G-Cocktail: An Algorithm to Address Cocktail Party Problem of Gujarati Language using CatBoost

Monika Gupta (monikaguptautu@gmail.com)
Uttarakhand Technical University
R K Singh
BTKIT Dwarahat
Sachin Singh
NIT Delhi

Research Article

Keywords: Voice identification, Gujarati, CatBoost, Indian Voice, Features

Posted Date: March 17th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-305722/v1

Noname manuscript No.
(will be inserted by the editor)

*G-Cocktail: An algorithm to address cocktail party problem of Gujarati language using CatBoost

Monika Gupta · Dr R K Singh · Dr Sachin Singh

Received: date / Accepted: date

Abstract The pandemic caused due to COVID-19, has seen things going online. People tired of typing prefer to give voice commands. Most of the voice based applications and devices are not prepared to handle the native languages. Moreover, in a party environment it is difficult to identify a voice command as there are many speakers. The proposed work addresses the Cocktail party problem of Indian language, Gujarati. The voice response systems like, Siri, Alexa, Google Assistant as of now work on single voice command. The proposed algorithm G- Cocktail would help these applications to identify command given in Gujarati even from a mixed voice signal. Benchmark Dataset is taken from Microsoft and Linguistic Data Consortium for Indian Languages(LDC-IL) comprising single words and phrases. G-Cocktail utilizes the power of CatBoost algorithm to classify and identify the voice. Voice print of the entire sound files is created using Pitch, and Mel Frequency Cepstral Coefficients (MFCC). Seventy percent of the voice prints are used to train the network and thirty percent for testing. The proposed work is tested and compared with K-means, Naïve Bayes, and LightGBM.

Keywords Voice identification · Gujarati · CatBoost · Indian Voice · Features

Monika Gupta
Electronics Engineering, Uttarakhand Technical University, Dehradun, Uttarakhand, India
E-mail: monikaguptautu@gmail.com
Dr R K Singh
Electronics and communication Engineering, BTKIT, Dwarahat, Uttarakhand, India E-mail: rksingh12@rediffmail.com
Dr Sachin Singh
Electronics and Electrical Engineering, NIT, Delhi, Delhi, India
E-mail: sachinsingh@nitdelhi.ac.in

1 Introduction

The pandemic in 2020 not only increased the internet usage but also increased the use of native languages. A user of a voice enabled device would prefer to use its native language. Expressions spoken in a natural language is a constantly shifting acoustic signal. These 'acoustic- phonetic parts' (APS) are demarcated based on distinct variations in time and frequency domains [1]. Any alteration in these realms should not be perceived as a segment boundary therefore segment demarcation decisions are based on phonetic criteria [2] [3]. We can associate the same signal that is demarcated in APSs with linguistic abstractions such as allophones, phonemes, morphophonemes, etc. Phoneme is a sound element that differentiates one word from another in each language. Allophones are the sounds of the same phoneme that vary but do not cause a significant difference in the expression. The similarity of strings with one or more APSs with allophones, however, may never be one-to-one. This inherent lack of similarities is compounded by the fact that the speech signal generator, i.e., the human speaker, does not produce invariable and exact copies of the signal for several instances of the 'same' allophone [4]. Consequently, not all languages have the same phenomes.

India is home to over 19,565 mother tongues/dialects, according to 2011 Census. Every language has its own phonotactic, prosodic, and acoustic characteristics [5]. As a result, identifying these languages, each with its own vernacular, cadence, semantics, and ambiance, becomes exceedingly difficult [6][7][8]. The focus of the paper is on Gujarati.

Gujarati has many dialects; the main ones are spoken in Mumbai and Ahmedabad. Others are: Surati, Kathiyawadi, Kharua, Khakari, Tarimukhi, and East African Gujarati. Since many dialects exist, there are many loan words from other languages. The dialects of southern Gujarati have borrowed words from Hindi, English, and Portuguese. Gujarati has eight vowels. Excluding [e] and [o], vowels are nasalised and in murmured and non-murmured forms. Gujarati has vowels that are short and long, but they do not conflict. Gujarati has 31 consonants, including 20 stops, 3 fricatives, 3 nasals, and 5 glides and liquids. At five distinct positions, the stops and nasals are articulated and graded as: labial, dental, retroflex, palatal, and velar [9]. In fact, the palatal stops are affricated. The four-way difference between Indo-Aryan and Indo-European languages (Proto-Indo-European had a three-way difference only) involves voiceless and voiced consonants, unaspirated and aspirated in each sequence of stops. Despite a large speaker base of Gujarati not much work is done in its speech enhancement, Text to Speech conversion and separation. Gujarati, like many Indian languages, has many dialects. Analyzing Gujarati requires different phonetic distribution. The current paper focuses on separating Gujarati voices from a mixed signal. More commonly called "cocktail party scene" problem.

Simultaneous and sequential arrangement are the two forms of auditory scene analysis. Simultaneous organization (or grouping) incorporates sounds that overlap, while sequential organization combines sounds that occur at different times [10]. Proximity in frequency and duration, harmonicity, common amplitude and frequency modulation, onset and offset synchrony, commonplace, and prior information are the key organizational concepts responsible for ASA when we express audio on a time-frequency image such as a spectrogram. These grouping rules often regulate speech separation [11][12][13][14]. The key point in speech recognition is that the sounds made by a human being are filtered by the vocal tract's shape, including tongue, teeth, etc. This shape defines the sound that comes out. The accurate assessment of the shape would give us an accurate representation of the phoneme being produced. The frame of the vocal tract embodies itself in the envelope of the short time power range. MFCC can precisely represent this envelope. Because of its ability of representing the envelope and high accuracy, MFCC is widely used feature for the voice signals [15]. The MFCC computation is a simulation of the human auditory system that seeks to mechanically enforce the ear's operating principle, believing that the human ear is a reliable speaker recognizer [16]. The other features are Linear Prediction Coefficient (LPC), Discrete Wavelet Transform (DWT), Linear Predictive Cepstral Coefficients (LPCC), and deep learning-based features [17].

LPC are a type of speech features that imitates the human vocal tract. It estimates the concentration and frequency of the left-over residue by approximating the formants, removing their effects from the speech signal, and evaluating the speech signal [18] [19]. Each sample of the signal is stated to be a direct incorporation of previous samples in the result. The formants are defined by the coefficients of the difference equation, so LPC must approximate these coefficients. LPC is a popular formant estimation method as well as a powerful speech analysis method. It provides very precise speech parameter estimates and is computationally efficient. Autocorrelation coefficients are aliased in conventional linear prediction. The susceptibility of LPC estimates to quantization noise is high, so they are not well suited for generalization [19].

DWT is an extension of Wavelet Transform (WT). It can derive information from latent signals in both the time and frequency domains at the same time. Many wavelets are orthogonal, which is an outstanding feature for compact signal representation [19] [20]. The wavelet transform breaks down a signal into a set of simple functions known as wavelets. It uses Dilations and flipping to build wavelets from a single template wavelet called mother wavelet. The WT's key feature is that it searches the frequency spectrum with a variable window, improving the temporal resolution [21]. Its parameters include different frequency scales. This enhances the speech information received in the related frequency. It provides enough frequency bands for accurate speech processing, but since the input signals are of finite duration, the wavelet coefficients may have excessively broad differences at the boundaries because of discontinuities [20] [21].

LPCC are Cepstral Coefficients derived from LPC. They illustrate the coefficients of the Fourier transform of the logarithmic magnitude spectrum of LPC [6][17][22]. The susceptibility of LPCC calculations to quantization noise is well documented. In the quefrency domain, cepstral analysis on a high-pitch speech signal yields poor source-filter separability. Lower-order cepstral coefficients are sensitive to spectral slope, whereas higher-order cepstral coefficients are sensitive to noise.

From the above study we observed that for separating the voices it is important to choose the right features. Deep Learning based i-vector [24] and x-vector [25] features; a fusion of MFCC, DWT and MFCC, GFCC [17][26] respectively, are good for language recognition. They work well in noisy environment but have overfitting problem with a small dataset. Gujarati like many Indian languages does not have a large data corpus. MFCC best suits and results shows that it has higher accuracy.

For separating the voices several, enhancing the voice signal, and Automatic Speech Recognition (ASR) scientist use various techniques [10] [25-46]. The latest is knowledge distillation [25]. For separating the voices Deep Learning offers Multilayer Perceptron (MLPs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Generative Adversarial Networks (GANs) are commonly used [10]. All the algorithms are good for Automatic Speech Recognition (ASR) but when it comes to smaller dataset they land with overfitting problem.

There is no historical evidence of Cocktail-party scene with Gujarati language [47- 57]. For ASR in Gujarati, methods like Statistical, Neural Networks and End-to-end recognition are used [35]. We considered Catboost [58] to solve the Cocktail-party issue with Gujarati voices. Catboost is:

1. A revolutionary categorical function processing algorithm. There is no need to manually pre-process features because it is done for you. In contrast to other algorithms, the performance of data with categorical features is higher.
2. Ordered boosting is a permutation-driven solution to the classic bosting algorithm. Gradient Boosting easily overfits on tiny datasets. There is a special modification in Catboost to handle this. On datasets where other algorithms struggled with overfitting, Catboost does not have the same issue. We solved the overfitting issue by adjustment the parameters. The details are explained in the model.
3. It is Fast and works on GPU.
4. It handles missing values as well.

2 Methodology

The objective of the paper is to separate voices from a mixed speech in Gujarati and predict the voice.

2.1 Experimental set-up

The setup includes hardware configuration, and dataset used.

2.1.1 Hardware

Intel core i5, fifth generation with 16 GB RAM, NVIDIA Graphic card running on windows 10.

2.1.2 Dataset

Microsoft Indian language Corpus and Linguistic Data Consortium for Indian Languages(LDC-IL). The dataset contains voices of adults only. We recorded few voice samples of the kids keeping the same set of dialogues and the parameters (number of channels, and sampling rate).

2.2 Model

G-cocktail first reads every voice file, filters it by removing the leading and training spaces. It then extracts the MFCC features and calculates the pitch. A method is designed to retain only the relevant features. Pitch and the retained MFCC features of every voice file are stored as a vector to create the testing dataset. Figure 1 graphically illustrates the working.

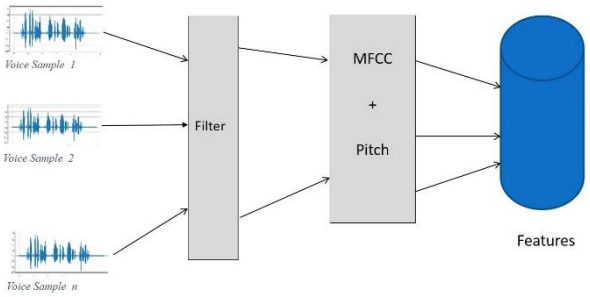

Fig. 1 Feature Extraction

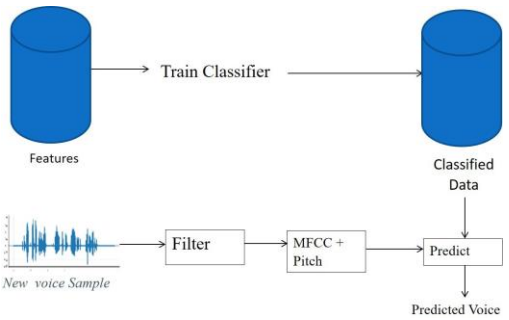

Fig. 2 Classification Model

As shown in fig. 2 the classifier (Catboost in our case) trains on the training set. For prediction, we create a mixed voice as shown in fn(2) and fn(2.1). The overall working of G-cocktail is summarized in the following steps:

Step 1. Filter the voice by removing trailing and leading blanks
Step 2. Calculate pitch of every voice file
Step 3. Extract MFCC features
Step 4. Retain only the relevant features
Step 5. Store features and pitch as a vector of all the voice files
Step 7. Create labels for each record
Step 8. Create a Catboost model
Step 9. Train the model with features and labels.
Step 10. Create a mixed voice by appending different voices
Step 11. Window the mixed voice of 15 seconds
Step 12. Calculate pitch and MFCC features.
Step 13. Using the trained model predict the voice with the new features and pitch.

2.3 Objective function

The objective of the paper is to design a predictive function P such that it can extract individual voice from a mixed voice signal.

$$Detected_v = P(G_{mixed}) \text{ ..... } fn(1)$$

The mixed signal G_{mixed} is created by appending Male, Female or assorted voices. The combinations could be just male or female voices or a mix of both.

$$G_{mixed} = \{G_{male} \vee G_{femal} \vee G_{assorted}\} \text{ ..... fn(2)}$$

Assorted is described in fn(2.1)

$$\{G_{male} \vee G_{femal} \vee G_{female\_kid} \vee G_{male\_kid}\} \text{ ..... fn(2.1)}$$

2.3.1 Filter the voices

The data set considered is benchmark and is free of noises. We removed the trailing and leading silence from the speech samples using the following equation:

$$S_G = \begin{Bmatrix} 0, \ V_G \ is \ not \ speech \\ 1, \ V_G \ is \ speech \end{Bmatrix} \text{ .....} eq(1)$$

For leading silence the eq(1) runs from start till voice is detected. For trailing silence the equation is run from end till the voice is detected. V_G is the voice sample in Gujarati. The sliced

speech is stored in S_G.

2.3.2 Calculate Pitch

Once the signal is trimmed, we calculate pitch using fn(3).

$$P_G = Pitch(S_G) \text{ .....} fn(3)$$

For calculating pitch we used autocorrelation. The equation used is:

$$Pitch = \frac{SamplingFrequncy}{ml + index} \text{ .... } eq(2)$$

Here,
Ml is the maximum lag and index is the index of the maximum peak.

Algorithm 1 explains the implementation of autocorrelation.

Algorithm 1: Pitch
Setup
Initialize required variables
Start
Step 1. Fs, d = read a sound file get frequency and data
Step 2. min = Fs/75
Step 3. max = Fs/700
Step 4. Plot autocorrelation plot with Maximum lags = max, normalize input vector to unit length
Step 5. Y = take the first peak
Step 6. Z = take data from half of Y
Step 7. Z = take Z from min to max
Step 8. Maximum_z = take maximum peak of Z
Step 9. index = get the index of the vector where Z= Maximum_z (index of maximum peak)
Step 10. Peak = Fs/(max+index)

2.3.3 Extract MFCC features

$$G_F = MFCC(S_G) \text{ .....} fn(4)$$

Function fn(4) computes first and second derivatives of Cepstral coefficient to get the temporal dynamics of the signal. Algorithm 2 explains the implementation:

Algorithm 2:MFCC implementation
Setup:
Size_FFT = 2048
Size_hop = 10 milli seconds
SamplingRate = 16000
f_min=0
f_max=SamplingRate/2
mel_filters=10
number_dct_filters=20
Start:
Step 1. Frame the audio_signal
Frame_length =(SamplingRate * Size_hop/1000)
Number_Frames = ((audio_length-Size_FFT)/Frame_length) +1
while Number_Frames:
Frames[n]= audio_signal[n*Frame_lenght:n*Frame_length+Size_FFT]
end of while
Step 2. Window the audio_signal convert to frequency domain
window = apply hann filter
audio_win = Frames * window
Step 3. T_window = Transpose audio_win
Step 4. Fft_audio = fft(T_window)
Step 5. Fft_audio_T= transpose (Fft_audio)
Step 6. audio_power = (Fft_audio_T)^2 # calculate power of the signal

Step 7. Compute Mel Spectrum
Get filter points
Convert Frequency to Mel
Mel_max = 2595 *log10(1 + f_max/700)
Mel_min = 2595 *log10(1 + f_min/700)
mels = generate mel_filters between Mel_max and Mel_min
Step 10. Convert Mel to frequency
frequency=700 * ($exp^{(mels/2595)}$ - 1)
Step 11. Create filter bank
FilterPoints, Melfreq= (Size_FFT+1)/(SamplingRate*frequency)
Step 12. for i in range 1 to length(FilterPoints)-1 :
filters[i,i:i+1] = FilterPoints[i+1] - FilterPoints[i]
filters[i, i+1:i+2] = FilterPoints[i+2] - FilterPoints[i+1] end for
Step 13. Normalize the filter to manage the noise. As the filter width is high noise would rise with the frequency.
filters = filters* (Triangular Mel Weight / width Mel Bands)
Step 14. filtered_audio = filters * transpose(audio_powser)
Step 15. Compute Discrete Cosine Transformation (DCT) to convert Mel to Cepstral
start=1
stop = 2*mel_filters
step = 2
sample= generate an array(start, stop, step)*(pi/(2*mel_filters)
for i between 1 and number_dct_filters :
dctFilters[i]= cos(i*sample) * sqrt(2/mel_filters)
end for
Step 16. Cepstral_coeff= dctFilters* (10* log10(filtered_audio))
Step 17. Compute second derivatives of Cepstral_coeff to get temporal dynamics of the signal
for i in range -F to F :
$G_F[n]$ = W[i] * Cepstral_coeff[n+i] / |i|
end for
n is the n^{th} time frame
W is the i^{th} weight
F is number of successive frames

2.3.4 *Prepare Data*

$$G_{dataset} = \sum_{i=1}^{n} \{P_G(i), G_F(i)\} \ .....fn(5.1)$$

$$label = \sum_{i=1}^{n} \{l_i\} \ .....fn(5.2)$$

For every sound file we calculate Pitch (P_g) and MFCC (G_F) Features using Algorithm1 and 2. Every record in the dataset is assigned a numeric label (l_i) ranging from 1 to number of files (n).

2.3.5 *Design Deep Learning Model*

$$Cat_m(\{p\}) \ .....eq(3)$$

Here,
Cat_m : *Catboost regressor model*
p : *parameters*

Model creates a structure with the following parameters and values:
p ={
objective = regression #type of model
boosting = goss # gradient based one sided
sampling depth = 10 # limits maximum depth of a tree
feature fraction = 1.0 # 100% features are selected
min number data in leaf =20 # controls overfitting

number of iterations = 150 # number of boosting iterations
early stopping round = 25 # boosting will not give up till 25 rounds
helps to overcome the problem of validation
learning rate= 0.1 # improves training loss
verbosity= 1 # provides information about training and scoring
}

The parameters are selected using hit-error-trial technique.

2.3.6 *Fit and Predict*

$$Cat_m.fit(G_{dataset}, label) \ .....eq(4)$$

The model in eq(3) is trained using eq(4). We created $G_{dataset}$ and labels using fn(5.1) and fn(5.2).

$$G_{eval} = \{Pitch(G_{mixed}), MFCC(G_{mixed})\} \ .....eq(5)$$

$$P_v = Cat_m.predict(G_{eval}) \ .... \text{eq(6)}$$

To get the pitch and Features of the mixed voice data, we again applied algorithms 1 and 2 on G_{mixed}. The trained model using eq(6) predicts the voice from G_{eval}. The predicted voice is store in P_v. The input for the catboost model is the MFCC features extracted from the standalone voices and labels assigned to them. Once the model is trained mixed data is evaluated by windowing it at 15 second interval.

3 Results

The results are tabulated with the mathematical formulations used to arrive at the results.

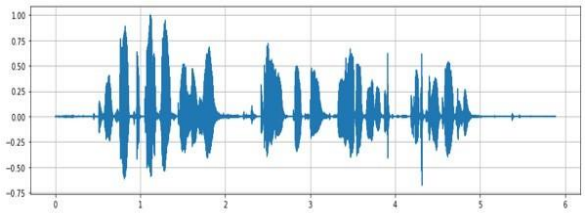

Fig. 3 Original voice of a Gujarati Male

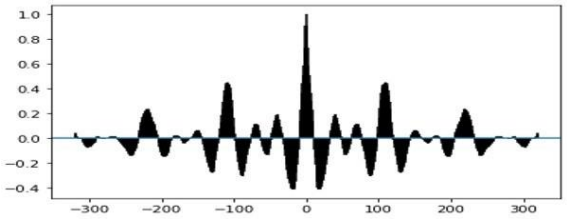

Fig. 4 Pitch estimation plot

Figure 3 is a frequency plot of an adult male voice in Gujarati. The plot is of the original voice before applying any filters. The result of algorithm 1, to calculate pitch, is plotted in Figure 4. The voice is again a male voice in Gujarati. All the plots are of adult male Gujarati voice.

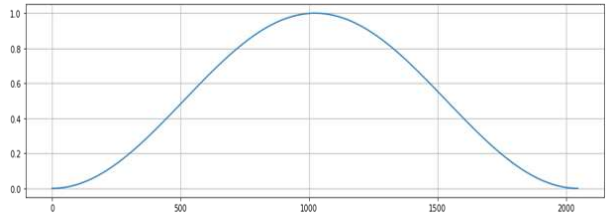

Fig. 5 Hanning window

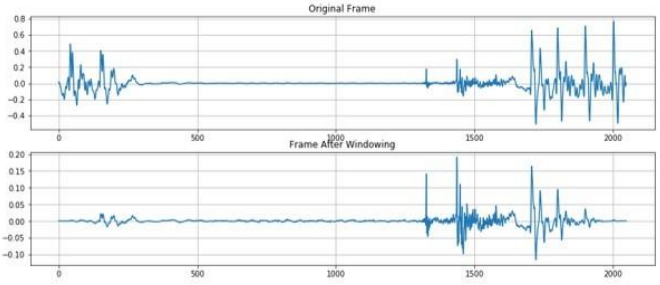

Fig. 6 Effect of Windowing

The steps in Algorithm 2 to calculate MFCC are plotted in Figure 5, 6, 7, and 8. For calculating MFCC the data is windowed as shown Fig. 5. We have used Hanning window. The plot shows the effect of windowing on the voice data as in step 2 of Algorithm 2. Steps 3-5 converts the signal into frequency domain and window the signal. Figure 6 shows the effect of windowing on the complete wave form.

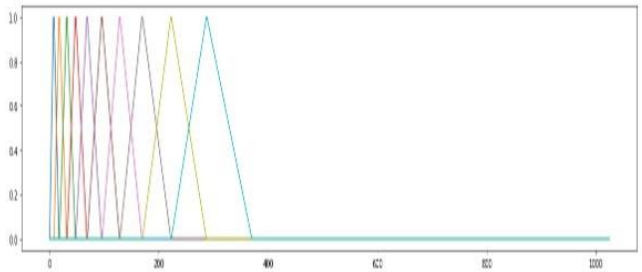

Fig. 7 Filter points

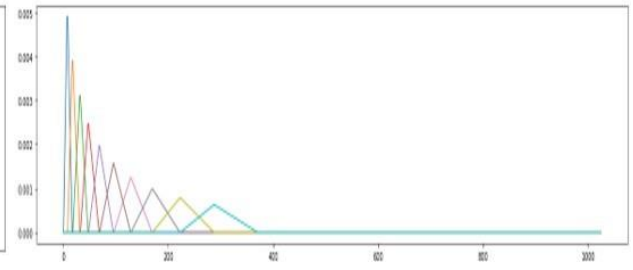

Fig. 8 Mel bands

Step 6-17 in Algorithm 2 compute the Cepstral Coefficient. First, filter points are created and then mel bands are computed. Figure 7. Figure 8 shows the plot of the filter points and filter bands.

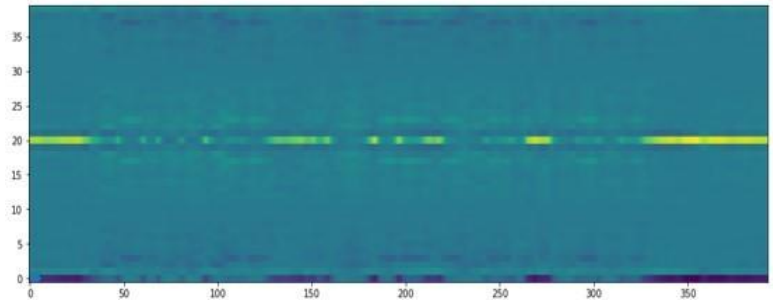

Fig. 9 Cepstral Plot

Step 17 of Algorithm 2 computes the Cepstral Feature same is plotted in Figure 9.

3.1. Mathematical Evaluation of Results

We draw qualitative comparison with other models using source-to-distortion ratio (SDR) [43]. Other measurement measures include signal-to-distortion ratio improvement (SI-SDR) [48], perceptual estimate of speech efficiency (PESQ) scores [40], scale-invariant signal-to-noise ratio (SI-SNR) [47]. SDR, SI-SDR, PESQ, SI-SNR higher values reflect better quality of separation.

SDR :

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

SI-SDR :

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

S-I SNR :

$$T_s = \frac{\langle es, S\rangle S}{\|P\|} \quad \text{.... eq(7)}$$

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

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

PESQ :

$$PESQ = 4.5 - 0.1d_s\text{-}0.0309\mathrm{d}_A \quad \text{.... eq(10)}$$

Here, S and es represent original and estimated clean source, respectively. L represents the length of the signal. e_{inter}, e_{noise}, e_{artif} represent interferences, noise and artifacts error terms, respectively. P represents the 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.

Table 1 Results of the Quality of the extracted voice

Model	SDR	PESQ	SI-SDR	SI-SNR
G-cocktail	13.32	3.15	12.89	12.72

We windowed the mixed signal at 15 seconds interval to calculate pitch and MFCC features. The features were iterated through the model to predict the voices.

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

$$\mathrm{Pr}ecision = \frac{TP}{TP + FP} \quad \text{.... eq(1)}$$

$$\mathrm{Re}call = \frac{TP}{TP + FN} \quad \text{.... eq(2)}$$

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

TP is the number of correctly detected sounds (predicted), FN is the number of voices that do not match. 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. We have tabulated the results manually but repeating the process for all the voice files and possible combinations. The results in table 2 are based on 100 voice samples.

Table 2 Evaluation of Different algorithms

Algorithm	Precision	Recall	Accuracy
SVM	0.769230769	1	0.85
KNN	0.833333333	1	0.9
XBoost	0.810810811	1	0.883333333
LightGBM	0.857142857	1	0.916666667
G-Cocktail	0.967741935	1	0.983333333

We experimented with SVM, KNN, XBoost, LightGBM, and CatBoost with the dataset. Our results proved that G-cocktail using XBoost performed better than the others under the given scenario.

There is little work done in enhancing and classification of Gujarati. We have compared or accuracy with the published papers on Gujarati. For experimental purpose, the assorted data contained Hindi, English, Tamil, and Telugu voices. The proposed model as able to recognize the Gujarati voice. Assorted voices were filtered out.

Table 3 Accuracy comparison with work on Gujarati

Work	model	Accuracy
[52]	Bootstrapping	88.71
[53]	HMM, HTK	95.71
[54]	HMM	87.23
[56]	TDNN, RNNLM	85.9
[51]	RNN-CTC	78.93
[49]	HMM, ANN	90.4
[57]	LSTM	80.89
[50]	LAS model	82.7
[55]	HMM, ANN	79.14
G-cocktail	Catboost with MFCC and pitch	98.33

Table 3 gives a comparison of different techniques used thus far for Gujarati language. The data sets are different. [52], [53], [49], and [55] used isolated Gujarati words, [54] used 25-word sentences, [56] did not limit to number of words in the sentences, [51], [57] used continuous speech of three Indian languages, and [50] used continuous speech of 9 Indian languages. The table highlights the accuracy achieved with Gujarati language.

Table 4 Accuracy comparison with different features

G-Cocktail	model	Accuracy
1	LPC and Pitch	94.23
2.	LPCC and Pitch	89.43
3.	i-vector and Pitch	86.23
4.	v-vector and Pitch	87.91
5.	MFCC and pitch	98.33

The experiment was done with other features like LPC, LPCC, i-vctor, and v-vector. The results obtained in 4 clearly reflect that MFCC performed better than rest of the features. The experiment was done for Gujarati voices only. The results of other languages with different dialects may be different.

4 Conclusion

G-cocktail is a step to address the languages used in India. Due to lesser speakers, the available dataset is small. It is not easy to train and classify a small dataset. G-cocktail can extract and identify the voices from a cocktail party like situation. The results show an accuracy of 96.2% which is more than the existing models for Indian languages. G-cocktail can be used to develop a voice bot in Gujarati. It will work even at a party. One will not have to silence everyone to give a command or speak close to the device.

Authors in future would add more languages and noises in the data to make it multilingual. Wewould like to experiment with FFT and wavelets to convert the signal to the frequency domain. The future model should be able to enhance and separate the voices from a party environment. It could help the doctors to develop hearing aids also. Voice bots would also work in the party environment. It would be able to identify the owner of the device and take the command.

5 Authors' Contribution

1. This is the first attempt to address the cocktail party proble with Gujarati language.
2. We have designed a code to remove the blanks which is represented through eq(1). A code in Matlab is written for the purpose.
3. Pitch calculation is done using algorithm 1. Equation 2 mathematically represents the calculation. The calculations are based on autocorrelation method but since this method is slow. For a conventional method time complexity is $O(N^2)$. The complexity I our method is O(N log N).
4. We have designed Algorithm 2 to calculate Temporal dynamics of the signal.
5. To avoid overfitting, we have adjusted the parameters of CatBoost.
6. We recorded few voices in studio environment to match the parameters of the available data sources.

6 Competing interests

There is no competing interest

References

1. P. Bhaskararao, “Salient phonetic features of Indian languages in speech technology,” Sadhana, vol. 36, no. 5, pp. 587–599, Oct. 2011, doi: 10.1007/s12046-011-0039-z.
2. G. S. Bhat, N. Shankar, C. K. A. Reddy, and I. M. S. Panahi, “A Real-Time Convolutional Neural Network Based Speech Enhancement for Hearing Impaired Listeners Using Smartphone,” IEEE Access, vol. 7, pp. 78421–78433, 2019, doi: 10.1109/ACCESS.2019.2922370.
3. C. Yarra, R. Aggarwal, A. Rajpal, and P. K. Ghosh, “Indic TIMIT and Indic English lexicon: A speech database of Indian speakers using TIMIT stimuli and a lexicon from their mispronunciations,” in 2019 22nd Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA), Cebu, Philippines, Oct. 2019, pp. 1–6, doi: 10.1109/O-COCOSDA46868.2019.9041230.
4. M. P. A. Jeeva, T. Nagarajan, and P. Vijayalakshmi, “Adaptive multi-band filter structure-based far-end speech enhancement,” IET signal process., vol. 14, no. 5, pp. 288–299, Jul. 2020, doi: 10.1049/iet-spr.2019.0226.
5. S. P. Panda, A. K. Nayak, and S. C. Rai, “A survey on speech synthesis techniques in Indian languages,” Multimedia Systems, vol. 26, no. 4, pp. 453–478, Aug. 2020, doi: 10.1007/s00530-020-00659-4.
6. P. Sarkar et al., “Designing prosody rule-set for converting neutral TTS speech to storytelling style speech for Indian languages: Bengali, Hindi and Telugu,” in 2014 Seventh International Conference on Contemporary Computing (IC3), Noida, Aug. 2014, pp. 473–477, doi: 10.1109/IC3.2014.6897219.
7. N. Mishra, M. Tech, U. Shrawankar, and D. V. M. Thakare, “AN OVERVIEW OF HINDI SPEECH RECOGNITION,” p. 6, 2010
8. P. P.Shrishrimal, R. R. Deshmukh, and V. B. Waghmare, “Indian Language Speech Database: A Review,” IJCA, vol. 47, no. 5, pp. 17–21, Jun. 2012, doi: 10.5120/7184-9893.
9. S. ud D. Khan, “The phonetics of contrastive phonation in Gujarati,” Journal of Phonetics, vol. 40, no. 6, pp. 780–795, Nov. 2012, doi: 10.1016/j.wocn.2012.07.001.
10. D. Wang and J. Chen, “Supervised Speech Separation Based on Deep Learning: An Overview,” IEEE/ACM Trans. Audio Speech Lang. Process., vol. 26, no. 10, pp. 1702–1726, Oct. 2018, doi: 10.1109/TASLP.2018.2842159.
11. Richard Sproat. Brahmi scripts. In Constraints on Spelling Changes: Fifth International Workshop on Writing Systems, Nijmegen, The Netherlands, 2002.
12. Richard Sproat. A formal computational analysis of indic scripts. In International Symposium on Indic Scripts: Past and Future, Tokyo, Dec. 2003.
13. N. Upadhyay and A. Karmakar, “Speech Enhancement using Spectral Subtraction-type Algorithms: A Comparison and Simulation Study,” Procedia Computer Science, vol. 54, pp. 574–584, 2015, doi: 10.1016/j.procs.2015.06.066.
14. N. Upadhyay, “An Improved Multi-band Speech Enhancement Utilizing Masking Properties of Human Hearing System,” in 2014 Fifth International Symposium on Electronic System Design, Surathkal, Mangalore, India, Dec. 2014, pp. 150–155, doi: 10.1109/ISED.2014.38.
15. J. Jo, H. Yoo and I. Park, "Energy-Efficient Floating-Point MFCC Extraction Architecture for Speech Recognition Systems," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 2, pp. 754-758, Feb. 2016, doi: 10.1109/TVLSI.2015.2413454.
16. S. Chakroborty, A. Roy and G. Saha, "Fusion of a Complementary Feature Set with MFCC for Improved Closed Set Text-Independent Speaker Identification," 2006 IEEE International Conference on Industrial Technology, Mumbai, India, 2006, pp. 387-390, doi: 10.1109/ICIT.2006.372388.
17. A. Das, S. Guha, P. K. Singh, A. Ahmadian, N. Senu, and R. Sarkar, “A Hybrid Meta-Heuristic Feature Selection Method for Identification of Indian Spoken Languages From Audio Signals,” IEEE Access, vol. 8, pp. 181432–181449, 2020, doi: 10.1109/ACCESS.2020.3028241.
18. K. Garg and G. Jain, “A comparative study of noise reduction techniques for automatic speech recognition systems,” in 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, India, Sep. 2016, pp. 2098–2103, doi: 10.1109/ICACCI.2016.7732361.
19. S. A. Alim and N. K. A. Rashid, “Some Commonly Used Speech Feature Extraction Algorithms,” From Natural to Artificial Intelligence - Algorithms and Applications, Dec. 2018, doi: 10.5772/intechopen.80419.
20. N. S. Nehe and R. S. Holambe, “DWT and LPC based feature extraction methods for isolated word recognition,” EURASIP Journal on Audio, Speech, and Music Processing, vol. 2012, no. 1, p. 7, Jan. 2012, doi: 10.1186/1687-4722-2012-7.
21. J. Hung and H. Fan, "Subband Feature Statistics Normalization Techniques Based on a Discrete

Wavelet Transform for Robust Speech Recognition," in IEEE Signal Processing Letters, vol. 16, no. 9, pp. 806-809, Sept. 2009, doi: 10.1109/LSP.2009.2024113.
22. O. Eltiraifi, E. Elbasheer and M. Nawari, "A Comparative Study of MFCC and LPCC Features For Speech Activity Detection Using Deep Belief Network," 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), Khartoum, Sudan, 2018, pp. 1-5, doi: 10.1109/ICCCEEE.2018.8515821.
23. N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language Recognition via i-vectors and Dimensionality Reduction," p. 4.
24. M. Mohammad Amini and D. Matrouf, "Data augmentation versus noise compensation for x-vector speaker recognition systems in noisy environments," 2020 28th European Signal Processing Conference (EUSIPCO), Amsterdam, Netherlands, 2021, pp. 1-5, doi: 10.23919/Eusipco47968.2020.9287690.
25. J. Wu, Y. Hua, S. Yang, H. Qin, and H. Qin, "Speech Enhancement Using Generative Adversarial Network by Distilling Knowledge from Statistical Method," Applied Sciences, vol. 9, no. 16, p. 3396, Aug. 2019, doi: 10.3390/app9163396.
26. B. Pulugundla et al., "BUT System for Low Resource Indian Language ASR," in Interspeech 2018, Sep. 2018, pp. 3182–3186, doi: 10.21437/Interspeech.2018-1302.
27. S. Gogoi and U. Bhattacharjee, "Vocal tract length normalization and sub-band spectral subtraction based robust assamese vowel recognition system," in 2017 International Conference on Computing Methodologies and Communication (ICCMC), Erode, Jul. 2017, pp. 32–35, doi: 10.1109/ICCMC.2017.8282709.
28. J. Wang, J. Zhang, K. Honda, J. Wei, and J. Dang, "Audio-visual speech recognition integrating 3D lip information obtained from the Kinect," Multimedia Systems, vol. 22, no. 3, pp. 315–323, Jun. 2016, doi: 10.1007/s00530-015-0499-9.
29. M. Varalwar and N. Patel, "CHARACTERISTICS OF INDIAN LANGUAGES," p. 6.
30. H. Sirsa and M. A. Redford, "The effects of native language on Indian English sounds and timing patterns," Journal of Phonetics, vol. 41, no. 6, pp. 393–406, Nov. 2013, doi: 10.1016/j.wocn.2013.07.004.
31. J. Singh and K. Kaur, "Speech Enhancement for Punjabi Language Using Deep Neural Network," in 2019 International Conference on Signal Processing and Communication (ICSC), NOIDA, India, Mar. 2019, pp. 202–204, doi: 10.1109/ICSC45622.2019.8938309.
32. M. G. Reddy et al., "Automatic pitch accent contour transcription for Indian languages," in 2015 International Conference on Computer, Communication and Control (IC4), Indore, India, Sep. 2015, pp. 1–6, doi: 10.1109/IC4.2015.7375669.
33. P. K. Polasi and K. Sri Rama Krishna, "Combining the evidences of temporal and spectral enhancement techniques for improving the performance of Indian language identification system in the presence of background noise," Int J Speech Technol, vol. 19, no. 1, pp. 75–85, Mar. 2016, doi: 10.1007/s10772-015-9326-0.
34. A. Patil, P. More, and M. Sasikumar, "Incorporating finer acoustic phonetic features in lexicon for Hindi language speech recognition," Journal of Information and Optimization Sciences, vol. 40, no. 8, pp. 1731–1739, Nov. 2019, doi: 10.1080/02522667.2019.1703266.
35. R. B. Parikh and D. H. Joshi, "Gujarati Speech Recognition – A Review," no. 549, p. 6, 2020.
36. S. Nath, J. Chakraborty, and P. Sarmah, "MACHINE IDENTIFICATION OF SPOKEN INDIAN LANGUAGES," p. 6, 2018.
37. H. U. Mullah, F. Pyrtuh, and L. J. Singh, "Development of an HMM-based speech synthesis system for Indian English language," in 2015 International Symposium on Advanced Computing and Communication (ISACC), Silchar, India, Sep. 2015, pp. 124–127, doi: 10.1109/ISACC.2015.7377327.
38. A. C. Morris, V. Maier, and P. Green, "From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition," p. 5.
39. N. D. Londhe, M. K. Ahirwal, and P. Lodha, "Machine learning paradigms for speech recognition of an Indian dialect," in 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, Tamilnadu, India, Apr. 2016, pp. 0780–0786, doi: 10.1109/ICCSP.2016.7754251.
40. Q. Li et al., "MSP-MFCC: Energy-Efficient MFCC Feature Extraction Method With Mixed-Signal Processing Architecture for Wearable Speech Recognition Applications," IEEE Access, vol. 8, pp. 48720–48730, 2020, doi: 10.1109/ACCESS.2020.2979799.
41. T. Lavanya, T. Nagarajan, and P. Vijayalakshmi, "Multi-Level Single-Channel Speech Enhancement Using a Unified Framework for Estimating Magnitude and Phase Spectra," IEEE/ACM Trans. Audio Speech Lang. Process., vol. 28, pp. 1315–1327, 2020, doi: 10.1109/TASLP.2020.2986877.
42. S. Kiruthiga and K. Krishnamoorthy, "Design issues in developing speech corpus for Indian languages — A survey," in 2012 International Conference on Computer Communication and Informatics, Coimbatore, India, Jan. 2012, pp. 1–4, doi: 10.1109/ICCCI.2012.6158831.

43. M. K. S. Khan and W. G. Al-Khatib, "Machine-learning based classification of speech and music," Multimedia Systems, vol. 12, no. 1, pp. 55–67, Aug. 2006, doi: 10.1007/s00530-006-0034-0.
44. G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," p. 9.
45. M. Joshi, M. Iyer, and N. Gupta, "Effect of Accent on Speech Intelligibility in Multiple Speaker Environment with Sound Spatialization," in 2010 Seventh International Conference on Information Technology: New Generations, Las Vegas, NV, USA, 2010, pp. 338–342, doi: 10.1109/ITNG.2010.11.
46. X. Hao, S. Wen, X. Su, Y. Liu, G. Gao, and X. Li, "Sub-Band Knowledge Distillation Framework for Speech Enhancement," in Interspeech 2020, Oct. 2020, pp. 2687–2691, doi: 10.21437/Interspeech.2020-1539.
47. Yang, C., Xie, L., Su, C., & Yuille, A. L. (2019). Snapshot Distillation: Teacher-Student Optimization in One Generation, in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2854–2863.
48. A. Desai Vijayendra and V. K. Thakar, "Neural Network Based Gujarati Speech Recognition for Dataset Collected by in-ear Microphone," Procedia Computer Science, vol. 93, pp. 668–675, 2016, doi: 10.1016/j.procs.2016.07.259.
49. H. N. Patel and Dr. P. V. Virparia, "A Small Vocabulary Speech Recognition for Gujarati," vol. 2, no. 1, 2011.
50. J. H. and D. B., "Speech Recognition System Architecture for Gujarati Language," Int. J. Comput. Appl., vol. 138, no. 12, pp. 28–31, 2016.
51. J. H. Tailor and D. B. Shah, "HMM-Based Lightweight Speech Recognition System for Gujarati Language," pp. 451–461, 2017.
52. H. B. Sailor, M. V. Siva Krishna, D. Chhabra, A. T. Patil, M. R. Kamble, and H. A. Patil, "DA-IICT/IIITV system for low resource speech recognition challenge 2018," Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH, vol. 2018-Septe, no. September, pp. 3187–3191, 2018.
53. H. K. Vydana, K. Gurugubelli, V. V. V. Raju, and A. K. Vuppala, "An exploration towards joint acoustic modeling for Indian languages: IIIT-H submission for Low Resource Speech Recognition Challenge for Indian languages, INTERSPEECH 2018," Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH, vol. 2018-Septe, no. September, pp. 3192–3196, 2018.
54. S. Valaki and H. Jethva, "A hybrid HMM/ANN approach for automatic Gujarati speech recognition," Proc. 2017 Int. Conf. Innov. Information, Embed. Commun. Syst. ICIIECS 2017, vol. 2018-Janua, pp. 1–5, 2018.
55. J. Billa, "ISI ASR system for the low resource speech recognition challenge for Indian languages," Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH, vol. 2018-Septe, no. September, pp. 3207–3211, 2018.
56. T. N. Sainath, R. J. Weiss, B. Li, P. Moreno, E. Weinstein, and K. Rao, "MULTILINGUAL SPEECH RECOGNITION WITH A SINGLE END-TO-END MODEL Shubham Toshniwal ∗ Toyota Technological Institute at Chicago," ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., pp. 4904–4908, 2018.
57. D. S. Pipalia Bhoomika Dave, "An Approach to Increase Word Recognition Accuracy in Gujarati Language," Int. J. Innov. Res. Comput. Commun. Eng. (An ISO Certif. Organ., vol. 3297, no. 9, pp. 6442–6450, 2007.
58. L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," arXiv:1706.09516 [cs], Jan. 2019, Accessed: Mar. 03, 2021. [Online]. Available: http://arxiv.org/abs/1706.09516.

Author biography

Monika Gupta Monika Gupta is working as assistant professor at Uttarak- hand Technical University, Dehradun, India in the department of electronics engineering. She has over 12 years of experience. Her area of interest is digital signal processing and VLSI circuit design.

Dr R K Singh Dr R.K.Singh is working as head of the department, Electronics and communication Engineering in BTKIT, Dwarahat, India. He has published more than 100 research papers in various international journals and conferences. His area of interest is VLSI circuit Design, and Digital signal processing.

Dr Sachin Singh Dr Sachin Singh is working as assistant professor in the department of Electronics and Electrical Engineering at NIT, Delhi. He did his M.Tech and P.hd from IIT Roorkee. He has published 23 research papers in various international journals and conferences. His area of interest is Digital image processing, Digital speech processing and electric motors.

Figures

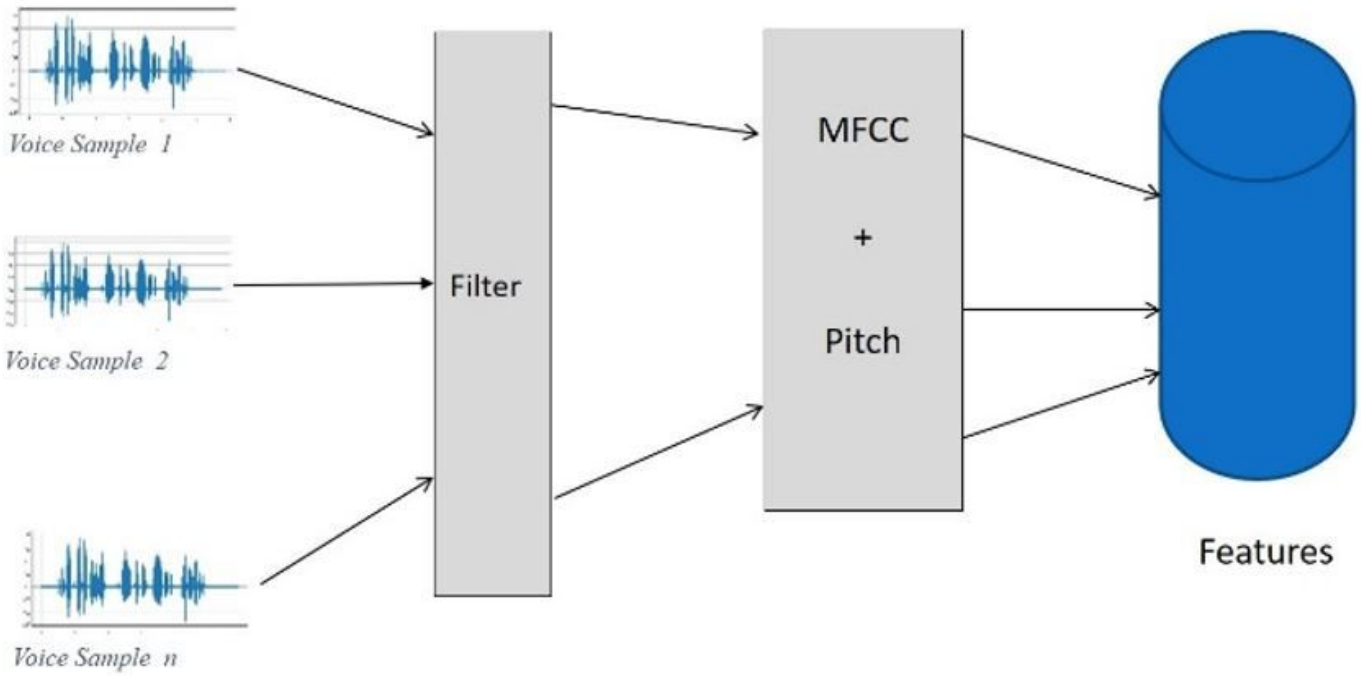

Figure 1

Feature Extraction

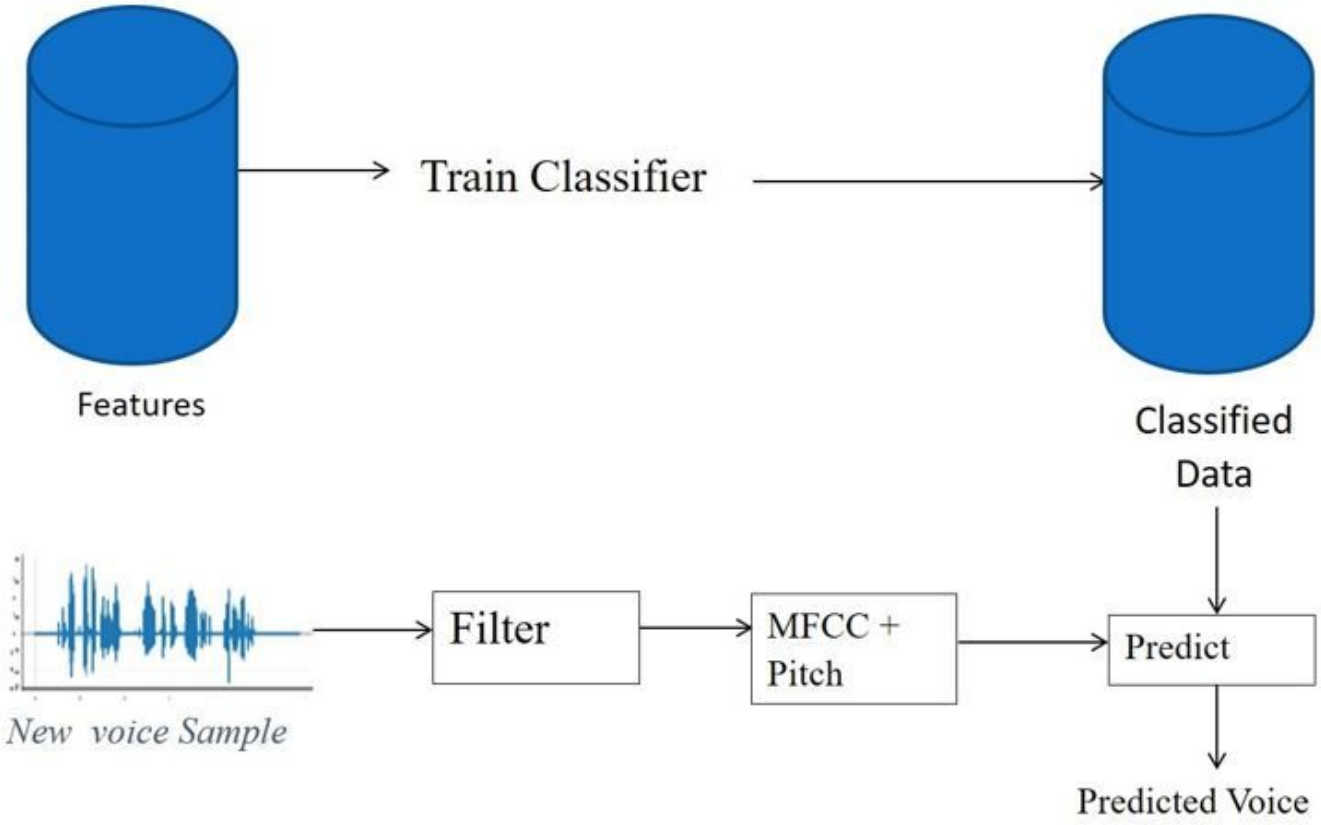

Figure 2

Classification Model

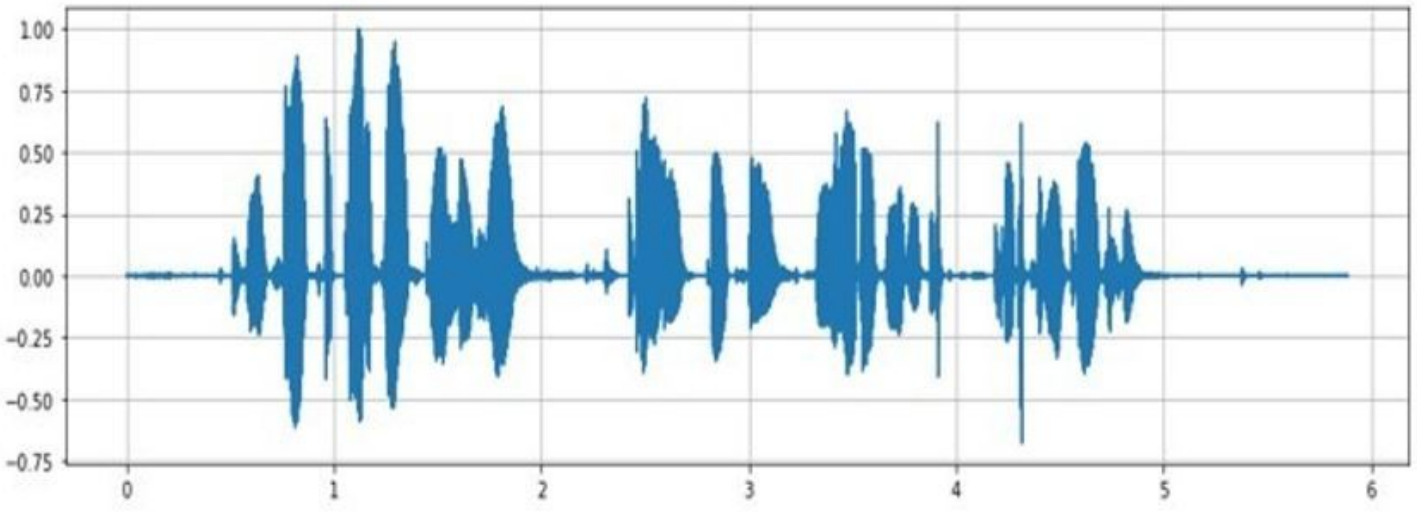

Figure 3

Original voice of a Gujarati Male

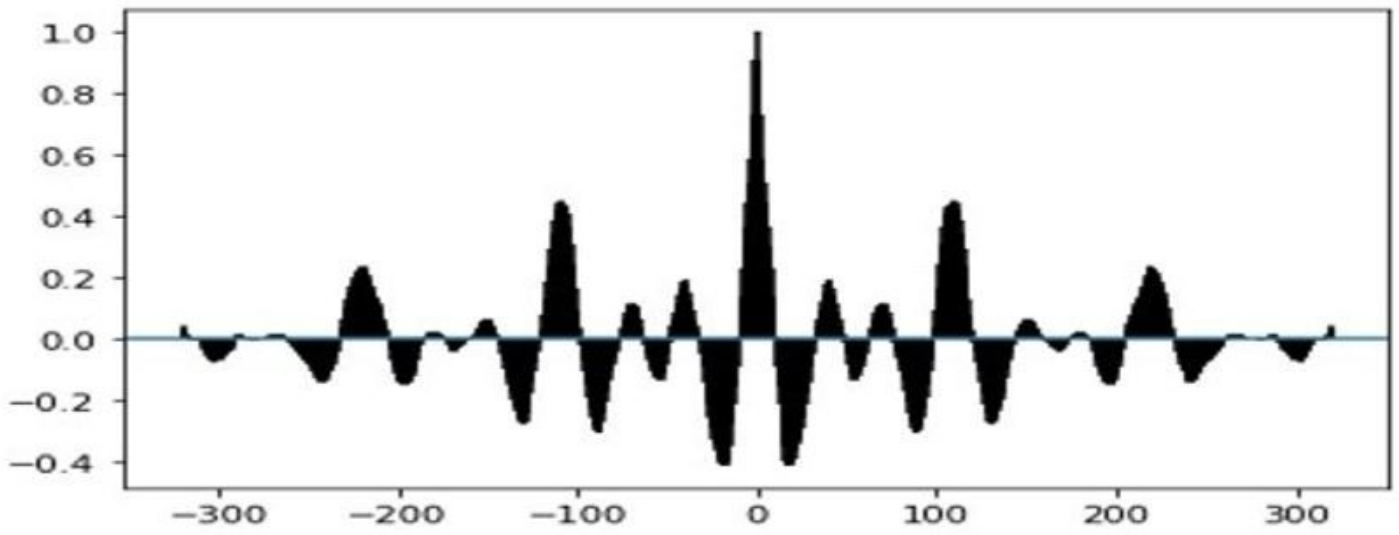

Figure 4

Pitch estimation plot

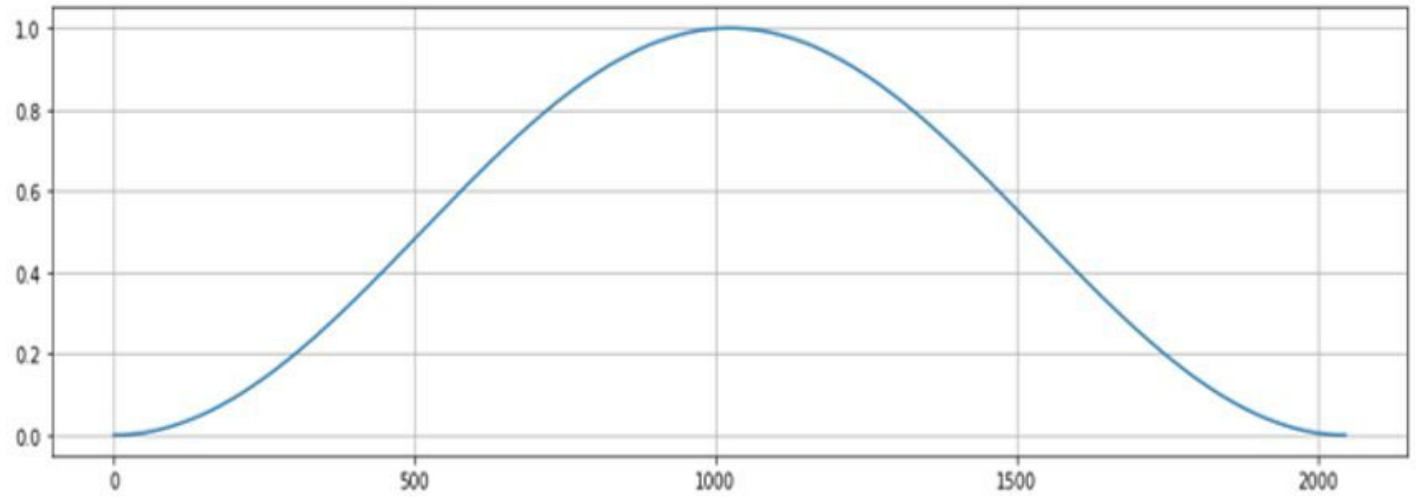

Figure 5

Hanning window

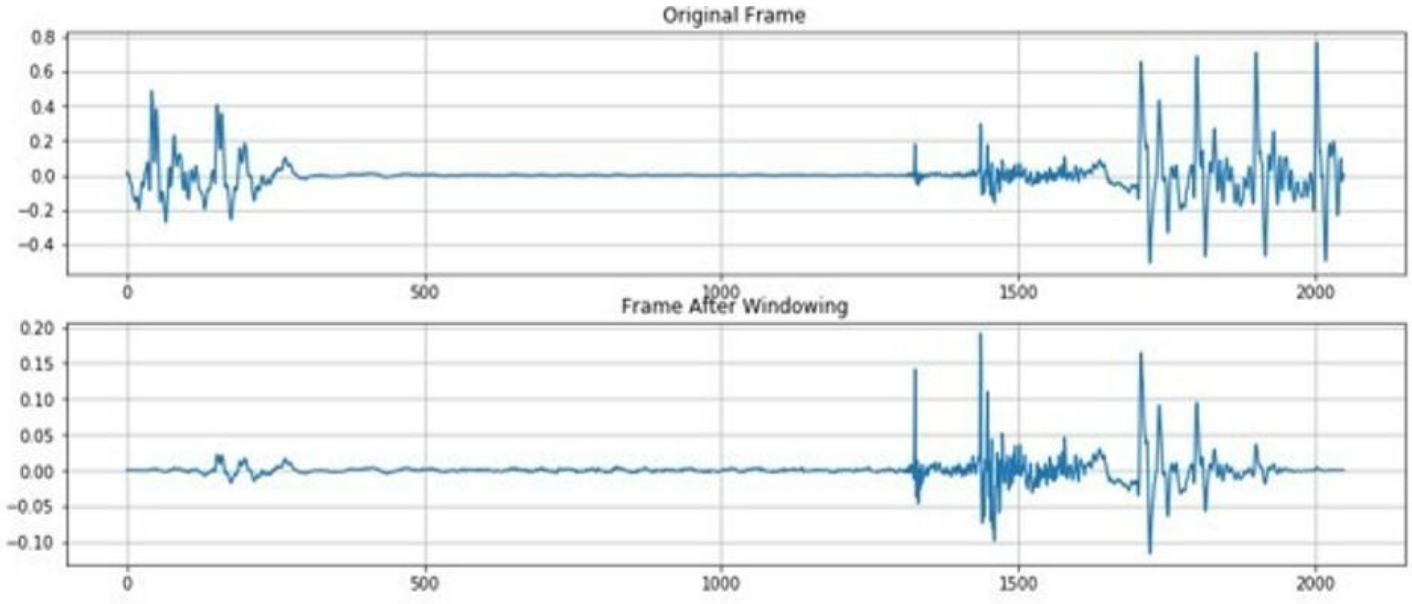

Figure 6

Effect of Windowing

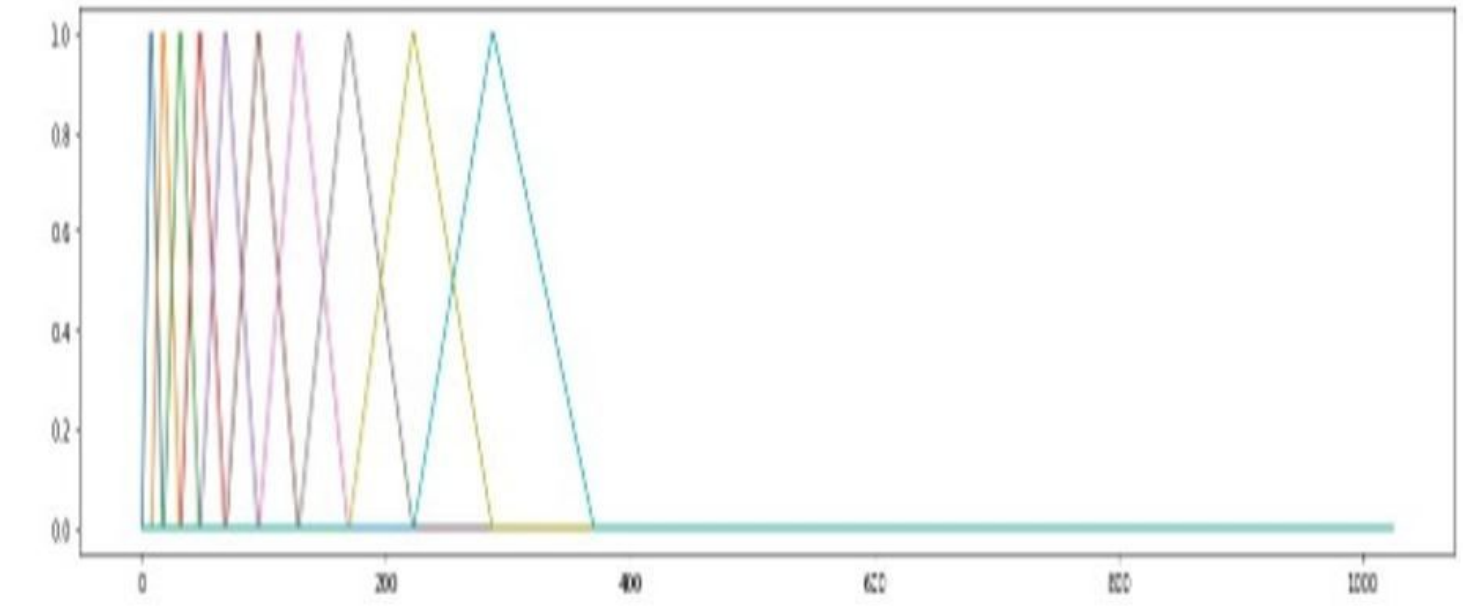

Figure 7

Filter points

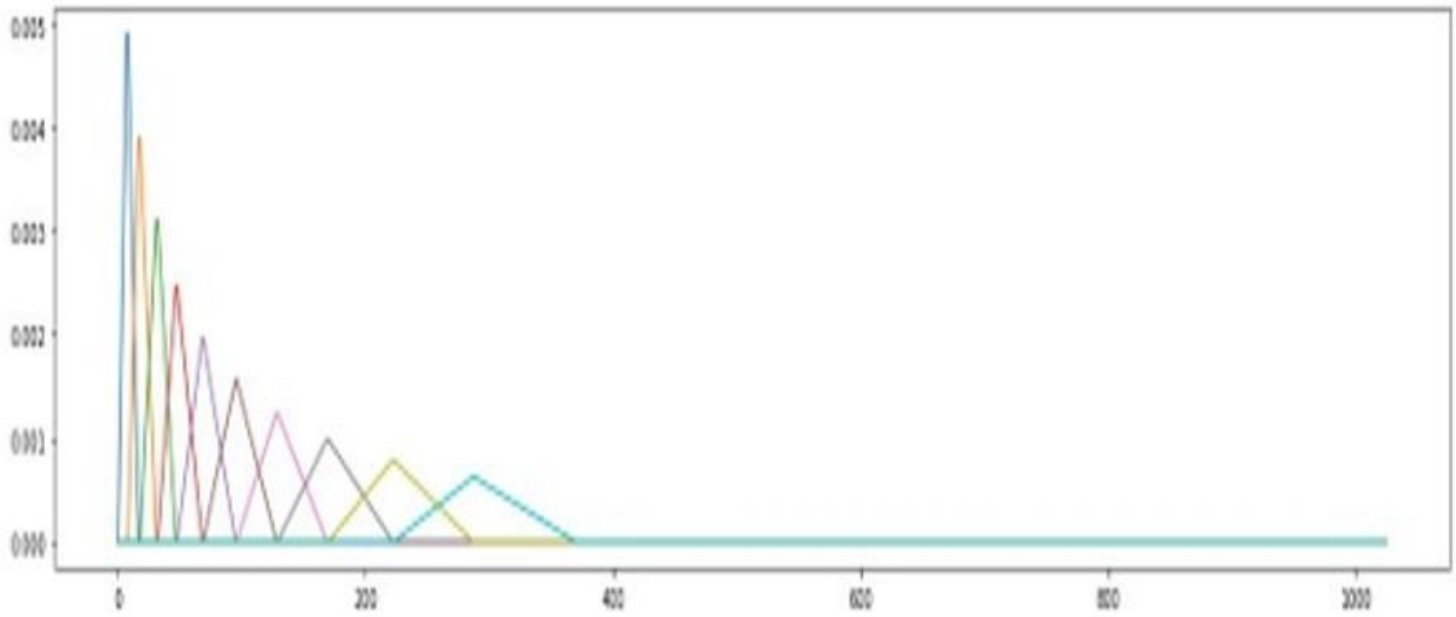

Figure 8

Mel bands

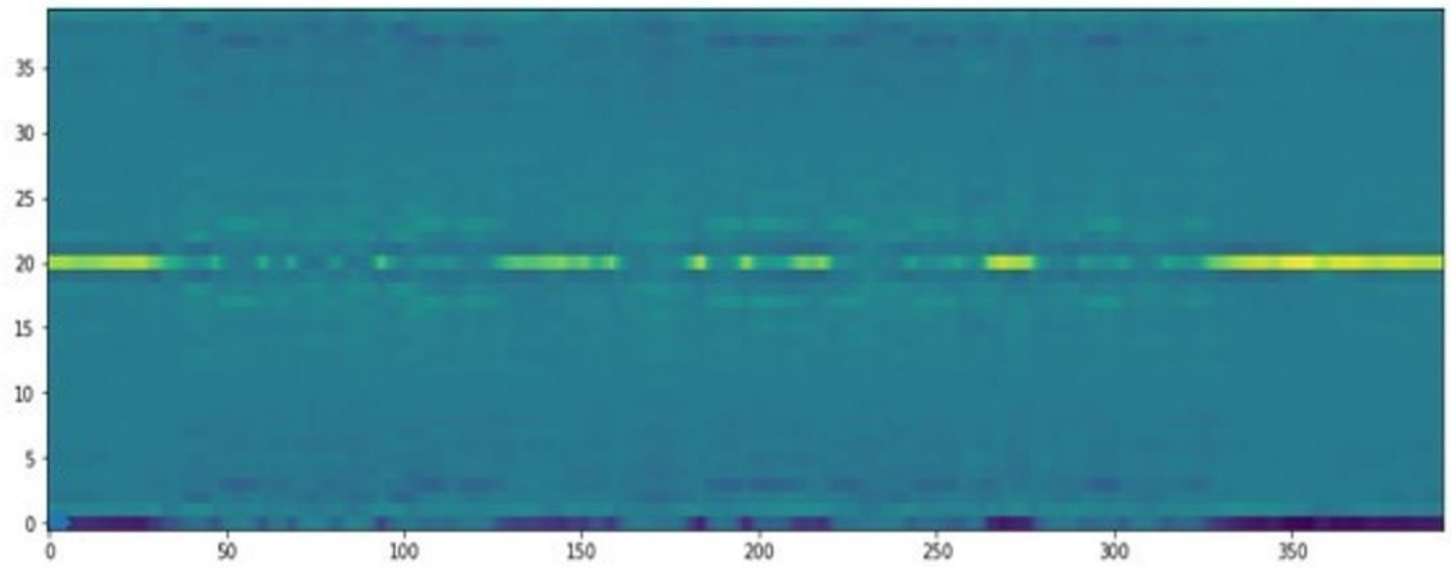

Figure 9

Cepstral Plot