Leveraging Region based Convolutional Neural Network (RCNN) with GMM Model Assisted by Hidden Markov Model for Speaker Detection

Aditya B Dhruva

Department of Instrumentation and Control Engineering, National Institute of Technology, Trichy, India 620015.

* Corresponding author: adityadhruva2003@gmail.com

doi: https://doi.org/10.21467/proceedings.7.4.2

ABSTRACT

This study suggests a novel deep learning and established methods-based speaker detection and identification system. A region-based CNN analyzes spectrograms for speaker detection, guiding a Gaussian Mixture Model (GMM) for improved speaker clustering. This approach aims to achieve higher accuracy and efficiency compared to traditional diarization methods.

Keywords: Gaussian Mixture model, Region based Convolutional Neural Network, Speaker Detection and Diarization.

1 Introduction

For processes to be optimized and productivity to be increased in the field of smart manufacturing, multiple stakeholders collaborated and communicated effectively. In applications like meeting summarization, real-time decision-making, and automated reporting, speaker diarization—the process of determining who spoke and when in a multi-speaker recording—was essential. Traditional methods frequently used libraries such as Speech Brain and Pyannote for segmentation and feature extraction (MFCCs, d-vectors). Even though these techniques worked well, they could be computationally costly and involved several processing steps, which could make real-time applications difficult in dynamic manufacturing settings.

This paper proposes a novel speaker diarization system for smart manufacturing, using deep learning to improve efficiency and accuracy. It presented a two-stage refinement strategy for speaker diarization to improve the clarity of communication in manufacturing meetings. In this paper, the predictions of a Region-based Convolutional Neural Network (R-CNN) were used to identify the initial speaker regions, which were crucial for determining the main contributors to the meeting content, such as production strategies or quality control. These regions were then used to guide the Gaussian Mixture Model (GMM), in which the GMM was used to represent a unique speaker in the manufacturing process. The GMMs were trained using the characteristics of the corresponding speaker region, which greatly improved the accuracy of the speaker identification.

Finally, a Hidden Markov Model (HMM) was used to ensure temporal consistency. This step guaranteed that the transitions between different Gaussian Mixture Models (GMMs) accurately represented the changes in the speakers over time. As a result, the approach delivered smooth speaker transitions and minimized errors that may have been caused by the initial R-CNN predictions or the GMM-based clustering. By integrating this state-of-the-art speaker diarization system into smart manufacturing environments, the authors sought to enhance communication efficiency, make better decisions, and, ultimately, increase productivity. The Research has successfully been completed and the findings recorded [1].

Proceedings DOI: 10.21467/proceedings.7.4; Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-984081-1-2

2 Functioning Of A Gaussian Mixture Model Assisted With Hidden Markov Model

The Hidden Markov Model (HMM) operated in conjunction with the Gaussian Mixture Model (GMM) by incorporating GMMs at each stage of the HMM process. The transition matrix, which was learned from training data, helped determine the probability of transitioning from one state to another. The following parameters were utilized to improve the accuracy and effectiveness of voice recognition:

1. **Pre-Emphasis:** Used a variety of methods, such as DC offset removal and silence words removal, to eliminate the noisy data.
2. **Windowing:** Windowing was the next step. From the start of each frame to the finish, we aimed to reduce the discontinuities of the signals. Each frame's window function was used for spectral analysis. In this the step window is: $b(m) = a(m) \times w(m), 0 \leq m \leq M-1$, where the number of samples in each frame is indicated by M.
3. **Feature Extraction:** For accurate recognition, the audio and voice signals were transformed into vector coefficients. The voice and audio signals are represented using the Mel Frequency Cepstral Coefficient.
4. **Pattern recognition:** It evaluates the similarities between each voice class's unseen test patterns. The hybrid model produced superior outcomes [2]-[4].

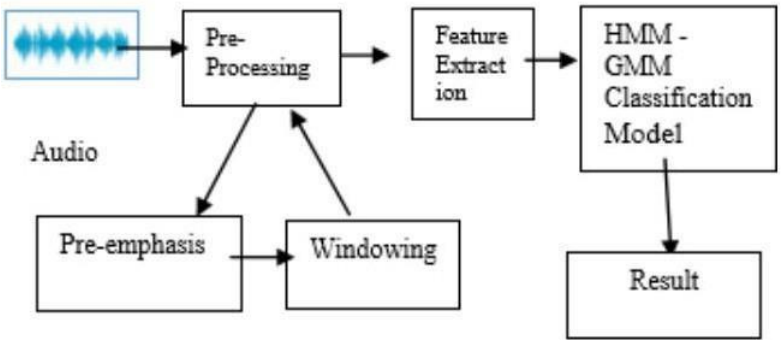

Fig 1: GMM-HMM Model Hybrid Model General Structure.

3 Basic structure of a region based convolutional neural network(R-CNN)

Let us understand the working of the Region-based Convolutional Neural Network. In computer vision, region-based convolutional neural networks, or R-CNNs, were revolutionary, especially for object detection applications. An R-CNN's basic structure entailed first producing region proposals that might contain objects. This was achieved through a selective search algorithm, which simplified the task by reducing the number of proposals to about 2000, while maintaining a high recall rate. Region proposals were then passed through a Convolutional Neural Network (CNN) to extract relevant features. A Support Vector Machine (SVM) was subsequently used to classify objects within these proposed regions. Additionally, a bounding box regressor helped accurately localize objects within the image. Over time, the original R-CNN evolved into several more efficient variants. Fast R-CNN enhanced processing speed by running the entire image through the CNN at once, while Faster R-CNN further optimized the process by integrating the region proposal mechanism directly into the network [5], [6].

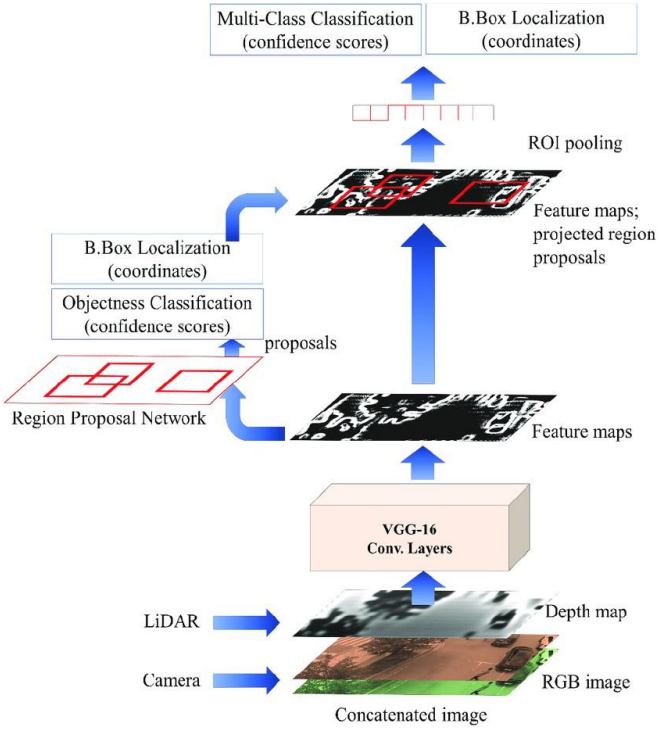

Fig 2: Working And Structure Of Region Based CNN.

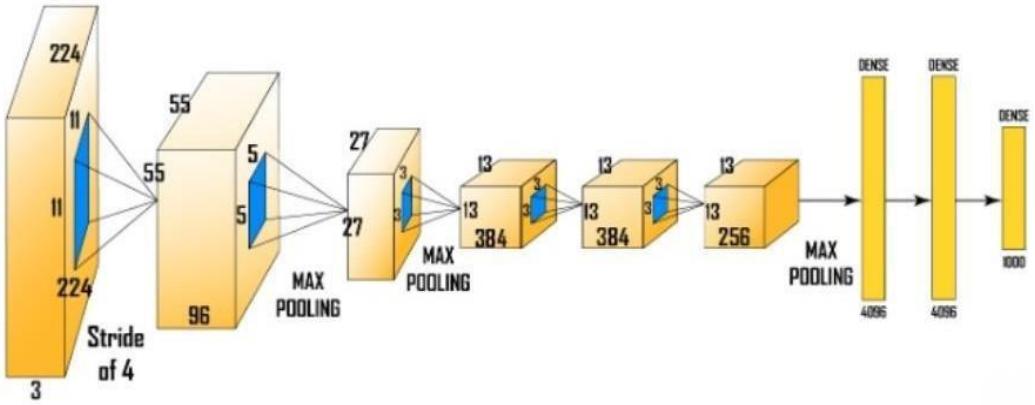

Fig 3: Working And Structure Of Region Based CNN By Taking A Dimension Value As An Example.

Proceedings DOI: 10.21467/proceedings.7.4
ISBN: 978-81-984081-1-2

Series: AIJR Proceedings
ISSN: 2582-3922

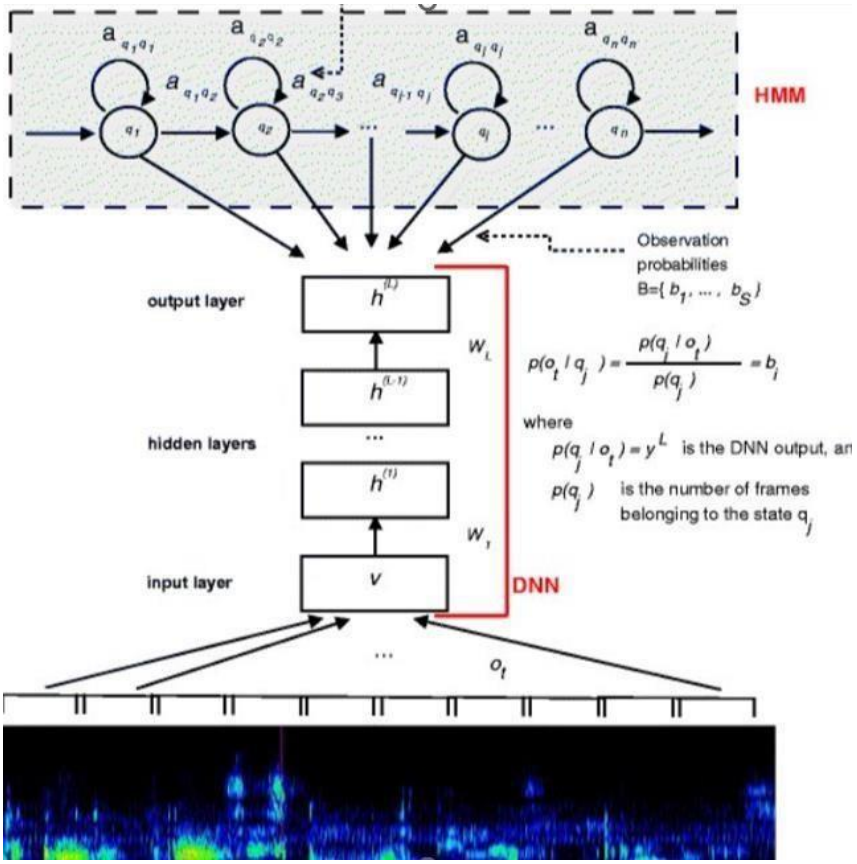

Fig 4: GMM-HMM Model Working

4 Mathematics and the working of Region based CNN with GMM-HMM model for speaker detection and diarization

In datasets where the true distribution is unknown, Gaussian Mixture Models (GMMs) are probabilistic models that are used to estimate the parameters of the underlying distributions. A more flexible and probabilistic method was offered by GMMs, which assign probabilities of belonging to each cluster as opposed to K-means clustering, which groups data points into discrete clusters. The core of GMMs lies in the Expectation-Maximization (EM) algorithm, an iterative procedure that alternately estimates the model parameters and the probability of data points belonging to each cluster. Let us look at the parameters of Gaussian Mixture Model upon which we built our model upon:

1. **Mean(μ)**: Randomly initialized for speaker diarization
2. **Covariance(Σ)**: Initialized randomly
3. **Weight** (it is also called the mixing coefficient) **(π):** Parameter that establishes each Gaussian component's relative importance or contribution to the mixture model as a whole. They showed the likelihood that a data point was a part of a specific component.
4. **K:** K is a hyper-parameter. This value was given by the Regional based Convolutional Neural Network.

Let us now go step by step to see how the GMM algorithm was designed:

1. **Initialization:** Initialization is the first step. With the exception of the

hyper-parameter K, whose value is determined by the RCNN prior to each speaker being detected and identified by the GMM-HMM model, all of the parameters in this were initialized at random.

2. **Expectation:** The main methodology that would be involved is:

$$r_{ic} = (\pi_c \times N(x_i/\mu_c, \Sigma_c) \Big/ \sum_{k=1}^{k} \pi_k \times N(x_i/\mu_k, \Sigma_k))$$

r_{ic} is the probability that the data point belongs to cluster (c) using the above equation.

π_c is the mixing coefficient which is the weight

$N(x_i/\mu_c, \Sigma_c)$ is called the probability density function for the vectors that the audio segments of speakers that are converted and plotted.

3. **Maximization Step (M step):** Here we update the parameters in order to get better accuracy than before:

$\pi_c = \sum_{i=1}^{m} r_{ic} / m$, here m is the number of data points present.

$\mu_c = \sum_{i=1}^{m} r_{ic} x_i \Big/ \sum_{i=1}^{m} r_{ic}$

$\Sigma_c = \sum_{i=1}^{m} r_{ic} \times (x_i - \mu_c)^2 \Big/ \sum_{i=1}^{m} r_{ic}$

The likelihood of any observation can be calculated using two ways.

1. **Forward algorithm:** Determines the likelihood of being in a specific state at a specific moment based on the sequence that has been observed up to that point.
2. **Backward algorithm:** Determines the likelihood of seeing the remainder of the sequence if we were in a specific state at a specific moment.

The parameters involved in a Hidden Markov Model is as follows:

1. Hidden States (S): A limited range of states that a system is capable of occupying.
2. Observables (O): A finite set of possible observations that can be emitted by the system.
3. Transition Probabilities (A): Likelihood of changing from one condition to another.
4. Emission Probabilities (B): Given that the system is in a specific state, the likelihood/probability of making a specific observation.
5. Initial Probabilities (π): How likely it was to begin in a specific state. Immediately

following this, we must overcome three obstacles.:

1. The Likelihood problem
2. The Decoding problem
3. The Learning Problem Let us go over each step by step:

1. **The Likelihood problem:** I used the forward and backward algorithms to

Proceedings DOI: 10.21467/proceedings.7.4
ISBN: 978-81-984081-1-2

Series: AIJR Proceedings
ISSN: 2582-3922

solve the problem, which was to determine the probability that a specific observation on a vector that it was this specific speaker can be derived from the HMM.

1. **Initialization:** $\boldsymbol{\alpha_i(i) = \pi_i \times b_i \times (O_1)}$. Given the observable O at time 1, multiplying the initial probability of state I by the emission probability b of that state. This is for the forward algorithm. For the Backward algorithm we execute the following formula: $\boldsymbol{\beta_T(i) = 1.}$ This means that the backward variables at time T of each state is equal to 1.

2. **Recursion:** $\boldsymbol{\alpha_{t+1}(j) = \sum_{i=1}^{N} \alpha_t(i) \times a_{ij} \times b_j(O_{t+1})}$. The forward variable is calculated recursively by multiplying the previous forward variable by the transition probability and the emission probability. Coming to the backward algorithm what we would be doing is: $\boldsymbol{\beta_t(i) = \sum_{j=1}^{N} a_{ij} \times b_j(O_{t+1}) \times \beta_{t+1}(j).}$
3. **Termination:** The forward algorithm gives us the formula: $\boldsymbol{P(O \mid \lambda) = \Sigma^N_{i=1} \alpha_T(i),}$ where λ denotes the HMM model present. Coming to the backward algorithm we have the termination step as: $\boldsymbol{P(O \mid \lambda) = \Sigma^N_{i=1} \pi_i \times b_i(O_1) \times \beta_1(i).}$

2. **The Decoding problem:** Here the Viterbi algorithm is used for decoding process. It works by finding the most likely path through the HMM that explains the observed sequence. It does this by considering all possible paths and choosing the one with the highest probability.
$\boldsymbol{X^*_{0:T} = argmax_X P[X_{0:t}/Y_{0:T}]}$ and $\boldsymbol{\mu(X_k) = max_X P[X_{0:k}/Y_{0:k}]}$ Given the observed initial data, a probability distribution for the potential initial states is produced by the first formula. The second formula maximizes the product of the terms on the right-hand side to choose the most likely initial state. The third formula, which establishes the following states, then used this ideal initial state as a fixed parameter.

3. **The Learning problem:** Because it is a dynamic programming solution and uses the forward-backward algorithm to res-estimate the model parameters, the Baum-Welch algorithm is the primary algorithm I used for parameter estimation. Usually, this issue is investigated first and then revisited at the end.

This method finds unique vocal signatures throughout the audio spectrum by utilizing RCNN's object detection capabilities. Following the detection of these vocal entities, the data can be used to improve the results of a Gaussian Mixture Model-Hidden Markov Model, which is commonly used for tasks involving speaker identification and verification. My projected speaker diarization method leverages the strengths of both electronic computer vision and speech processing. By applying R CNN to the spectrograph of the audio signal, we can visually detect prospective speaker system regions. The number of detected regions provides a first count on for the number of speakers [K], an essential hyperparameter for the Gaussian concoction Model. The GMM is then integrated into the Hidden Markov Model (HMM) at each stage. Each GMM represents a unique speaker, and the HMM transitions between these GMMs to model the sequence of speakers in the audio. This approach benefits from the complementary strengths of the two techniques, allowing for both visual and acoustic-based speaker

recognition [4].

5 Coding and experimental results

Although the code was purposefully simple, it was a new idea in this area. To keep the experimentation straightforward, the suggested solutions were created using Jupyter Notebook, and the diarization error constant was the evaluation metric. Processing a 5-minute audio file took about 4 to 5 minutes, which was roughly the same amount of computation time as the most recent speaker diarization models for a single audio which is pyannote present currently on hugging face for usage. The user input four audio files' spectrogram images with 34×50 pixels. The audio duration was 8, 11, 13, and 18 seconds, and the entire image was passed as a single frame to the RCNN. The audio file consisted of two speakers, labelled as 0 and 1, and the following table is the prediction of whether the speaker:

Table I: Glimpse Of The Precisions Of The Model For Two Speakers Over A Range Of Audio File Lengths.

SlNo.	Duration (in seconds)	Precision of model for speaker 1 (in percentage)	Precision of model for speaker 2 (in percentage)
1	18	97	96
2	11	89	94
3	13	92	95
4	8	85	89

The Classification Metrics of the RCNN overall for the above spectrogram images were as follows:

1. Recall: 97%
2. Precision: 92%

But this was the case when speaker 1 had a very high-pitched voice while speaker 2 had a low pitch and a deep voice. If we compare the spectrogram images results when two speakers with similar pitch, tone and loudness then the results were as follows:

Table II: Speakers With Similar Pitch And Voice.

Sl no.	Duration (in seconds)	Precision of model for speaker 1	Precision of model for speaker 2
1	12	79	84
2	9	71	83
3	8	70	81
4	10	78	82

Thus, as it can be seen, the results were not up to the expectation, but when it was used for boosting the GMM-HMM model, then the overall prediction significantly increased as it can be seen below.

The reason only the rows 2,5, and 9 have probabilities listed was because only during these durations in the entire audio file that the speakers have spoken.

Proceedings DOI: 10.21467/proceedings.7.4
ISBN: 978-81-984081-1-2

Series: AIJR Proceedings
ISSN: 2582-3922

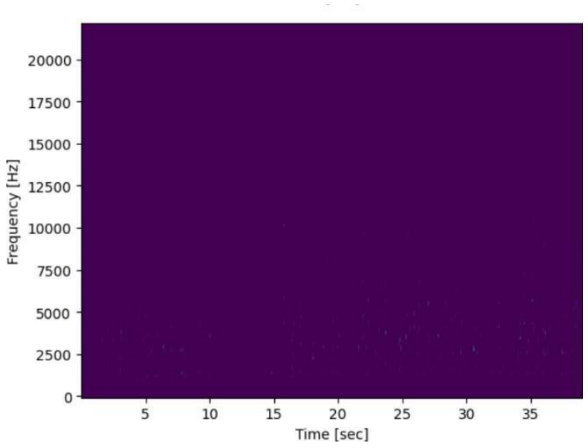

Fig 5

Fig 5: A Picture Of An Audio File's Spectrogram, Which Would Be Used To Run An RCNN And Determine How Many Speakers Are Present.

Table III: The Time And Error Of The Model In Assessing Audio Files Of Different Lengths.

Sl.No.	Length of audio file	Time required processing	Diarization error Constant
1	8.54 seconds	23.54 seconds	0.0121
2	38.64 seconds	71.32 seconds	0.023
3	62.54 seconds	159.67 seconds	0.054
4	81.44 seconds	201.34 seconds	0.1

According to the experiment conducted, the diarization error rate increases when the length of the audio file increases, this experimentally hypothesized is mainly due to the fact that the R-CNN had to iterate over multiple segmented images of the spectrogram thus making errors. Language change had been considered here. It would not directly impact the proceedings of the experiments, but might slightly indirectly impact the working of the GMM-HMM model due to difference in speaking style and pitch in different languages.

6 Usage of this model

This model was especially useful for human-Robot interactions, transcribing meetings and lectures security and surveillance, and providing captions for people with hearing impairments. However, the current limitations that a user might have encountered when using this was that it would have been extremely hard to put in place in real time due to the high processing power required, which may not always have been available.

7 Conclusion

This study offers a fresh method for speaker diarization in the context of smart manufacturing. Through the smooth integration of a Region-based Convolutional Neural Network (CNN) with a Gaussian

Mixture Model and Hidden Markov Model, the suggested system capitalizes on the robustness of conventional clustering techniques for efficient speaker differentiation while utilizing the power of deep learning for accurate speaker detection. This innovative approach has the potential to significantly advance speaker diarization in Smart Manufacturing applications. By enabling accurate and efficient speaker identification, this technology can unlock new possibilities in areas such as:

1. Real-time Quality Control: Monitoring and analyzing conversations between workers and machines to identify potential issues and improve production processes.
2. Enhanced Worker Safety: Identifying and responding to distress calls or unusual sounds in real- time to ensure worker safety.
3. Improved Communication: Facilitating seamless communication and collaboration between workers and machines, leading to increased productivity and efficiency.

More intelligent, effective, and secure industrial operations are made possible by this research, which establishes the foundation for future developments in speaker diarization within the framework of smart manufacturing.

8 Declarations

8.1 Acknowledgment

This research was conducted using Jupyter Notebook as the primary development environment. Spectrogram images were annotated using MATLAB and Roboflow, facilitating the training process. The audio signals used in this study were collected and processed by the author. To gain a thorough understanding of the GMM-HMM and R-CNN models, extensive research was conducted on relevant articles and papers. The references for these sources are provided below. Additionally, the images included in this work are sourced from reputable articles and research papers, with their respective citations listed.

8.2 Study Limitations

The author only had a limitation of computation power when processing and running on large audio files due to lack of dedicated hardware.

8.3 Competing Interests

There has been no conflict of interest.

8.4 AI-usage

No AI was used for writing this research paper.

8.5 Publisher's Note

AIJR remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

How to Cite

A. B. Dhruva, "Leveraging Region based Convolutional Neural Network (RCNN) with GMM Model Assisted by Hidden Markov Model for Speaker Detection", *AIJR Proc.*, vol. 7, no. 4, pp. 12–21, Jun. 2025. doi: 10.21467/proceedings.7.4.2

References

[1] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, and G. Friedl, "Speaker diarization: A review of recent research," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 2, pp. 357–370, Feb. 2012. doi: 10.1109/TASL.2011.2143856.

[2] M. Diez, L. Burget, F. Landini, and J. Černocký, "Analysis of speaker diarization based on Bayesian HMM with eigenvoice priors," in *Proc. 2019 IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Singapore, Singapore, Dec. 2019, pp. 913–920. doi: 10.1109/ASRU46091.2019.9003823.

[3] J. I. De La Rosa and A. Becerra, "Speech recognition in a dialog system: From conventional to deep processing. A case study a pplied to Spanish," *Multim. Tools Appl.*, vol. 77, no. 10, pp. 13019–13043, May 2018. doi: 10.1007/s11042-017-5160-5.

[4] S. Nemade, Y. K. Sharma, and R. D. Patil, "To improve voice recognition systems using GMM and HMM classification models," *Int. J. Innov. Technol. Explor. Eng. (IJITEE)*, vol. 8, no. 11, pp. 4683-4686, Sep. 2019. doi: 10.35940/ijitee.K2204.0981119.

[5] P. C. Woodland, H. Y. Yu, and C. V. Ramamurthy, "Speaker recognition and diarization," in *The Handbook of Speech Production*, M. A. Redford, Ed. Hoboken, NJ, USA: Wiley-Blackwell, 2015, pp. 461–486.

[6] R. Girshick, "Fast R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 144–154, Jan. 2016. doi:10.1109/TPAMI.2015.2437391.