VoSE: An algorithm to Separate and Enhance Voices from Mixed Signals using Gradient Boosting

Monica Gupta (✉ research_student@hotmail.com)

Uttarakhand Technical University

Dr R K Singh

BTKIT Dwarahat

Dr Sachin Sinha

NOT, New Delhi

Research Article

Keywords: Speech Enhancement, Psychoacoustic Model, Separation, Recognition

Posted Date: October 19th, 2020

DOI: <https://doi.org/10.21203/rs.3.rs-93561/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

VoSE: An algorithm to Separate and Enhance Voices from Mixed Signals using Gradient Boosting

Monika Gupta¹, Dr R.K. Singh², Dr Sachin Singh³

Monika Gupta¹ “, Uttarakhand Technical University, Dehradun, India, monikaguptautu@gmail.com

, Dr R.K. Singh² “, BTKIT, Dwarahat, India, rsingh12@rediffmail.com

Dr Sachin Singh³ “, NIT, New Delhi, India, sachinsingh@nitdelhi.ac.in

Abstract

Voice Separation and Enhancement (VoSE) algorithm aims at designing a predictive model to solve the problem of speech enhancement and separation from a mixed signal. VoSE can be used for any language, with or without a large Datasets. VoSE can be utilized by any voice response system like, Siri, Alexa, Google Assistant which as of now work on single voice command. The pre-processing of the voice is done using a Trimming Negative and Nonzero voice filter (TNNVF), designed by the authors. TNNVF is independent of language, it works on any voice signal. The segmentation of a voice is generally carried out on frequency domain or time domain. Independently they are known to have ripple or rising effect. To rule out the ripple effect, data is filtered in the time-frequency domain. Voice print of the entire sound files is created for the training and testing purpose. 80% of the voice prints are used to train the network and 20% are kept for testing. The training set contains over 48,000 voice prints. LightGBM with TensorFlow helps in generating unique voice prints in a short time. To enhance the retrieved voice signals, Enhance Predictive Voice (EPV) function is designed. The tests are conducted on English and Indian languages. The proposed work is compared with K-means, Decision Stump, Naïve Bayes, and LSTM.

Keywords- Speech Enhancement, Psychoacoustic Model, Separation, recognition

I. Introduction

Much research in source separation is centred on the famous cocktail party issue [1], where a listener has to attend to speech selectively in a context of competing speech noise. Human's auditory brain is capable of selectively recognising the voices. Brain is able to separate spectral-temporal representations for concurrent speeches [2]. To understand it in simpler terms, consider a party. Depending on the situation, given a variety of distracting voices and other sounds, one can isolate friend's speech and remember the words with little effort [3]. This is a strong indication that the human auditory system has a function to distinguish incoming signals. The principle is known as psychoacoustics. To correctly describe psychoacoustics, there are solutions such as [4].

Audio command enabled devices like Google Assistant, Siri, and Alexa [5] have taken the technology to the next level. These systems accept audio inputs and execute the process after the voice information is decoded. Speech inputs/commands are given in a natural environment. The natural environment has high acoustic noises in the background. The noises could be of different levels and may have one or more interfering sources [6]. Cocktail party problem introduced by Cherry in 1953, is one such problem where recording is done in a natural environment. Cherry introduced automatic voiceprint and speech recognition [7,8].

For speech separation, different methods have been designed [9-14]. Approaches such as Computational Auditory Scene Analysis (CASA) [15-18], Hidden Markov Model (HMM) [19-21], HMM in conjunction with Cepstral Coefficients for Mel Frequency [22-24], Non-negative Factorization of Matrix (NMF) [25-28] and Minimal Mean Square Error (MMSE) [29-

32]. However, these strategies have seen relatively little success. For large databases, these models could not perform well. In addition, most of them do not recognise the human auditory system's psychoacoustic properties, such as the temporal and spectral masking effects, and are thus unable to distinguish between a real sound and what a human would perceive. Deep learning has bridged the gap between what human perceive and what a computer understands. It has significantly improved speech recognition [33-42]. These approaches are to make the computer think like human. It is observed that researchers prefer to use MFCC with deep learning [41] or Principal Component Analysis (PCA) with Deep Convolution Neural Network (DCNN) [42].

The algorithms which are developed thus far are no substitute for what humans can do. To solve the different problems and pay attention to the speaker of importance, people use many patterns in a group. When it comes to a gathering of heavy music, the differences are larger. One has to filter the music out and stretch the ears to understand. Patterns in such circumstances play an essential role. The patterns include accuracy, continuity of tone, language and position of the speaker. To resolve the pattern issue, Permutation Invariant Training(PIT), and utterance level Pattern Invariant Training(uPIT) were proposed[43,44] to separate the signals. PIT and uPIT, however, only use the mixed amplitude range as input features. PIT and uPIT fail to accurately discriminate between each speaker. uPIT suffers from permutation problem. To overcome the issue of permutation authors proposed Deep Clustering (DC) with uPIT [45,46]. DC, Deep Attractor Network [47,] and uPIT can predict the assignments at the utterance level of all TF bins at once, without the need for frame based assignment, which is the main cause of the permutation problem. Nevertheless, when vocal features of speakers are similar; these methods also suffer from the issue of permutation.

To exploit the recently developed techniques in Artificial Intelligence(AI), Deep learning based on audio-visual data is introduced in recent years[48,49]. It is widely known that humans not only listen to the sound but also note the speaker 's emotions, they read lips, eyes and body gestures. The proposed work in [48,49] is speaker-dependent, and the effects of the separation too is not satisfactory.

The above finding suggests the need for source separation, especially in cases where an unidentified mixed signal is transmitted and registered in a sensor array. Speech signs have silent spaces and meaningless noises as well. To overcome the issues, authors developed Trimming Negative and Nonzero voice filter (TNNVF). It is also observed that there is a ripple effect in the above mentioned models as speech segmentation is conducted either on a frequency domain or a time domain. VoSE translates the speech data into time-frequency domain. To isolate the voices, the suggested model uses LightGBM[51-52] with TensorFlow running in the background. TensorFlow[53] helps in producing the individual voice prints in the shortest possible time.

Why LightGBM?

Decision tree learning algorithms[50-54] construct trees level(depth)-wise. LightGBM, a gradient boosting algorithm, builds trees leaf-wise as a result there is a lesser loss. LightGBM uses an optimized histogram algorithm. It splits the continuous individual values into n intervals and selects the dividing points among the n values. The use of the histogram algorithm has a regularization effect and can avoid overfitting effectively. LightGBM after the first split, accomplishes the second split only on the leaf node. The leaf-wise isolation of the LightGBM algorithm allows it to operate on large data sets as well. LightGBM has a maximum depth parameter, it expands like a tree but prevents overfitting.

Gradient boosting, due to its tree structure, is known to be good for tabular data but recently researchers have found it useful in a various applications[55-67].

The models in [1-69] are either specific to application or address a single language but none of them address the issue of speech translation. Once the speech is separated the voice is not converted into text. For building robust acoustic models for speech recognition[68,69], accurate phonetic transcriptions are important. VoSE after enhancing the predicted voice converts the speech to text to make sure that the converted text matches the original speech's text.

II. Methodology

The methodology involves two processes – Experimental Setup, and the implementation. Implementation is explained through objective functions and related algorithms.

1.1 Experimental Setup

i. Hardware

Processor: Intel core i5, fifth generation

RAM: 8 GB

Graphic card: Nvidia

HDD: 1 TB

OS: Windows 10

ii. Dataset

Festvox CMU_ARCTIC databases, VoxForge Speech Corpus, Wall Street Journal Dataset (WSJ0) , Microsoft Indian language Corpus, and Linguistic Data Consortium for Indian Languages(LDC-IL).

The methodology is summarized into objective functions, which are explained in following section. It is followed by steps and the algorithms designed.

1.2 Objective functions

There are three main objectives of the proposed work. The objectives functions are represented mathematically below:

$$E_v = EPV(P_v) \dots\dots\dots \text{fn}(1)$$

E_v : Enhanced Voice

EPV : Enhancement function

P_v : Predictive voice

When a sound is retrieved from a mixed signal, the sound files are first filtered, normalized and then predictive analysis is run over them. The process returns similar but not the same sound. Enhance Predictive Voice(EPV) function utilizes the multi-class classification capabilities of LightGBM to retrieve the near original voice. The function is explained further in the paper with results.

$$P_v = P_{fn}(F_{dataset}) \dots\dots\dots \text{fn}(2)$$

$F_{dataset}$: filtered dataset

P_{fn} : Predictive function

P_v : Predictive voice

P_{fn} function is to be reduced to classify and predict the voice in the least possible time. The function is explained with the help of algorithm in the following sections.

$$F_{dataset} = TNNVF(V_{dataset}) \dots\dots\dots \text{fn}(3)$$

$F_{dataset}$: filtered dataset

$V_{dataset}$: Voice Dataset

Trimming Negative and Nonzero voice filter (TNNVF) is based on two algorithms, one is to detect the voice and the other to detect speech.

a. Detect a voice

$$S_p = \begin{cases} 0, & v_i \text{ not a voice} \\ > 0, & v_i \text{ is a voice} \end{cases} \dots\dots \text{eq}(1)$$

Here,

S_p : Retained Signal

v_i : voice

The voice sample is iterated to check for 0, any zero value found is removed from the data. The process removes the leading, training spaces, and in between silence. The trimmed signal is further tested to retain only the speech using eq(2)

b. Detect Speech

$$S_t = \begin{cases} 0, & \sigma^2(S_i) < Q_s \\ 1, & \sigma^2(S_i) > Q_s \end{cases} \dots\dots \text{eq}(1)$$

Here,

S_t : Speech

S_i : Signal

Q_s : Threshold

The threshold value is arrived at after iterating through the dataset. After eq(1) the signal does not have any silence therefore, now the signal either has voice or noise. For the purpose, average of each signal is calculated and added up. The sum is then divided by the number of samples to arrive at a threshold value Q_s . The voice is iterated and if the variance (σ^2) of the data is more than Q_s then it is considered as voice otherwise it is taken as silence. This data is removed from the voice sample.

1. Working

Following steps briefly explains the working of VoSE:

a. Seperate Voices

1. Voice files of different languages are stored in related folders.

(The languages are not mixed they are tested individually)

2. Each folder is read, and the data is filtered using TNNVF.
3. A dataset is created using all the voice prints.
4. Data is then split into Training and Testing set.
5. Training and Testing labels are created.
6. Network is trained.
7. Voices from different folders are fused to create a mixed voice dataset.
8. Predict Trained network with fused voices

b. Enhance Voices

9. Read raw voice data
10. Create labels
11. Split data into training and testing set
12. Train model
13. Predict trained network with output after step 8.

Training and validation are performed on the sample after pre-processing using eq 1 and 2. The voices filtered are housed in three directories that have male, female and assorted voices. The various voices represent voices of children, elders, men and women. The datasets have utterances from 100 speakers. The total utterances are over 48000, in 5275 files. The length of each speech is between six and seven seconds. The bit rate of the voices is 256 kb/s. The sampling rate is 16 KHz.

The mixed voices have a combination of voices from each dataset. The limit of speaker is set to four (to male and two female) for the purpose of experiment. A sample mixed voice would have a voice each from WSJ0, Festvox, VoxForge, and raw folders. Festvox has the largest corpus of 1132x4 (4 different speakers) voice samples. Indian languages, Hindi and Bengali are taken from Microsoft Indian language Corpus, and Linguistic Data Consortium for Indian Languages(LDC-IL).

A code is written which automatically reads the files from different folders and fuse the voices. The fused data is stored in a mixed voice folder.

2. Model

Figure 1: Working model graphically represented. Here V_s is the single voice used for prediction.

The model in figure 1 graphically represent the working VoSE. The algorithms explain the working in detail. The algorithms are based on the code written for the purpose.

Algorithm 1: Prepare sound files

Setup

Initialize required variables

Read folder having sound files

Start

Step 1 While not end of folder do

Step 2 $v_i \leftarrow$ read sound file

Figure 2: Voice plot with leading, trailing and in between silence

Step 3 Detect voice using eq (1)

Step 4 Remove unwanted data

Step 5 Detect speech using eq (2)

Step 6 $\text{clean_speech} \leftarrow$ Retain only speech

Step 7 $\text{clean_voice} \leftarrow$ Save clean file to folder

Step 8 end while

Figure 3: Trimmed Voice plot

Algorithm 1 helps in reading all the sound files from a folder. It cleans the blank signals and retain only speech.

Algorithm 2: Prepare Training set

Setup

$N \leftarrow 0$

Start

Step 1 for i 1 to end of male voice folder:

Step 2 $V_{\text{dataset}}(i) \leftarrow$ read male voice

Step 3 $\text{label}(i) \leftarrow N$

Step 4 $N \leftarrow$ increment N by 1

Step 5 $\text{voice} \leftarrow [V_{\text{dataset}}[i], \text{label}[i]]$ #write to a csv file

Step 6 end for

Step 7 repeat steps 1 to 6 for female and assorted folders

Algorithm 2 prepares the training set. The dataset contains male, female and assorted voices. Single channel is read, digitized and stored in the csv file with appropriate label. Numeric labels are assigned for the uniformity in the data types.

Algorithm 3: Prepare Testing set

```

Step 1 index  $\leftarrow$  random 1 to length of voice folder
Step 2 for i in index:
Step 3      $V_f \leftarrow$  female_voice[i]
Step 4      $V_m \leftarrow$  male_voice[i]
Step 5      $M_v \leftarrow$  read_assorted[i]
Step 6      $M_f \leftarrow \{V_m + V_f + M_v\}$ 
Step 7      $Avg \leftarrow (std(V_m) + std(V_f) + std(M_v)) / 3$ 
Step 8      $N \leftarrow M_f / Avg$  #Normalize mixed signal
Step 9     label[i]  $\leftarrow$  i
Step 10 end for

```

Testing set is a fusion of different voices. Since, amplitude and pitch of these sounds are different the data is normalized using steps 6 and 7. Average of standard deviation of V_m , V_f , and M_v is calculated in Step 6 . To normalize the fused voice (M_f) is divided by it in 7.

Algorithm 4: LightGBM model

Setup: Initialize Model

Start:

```

Step 1  $X_{train}, X_{label}, Y_{train}, Y_{label} \leftarrow \text{split}(\{V_m, V_f, M_v\}, label)$ 
Step 2 parameters  $\leftarrow \{$ 
    objective  $\leftarrow$  multiclass           #type of model
    metric  $\leftarrow$  null                  # metric corresponding to objective
    boosting  $\leftarrow$  goss                # gradient based one sided sampling
    depth  $\leftarrow$  10                     # limits maximum depth of a tree
    number of leaves  $\leftarrow 2^{\text{depth}} - 1$  # maximum number of leaves in one tree
    feature fraction  $\leftarrow$  1.0          # 100% features are selected
    bagging fraction  $\leftarrow$  1.0          # randomly selects data without resampling
    bagging frequency  $\leftarrow$  0          # disable row sampling
    min number data in leaf  $\leftarrow$  20    # controls overfitting
    number of iterations  $\leftarrow$  150    # number of boosting iterations
    early stopping round  $\leftarrow$  25      # boosting will not give up till 25 rounds
                                           # helps to overcome the problem of validation
    learning rate  $\leftarrow$  0.1            # improves training loss
    verbosity  $\leftarrow$  1                  # provides information about training and
                                           scoring
    }
Step 3 Train_Dataset  $\leftarrow$  model.Dataset( $X_{train}, Y_{label}$ , feature_name=label,
    categorical_feature=['Class'])
Step 5 model  $\leftarrow$  model.train(parameters, Train_Dataset, num_boost_round=50)

```

Algorithm 5: Segment Voice

Setup: $i \leftarrow 0$

Step 1 prediction $\leftarrow$ model.predict(N)

Step 2 for predict in prediction:


```

Step 3      if max(predict)==Ylabel[i] :
Step 4          v ← Xtrain[i]
Step 5          l ← Ylabel[i]
Step 6          play(v)
Step 7      else
Step 8          other_voice ← Xtrain[i]
Step 9      end if
Step 10     i ← increment by 1
Step 11 end for

```

In Algorithm 4 voice samples of male, female, mixed are split into training and testing sets with labels assigned to each voice print in algorithm 3. The parameters are selected according to Laurance[70]. Several test runs were carried out before arriving at the optimum parameters and their values. At the end of algorithm model is ready for prediction. For prediction Normalized fused voice sample N is used. Maximum of predicted output is matched with the label stored in Y_{train}, if a match is found the voice signal is retained from the dataset using the label. The voice retrieved is a processed voice. To get the original voice print EPV is used.

Algorithm 6: EPV

```

Setup: model ← [] Initialize Model
      D ← [] initialize structure to hold data with labels
      index ← 0
Start:
Step 1 D ← ['Male':{Male},'Female':{Female},'Assorted':{Assorted}]
Step 2 D ← append class 'signature'
Step 3 while not end of D:
Step 4     signature[index]=index
Step 5     index ← +1
Step 6 X ← {Male, Female, Assorted}
Step 7 Y ← signature
Step 8 parameters ← {
                    boosting_type : gbd
                    ,objective: multiclass
                    , metric' : multi_logloss
                    , min_data: 1
                    , num_class : length of signature
                    }
Step 9 Train_Dataset ← model.Dataset(Xtrain, Y, feature_name=
                    ['Male', 'Female' , 'Assorted'],
                    categorical_feature=['Signature'])
Step 10 model ← model.train(parameters, Train_Dataset, num_boost_round=50)
Step 11 predicted ← model.predict(predict)
      reset index to ← 0
Step 12 for predict in predicted:
Step 13     if max(predict) matches with Y[index] then
Step 14         Ev ← X[index]
Step 15     end if
Step 16     index ← +1
Step 17 end for

```

EPV is a simple takes multiclass parameter for multiclass classification. Original voices are taken as the input dataset and labels are assigned from 1 to the length of the samples. Once the model is trained, prediction analysis is done on the predicted output received after running algorithm 5. The predicted voice print Ev is the enhanced voice print which is very close to the original voice. Received voice is converted into text to ascertain the claim.

III. Accuracy and Comparison

For accuracy True Negative(TN), False Negative(FN), True Positive(TP), and True Negative(TN) are measured. These are further used to calculate Precision, Recall, Accuracy and F1 score. The mathematical formulation used is:

$$Precision = \frac{TP}{TP + FP} \dots \text{eq(1)}$$

$$Recall = \frac{TP}{TP + FN} \dots \text{eq(2)}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \dots \text{eq(3)}$$

$$F1 = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)} \dots \text{eq(4)}$$

TP is the number of correctly detected sounds (predicted), FN is the number of voices that have not been correctly identified or one may claim that they have been wrongly identified. FP is the number of speech signals known as voice signals, but they are not. TN is the number of not a Speech Signal correctly defined.

Qualitative comparison is drawn with other models using source-to-distortion ratio (SDR)[44]. Other measurement measures include signal-to-distortion ratio improvement (SI-SDR)[45], perceptual estimate of speech efficiency (PESQ) scores[46], scale-invariant signal to-noise ratio (SI-SNR)[47]. SDR, SI-SDR, PESQ, SI-SNR higher values reflect better quality of separation.

SDR is represented as:

$$SDR = 10 \log_{10} \left(\frac{\|S\|^2}{\|e_{interf} + e_{noise} + e_{artif}\|^2} \right) \dots \text{eq(5)}$$

SI-SDR is represented as:

$$SI - SDR = 10 \log_{10} \left(\frac{\left\| \frac{es^L \times s}{P} \times S \right\|^2}{\left\| \frac{es^L \times s}{P} \times S - es \right\|^2} \right) \dots \text{eq(6)}$$

S-I SNR is represented as:

$$T_s = \frac{\langle es, S \rangle_S}{\|P\|} \dots \text{eq(7)}$$

$$e_n = es - T_s \dots \text{eq(8)}$$

$$SI - SNR = 10 \log_{10} \frac{\|T_s\|^2}{\|e_n\|} \dots \text{eq(9)}$$

PESQ is represented as:

$$PESQ = 4.5 - 0.1d_s - 0.0309d_A \dots \text{eq(10)}$$

Here, S and es represent original and estimated clean source respectively. L represents length of the signal. e_{interf} , e_{noise} , e_{artif} represent interferences, noise and artifacts error terms respectively. P represents power of the signal (S, S). T_s , e_n represent Target noise and estimated noise respectively. d_s , d_A represent symmetric and asymmetric disturbances. S and es are both normalized to have zero-mean to ensure scale-invariance.

IV. Results

The discusses the results obtained. Spectrograms of original voices with fused voice is first produced followed by the quality and time check.

Figure 4: Fused voice Spectrogram

Figure 5: Original male voice spectrogram

Figure 6: Estimated male voice spectrogram

Figure 7: Spectrogram of original female voice

Figure 8: Spectrogram of estimated female voice

Figure 9: Spectrogram of original assorted voice data(1)

Figure 10: Spectrogram of estimated assorted voice data(1)

Figure 11: Spectrogram of original male voice from assorted data(2)

Figure 12: Spectrogram of estimated male voice from assorted data(2)

Figures 4, displays the spectrogram of the fused voices. Four voices are fused from each dataset. 5, 7, 9, 11 show spectrogram of original male, female, assorted voice(1) and assorted voice(2). Assorted voice is taken from the recorded voices and from the benchmark datasets. The plots are between time and amplitude and between time and frequency domain. Figures 6, 8, 10, 12 are estimated speech signals of male, female, assorted voice(1) and voice(2). They are predicted from the fused signal. The recovered voices are almost similar to the original voices. Although, from the plots one can, be assured that the retrieved voices are of good quality. To confirm the claim robustness tests are carried out.

Table 1: Test results on WSJ dataset

Model	SDR	SI-SDR	PESQ	SI-SNR
VoSE	12.52	11.94	2.99	11.62
Y. Jin [46]	10.94	10.75	2.89	10.72
Chen [47]	10.8	10.4	2.82	10
M. Kolbæk [45]	10	-	2.64	-
M. Kolbæk [45]	9.4	-	-	-

Higher values of SDR, SI-SDR, PESQ, SI-SNR represents better quality of the signal. Table 3 shows the values of the parameters tested on WSJ dataset. The values are calculated on the detected voices.

Table 2: Test results on Festvox CMU_ARCTIC dataset

Model	SDR	SI-SDR	PESQ	SI-SNR
VoSE	11.42	10.94	2.92	10.92

Table 3: Test results on VoxForge Speech Corpus

Model	SDR	SI-SDR	PESQ	SI-SNR
VoSE	13.32	12.89	3.15	12.72

Table 4: Test results on Microsoft Indian Language Corpus

Model	SDR	SI-SDR	PESQ	SI-SNR
VoSE	11.25	10.76	2.75	10.52

Table 5: Test results on Linguistic Data Consortium for Indian Languages(LDC-IL)

Model	SDR	SI-SDR	PESQ	SI-SNR
VoSE	11.12	10.28	2.14	10.02

Table 2, 3,4 and 5 shows the values of SDR, SI-SDR, PESQ, and SI-SNR on Festvox, VoxForge, Microsoft Indian Language Corpus, and LDC-IL dataset. To the best of the knowledge of the authors, the datasets have not been tested on the above parameters for cocktail party problem.

Table 6: Different Classification Algorithms Tested

Type	TP	FP	TN	FN
Kmeans	75	0	74	1
Decision Stumps	75	0	73	2
Naïve Bayes	75	0	73	2
LSTM	75	0	74	1
VoSE	75	0	75	0

Table 7: Precision, recall, accuracy and F1-score

Type	Precision	Recall	Accuracy	F1-score
Kmeans	0.986842105	1	0.9933333	0.9933775
Decision Stumps	0.974025974	1	0.9866667	0.9868421
Naïve Bayes	0.974025974	1	0.9866667	0.9868421
LSTM	0.986842105	1	0.9933333	0.9933775
VoSE	1	1	1	1

To test the robustness further Precision, Recall, Accuracy, and F1 score are calculated. Table 6 shows the number of samples tested on different algorithms. FP, TP, FN, and TN are recorded for each algorithm. The values Table 7 are based on Table 6 values. the mathematical formulations are explained in equations 1-4.

Figure 13: Time taken

Along with accuracy it is equally important to calculate the time. Higher time consumed would defeat the purpose of the model even if the Accuracy is high. The proposed model VoSE, using LightGBM consumes much lesser time for all the three datasets.

4.1 Speech to Text

The output of the enhanced predicted voices are :

Converting audio transcripts into text ...(English)

whenever his friends ask him if you would like to go with them

Converting audio transcripts into text ...(Hindi)

श्रीनगर टोही उपग्रहों को मार गिरा सकता है तो भारत अपने ऊपर धरती पर सकुशल उतार सकता है

Converting audio transcripts into text ...(Tamil)

டெல்லியில் தேசிய கொடியை ஏற்றி வைத்து நாட்டு மக்களுக்கு பிரதமர் ந
ரேந்திர மோடி உரை

Three outputs of voice to speech conversion are reproduced above. English and Tamil conversions are perfect but there is a small error with Hindi. First word is not correct the correct words were Yadi Cheen (if China). Out of 100 Hindi voice samples only the above voice shows an error. The above outputs are from the code written in python. The code uses speech recognition module. The module is utilizes google speech recognition for the conversion.

V. Conclusion

This paper presents a model based on gradient boosting algorithm. The objective of VoSE is to separate the voices from a mixed signal and enhance them. The model is able to successfully separate male, female, assorted voices, and other voices from a mixed signal. The algorithm is compared with benchmark algorithm like, Kmeans, Decision Stumps, Naïve Bayes, and LSTM. The comparison is drawn by running the algorithms on the dataset created for the proposed work. Two main objectives of VoSE- to separate the voices from a mixed signal, and to enhance the separated voices are achieved in good time. The results show that VoSE consumes lesser time than K-means, Decision Stumps, Naïve Bayes, and LSTM. An accuracy of 99.99% shows that it performs better than the considered algorithms. The quality of the recovered voices is measured using SI, SI-SDR, PESQ, and SI-SNR. Higher values indicate that the quality of the recovered voice is good.

VoSE can be used to design hearing aid which can give crystal clear sound to the hearing impaired. The scope of the model is not limited to one application. VoSE can be utilized by any voice response system like, Siri, Alexa, Assistant which as of now work on single voice command. VoSE can also be used for audio Bots. In future authors plan to develop a self-learning algorithm that can decode the voices from any source and silence the noises completely. The current research is limited to separation and enhancement of known mixed voices. VoSE is the first step towards the final goal of designing a robust system which would be able to identify the voices from unknown speakers and sources.

VI. References

- [1] B. Sagi, S. C. Nemat-Nasser, R. Kerr, R. Hayek, C. Downing and R. Hecht-Nielsen, "A Biologically Motivated Solution to the Cocktail Party Problem," in Neural Computation, vol. [2] 13, no. 7, pp. 1575-1602, 1 July 2001, doi: 10.1162/089976601750265018.
- [2] S. Haykin and Z. Chen, "The Cocktail Party Problem," in Neural Computation, vol. 17, no. 9, pp. 1875-1902, 1 Sept. 2005, doi: 10.1162/0899766054322964.

- [3] Stages of Listening. https://saylordotorg.github.io/text_stand-up-speak-out-the-practice-and-ethics-of-public-speaking/s07-04-stages-of-listening.html. Accessed 25 July 2020.
- [4] I. Yasin, V. Drga, F. Liu, A. Demosthenous and R. Meddis, "Optimizing Speech Recognition Using a Computational Model of Human Hearing: Effect of Noise Type and Efferent Time Constants," in *IEEE Access*, vol. 8, pp. 56711-56719, 2020, doi: 10.1109/ACCESS.2020.2981885.
- [5] L. Burbach, P. Halbach, N. Plettenberg, J. Nakayama, M. Ziefle and A. Calero Valdez, "'Hey, Siri', 'Ok, Google', 'Alexa'. Acceptance-Relevant Factors of Virtual Voice-Assistants," 2019 IEEE International Professional Communication Conference (ProComm), Aachen, Germany, 2019, pp. 101-111, doi: 10.1109/ProComm.2019.00025.
- [6] K. T. Deepak and S. R. M. Prasanna, "Foreground Speech Segmentation and Enhancement Using Glottal Closure Instants and Mel Cepstral Coefficients," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 24, no. 7, pp. 1205–1219, Jul. 2016, doi: 10.1109/TASLP.2016.2549699.
- [7] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," *J. Acoust. Soc. Amer.*, vol. 25, no. 5, pp. 975_979, Sep. 1953.
- [8] M. Cooke, J. R. Hershey, and S. J. Rennie, "Monaural speech separation and recognition challenge," *Comput. Speech Lang.*, vol. 24, no. 1, pp. 1_15, Jan. 2010.
- [9] H. Kamper, A. Jansen, and S. Goldwater, "Unsupervised Word Segmentation and Lexicon Discovery Using Acoustic Word Embeddings," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 24, no. 4, pp. 669–679, Apr. 2016, doi: 10.1109/TASLP.2016.2517567.
- [10] T. T. Chan and Y. Yang, "Complex and Quaternionic Principal Component Pursuit and Its Application to Audio Separation," in *IEEE Signal Processing Letters*, vol. 23, no. 2, pp. 287-291, Feb. 2016, doi: 10.1109/LSP.2016.2514845.
- [11] W. Biesmans, N. Das, T. Francart, and A. Bertrand, "Auditory-Inspired Speech Envelope Extraction Methods for Improved EEG-Based Auditory Attention Detection in a Cocktail Party Scenario," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 5, pp. 402–412, May 2017, doi: 10.1109/TNSRE.2016.2571900.
- [12] A. H. Abo Absa, M. Deriche, M. Elshafei-Ahmed, Y. M. Elhadj, and B.-H. Juang, "A Hybrid Unsupervised Segmentation Algorithm for Arabic Speech Using Feature Fusion and a Genetic Algorithm (July 2018)," *IEEE Access*, vol. 6, pp. 43157–43169, 2018, doi: 10.1109/ACCESS.2018.2859631.
- [13] R. Lu, Z. Duan and C. Zhang, "Audio–Visual Deep Clustering for Speech Separation," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1697-1712, Nov. 2019, doi: 10.1109/TASLP.2019.2928140.
- [14] B. Wiem, B. M. Mohamed Anouar and B. Aïcha, "Phase-aware subspace decomposition for single channel speech separation," in *IET Signal Processing*, vol. 14, no. 4, pp. 214-222, 6 2020, doi: 10.1049/iet-spr.2019.0373.
- [15] D. Ellis, "Computational auditory scene analysis exploiting speech-recognition knowledge," *Proceedings of 1997 Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, USA, 1997, pp. 4 pp.-, doi: 10.1109/ASPAA.1997.625625.
- [16] P. Li, Y. Guan, B. Xu and W. Liu, "Monaural Speech Separation Based on Computational Auditory Scene Analysis and Objective Quality Assessment of Speech," in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 6, pp. 2014-2023, Nov. 2006, doi: 10.1109/TASL.2006.883258.
- [17] P. Li, Y. Guan, W. Liu and B. Xu, "Combining Machine Learning and Computational Auditory Scene Analysis to Separate Monaural Speech of Two-Talker," 2007 International Conference on Natural Language Processing and Knowledge Engineering, Beijing, 2007, pp. 280-284, doi: 10.1109/NLPKE.2007.4368044.
- [18] Q. Kong, Y. Wang, X. Song, Y. Cao, W. Wang and M. D. Plumbley, "Source Separation with Weakly Labelled Data: an Approach to Computational Auditory Scene Analysis," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 101-105, doi: 10.1109/ICASSP40776.2020.9053396.
- [19] A. Erell and D. Burshtein, "Noise adaptation of HMM speech recognition systems using tied-mixtures in the spectral domain," in *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 1, pp. 72-74, Jan. 1997, doi: 10.1109/89.554271.
- [20] S. E. Bou-Ghazale and J. H. L. Hansen, "HMM-based stressed speech modelling with application to improved synthesis and recognition of isolated speech under stress," in *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 3, pp. 201-216, May 1998, doi: 10.1109/89.668815.
- [21] C. Lee and S. Lee, "Noise-Robust Speech Recognition Using Top-Down Selective Attention With an HMM Classifier," in *IEEE Signal Processing Letters*, vol. 14, no. 7, pp. 489-491, July 2007, doi: 10.1109/LSP.2006.891326.

- [22] C. Do, D. Pastor and A. Goalic, "On the Recognition of Cochlear Implant-Like Spectrally Reduced Speech With MFCC and HMM-Based ASR," in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 1065-1068, July 2010, doi: 10.1109/TASL.2009.2032945.
- [23] K. Naithani, V. M. Thakkar and A. Semwal, "English Language Speech Recognition Using MFCC and HMM," 2018 International Conference on Research in Intelligent and Computing in Engineering (RICE), San Salvador, 2018, pp. 1-7, doi: 10.1109/RICE.2018.8509046.
- [24] A. D. S. Dm, R. D. Souza and K. Mohan, "Speech Based Emotion Recognition Using Combination of Features 2-D HMM Model," 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2019, pp. 381-385, doi: 10.1109/I-SMAC47947.2019.9032453.
- [25] M. Novak and R. Mammone, "Improvement of non-negative matrix factorization based language model using exponential models," *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2001. ASRU '01., Madonna di Campiglio, Italy, 2001, pp. 190-193, doi: 10.1109/ASRU.2001.1034619.
- [26] A. Bertrand, K. Demuynck, V. Stouten and H. Van hamme, "Unsupervised learning of auditory filter banks using non-negative matrix factorisation," 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, Las Vegas, NV, 2008, pp. 4713-4716, doi: 10.1109/ICASSP.2008.4518709.
- [27] S. U. N. Wood, J. Rouat, S. Dupont, and G. Pironkov, "Blind Speech Separation and Enhancement With GCC-NMF," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 4, pp. 745-755, Apr. 2017, doi: 10.1109/TASLP.2017.2656805.
- [28] N. C. Nag and M. S. Shah, "Investigating Single Channel Source Separation Using Non-Negative Matrix Factorization and Its Variants for Overlapping Speech Signal," 2019 International Conference on Nascent Technologies in Engineering (ICNTE), Navi Mumbai, India, 2019, pp. 1-6, doi: 10.1109/ICNTE44896.2019.8946013.
- [29] A. M. Peinado, V. Sanchez, J. L. Perez-Cordoba and A. J. Rubio, "Efficient MMSE-based channel error mitigation techniques. Application to distributed speech recognition over wireless channels," in *IEEE Transactions on Wireless Communications*, vol. 4, no. 1, pp. 14-19, Jan. 2005, doi: 10.1109/TWC.2004.840198.
- [30] H. K. Kim and R. C. Rose, "Cepstrum-Domain Model Combination Based on Decomposition of Speech and Noise Using MMSE-LSA for ASR in Noisy Environments," in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 4, pp. 704-713, May 2009, doi: 10.1109/TASL.2008.2012319.
- [31] J. A. González, A. M. Peinado, A. M. Gomez, J. L. Carmona and J. A. Morales-Cordovilla, "Efficient VQ-based MMSE estimation for robust speech recognition," 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, Dallas, TX, 2010, pp. 4558-4561, doi: 10.1109/ICASSP.2010.5495566.
- [32] C. H. You and B. Ma, " β -Masking MMSE speech enhancement for speech recognition," 2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP), Singapore, 2017, pp. 341-345, doi: 10.1109/SIPROCESS.2017.8124561.
- [33] D. Yu, G. Hinton, N. Morgan, J. Chien and S. Sagayama, "Introduction to the Special Section on Deep Learning for Speech and Language Processing," in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 4-6, Jan. 2012, doi: 10.1109/TASL.2011.2173371.
- [34] L. Deng and X. Li, "Machine Learning Paradigms for Speech Recognition: An Overview," in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 5, pp. 1060-1089, May 2013, doi: 10.1109/TASL.2013.2244083.
- [35] G. Wang and K. C. Sim, "Regression-Based Context-Dependent Modeling of Deep Neural Networks for Speech Recognition," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 11, pp. 1660-1669, Nov. 2014, doi: 10.1109/TASLP.2014.2344855.
- [36] P. Zhou, H. Jiang, L. Dai, Y. Hu and Q. Liu, "State-Clustering Based Multiple Deep Neural Networks Modeling Approach for Speech Recognition," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 4, pp. 631-642, April 2015, doi: 10.1109/TASLP.2015.2392944.
- [37] M. Kolbæk, D. Yu, Z. Tan and J. Jensen, "Joint separation and denoising of noisy multi-talker speech using recurrent neural networks and permutation invariant training," 2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP), Tokyo, 2017, pp. 1-6, doi: 10.1109/MLSP.2017.8168152.
- [38] H. Meng, T. Yan, F. Yuan and H. Wei, "Speech Emotion Recognition From 3D Log-Mel Spectrograms With Deep Learning Network," in *IEEE Access*, vol. 7, pp. 125868-125881, 2019, doi: 10.1109/ACCESS.2019.2938007.

- [39] G. Zhong, K. Zhang, H. Wei, Y. Zheng and J. Dong, "Marginal Deep Architecture: Stacking Feature Learning Modules to Build Deep Learning Models," in *IEEE Access*, vol. 7, pp. 30220-30233, 2019, doi: 10.1109/ACCESS.2019.2902631.
- [40] Y. Tu, J. Du and C. Lee, "Speech Enhancement Based on Teacher–Student Deep Learning Using Improved Speech Presence Probability for Noise-Robust Speech Recognition," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2080-2091, Dec. 2019, doi: 10.1109/TASLP.2019.2940662.
- [41] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh and K. Shaalan, "Speech Recognition Using Deep Neural Networks: A Systematic Review," in *IEEE Access*, vol. 7, pp. 19143-19165, 2019, doi: 10.1109/ACCESS.2019.2896880.
- [42] Mustaqeem, M. Sajjad and S. Kwon, "Clustering-Based Speech Emotion Recognition by Incorporating Learned Features and Deep BiLSTM," in *IEEE Access*, vol. 8, pp. 79861-79875, 2020, doi: 10.1109/ACCESS.2020.2990405.
- [43] D. Yu, M. Kolbæk, Z. Tan and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, 2017, pp. 241-245, doi: 10.1109/ICASSP.2017.7952154.
- [44] M. Kolbæk, D. Yu, Z. Tan and J. Jensen, "Multitalker Speech Separation With Utterance-Level Permutation Invariant Training of Deep Recurrent Neural Networks," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901-1913, Oct. 2017, doi: 10.1109/TASLP.2017.2726762.
- [45] Fan, Cunhang & Liu, Bin & Tao, Jianhua & Yi, Jiangyan & Wen, Zhengqi. (2019). Discriminative Learning for Monaural Speech Separation Using Deep Embedding Features. 4599-4603. 10.21437/Interspeech.2019-1940.
- [46] Chen, Zhuo & Luo, Yi & Mesgarani, Nima. (2017). Speaker-independent Speech Separation with Deep Attractor Network. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. PP. 10.1109/TASLP.2018.2795749.
- [47] Y. Jin, C. Tang, Q. Liu, and Y. Wang, "Multi-Head Self-Attention-Based Deep Clustering for Single-Channel Speech Separation," *IEEE Access*, vol. 8, pp. 100013–100021, 2020, doi: 10.1109/ACCESS.2020.2997871.
- [48] Ephrat, Ariel & Mosseri, Inbar & Lang, Oran & Dekel, Tali & Wilson, Kevin & Hassidim, Avinatan & Freeman, William & Rubinstein, Michael. (2018). Looking to Listen at the Cocktail Party: A Speaker-Independent Audio-Visual Model for Speech Separation. *ACM Transactions on Graphics*. 37. 10.1145/3197517.3201357.
- [49] R. Lu, Z. Duan, and C. Zhang, "Listen and Look: Audio–Visual Matching Assisted Speech Source Separation," *IEEE Signal Process. Lett.*, vol. 25, no. 9, pp. 1315–1319, Sep. 2018, doi: 10.1109/LSP.2018.2853566.
- [50] "Machine Learning Algorithms Mindmap." Jixta, 17 July 2015, <https://jixta.wordpress.com/2015/07/17/machine-learning-algorithms-mindmap/>.
- [51] Features — LightGBM 2.3.2 Documentation. <https://lightgbm.readthedocs.io/en/latest/Features.html>. Accessed 27 July 2020.
- [52] Choudhury, Ambika. "Comparing The Gradient Boosting Decision Tree Packages: XGBoost vs LightGBM." *Analytics India Magazine*, 1 Aug. 2019, <https://analyticsindiamag.com/comparing-the-gradient-boosting-decision-tree-packages-xgboost-vs-lightgbm/>.
- [53] "Why TensorFlow." TensorFlow, <https://www.tensorflow.org/about>. Accessed 27 July 2020.
- [54] Experiments—LightGBM 2.3.2 Documentation. <https://lightgbm.readthedocs.io/en/latest/Experiments.html>. Accessed 29 July 2020.
- [55] Y. Zhou, Q. Sun, and S. Lin, "Link State Aware Dynamic Routing and Spectrum Allocation Strategy in Elastic Optical Networks," *IEEE Access*, vol. 8, pp. 45071–45083, 2020, doi: 10.1109/ACCESS.2020.2977612.
- [56] C. Zhang et al., "Weather Visibility Prediction Based on Multimodal Fusion," *IEEE Access*, vol. 7, pp. 74776–74786, 2019, doi: 10.1109/ACCESS.2019.2920865.
- [57] X. Yang and J. Ding, "A Computational Framework for Iceberg and Ship Discrimination: Case Study on Kaggle Competition," *IEEE Access*, vol. 8, pp. 82320–82327, 2020, doi: 10.1109/ACCESS.2020.2990985.
- [58] Y. Xia, "A Novel Reject Inference Model Using Outlier Detection and Gradient Boosting Technique in Peer-to-Peer Lending," *IEEE Access*, vol. 7, pp. 92893–92907, 2019, doi: 10.1109/ACCESS.2019.2927602.

- [59] A. A. Taha and S. J. Malebary, "An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine," *IEEE Access*, vol. 8, pp. 25579–25587, 2020, doi: 10.1109/ACCESS.2020.2971354.
- [60] Y. Qu, Z. Lin, H. Li, and X. Zhang, "Feature Recognition of Urban Road Traffic Accidents Based on GA-XGBoost in the Context of Big Data," *IEEE Access*, vol. 7, pp. 170106–170115, 2019, doi: 10.1109/ACCESS.2019.2952655.
- [61] S. M. Krishna Moorthy, K. Calders, M. B. Vicari, and H. Verbeeck, "Improved Supervised Learning-Based Approach for Leaf and Wood Classification From LiDAR Point Clouds of Forests," *IEEE Trans. Geosci. Remote Sensing*, vol. 58, no. 5, pp. 3057–3070, May 2020, doi: 10.1109/TGRS.2019.2947198.
- [62] Y. Ju, G. Sun, Q. Chen, M. Zhang, H. Zhu, and M. U. Rehman, "A Model Combining Convolutional Neural Network and LightGBM Algorithm for Ultra-Short-Term Wind Power Forecasting," *IEEE Access*, vol. 7, pp. 28309–28318, 2019, doi: 10.1109/ACCESS.2019.2901920.
- [63] G. Joo, Y. Song, H. Im, and J. Park, "Clinical Implication of Machine Learning in Predicting the Occurrence of Cardiovascular Disease Using Big Data (Nationwide Cohort Data in Korea)," *IEEE Access*, vol. 8, pp. 157643–157653, 2020, doi: 10.1109/ACCESS.2020.3015757.
- [64] X. Fei, Q. Zhang, and Q. Ling, "Vehicle Exhaust Concentration Estimation Based on an Improved Stacking Model," *IEEE Access*, vol. 7, pp. 179454–179463, 2019, doi: 10.1109/ACCESS.2019.2958703.
- [65] C. Dong, G. He, X. Liu, Y. Yang, and W. Guo, "A Multi-Layer Hardware Trojan Protection Framework for IoT Chips," *IEEE Access*, vol. 7, pp. 23628–23639, 2019, doi: 10.1109/ACCESS.2019.2896479.
- [66] J. Cao et al., "A Novel False Data Injection Attack Detection Model of the Cyber-Physical Power System," vol. 8, p. 17, 2020.
- [67] Md. W. Ahmad et al., "Mal-Light: Enhancing Lysine Malonylation Sites Prediction Problem Using Evolutionary-based Features," *IEEE Access*, vol. 8, pp. 77888–77902, 2020, doi: 10.1109/ACCESS.2020.2989713.
- [68] Jeena J. Prakash, Golda Brunet Rajan, and Hema A. Murthy. 2019. Importance of Signal Processing Cues in Transcription Correction for Low-Resource Indian Languages. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 19, 1, Article 14 (January 2020), 26 pages. DOI:<https://doi.org/10.1145/3342352>
- [69] C. Ding et al., "Towards Burmese (Myanmar) Morphological Analysis: Syllable-based Tokenization and Part-of-speech Tagging," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 19, no. 1, p. 5:1–5:34, May 2019, doi: 10.1145/3325885.
- [70] *Laurae++: Xgboost / LightGBM - Parameters*.<https://sites.google.com/view/lauraepp/parameters>. Accessed 30 July 2020.

Figures

Figure 1

Working model graphically represented. Here V_s is the single voice used for prediction.

Figure 2

Voice plot with leading, trailing and in between silence

Figure 3

Trimmed Voice plot

Figure 4

Fused voice Spectrogram

Figure 5

Original male voice spectrogram

Figure 6

Estimated male voice spectrogram

Figure 7

Spectrogram of original female voice

Figure 8

Spectrogram of estimated female voice

Figure 9

Spectrogram of original assorted voice data(1)

Figure 10

Spectrogram of estimated assorted voice data(1)

Figure 11

Spectrogram of original male voice from assorted data(2)

Figure 12

Spectrogram of estimated male voice from assorted data(2)

Figure 13

Time taken