A Comparative Study of Resource Usage for Speaker Recognition Techniques

Pulkit Verma and Pradip K. Das
Department of Computer Science and Engineering
Indian Institute of Technology Guwahati
Guwahati, Assam, India

***Abstract*—Resource usage of a software is an important factor to be taken into consideration while developing speaker recognition applications for mobile devices. Sometimes usage parameters are considered as important as accuracy of such systems. In this work, we propose an analysis of resource utilization in terms of power consumption, memory and space requirements of three standard speaker recognition techniques, viz. GMM-UBM framework, Joint Factor Analysis and i-vectors. Experiments are performed on the MIT MDSVC corpus using the Energy Measurement Library (EML). It is found that though i-vector approach requires more storage space, it is superior to the other two approaches in terms of memory and power consumption, which are critical factors for evaluating software performance in resource constrained mobile devices.**

***Index Terms*—GMM, UBM, JFA, i-vectors, speaker recognition, resource usage, power consumption.**

I. Introduction

Resource usage in speaker verification systems is a major bottleneck in providing real time performance of such systems. It becomes increasingly important in the current scenario where the speeds of devices have been increasing at a higher rate than ever before and smaller devices are replacing the large static systems. Optimal utilization of resources becomes an important factor in the success of any technique in such environments. Despite the advances in speech processing and recognition technology, resource usage remains a major bottleneck in providing real time experience to the users.

Gaussian Mixture Model (GMM) based speaker verification approach was proposed by Reynolds [1] long time back. It remained popular in the form of the GMM-UBM framework. It was modified so as to take into account the inter-speaker (speaker) variability and intra-speaker (channel) variability in the modeling process. This approach was termed as Joint Factor Analysis (JFA) [2], [3].

JFA considers that these variabilities are independent of each other. But it was later found that channel variabilities were also modeling the inter-speaker variations [4]. Dehak et. al. [5] resolved this issue by proposing a channel-blind approach where speaker and channel variabilities are modeled together in a lower dimensional total variability space. The vectors represented in this space are called i-vectors. This has now become the state of the art approach in nearly all speech processing applications [6] because of its low complexity and better performance.

In these days of energy aware devices, the efficacy of a software based on its power requirements has gained significance. So power consumption has become an important dimension of software performance and quality measurement. Calculating the power consumption of a software was first proposed for embedded software [7]. Making such systems useful for application software is challenging and many methods have been developed but few of them are actually used in practice. A review of energy consumption of software libraries is presented in [8] and [9]. PowerAPI [10] and Energy Measurement Library (EML) [11] are some of the tools which contain successful implementations to solving this problem.

SpeakerSense [12] was the first major approach in calculating energy efficiency for a continuous background sensing speaker identification prototype. The work presented a detailed analysis for variation in energy efficiency and accuracy vs length of smoothing windows. A major drawback of this approach is that it is tested on a very small dataset of 17 speakers and implements on GMM-UBM framework.

In this paper, we present a comparative study of power, memory and storage space consumption of GMM-UBM framework, JFA and i-vector approach in the domain of speaker recognition.

The rest of the paper is organized in the following manner: in Section 2, the three speaker recognition techniques are explained; in Section 3, approaches used for resource usage analysis are described; in Section 4, the experimental setup is explained; in Section 5, the results are presented and discussed; and in Section 6, conclusions and future work are discussed.

II. Speaker Recognition Approaches

The general framework of any speech recognition approach can be explained using a speaker verification system as shown in Figure 1. This generic approach can be subdivided into two main steps, speaker enrollment and speaker verification.

During the speaker enrollment process, a background model is generated using the data collected from non-target utterances. In this paper, for all the approaches this model will be Universal Background Model (UBM). Using this background

DOI: https://doi.org/10.1109/ICSPCom.2016.7980598

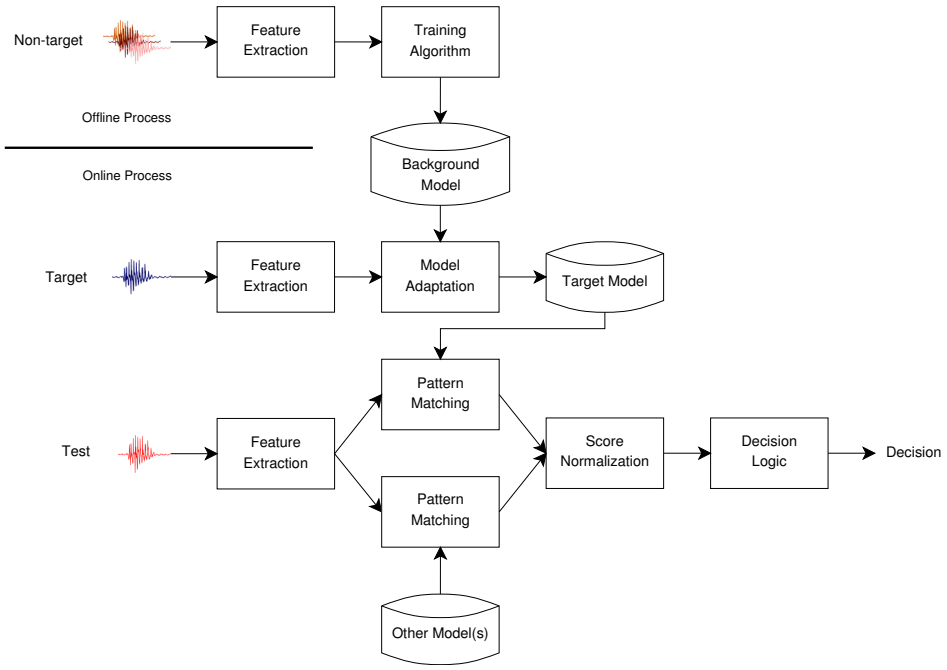


Fig. 1: Flowchart for a generic Speaker Recognition Process

model, a target model is generated using the target utterances by adapting the background model according to the target data. Now this target model will act as the single point of reference for the pattern recognition algorithms. Whenever any test utterance is given as input to the system, features are extracted from it and pattern matching algorithms are applied on it using one or more kinds of target models. The resultant similarity score is then normalized and final decision is taken after applying some decision logic to this normalized score.

A. GMM-UBM Framework

In this method GMMs are created for each targeted speaker by training a feature specific model on each speaker. Estimation Maximization (EM) algorithm is used to train the Universal Background Model (UBM). This UBM is changed using speaker specific data by applying the MAP algorithm to generate the speaker specific GMMs. A particular test utterance is compared against these models and UBM to complete the recognition process.

A major drawback in this method is that the MAP algorithm may adapt to the channel and other environmental factors. Also if the training data is limited, the speaker models may fail to capture all the speaker characteristics.

B. Joint Factor Analysis

In JFA [2], [3], [13], we assume that for a particular speaker the speaker and channel dependent supervector μ used to represent any speech utterance is generated by the vector sum of speaker dependent supervector and channel dependent supervector. Also these speaker and channel supervectors are distributed normally and are statistically independent. If $\mathbf{s}$ is speaker dependent supervector, $\mathbf{c}$ in channel dependent supervector, then

$$\mu = \mathbf{s} + \mathbf{c} \tag{1}$$

We assume that the distribution of $\mathbf{s}$ and $\mathbf{c}$ has a hidden variable description as depicted by Equations 2 and 3 respectively, given by:

$$\mathbf{s} = \mathbf{m} + \mathbf{Vy} + \mathbf{Dz} \tag{2}$$

where, $\mathbf{s}$ is speaker dependent supervector, $\mathbf{m}$ is a speaker and channel independent supervector (from UBM), $\mathbf{V}$ is eigenvoice matrix (rectangular matrix of low rank), $\mathbf{D}$ is a residual diagonal matrix, $\mathbf{y}$ is a vector representing speaker factors, $\mathbf{z}$ is a normally distributed random vector representing speaker specific residual factors.

$$\mathbf{c} = \mathbf{Ux} \tag{3}$$

where, $\mathbf{c}$ is channel dependent supervector, $\mathbf{U}$ is eigenchannel matrix (matrix of low rank), $\mathbf{x}$ is a normally distributed random vector representing channel factors. Hence it can be inferred that $\mathbf{c}$ is normally distributed vector whose mean is 0.

C. i-vectors

Dehak [4] observed that the channel dependent supervector in JFA also models the speaker features. He introduced a new low dimensional total variability space $\mathbf{T}$ to account for both the variabilities, where μ is given by:

$$\mu = \mathbf{m} + \mathbf{Tx} \tag{4}$$

where, $\mathbf{m}$ is the UBM supervector, $\mathbf{x}$ is a normally distributed random vector in this space. The factors of $\mathbf{x}$ also called as total factors and are generally known as identity vectors or i-vectors.

One of the additional advantage of this approach is that supervised training is not needed in this model unlike JFA and GMM-UBM.

D. Comparison

The three methods compared in this paper differ in implementation of various steps shown in Figure 1. Though the basic flow remains same, many options are available to implement each parameter of the generic method. Table I shows the theoretical comparison of the implementation of three methods used in this paper.

III. Resource Usage Analysis

A. Power Consumption

We have used Energy Measurement Library (EML) [11] to measure the power requirements of the various approaches. It acts as a middleware between code for measuring power consumption and hardware based measurement tools. EML uses Performance API (PAPI) [14] to get the power readings via RAPL register interfaces. Single multi-core Intel® Xeon® CPU was used for the experiments. EML also provides abstracted interface to simplify data collection and representation. It is currently available for a limited number of hardware platforms.

B. Memory and Space Consumption

For the memory requirements of the applications, we have calculated the resident set size (RSS) portion of the memory as it is the actual amount of memory that is occupied by the process in main memory. We have used the **`pmap`** utility available in UNIX® based operating systems for this purpose.

For the space requirements we have calculated the amount of space occupied by the executables on the secondary disk. **`ls`** command was used for this purpose.

IV. Experimental Setup

A. Corpus

We have used the MIT Mobile Device Speaker Verification Corpus (MDSVC) [15], aimed at supporting speaker verification research using mobile devices, in this work. Recordings were done using microphones and internal headsets in different environments to generate multi-style trained models. Since the data was collected over 2 sessions it also contains inter-session variability as well.

The corpus has an enrolled data of 48 speakers out of whom 26 were male and 22 were female. A total of 54 recordings per person were done in each session. Hence over 2 sessions, 5,184 recordings were collected for enrolled users. Imposter data was also recorded which consisted of 40 speakers, out of which 23 were male and 17 female.

The text used consisted of names and ice cream flavor phrases. The average length of each recording is about 2 seconds.

B. Scoring Techniques

Zero-dependent test-score normalization (ZT-norm) [16] has been used in GMM-UBM and JFA approach. In this technique the T-norm of each test utterance is calculated against each imposter model. These scores are further normalized by applying Z-norm to get the output of ZT-norm. Here it should be noted that Z-norm is calculated by scoring each target speaker model with each imposter utterance.

Probabilistic Linear Discriminant Analysis (PLDA) [17] has been used for similarity scoring in the i-vector approach. Given two i-vectors $\mathbf{x_1}$ and $\mathbf{x_2}$, the PLDA score measures the log-likelihood ratio between the two hypotheses $\{H_0, H_1\}$, where H_0 hypotheses that $\mathbf{x_1}$ and $\mathbf{x_2}$ belong to the same speaker while H_1 hypotheses otherwise. The score is given as:

$$score_{PLDA} = log(p(\mathbf{x_1}, \mathbf{x_2}|H_0)) - log(p(\mathbf{x_1}, \mathbf{x_2}|H_1)) \quad (5)$$

This can be explained by the fact that if the i-vectors belong to the same speaker then their latent variables will be same, otherwise they will be different. The solution to (5) can be found in [17]. For normalization we have used standard Length Normalization prior to applying PLDA.

C. Tools and Libraries

For the calculation of power consumption Energy Measurement Library (EML) [11] is used. For the speaker recognition techniques, program development was based on ALIZE [18] toolkit. This was to ensure that standard code-base is used for the experiments.

D. Platform

The output of power requirements vary across systems and might change significantly if the measurement platform is altered. But the relative power requirements of different processes generally does not change.

The system used for the experiments has the following specifications:

- 1 x Intel® Xeon® CPU E5530 @ 2.40GHz
- 16 Cores
- 48 MB L3 Cache
- 32 GB RAM
- gcc 4.1.2
- Intel® MSR RAPL interface
- Intel® MPSS 3.4.2

E. Configuration

First 19 Mel frequency cepstral coefficients have been extracted along with log energy feature. 25 ms Hamming window was used along with a frame advance of 10 ms for this purpose. Using windows of 5 frame, delta and double-delta coefficients were calculated to give a total of 60 dimensional features.

One gender-independent UBM was trained with 32 Gaussian components from a training set consisting of randomly selected recordings from MIT MDSVC corpus. This UBM is used for the extraction of zero and first order Baum-Welch statistics. Speaker and channel variability subspaces of 32 and 16 dimensions respectively are used for JFA experiments. For the i-vector approach 48 dimensional total variability space is used, along with 32 dimensional speaker subspace being used for PLDA.

TABLE I: Different Parameters used in the experiments for the three approaches

Parameter	GMM-UBM	JFA	i-vector
Feature Extraction	MFCC	MFCC	MFCC
Training Algorithm	EM	EM + JFA Hyperparameter	EM + TVA training
Background Model	UBM	UBM + JFA Parameters	UBM + TV Parameters
Model Adaptation	BW + MAP	BW + JFA Adaptation	BW + FA training
Pattern Matching	BW	BW	BW + Factor extraction
Score Normalization	Log likelihood	Log likelihood	Cosine Distance

While evaluating each system's performance, the time taken for a method comprises of the total time involved, from the feature extraction step to the execution of final decision logic for the test sets. Any precomputation of i-vectors is not done, so as to simulate the real conditions in which speaker recognition application will be used.

The accuracies are calculated as the percentage of speakers the method is able to identify accurately out of 48 speakers on whose utterances the experiments were performed. The inaccuracies are reported when the confidence values generated after the scoring step are within 2% of each other for multiple speakers. Hence in such a case the method cannot tell the test utterance is associated with which recording with sufficient confidence.

V. Results

We compared the memory requirements of GMM-UBM approach, JFA approach, and i-vector technique using the `pmp` command[1]. The time interval at which the readings were taken is 50 milliseconds. The graph in Figure 2a shows the variation of memory usage with time. It can be clearly seen that the i-vector approach is much faster as compared to the JFA and GMM-UBM approach. Also the memory requirements are relatively lesser than the other two. The maximum and average memory used during the experiment is also reported in Table II.

Table II also shows the power consumption of the three approaches. Though i-vector approach seems to perform slightly better than the other two approaches, it can be seen that all of them perform nearly equally in this respect. The unit of power consumption is same as mentioned in [11].

It should be noted that EML gives better results on parallelizable programs. JFA and GMM-UBM approaches implemented in the experiments do not have any preprocessing to be done, whereas in the i-vector approach normalization was done prior to actual scoring. Hence the results for i-vector approach might be improved if power consumption calculation is done without any parallelization.

The space requirements of the approaches keep increasing as we move from GMM-UBM to i-vector owing to the large code size and complexity for i-vector calculation leading to larger executables.

On the basis of these experiments we observed that the i-vector approach performs significantly better in terms of speed. The power consumption of all 3 approaches is comparable with i-vector approach performing slightly better. The i-vector technique also exhibit slight optimality in terms of memory consumption but due to its large code-base it uses the maximum space to the order of thrice the space used by the other 2 methods.

[1]The results are based on [19]

VI. Conclusion

In this paper, a comparison of the current state of the art techniques in terms of their resource utilization targeted towards speaker recognition is conducted. Experiments were performed on a standard corpus collected using mobile devices in a noisy environment. This dataset emulates the data likely to be encountered in real life environments. The algorithm tested using this dataset reflects the true nature of performance in actual situations.

The results show that the i-vector approach is more efficient in terms of memory usage and power consumption, which are the major focus areas for today's computing environments. Specifically in case of mobile applications, these two factors can play an important role in the success and usability of such software. Even in terms of running time the i-vector approach outperforms the other two approaches.

The current work only analyzes the resource usage on laptops and desktops in the lab environment. This will be extended to mobile devices in future to get more specific readings for various mobile platforms. Work is underway to port the currently developed C++ code to Android platform.

Future enhancements to this work include incorporating these experiments on heterogeneous operating systems like Android™, Windows® Phone, etc. as well as experimenting with different sized feature vectors for all these speaker recognition approaches.

Acknowledgment

The authors would like to thank MIT Computer Science and Artificial Intelligence Laboratory for providing MIT MDSVC Corpus and also appreciate the support of UNESCO and Government of India for providing funds for the project, which helped greatly in the work reported herein.

References

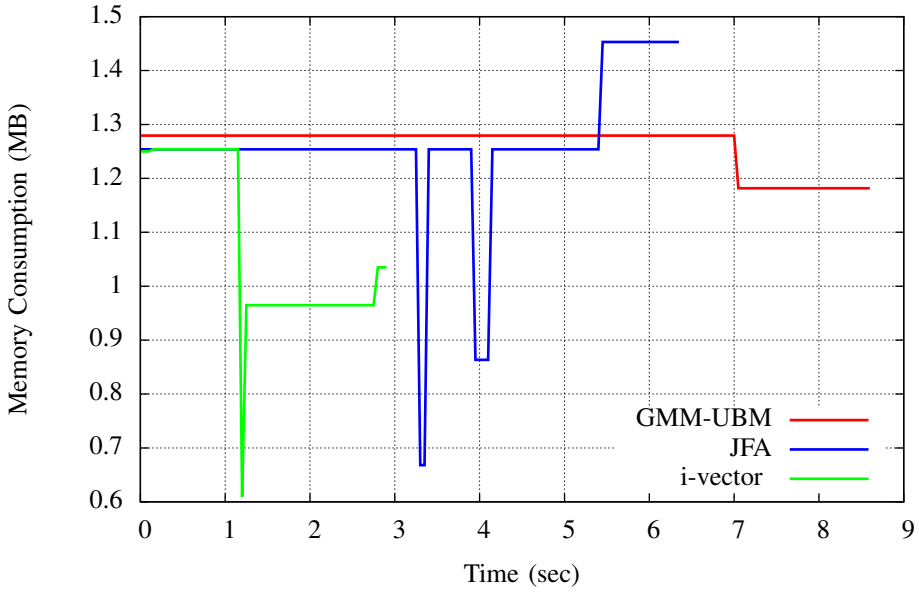
[1] D. A. Reynolds, "Speaker identification and verification using gaussian mixture speaker models," *Speech Communication*, vol. 17, no. 1-2, pp. 91–108, 1995.

[2] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," Tech. Rep., 2005.

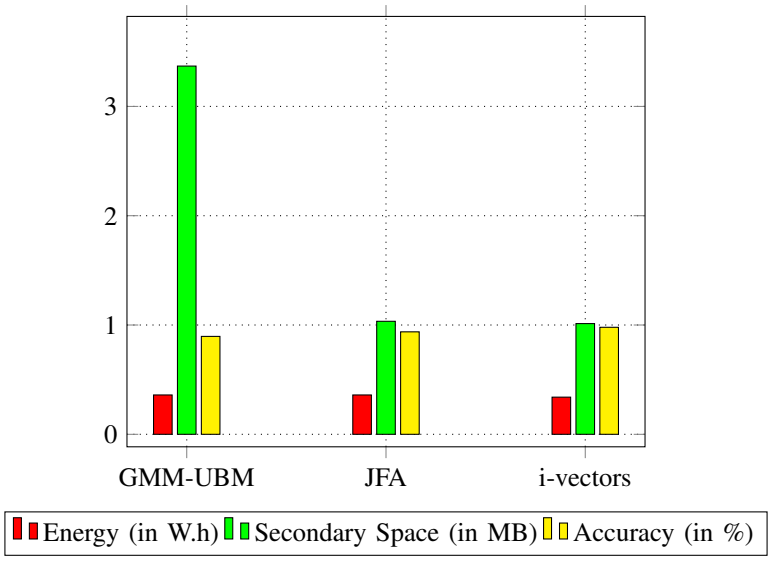
[3] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint Factor Analysis Versus Eigenchannels in Speaker Recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 4, pp. 1435–1447, May 2007.

TABLE II: Resource consumption for GMM-UBM, JFA and i-vector approaches for speaker recognition

Approach	Memory Consumption (MB)		Energy	Space	Time	Accuracy
	Average	Maximum	(W.h)	(KB)	(sec)	(%)
GMM-UBM	1291.5	1310.0	0.3648	1013	8.65	89.58
JFA	1292.0	1488.0	0.3641	1034	6.45	93.75
i-vectors	1106.0	1284.0	0.3472	3369	2.95	97.91



(a) Memory Consumption



(b) Energy and Space Requirements

Fig. 2: Resource consumption for GMM-UBM, JFA, and i-vector approaches for speaker recognition

[4] N. Dehak, "Discriminative and Generative Approaches for Long- and Short-term Speaker Characteristics Modeling: Application to Speaker Verification," Ph.D. dissertation, 2009, aAINR50490.

[5] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, May 2011.

[6] P. Verma and P. K. Das, "i-Vectors in Speech Processing Applications: A Survey," *International Journal of Speech Technology*, vol. 18, no. 4, pp. 529–546, 2015. [Online]. Available: http://dx.doi.org/10.1007/s10772-015-9295-3

[7] V. Tiwari, S. Malik, and A. Wolfe, "Power analysis of embedded software: a first step towards software power minimization," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 2, no. 4, pp. 437–445, Dec 1994.

[8] A. Noureddine, R. Rouvoy, and L. Seinturier, "A Review of Energy Measurement Approaches," *SIGOPS Oper. Syst. Rev.*, vol. 47, no. 3, pp. 42–49, Nov. 2013. [Online]. Available: http://doi.acm.org/10.1145/2553070.2553077

[9] A. Noureddine, R. Rouvoy, and L. Seinturier, "Unit Testing of Energy Consumption of Software Libraries," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, ser. SAC '14. New York, NY, USA: ACM, 2014, pp. 1200–1205. [Online]. Available: http://doi.acm.org/10.1145/2554850.2554932

[10] A. Bourdon, A. Noureddine, R. Rouvoy, and L. Seinturier, "PowerAPI: A Software Library to Monitor the Energy Consumed at the Process-Level," *ERCIM News*, vol. 92, pp. 43–44, Jan. 2013.

[11] A. Cabrera, F. Almeida, J. Arteaga, and V. Blanco, "Measuring energy consumption using EML (energy measurement library)," *Computer Science - Research and Development*, pp. 1–9, 2014.

[12] H. Lu, A. Bernheim Brush, B. Priyantha, A. Karlson, and J. Liu, "SpeakerSense: Energy Efficient Unobtrusive Speaker Identification on Mobile Phones," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, K. Lyons, J. Hightower, and E. Huang, Eds. Springer Berlin Heidelberg, 2011, vol. 6696, pp. 188–205.

[13] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 5, pp. 980–988, July 2008.

[14] S. Browne, J. Dongarra, N. Garner, G. Ho, and P. Mucci, "A portable programming interface for performance evaluation on modern processors," *Int. J. High Perform. Comput. Appl.*, vol. 14, no. 3, pp. 189–204, Aug. 2000. [Online]. Available: http://dx.doi.org/10.1177/109434200001400303

[15] R. Woo, A. Park, and T. Hazen, "The MIT Mobile Device Speaker Verification Corpus: Data Collection and Preliminary Experiments," in *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The*, June 2006, pp. 1–6.

[16] R. Zheng, S. Zhang, and B. Xu, "A comparative study of feature and score normalization for speaker verification," in *Advances in Biometrics*, ser. Lecture Notes in Computer Science, D. Zhang and A. Jain, Eds. Springer Berlin Heidelberg, 2005, vol. 3832, pp. 531–538.

[17] Y. Jiang, K. Lee, Z. Tang, B. Ma, A. Larcher, and H. Li, "PLDA modeling in i-vector and supervector space for speaker verification," in *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, 2012, pp. 1680–1683.

[18] A. Larcher, J. Bonastre, B. G. B. Fauve, K. Lee, C. Lévy, H. Li, J. S. D. Mason, and J. Parfait, "ALIZE 3.0 - open source toolkit for state-of-the-art speaker recognition," in *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association*, Lyon, France, August 25-29, 2013, 2013, pp. 2768–2772.

[19] P. Verma, "Resource Usage Analysis for Speech Recognition Techniques," Master's thesis, Department of Computer Science & Engineering, Indian Institute of Technology Guwahati, India, 2015.