iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IV **Month of publication:** April 2024

DOI: https://doi.org/10.22214/ijraset.2024.60714

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 12 Issue IV Apr 2024- Available at www.ijraset.com

Speech to Text Transcription

Sai Teja Ramacharla[1], Vustepalle Aniketh[2], Dr. M. Senthil Kumaran[3]
[1, 2]*Computer Science and Engineering SCSVMV University, Kanchipuram, Tamilnadu*
[3]*Assistant Professor, Dept of CSE, SCSVMV University Kanchipuram, Tamilnadu*

Abstract: ***This paper aims to enhance speech recognition and audio processing, converting spoken sentences into text and taking input from various input sources like microphones, audio, and video files. Notably, it offers robust audio conversion capabilities, supporting MP3 to WAV and other formats.***
To enhance the scalability and user experience, the system is implemented as a Flask-powered web application, providing users with a seamless interface accessible through a Flask-powered web browser serves to facilitate intuitive and user-friendly communication, making the application versatile, different users -and adaptable to features here the first. This comprehensive design meets the needs of users looking for efficient speech recognition and audio processing solutions, especially for web-based applications.
Keywords: Audio Conversion, Speech Recognition, Flask, User-friendly Interface, Web Application.

I. INTRODUCTION

This project introduces the Speech to Text Transcript solution, aiming to convert spoken words into text using sophisticated speech recognition techniques implemented in Python. It utilizes specialized libraries like Speech Recognition and PyDub, offering versatile input options including microphones, audio files, and video files.

A. Key Features

1) Seamless audio file conversion across various formats, ensuring compatibility with a wide range of inputs. Supports popular formats such as MP3 to WAV for flexible handling.
2) Implemented as a user-friendly web application using Flask, making it accessible through web browsers for easy usage.
3) Integration of speech recognition capabilities using the Speech Recognition library, supporting both microphone input and audio file input for accurate conversion. Further enhanced with Google Web Speech API integration.
4) Extends functionality to video files, allowing annotation of comments from videos by intelligently extracting audio and applying speech recognition techniques.
5) From a coding perspective, the primary functionality is encapsulated within the app.py file. It orchestrates tasks such as audio extraction from videos, MP3 to WAV conversion using PyDub, and speech recognition configuration.

II. EXISTING SYSTEM

Existing systems mainly focus on speech recognition and convert spoken language into transcriptions for transcription and text-based analysis.
State-of-the-art technologies consider performance against hardware requirements, while incorporating methods and technologies for efficient human-computer interaction into the multidisciplinary field of speech recognition.

Draw backs of Existing System

1) Manual file conversion leads to time and effort consumption.
2) Lack of integration among different tools for processing tasks.
3) User experience is less than optimal due to the system's drawbacks.
4) Inconsistent results and errors due to manual processes.
5) Overall inefficiency that hampers content creation and professional work.
6) The existing systems does not support the saving of the transcript in a .txt format.

III. LITERATURE REVIEW

In the literature survey, several papers have been explored different strategies and items :

1) "Adapting Large Language Model with Speech for Fully Formatted End-To-End Speech Recognition" presented by Shaoshi Ling, Yuxuan Hu and etal. This research builds upon the existing landscape of end-to-end (E2E) speech recognition models, which typically consist of encoder and decoder blocks for acoustic and language modeling functions. While pretrained large language models (LLMs) have demonstrated potential to enhance E2E automatic speech recognition (ASR) performance, integrating them has faced challenges due to mismatches between text-based LLMs and those used in E2E ASR. This paper takes a novel approach by adapting pretrained LLMs to the domain of speech. The authors explore two model architectures: an encoder-decoder-based LLM and a decoder-only-based LLM. The proposed models leverage the strengths of both speech and language models while minimizing architectural changes.

2) The "End-End Speech-to-Text translation with modality agnostic meta-learning" presented by Sathish Indurthi,Houjeung Han and etal. This research addresses the challenge of training end-to-end Speech Translation (ST) models with limited data, a common issue in the field due to the difficulty in collecting large parallel speech-to-text datasets. The authors propose a novel modality-agnostic meta-learning approach that leverages transfer learning from source tasks such as Automatic Speech Recognition (ASR) and Machine Translation (MT). Unlike previous transfer learning methods, the proposed approach employs a meta-learning algorithm, specifically the Model-Agnostic Meta-Learning (MAML) algorithm, to update parameters in a way that serves as a robust initialization for the target ST task.

3) "fairseq S2T: Fast Speech-to-Text Modeling with fairseq" presented by Changhan Wang, Yun Tang and etal. This research introduces FAIRSEQ S2T, an extension of the FAIRSEQ toolkit designed for speech-to-text (S2T) tasks, encompassing end-to-end modeling for speech recognition and speech-to-text translation. The authors highlight the growing importance of end-to-end sequence-to-sequence (S2S) models in the realm of S2T applications, citing their success in automatic speech recognition (ASR) and the resurgence of speech-to-text translation (ST) research. The paper emphasizes the interconnectedness of ASR, ST, machine translation (MT), and language modeling (LM), advocating for comprehensive S2S modeling toolkits to address the evolving landscape of these tasks.

4) "Arabic Automatic Speech Recognition: A Systematic Literature Review" presented by Amira Dhouib, Achraf Othman and etal. The systematic literature review (SLR) comprehensively explores the landscape of Automatic Speech Recognition (ASR) with a specific focus on the Arabic language. Covering a period from 2011 to 2021, the study addresses seven key research questions to shed light on the trends and advancements in Arabic ASR research. The authors identified 38 relevant studies across five databases that met their inclusion criteria. The results showcase the utilization of various open-source toolkits for Arabic ASR, with KALDI, HTK, and CMU Sphinx emerging as the most prominent ones. Notably, the review highlights the predominant use of Modern Standard Arabic (MSA) in 89.47% of the studies, while 26.32% explore different Arabic dialects.

5) The "Leveraging weakly supervised data to improve end-to-end speech-to-text translation" presented by Jia, Y., Johnson, and etal. This research delves into the realm of end-to-end Speech Translation (ST) models, emphasizing their potential advantages over traditional cascaded models involving Automatic Speech Recognition (ASR) and text Machine Translation (MT). The study acknowledges the challenges in training robust end-to-end ST models due to the scarcity of large parallel corpora containing speech and translated transcript pairs. It builds upon prior research efforts that leverage pre-trained components and multi-task learning to utilize weakly supervised training data, such as speech-to-transcript or text-to-foreign-text pairs.

The authors propose a novel approach, demonstrating that using pre-trained MT or text-to-speech (TTS) synthesis models to convert weakly supervised data into speech-to-translation pairs for ST training can be more effective than multi-task learning.

IV. PROPOSED METHODOLOGY

The proposed technique introduces a speech recognition system designed to deal with a variety of audio inputs seamlessly. It offers VI three most important methods to input records: through a microphone for real-time audio, via audio documents, or via processing video documents.

This system is predicated at the speech_ recognition library to carry out speech recognition, which smoothly integrates with the Google Web Speech API. Depending on the input approach chosen, the system captures audio, strategies it, and then works to decipher the spoken phrases. Furthermore, users can shop the transcripts in a convenient .txt format for smooth access and reference.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

V. SYSTEM ARCHITECTURE

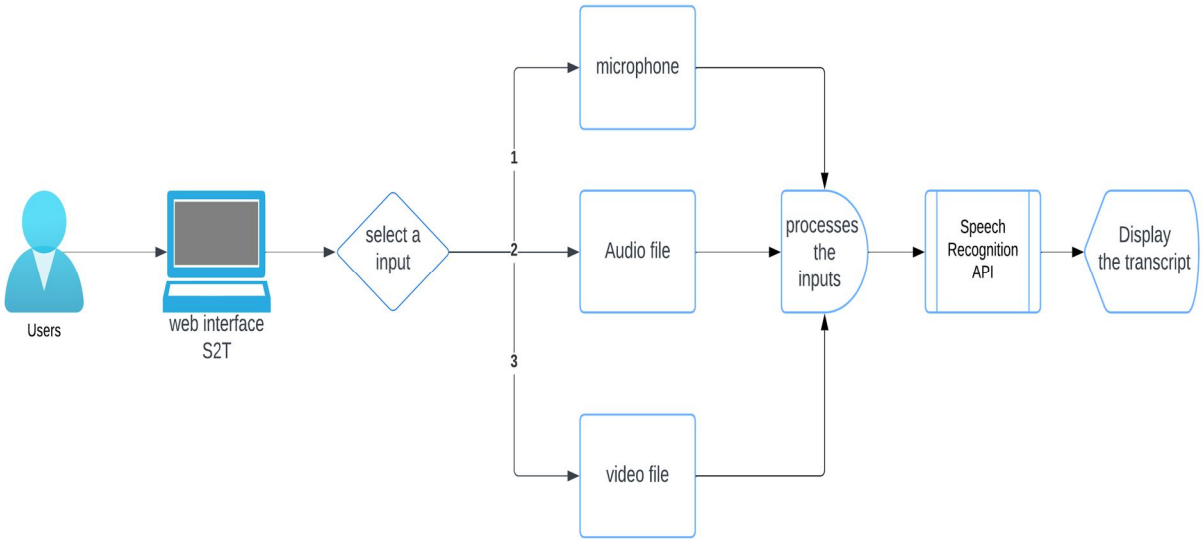


Fig. 1. System Architecture

VI. IMPLEMENTATION PROCESS

We have used python programming language to build our project in the backend. This implements a web application for speech recognition using Flask, a Python web framework, and various libraries such as Speech Recognition, PyDub, and MoviePy. The application allows users to choose between different input types: microphone, audio file, or video file. The front-end is designed using HTML and CSS, and the logic for handling user interactions is implemented in JavaScript.

Here's a breakdown of the implementation process:

1) *HTML and CSS (index.html):* The HTML file defines the structure of the web page, including a form with options to select the input type (microphone, audio file, or video file). It uses CSS to style the page and link to an external stylesheet (styles.css) for additional styling.
2) *JavaScript (script.js):* The JavaScript file is responsible for dynamically displaying and hiding input options based on the selected input type. It listens for changes in the input type dropdown and adjusts the visibility of input divs accordingly. It also provides a theme switcher button to switch between light and dark themes.
3) *Flask Backend (app.py):* The Flask application is set up with routes to handle both GET and POST requests. The main route ('/') renders the index.html template. The server-side logic is implemented in Python, utilizing the Speech Recognition library for speech-to-text functionality and other libraries for audio and video processing.The index route handles form submissions, extracting audio from the selected input type (microphone, audio file, or video file), and performing speech recognition using Google Web Speech API. The resulting transcript is displayed on the web page.
4) *Audio and Video Processing (app.py):* The application includes functions to extract audio from a video file and convert MP3 audio to WAV format using MoviePy and PyDub libraries, respectively. It uses a pre-defined trigger and response for an easter egg scenario, where if the recognized speech contains a specific phrase, the transcript is replaced with a different response.
5) *Running the Application:* The application is run using app.run(debug=True) in the __main__ block. This starts the Flask development server. Users can access the web application through a web browser, interact with the input form, and observe the real-time transcription based on the selected input type. In summary, this implementation integrates front-end elements with a Flask backend to create a user-friendly web application for speech recognition, supporting different input sources. The server-side logic leverages various libraries to handle audio and video processing, making it a comprehensive solution for speech recognition tasks.

VII. RESULTS

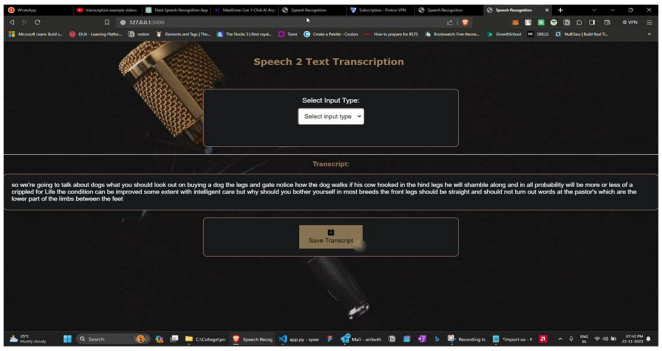


Fig2. (Audio File)

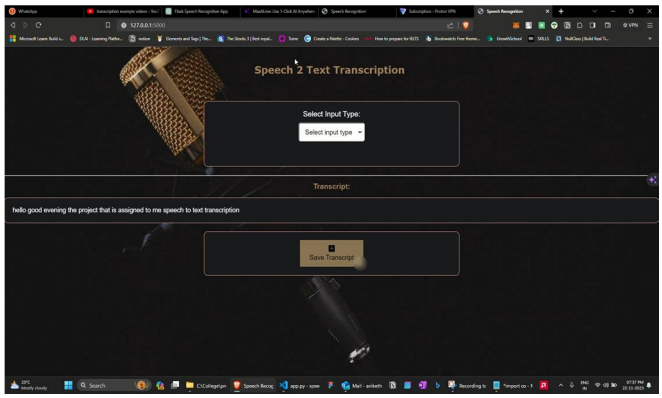


Fig 3. (Microphone Output)

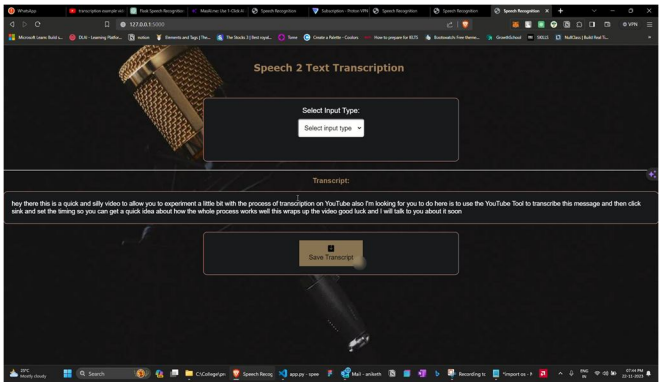


Fig 4. (Video File Transcription)

VI. CONCLUSION

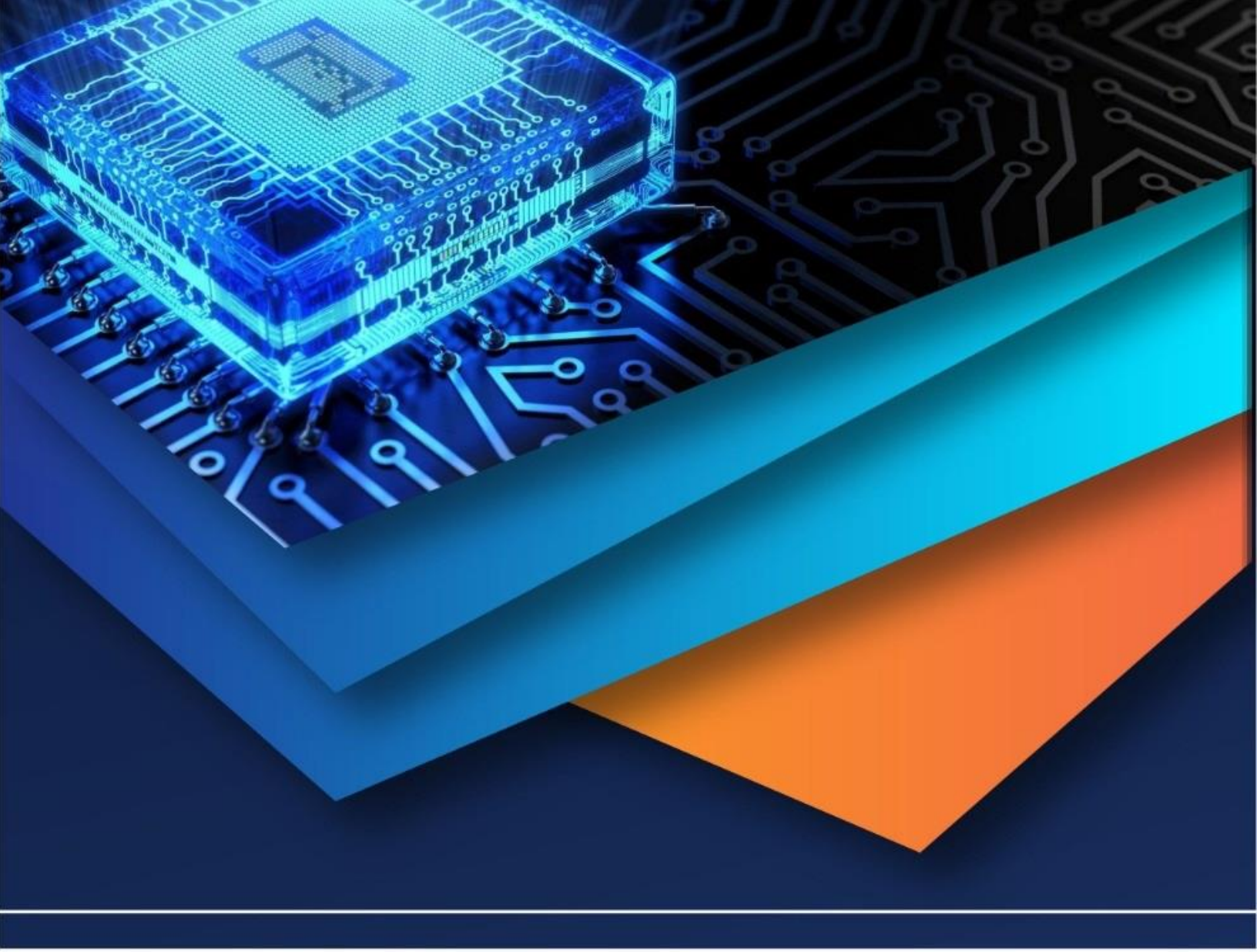
In summary, the implemented speech recognition system is an exceptionally versatile tool, capable of handling a variety of inputs including microphones, audio files, video files and cameras handle Leveraging the speech_recognition library, the system captures, processes and accurately recognizes spoken words , provides a comprehensive solution for various applications The addition of features such as error handling ensures robustness, and increases system reliability in in a real-world setting. The functionality of the system is also highlighted by the integration of Flask, which enables web-based interaction. Users can select their preferred input method and participate in the system easily using an intuitive interface. The flexibility and interactivity of the system positions it as a valuable tool for a wide range of speech. The web application provides an intuitive and dynamic experience for those seeking more accurate and responsive speech reading capabilities Overall, this speech recognition system, with its useful features , and its customizable structure, stands as a powerful and user-centered application in artificial intelligence.

VII. FUTURE SCOPE

To implement more Transformational enhancements in efforts to improve speech and text technology. First, the aim is to introduce multilingual support, allowing users to easily switch between preferred languages. At the same time, the use of model fine-tuning, noise reduction algorithms to increase accuracy in speech recognition. Furthermore, we can add voice command functionality to various applications, devices, and services, empowering users with automation and control capabilities. We are committed to empowering By leveraging the power of natural language processing (NLP), our system will skillfully extract meaning and information relevant from a receptive language, to facilitate more nuanced communication. Additionally, real-time transcription capabilities become essential, providing valuable support for increasing note-taking and coherence during lectures, speeches. Finally, by integrating cloud-based speech recognition services, we will ensure scalability and flexibility and further increase insight accuracy, we will prioritize the system in up-to-date coding.

REFERENCES

[1] “Arabic Automatic Speech Recognition: A Systematic Literature Review” Amira Dhouib ,Achraf Othman ORCID,Oussama El GhoulORCID,Mohamed Koutheair KhribiORCID andAisha Al Sinani - 2022.
[2] ”fairseq S2T: Fast Speech-to-Text Modeling with fairseq”. Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Sravya Popuri, Dmytro Okhonko, Juan Pino.- 2020.
[3] “Adapting Large Language Model with Speech for Fully Formatted End-to-End Speech Recognition” Shaoshi Ling, Yuxuan Hu, Shuangbei Qian, Guoli Ye, Yao Qian, Yifan Gong, Ed Lin, Michael Zeng - 2021.
[4] “Leveraging Weakly Supervised Data To Improve End-To-End Speech-To-Text Translation”. Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J. Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, Yonghui Wu - 2019.
[5] “End-end Speech-to-Text Translation with Modality Agnostic Meta-Learning” Sathish Indurthi; Houjeung Han; Nikhil Kumar Lakumarapu – 2020.
[6] “wav2vec 2.0 : A Framework for Self-Supervised Learning of Speech Representations” Alexei Baevski, Henry Zhou, Abdel-rahman Mohamed, Michael Auli - 2020.
[7] “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition” - Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, Quoc V. Le – 2019.
[8] “Conformer: Convolution-augmented Transformer for Speech Recognition”. Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, Ruoming Pa - 2020.
[9] Hybrid CTC/attention architecture for end-to-end speech recognition. Watanabe, S., Sainath, T. N., Prabhavalkar, R., Pratap, V., & Variani, E. (2017).
[10] Audio augmentation for speech recognition. Ko, T., Peddinti, V., Povey, D., Khudanpur, S., & Zhang, Z. (2015).
[11] Speech recognition with deep recurrent neural networks. Graves, A., Mohamed, A. R., & Hinton, G. (2013).
[12] Deep neural networks for acoustic modeling in speech recognition. Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. (2012).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY