

**Sector 29, Pradhikaran, Akurdi, Pune - Maharashtra, INDIA 411044**

(Establishment by Maharashtra Act No. LXIII of 2017)

**School of Computer Science, Engg. & Applications**

**Research Article**

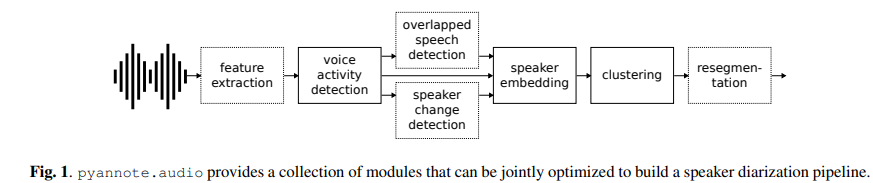
1. **Name of student: Akshita Nayak**
   1. **PRN: 20190802100**
   2. **Email Id: 20190802100@dypiu.ac.in**
2. **Title: Real time Speech Recognition & Diarization using Python Libraries and its Frameworks**
3. **Abstract:**

Creating of speech recognition application requires advanced speech processing techniques realized by specialized speech processing software. It is very possible to improve the speech recognition research by using frameworks based on open-source speech processing software.

In this project, we'll build a system that can record live speech using your microphone, and then transcribe it using speech recognition. This can be used to automatically record and transcribe meetings, lectures, and other events.

Speaker Diarization is the process which aims to find who spoke when in an audio and total number of speakers in an audio recording.

We introduce pyannote.audio, an open-source toolkit written in Python for speaker diarization. Based on PyTorch machine learning framework, it provides a set of trainable end-to-end neural building blocks that can be combined and jointly optimized to build speaker diarization pipelines. pyannote.audio also comes with pre-trained models covering a wide range of domains for voice activity detection, speaker change detection, overlapped speech detection, and speaker embedding – reaching state-of-the-art performance for most of them.



We'll be using Jupyter notebook to write our code, and to build the interactive widgets to start and stop recording. By the end, you'll be able to click a button, and have speech be automatically recorded and transcribed.

**Project Steps**

This project contains:

* Voice Activity Detection (webrtcvad)
* Speaker Segmentation based on Bi-LSTM

Voice activity detection (VAD) is a technique in which the presence or absence of human speech is detected. This part has been completed using a module developed by google called as WebRTC. It's an open framework for the web that enables Real-Time Communications (RTC) capabilities in the browser. The voice activity detector is one of the specific module present in WebRTC. This basic working of WebRTC based VAD is as,

* WebRTC VAD is a Gaussian Mixture Model(GMM) based voice activity detector
* GMM model using PLP features
* Two full covariance Gaussians: One for speech, and one for Non-Speech is used. To learn about PLP we followed this paper:  
  [link](http://www5.informatik.uni-erlangen.de/Forschung/Publikationen/2005/Hoenig05-RPL.pdf).

1. **Introduction:**

**What is speech recognition?**

The capacity of technology to obey spoken commands. Speech recognition provides input for automatic translation, generates print-ready dictation, and allows hands-free operation of various devices and equipment—all of which are especially helpful to many disabled people. Medical dictation software and automated telephone systems were some of the first speech recognition applications. In particular in professions that depend on specific vocabularies, it is frequently used for dictation, for accessing databases, and for issuing commands to computer-based systems. It also makes personal assistants like Apple's Siri possible for cellphones and vehicles.

**What advantages does speech recognition technology offer, then?** When typing is typically faster (and quieter), why exactly do we need computers to comprehend our speech? For many programmes that don't run on computers, which are becoming more and more widespread, speech offers a natural interface. Here are a few significant ways that speech recognition technology affects people's life.

**Digital devices** must be controlled verbally in order for digital personal assistants like Alexa and Google Home to function. They serve as excellent illustrations of how computers might learn to interpret your voice over time by using machine learning. However, the use of voice recognition technology, made possible by signal processing, is essential.

**Helping the Hearing- and Vision-Impaired:** Many people with visual impairments use screen readers and text-to-speech dictation programmes to communicate. Additionally, text transcription from audio can be a vital communication tool for hearing-impaired people.

**Enabling Hands-Free Technology:** Speech is tremendously helpful when your eyes and hands are occupied, like while you're driving. You'll be less likely to get lost and won't need to stop to figure out how to use your phone or read a map if you can communicate with Apple's Siri or Google Maps to get you where you need to go.

**The following reasons make speech recognition technology a growth skill:**

Although it is already a part of our daily lives, speech recognition technology is currently only capable of recognising relatively straightforward commands.

Researchers will be able to develop more intelligent systems that can comprehend conversational speech as technology develops (remember the robot job interviewers?). One day, your computer will be able to respond to your questions with logic in the same way that you would respond to a human being. The development of signal processing technology will enable all of this. Numerous businesses are seeking exceptional individuals who want to work in this industry because there are an increasing number of professionals that are needed. A speech signal's processing, interpretation, and understanding provides the foundation for many cutting-edge new technologies and communication techniques. In light of current developments, voice recognition technology will represent a rapidly expanding (and drastically altering) branch of signal processing for years to come.

1. **Literature Survey:**

**Paper Citation:** **pyannote.audio: neural building blocks for speaker diarization**

[Hervé Bredin](https://arxiv.org/search/eess?searchtype=author&query=Bredin%2C+H), [Ruiqing Yin](https://arxiv.org/search/eess?searchtype=author&query=Yin%2C+R), [Juan Manuel Coria](https://arxiv.org/search/eess?searchtype=author&query=Coria%2C+J+M), [Gregory Gelly](https://arxiv.org/search/eess?searchtype=author&query=Gelly%2C+G), [Pavel Korshunov](https://arxiv.org/search/eess?searchtype=author&query=Korshunov%2C+P), [Marvin Lavechin](https://arxiv.org/search/eess?searchtype=author&query=Lavechin%2C+M), [Diego Fustes](https://arxiv.org/search/eess?searchtype=author&query=Fustes%2C+D), [Hadrien Titeux](https://arxiv.org/search/eess?searchtype=author&query=Titeux%2C+H), [Wassim Bouaziz](https://arxiv.org/search/eess?searchtype=author&query=Bouaziz%2C+W), [Marie-Philippe Gill](https://arxiv.org/search/eess?searchtype=author&query=Gill%2C+M)

While pyannote.audio supports training models from the waveform directly (e.g. using SincNet learnable features [9]), the pyannote.audio.features module provides a collection of standard feature extraction techniques such as MFCCs or spectrograms using the implementation available in the librosa library [10]. They all inherit from the same FeatureExtraction base class that supports on-the-fly data augmentation which is very convenient for training neural networks. For instance, it supports extracting features from random audio chunks while applying additive noise from databases such as MUSAN [11]. Contrary to other tools that generate in advance a fixed number of augmented versions of each original audio file, pyannote.audio generates a virtually infinite number of versions as the augmentation is done on-the-fly every time an audio chunk is processed.

**Paper Citation: Speech Recognition Based on Open Source Speech Processing Software**

[Piotr Kłosowski](https://link.springer.com/chapter/10.1007/978-3-319-07941-7_31#auth-Piotr-K_osowski), [Adam Dustor](https://link.springer.com/chapter/10.1007/978-3-319-07941-7_31#auth-Adam-Dustor), [Jacek Izydorczyk](https://link.springer.com/chapter/10.1007/978-3-319-07941-7_31#auth-Jacek-Izydorczyk), [Jan Kotas](https://link.springer.com/chapter/10.1007/978-3-319-07941-7_31#auth-Jan-Kotas) & [Jacek Ślimok](https://link.springer.com/chapter/10.1007/978-3-319-07941-7_31#auth-Jacek-_limok)

SPRACH is an abbreviation for Speech Recognition Algorithms for Connection Hybrids. It involves usage of HMM (Hidden Markov Models), ANN (Artificial Neural Networks), statistical inference in said networks, as well as hybrid HMM-ANN technology in order to further improve current research on continuous speech recognition. The project was developed across multiple universities in Europe and therefore one of its main goals was to adapt hybrid speech recognitions to languages other than English.

1. **Methodology:**

**Voice activity detection:**

Voice activity detection is the task of detecting speech regions in a given audio stream or recording. It can be addressed in pyannote.audio using the above principle with K = 2: yt = 0 if there is no speech at time step t and yt = 1 if there is. At test time, time steps with prediction scores greater than a turnable threshold θVAD are marked as speech. Overall, this essentially implements a simplified version of the voice activity detector originally described in [12]. Pre-trained models are available, reaching state-of-the-art performance on a range of datasets

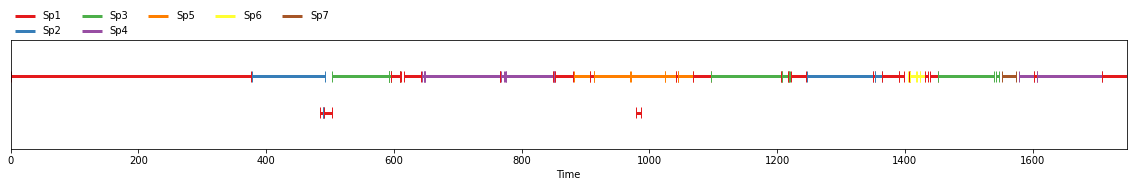
**Speaker change detection:**

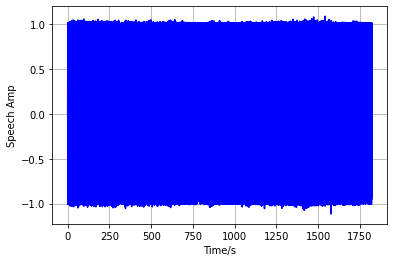
Speaker change detection is the task of detecting speaker change points in a given audio stream or recording. It can be addressed in pyannote.audio using the same sequence labelling principle with K = 2: yt = 0 if there is no speaker change at time step t and yt = 1 if there is. To address the class imbalance problem and account for human annotation imprecision, time steps {t | |t − t ∗ | < δ} in the close temporal neighbourhood of a speaker change point t ∗ are artificially labeled as positive for training. In practice, the order of magnitude of δ is 200ms. At test time, time steps corresponding to prediction scores local maxima and greater than a tunable threshold θSCD are marked as speaker change points. Overall, this essentially implements a version of the speaker change detector originally described in table. Pre-trained models are available, reaching state-of-the-art performance on a range of datasets.

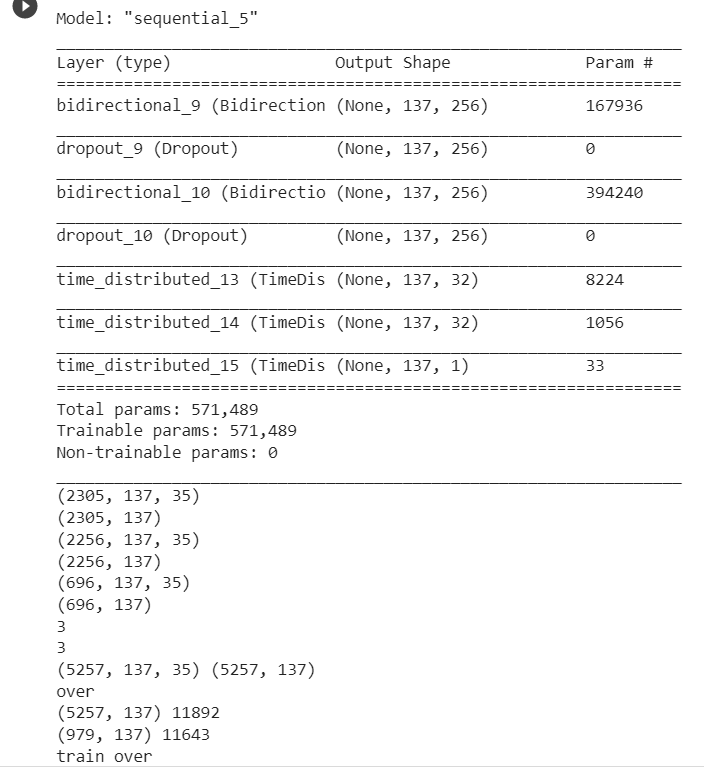
**Re-segmentation:**

Re-segmentation is the task of refining speech turns boundaries and labels coming out of a diarization pipeline. Though it is an unsupervised task, it is addressed in pyannote.audio using the same sequence labeling principle with K = κ + 1 where κ is the number of speakers hypothesized by the diarization pipeline: yt = 0 if no speaker is active at time step t and yt = k if speaker k ∈ J1; κK is active. Because the re-segmentation step is unsupervised by design, one cannot pre-train re-segmentation models. For each audio file, a new re-segmentation model is trained from scratch using the (automatic, hence imperfect) output of a diarization pipeline as training labels. Once trained for a number of epochs (one epoch being one complete pass on the file), the model is applied on the very same file it was trained from – making this approach completely unsupervised. Each time step is assigned to the class (non-speech or one of the κ speakers). with highest prediction scores. This essentially implements a version of the re-segmentation approach originally described where it was found that = 20 is a reasonable number of epochs. This re-segmentation step may be extended to also assign the class with the second highest prediction score to overlapped speech regions.

1. **Results:**







1. **Conclusion:**

Speaker Diarization is also a powerful analytic tool. By identifying and labeling speakers, product teams and developers can analyze each speaker’s behaviors, identify patterns/trends among individual speakers, make predictions, and more.

While there are clearly some limitations to Speaker Diarization today, advances in Deep Learning research are helping to overcome these deficiencies and to boost Speaker Diarization accuracy.

Clustering Algorithms such as K-means and K-shift can also be used try to find homogeneous subgroups within the data such that data points in each cluster are as similar as possible according to a similarity measure such as euclidean-based distance or correlation-based distance. The decision of which similarity measure to use is application-specific.

The Project can be extended for real time audio recognition as well.

1. **References:**

* <https://github.com/pyannote/pyannote-audio>
* <https://arxiv.org/abs/1710.10468>
* <https://www.isca-speech.org/archive/Interspeech_2017/pdfs/0411.PDF>
* <https://scikit-learn.org/stable/modules/clustering.html>
* <https://arxiv.org/abs/1710.10467>
* <https://github.com/Janghyun1230/Speaker_Verification>
* <https://github.com/yinruiqing/change_detection>
* <https://pypi.org/project/webrtcvad-wheels/>
* <https://github.com/wblgers/py_speech_seg/tree/master/BiLSTM>