, adjust the frequency response of the audio signal, and separate individual sound sources from a mixture of sound sources.
6) *CNN Algorithm:* The ResNet-based model architecture takes a 1D audio signal with a specified sample rate and channel as input. Residual blocks with varying numbers of convolutional layers and filters are used to process the input, with skip connections aiding in gradient flow. After passing through residual blocks, the output is downsampled, flattened, and passed through two fully connected layers before being fed into the output layer. The model is compiled using the Adam optimizer and sparse categorical cross-entropy loss function, and includes early stopping and model checkpoint callbacks. The build_model function takes the number of classes and model name as input arguments.
7) *Speaker Recognition:* To evaluate the performance of the speaker recognition model, a test dataset (test_ds) is created from the audio files and their corresponding labels. The audio samples are then batched, shuffled, and augmented with noise using the add_noise function. The audio waveforms are transformed into their frequency domain representation using audio_to_fft for each batch, and the model is used to predict the speaker labels with model.predict. A sample of audio files (SAMPLES_TO_DISPLAY) is randomly selected from the batch, and their predicted labels (y_pred) are compared with their true labels (labels). If the predicted label matches the true label, the message "Welcome" is printed, and the predicted label is printed along with the message "The speaker is." Otherwise, only the predicted label is printed with the message "Sorry.

8) *Sentiment Analysis:* The analyze_sentiment function is designed to perform sentiment analysis on an audio file. The function takes a filename as input, loads the audio file using the SpeechRecognition library, and transcribes the speech to text using the Google Speech Recognition API with Marathi language specified. The TextBlob class from the textblob library is used to analyze the sentiment of the transcribed text and returns a polarity score between -1 (negative sentiment) and +1 (positive sentiment). Finally, the sentiment score is printed with a label indicating whether the sentiment is positive, negative, or neutral.
9) *Graded Sentiment Analysis:* To analyze the sentiment of the transcribed text, the code uses the SentimentIntensityAnalyzer object from the nltk library. It loads the audio file of the speaker using the SpeechRecognition library and records the audio data with the record method of the AudioFile object. Then, the recognize_google method of the SpeechRecognition library is used to transcribe the speech, specifying the language of the speech as Marathi. The polarity_scores method of the SentimentIntensityAnalyzer object is then used to obtain sentiment scores, including a compound score that ranges from -1 (very negative) to 1 (very positive). The sentiment scores are printed to the console using the print function. The accuracy of the sentiment analysis may depend on the quality of the transcribed text and the accuracy of the speech recognition.

B. Algorithm

1) Converting audio from MP3 to Wave

a) Open WavePad and load the MP3 file you want to convert.
b) Go to the "File" menu and select "Save As".
c) Choose "Waveform Audio File Format (WAV)" as the output format.
d) Select the output location and file name.
e) Click "Save" to begin the conversion process.

Once the conversion is complete, use the editing tools to split the audio file at desired points and save the split files to the desired location.

2) Speaker Recognition

Step 1: Data Preparation
Prepare labeled audio recordings of speakers for training and evaluation
Split the data into training and validation sets

Step 2: Feature Extraction
Extract speech features from audio recordings, such as MFCC, PLP, or other relevant features
Normalize the feature values if needed

Step 3: Model Definition

- Define a function called `residual_block` that takes in a tensor, the number of filters, and other optional arguments such as the number of convolutional layers and the activation function.
- In the `residual_block` function, create a 1D convolutional layer with a filter size of 1 to downsample the tensor.
- Using a loop, add multiple 1D convolutional layers with the specified number of filters, followed by an activation function.
- Add another 1D convolutional layer with the specified number of filters.
- Add the downsampled tensor from step 2 to the output of the final convolutional layer.
- Apply the activation function to the output tensor.
- Add a max pooling layer to downsample the tensor further.
- Define a function called `build_model` that takes in the input shape and number of classes as arguments.

Step 4: Model Training
Compile the model by specifying the loss function, optimizer, and evaluation metrics
Train the model using the prepared training data and validation data
Monitor the training progress and tune hyperparameters as needed

Step 5: Model Evaluation
Evaluate the trained model on a separate test set or cross-validation set
Compute performance metrics such as accuracy, precision, recall, F1-score, etc.

Step 6: Prediction/Inference
Use the trained model to make predictions on new, unseen audio recordings.

3) Sentiment analysis
Step 1: Import Libraries
Import the TextBlob library in Python
Import SpeakerRecognition as sr from Python

Step 2: Sentiment Analysis
- Define a function analyze_sentiment that takes a filename as input.
- Create a speech recognizer object r from the Recognizer() class in the speech_recognition library.
- Open the audio file specified by the filename using the AudioFile() class of the speech_recognition library.
- Use the record() method of the Recognizer() object to record the audio from the AudioFile() object.
- Use the recognize_google() method of the Recognizer() object to transcribe the speech to text.
- Specify the language of the speech to be Marathi using the language parameter of the recognize_google() method.
- Create a TextBlob object blob from the transcribed text using the TextBlob() class of the textblob library.
- Compute the polarity of the sentiment using the sentiment.polarity attribute of the TextBlob object.
- Check if the sentiment polarity is positive, negative, or neutral, and print the corresponding sentiment score and label.
- Call the analyze_sentiment function with a sample audio file specified by the filename variable.

C. Mathematical Model
The residual block function takes an input `x` of shape `(n, c)` and produces an output `y` of the same shape, where `filters` is the number of filters and `conv_num` is the number of convolutional layers in the block. The operations in the block can be expressed mathematically as:

1) `s = Conv1D(filters, 1, padding='same')(x)`, where `s` is the result of applying a 1D convolution with filter size 1 to `x` with `filters` output filters and same padding.

2) `for i in range(conv_num - 1):`
`x = Conv1D(filters, 3, padding='same')(x)`, where `x` is the result of applying a 1D convolution with filter size 3 to `x` with `filters` output filters and same padding.

`x = Activation(activation)(x)`, where `activation` is the activation function used in the block (default is ReLU).

3) `x = Conv1D(filters, 3, padding='same')(x)`, where `x` is the result of applying a 1D convolution with filter size 3 to `x` with `filters` output filters and same padding.
4) `y = Add()([x, s])`, where `y` is the sum of `x` and `s`.
5) y = Activation(activation)(y)`, where `activation` is the activation function used in the block.

The model architecture is a series of residual blocks with increasing number of filters, followed by average pooling, flatten, and dense layers. Each residual block takes an input of shape `(n, c)` and produces an output of the same shape. The operations in the model can be expressed mathematically as:
- `x = residual_block(inputs, 16, 2)`
- `x = residual_block(inputs, 32, 2)`
- `x = residual_block(inputs, 64, 3)`

- `x = residual_block(inputs, 128, 3)`
- `x = residual_block(inputs, 128, 3)`
- `x = AveragePooling1D(pool_size=3, strides=3)(x)`, where `x` is the result of applying 1D average pooling with pool size 3 and stride 3 to the input.
- `x = Flatten()(x)`, where `x` is flattened to a 1D vector.
- `x = Dense(256, activation='relu')(x)`, where `x` is the result of applying a fully connected layer with 256 units and ReLU activation.
- `x = Dense(128, activation='relu')(x)`, where `x` is the result of applying a fully connected layer with 128 units and ReLU activation.
- `outputs = Dense(num_classes, activation='softmax', name='output')(x)`, where `outputs` is the output of the model, a probability distribution over the `num_classes` output classes, obtained by applying a fully connected

IV. TECHNOLOGY STACK

1) *OS:* Python's OS module is a part of the standard utility modules that provides a set of functions for working with the operating system. It allows developers to interact with the file system and execute system commands in a platform-independent manner. The OS module is built on top of platform-specific modules, such as posix or nt, to provide a consistent API for different operating systems.
2) *TenserFlow:* TensorFlow is a popular open-source machine learning framework that was developed by Google. It provides a flexible and efficient platform for developing a wide range of machine learning and deep learning applications, including speech and image recognition, natural language processing, recommendation systems, and more. TensorFlow comes with a comprehensive set of tools and APIs that enable developers to build, train, and deploy machine learning models at scale.
3) *Keras:* Keras is a Python-based open-source deep learning framework that provides a simple and intuitive API for building and training neural network models. It emphasizes ease of use, modularity, and fast prototyping, making it a popular choice for researchers and developers working on deep learning projects. With Keras, developers can easily build and train different types of models, including feedforward neural networks, convolutional neural networks (CNN), and recurrent neural networks (RNN), among others. It also supports multiple backends, such as TensorFlow, Theano, and Microsoft Cognitive Toolkit (CNTK).
4) *PyAudio:* Pyaudio is a Python library that enables developers to interact with audio devices, such as microphones and speakers, using the PortAudio library, which is a cross-platform audio I/O library. With Pyaudio, developers can record and play audio, and manipulate audio streams in real-time, all within Python code, making it a convenient and easy-to-use library for audio-related projects.
5) *Neural Net:* A neural network is a type of machine learning model that is inspired by the structure and function of the human brain. It consists of a network of interconnected nodes, or artificial neurons, that work together to recognize patterns and relationships in data. By adjusting the weights and biases of the connections between neurons, the network can learn to make accurate predictions or classifications based on input data.
6) *NLTK:* NLTK (Natural Language Toolkit) is a widely used Python library for natural language processing (NLP) tasks. It offers a comprehensive suite of tools and resources for processing and analyzing text data, including tokenization, part-of-speech tagging, sentiment analysis, named entity recognition, language modeling, and much more.
7) *NumPy:* NumPy is a powerful library for numerical computing in Python. It is widely used for scientific computing and data analysis due to its efficient array operations and mathematical functions. NumPy's multidimensional array object enables fast and easy manipulation of large datasets, making it a valuable tool for many applications, including machine learning, image processing, and more.
8) *Speech Recognition:* SpeechRecognition is a popular Python library that simplifies the use of speech recognition APIs, including Google Speech-to-Text, IBM Watson, Microsoft Azure, and more. With support for multiple audio formats and options for handling audio input from various sources, SpeechRecognition offers a straightforward and user-friendly interface for performing speech recognition tasks in Python.
9) *TextBlob:* TextBlob is a powerful Python library for natural language processing (NLP) that enables developers to perform various NLP tasks, such as sentiment analysis, noun phrase extraction, part-of-speech tagging, language translation, and more, using an easy-to-use API.

10) *Vader Lexicon:* VADER (Valence Aware Dictionary and Sentiment Reasoner) is a widely-used sentiment analysis tool that is specifically designed for analyzing text in social media. It is available as a pre-trained model in the NLTK (Natural Language Toolkit) library for Python. The VADER lexicon includes a list of words and phrases that have been labeled with sentiment scores ranging from -1 (most negative) to +1 (most positive). Additionally, the lexicon includes rules for handling negation, intensifiers, and other linguistic features that can impact sentiment analysis on social media text.
11) *Shutil:* The shutil module in Python offers high-level functions for working with directories and files. It provides a convenient way to perform various file-related operations such as copying, moving, and deleting files, and also enables archiving and compressing files and directories. This module is included in the Python Standard Library, which means that it can be used without requiring any additional installations.
12) *Subprocess:* The subprocess module in Python enables launching and interacting with new processes, and obtaining their exit codes. It allows executing system commands and programs, capturing their output, and connecting to their input/output/error pipes. With a high-level interface for executing external commands and a low-level interface for working with processes directly, the subprocess module is platform-independent, making it compatible with multiple operating systems, including Linux, Windows, and macOS.
13) *Multi-processing:* The multiprocessing module in Python enables the creation of multiple processes to run in parallel, similar to the threading module. It provides a Process class that allows a new child process to be spawned, which can then run concurrently with the parent process. The parent process can communicate with the child process and wait for it to complete its task using an API.
14) *Sentiment Intensity Analyzer:* The Sentiment Intensity Analyzer is a useful tool for performing sentiment analysis on text. It not only identifies the sentiment of the text (positive, negative, or neutral), but also measures the strength or intensity of the sentiment. The tool works by assigning numerical scores to individual words or phrases in the text based on their polarity and intensity. The polarity score ranges from -1 to 1, while the intensity score ranges from 0 to 1. The Sentiment Intensity Analyzer relies on a lexicon or dictionary of words and phrases that have been previously assigned polarity and intensity scores. This lexicon can be customized for a specific domain or context to enhance the accuracy of sentiment analysis results.
15) *Pathlib:* Python's pathlib module offers an object-oriented way to interact with the file system, replacing the os.path module's limitations. Introduced in Python 3.4, pathlib facilitates file paths, directories, and operations in a more intuitive way. It offers a range of classes and methods to represent paths as objects, simplifying file system operations.
16) *Random:* The Python Random module is a built-in module that generates pseudo-random numbers. It is commonly used to perform various random actions like generating random numbers, printing a random value for a list or string, and more.
17) *CSV:* The csv module is a tool that facilitates reading and writing of tabular data in CSV format. It enables developers to write data in the preferred format of Excel or read data from an Excel-generated file, without the need for detailed knowledge of the CSV format used by Excel.

V. CONCLUSION

Our proposed system is a deep neural network-based model for speaker recognition and tonal speech detection. It is designed to recognize speakers based on their speech signals and then analyze their speech to detect emotional tones, such as joy, sorrow, aggression, and others. The system is specifically developed for Marathi, one of the Indian Regional Languages, and other audio signals. While there have been numerous studies on tonal speech recognition systems for Asian Continental Languages and Indo-European Languages, very little work has been done for Indian Regional Languages in the tonal speech recognition system. Therefore, our system fills this gap and provides a secure and protected environment for speaker authentication and voice identification. As the area of speech recognition and speaker identification is continually evolving, there are vast possibilities for future enhancements. With improved speech recognition accuracy and speech quality, speech recognition can provide a secure terrain to various services that we use in our daily life by employing voice authentication. Communication will become easier and more reliable for everyone as technology advances in this field.

VI. FUTURE SCOPE

The project described above has the potential to be extended in various ways to support more advanced applications. For instance, it can be trained to perform speaker and sentiment analysis in multiple languages. Real-time analysis of audio streams can also be implemented, especially for use cases such as call centers, video conferencing, and customer support.

Additionally, the project can be integrated with popular voice assistants such as Siri, Alexa, and Google Assistant to provide personalized and accurate responses based on the user's voice. It can also be utilized in speech therapy applications, helping individuals with speech disorders or disabilities by providing feedback on their speech patterns, tone, and emotions. The project can also be utilized in various other applications such as political analysis, content analysis, and online reputation management. For instance, in political analysis, the project can be used to analyze the tone and emotions of politicians in their speeches, providing insights into their public image and their effect on the general public. In mental health applications, the project can be used to recognize and analyze the emotions expressed by patients in their speech, providing valuable insights for diagnosis and treatment. Overall, the project can be further developed and extended to cater to various future applications and advancements.

REFERENCES

[1] L. Kaushik, A. Sangwan and J. H. L. Hansen, "Sentiment extraction from natural audio streams," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 2013, pp. 8485-8489, doi: 10.1109/ICASSP.2013.6639321.

[2] Douglas O'Shaughnessy , " Automatic Speech Recognition ", Plenary in IEEE Chilecon 2015, 978-1-4673-8756-9/15/$31.00 © 2015 IEEE.

[3] T. C. Nguyen, J. Chaloupka and J. Nouza, "Study on incorporating tone into speech recognition of Vietnamese," 2015 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM), Liberec, Czech Republic, 2015, pp. 1-6, doi: 10.1109/ECMSM.2015.7208688.

[4] S. Maghilnan and M. R. Kumar, "Sentiment analysis on speaker specific speech data," 2017 International Conference on Intelligent Computing and Control (I2C2), Coimbatore, India, 2017, pp. 1-5, doi: 10.1109/I2C2.2017.8321795.

[5] Srujana K , R.Ramesh , Kiran G., Ch. Manikanta, " Artificial Intelligence Speech Recognition System using MATLAB " 978-1-5386-3243-7/17/$31.00 ©2017 IEEE.

[6] X. Zhao and Y. Wei, "Speaker Recognition Based on Deep Learning," 2019 IEEE International Conference on Real-time Computing and Robotics (RCAR), Irkutsk, Russia, 2019, pp. 283-287, doi: 10.1109/RCAR47638.2019.9044086.

[7] M. Wang, T. Sirlapu, A. Kwasniewska, M. Szankin, M. Bartscherer and R. Nicolas, "Speaker Recognition Using Convolutional Neural Network with Minimal Training Data for Smart Home Solutions," 2018 11th International Conference on Human System Interaction (HSI), Gdansk, Poland, 2018, pp. 139-145, doi: 10.1109/HSI.2018.8431363.

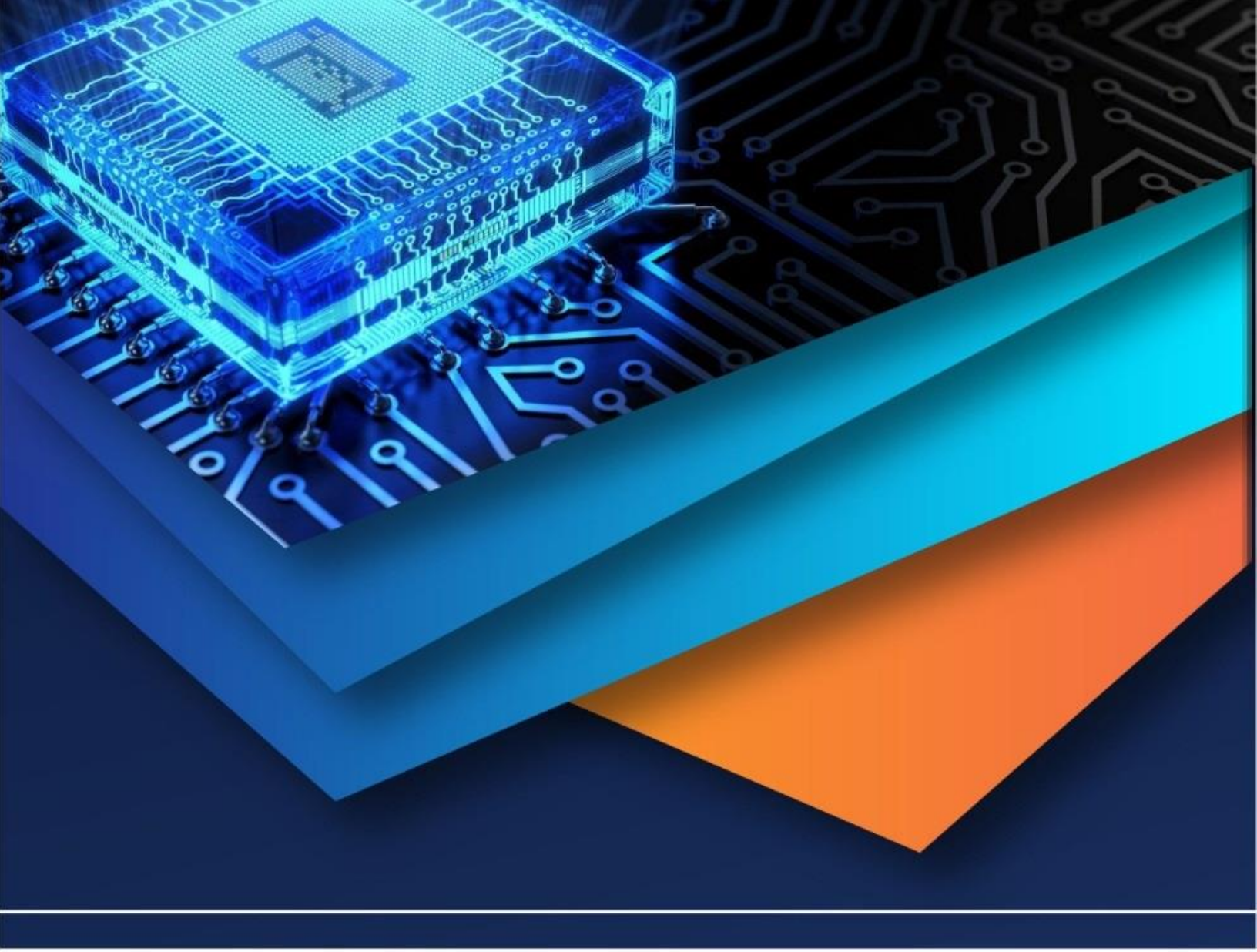
[8] M. Hamidi, H. Satori, N. Laaidi and K. Satori, "Conception of Speaker Recognition Methods: A Review," 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Meknes, Morocco, 2020, pp. 1-6, doi: 10.1109/IRASET48871.2020.9092118.

[9] Y. Chandra and A. Jana, "Sentiment Analysis using Machine Learning and Deep Learning," 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2020, pp. 1-4, doi: 10.23919/INDIACom49435.2020.9083703.

[10] Feng Ye 1,2 and Jun Yang 1,* , " A Deep Neural Network Model for Speaker Identification " , Appl. Sci. 2021, 11, 3603.

[11] M. Bansal, S. Yadav and D. K. Vishwakarma, "A Language-Independent Speech Sentiment Analysis Using Prosodic Features," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2021, pp. 1210-1216, doi: 10.1109/ICCMC51019.2021.9418357.

[12] Sakshi Dua 1, Sethuraman Sambath Kumar 1, Yasser Albagory 2, Rajakumar Ramalingam 3, Ankur Dumka 4,5, Rajesh Singh 6, Mamoon Rashid 7,*, Anita Gehlot 6, Sultan S. Alshamrani 8 and Ahmed Saeed AlGhamdi 2 , " Developing a Speech Recognition System for Recognizing Tonal Speech Signals Using a Convolutional Neural Network " , Appl. Sci. 2022, 12, 6223.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY