ERIL: An Algorithm for Emotion Recognition from Indian Languages Using Machine Learning

Pramod Mehra1 · Parag Jain2

Accepted: 28 May 2022 / Published online: 12 September 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
It is critical for a computer to understand the speaker’s mood during a human–machine conversation. Until now, we’ve only used neutral phrases or utterances to train robots. A person’s mood affects their performance. Machines have a hard time deciphering human mood from voice because humans can make fourteen distinct sounds in a second. For a machine to comprehend human behavior, it must first comprehend the human ear’s acoustic skills. Linear Prediction Coefficients (LPC) and Mel Frequency Cepstral Coefficients (MFCC) can simulate the human auditory system. Emotion Recognition from Indian Languages (ERIL) extracts emotions like fear, anger, surprise, sadness, happiness, and neutral. ERIL first pre-processes the voice signal, extracts selective MFCC, LPC, pitch, and voice quality features, then classifies the speech using Catboost. We tested ERIL on different benchmark classifiers to choose Catboost. ERIL is a multilingual emotion classifier, it is independent of any language. We checked it on Hindi, Gujarati, Marathi, Punjabi, Bangla, Tamil, Oriya, Kannada, Assamese, and Telugu. We recorded a speech dataset of various emotions in these languages. The accuracy of distinct emotions is 95.05 percent on average. The languages have a combined average of 95.05082 percent.

Keywords MFCC · LPC · Pitch · Indian speech · Emotion recognition · Emotion classification · Catboost

1 Introduction
Knowledge exchange is closely linked to interacting parties’ reciprocal knowledge. A normal two-person discussion begins with shared identification and concludes with mutual confidence. To apply human skills to computers, it is necessary to first understand how people perceive emotions. Over the years, a lot of effort has gone into isolating the right features from speech and correctly classifying them [1–4]. Technology has brought voice

✉ Pramod Mehra
pramodemehra11@gmail.com

Parag Jain
paragjain2k1@rediffmail.com

1 Uttarakhand Technical University, Dehradun, Uttarakhand, India
2 Roorkee Institute of Technology, Roorkee, Uttarakhand, India
bots, applications, and gadgets that understand human voice commands. Google assistant, Siri and Alexa responding to users’ voice orders are household names. These applications are multilingual and work brilliantly on voice commands, but cannot understand emotions. If I say, “I am in pain”, these applications will google on the keywords and land up with a song or a movie or an article. These applications will be far more useful if they can determine the emotion and search according to the emotion of the speaker [5].

Natural Language Processing (NLP) with emotion recognition helps the organizations to plan their marketing strategies. Instead of text people put voice messages on social platforms these days. When the voice messages replace text, existing NLP techniques do not work well, as we will have to first convert the speech to text and then apply NLP [6]. Emotion recognition from the speech finds its application in medicine, virtual-reality, education, business-development, and entertainment [7]. Capturing the emotions from phone calls made for emergency services or police is equally important. Emotion extraction from threats made over phones can help the cops to make a character sketch of the person.

We need a robust algorithm to decipher the emotions from the speech. Emotions Recognition from Speech (ERS) involves four steps: pre-processing, feature extractions, classification, and recognition [1–7]. Before the retrieval of the essential characteristics in the voice, we must analyze the speech signal to eliminate noise [8]. We achieve feature extraction by transforming the speech waveform to a parametric representation for subsequent processing and analysis at a lower data rate. Quality feature extraction helps to classify the data easily [9].

To extract the vocal features like human ear, the algorithm should replicate the human acoustics. MFCC, LPC, Linear Prediction Cepstral Coefficients (LPCC), Linear Spectral Frequencies (LSF), and Perceptual Linear prediction (PLP) imitate the human hearing and speaking tract and give relevant features [10]. MFCC filters frequencies linearly at low frequencies and logarithmically at high frequencies to preserve the phonetically vital properties of the speech signal. LPC helps in determining the position of formants and crest in the spectrum [11]. LSF considers the nasal cavity and mouth structure, laying the groundwork for the linear prediction of illustration’s physiological significance [12]. LPCC uses vocal tract features to collect emotion-specific information. PLP method incorporates vital bands, intensity-to-loudness compression, and equivalent loudness pre-emphasis to extract features from the speech. It is based on the nonlinear bark scale and helps in speech recognition by removing speaker-dependent functionality [13]. The other feature extraction algorithms are Discrete Wavelets, Hybrid models and Deep learning based.

DWT, an extension of Wavelets Transform (WT), can collect information from latent signals in both the time and frequency domains [14]. Many wavelets are orthogonal, which is an excellent characteristic of signal representation in small spaces. The wavelet transform divides a signal into wavelets, which are basic functions. The WT’s main feature is that it uses a variable window to scan the frequency range, which increases temporal precision. Different frequency ranges are among the parameters. This improves the quality of speech information got at the appropriate frequency. It provides enough frequency bands for effective speech recognition, but since the input signals are finite in the range, the wavelet coefficients can have overly large variations at the edges because of discontinuities.

Deep Learning offers i-vector and x-vector features [15, 16]. These features are a fusion of MFCC, DWT and MFCC, Gammatone Frequency Cepstral Coefficients (GFCC) respectively. With the advancement of deep learning (DL), there is a trend in speech synthesis to use DL to extract features from speech signals automatically. Convolutional Neural Network (CNN) is good at extracting local features from unprocessed data. CNNs is created specifically for visual recognition activities. Their success prompted researchers to
investigate 2-D CNNs in the field of SER [17]. To understand emotions in the voice, CNN models derive high-level salient information from speech signals. Similarly, some researchers have used CNNs to build fully convolutional networks (FCNs) that can accommodate fixed input of differing sizes. In time series classification tasks based on a set input variable scale, the FCNs provided good results. FCN, though, cannot learn temporal features. A CNN and Long Short-Term Memory (LSTM) model can read the spatio-temporal features and thus widely used in SER [7]. These techniques perform well in a noisy environment but suffer from over-fitting when dealing with tiny datasets. Deep learning, Transformers, Knowledge distillation, Bidirectional Encoder Representations from Transformers (BERT) or variants of BERT have shown good results in Automatic Speech recognition, classification, speech separation, speaker-recognition, and speech-enhancement [16, 18–39]. But there is very little historical evidence of emotional recognition of Indian speech. Table 1 illustrates work done in various Indian languages.

Our study conclude that the techniques used for sentiment analysis from speech thus far, work better on a larger dataset and on a single language. There is no historical evidence of emotion extraction from multilingual speech data of Indian languages [23–25, 27, 28, 40–47]. Our research is based on 10 Indian languages: Hindi, Gujarati, Marathi, Punjabi, Bengali, Tamil, Oriya, Kannada, Assamese, and Telugu. Most of them have a small data corpus.

ERIL’s efficiency is compared to that of state-of-the-art systems. Each dataset’s comprehensive investigational implications are explained in separate parts, along with discussions and comparisons of baseline state-of-the-art approaches. In this research, we experimented with various forms of AI architectures to find the most effective method for recognizing the emotion in speech signals. We proposed this system for the SER after extensive testing, which ensures high-level output with a higher prediction rate.

2 Methodology

The objective is to extract sentiments from Multilingual Indian voices.

2.1 Experimental Set-Up

The section is split into hardware and dataset.

2.1.1 Hardware

- Processor: 7th Gen Intel Core i5, 2.5 GHz.
- Operating System: Windows 10
- Memory & Storage: 16 GB DDR4 RAM with Nvidia Graphics
- Storage: 1 TB

2.1.2 Dataset

We created a dataset of semi-natural expression. UTU Semi Natural Emotion Speech Corpus (UTU-SNESC) was developed for training and validating emotion recognition models. The corpus contains dialogues from Indian films by well-known actors and actresses (Table 2).
| Paper | Indian language | Technique | Application |
|-------|----------------|-----------|-------------|
| [23]  | Telugu         | Linear mixed effects modeling | Distinguish between Hindi and Telugu |
| [24]  | Punjabi        | Deep Neural Network          | Language enhancement in a noisy environment |
| [25]  | 11 Indian languages | Pitch contour transcription is derived using tonal labels and the corresponding timing information from the pitch contour | Automatic pitch accent contour transcription |
| [40]  | IIT Kharagpur-Multilingual Indian Language Speech Corpus having 27 Indian languages | Gaussian Mixture Model (GMM), Hidden Markov Model (HMM), and neural networks | Automatic Speech Recognition (ASR) in a noisy environment |
| [27]  | Hindi          | GMM-HMM and MFCC to remove noise | ASR |
| [41]  | Bangla, English, Hindi, Marathi, Tamil, Telugu, Assamese, Gujarati, Kannada, and Malayalam | MFCC, LPC, Hybridized Binary Bat Algorithm (BBVA) with Late Acceptance Hill-Climbing (LAHC) for feature selection. Classifier used is Random Forest | ASR |
| [28]  | Indian English | HMM based speech synthesis system | Text to speech |
| [42]  | Assamese, Tamil, and Bengali | HMM-GMM, Kaldi toolkit | ASR |
| [43]  | Punjabi, URDU, English, Mandarin, Persian, and Italian | For feature extraction—MFCC, ZCR, Harmonic to Noise Ratio (HNR), and Teager Energy Operator (TER) were calculated. For classification—SVM was used | ERS |
| [44]  | Hindi, Marathi | Feature extraction was done using Gammatone Frequency Cepstral Coefficient (GFCC) based on Discrete Cosine Transform (DCT) and High Pass Filter. For Modified SVM (MSVM) was used | ERS |
| [45]  | Marathi        | Feature extraction was done using–LPC and energy. Classification was done using–KNN | ERS |
| [46]  | Tamil          | Feature extraction was done using–MFCC. Classification was done using–SVM | ERS |
| [47]  | Malayalam      | Feature extraction was done using–MFCC, and Short Time Energy (STE). Classification was done using–ANN and SVM | ERS |
| [47]  | Malayalam      | Feature extraction was done using–Deep Belief Network (DBN) | ERS |
Total sentences are 1960. Each language has an equal number of audio clips.

2.2 Model

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

Figure 1a shows the graphical representation of the model. Figure 1b is the predictive model. The output from classification is used to predict the emotion from the new voice. The overall working of ERIL is summarized in the following steps:

*Step 1* Read each file with different sentiments and languages

*Step 2* Filter the voices. We removed blanks

*Step 3* Extract LPC and MFCC features. Calculate Pitch. Club them all as features

*Step 4* Features satisfying a threshold are retained

*Step 5* Assign labels to features as per sentiment in the voice

*Step 6* Design a Catboost model having relevant parameters

*Step 7* Using features and labels train the model

*Step 8* To predict the sentiment:

- Take a new voice for output
- Repeat steps 2 through 4
- Run the model with the new features to predict the sentiment

We used Catboost [58] to extract the categorical features and to predict the sentiments from voices. Advantages of Catboost:

1. Works faster than other deep learning algorithm as it works on GPU.
2. It handles the missing value
3. It does not require pre-processing.
4. It can extract categorical features.
5. It works on tiny dataset. We overcame over fitting problem by adjusting the parameters of Catboost.

| Table 2   | UTU-SNESC Dataset |
|-----------|-------------------|
| Emotion   | Audio clips  | Participation (%) |
| Neutral   | 340          | 17.35            |
| Fear      | 260          | 13.27            |
| Anger     | 500          | 25.51            |
| Sadness   | 180          | 9.18             |
| Happiness | 420          | 21.43            |
| Surprise  | 260          | 13.27            |
2.3 Objective Function

The objective of the paper is to design predictive the sentiments from multilingual Indian voices. The function is represented mathematically:

$$S_v = P_s\left(\text{filter}(V_I) \rightarrow f\{MFCC(V_f) + LPC(V_f) + Pitch(V_f)\} \rightarrow \text{Classify}(\forall f)\right)$$  \hspace{1cm} (1)

Here, $S_v$ = Sentiment from voice; $V_I$ = Indian voice; $V_f$ = Filtered Indian voices, $P_s$ = Predictive function, $\forall f$ = All the features.

2.3.1 Filter the Voices

The data set considered is recorded in a studio environment. There are some silent spots in the recording which we removed from using the following equation:

$$V_f = \begin{cases} 0, & V_I \text{ is not speech} \\ 1, & V_I \text{ is speech} \end{cases} \hspace{1cm} (2)$$

Using Eq. (2), the code iterates through the voice sample and removes all the zeros. The filtered voice is stored in $V_f$.

2.3.2 MFCC Feature Extraction

$$V_{MF} = MFCC(V_f) \hspace{1cm} (3)$$

The function (3) calculates the first and second derivatives of the Cepstral coefficient to obtain the signal’s spatial features. The first algorithm discusses how it works:

![Proposed Classification Model](image)

**Fig. 1** a Proposed Classification Model. b Predictive model
Algorithm 1: MFCC implementation

Setup:
Initialize required variables
Size_Of_FFT = 2048
Hop_Size = 10 ms (milliseconds)
Sample_rate = 44100 (44.1 KHz)
min_freq=0
max_freq=Sample_rate/2
no_of_mel_filters=10
DCT_filters=20
Fr=12

Start:
Step 1. Audio signal framing
Lenght_of_Frame \leftarrow \frac{\text{sample_rate} \times \text{Hop Size}}{1000}
No_Of_Frame \leftarrow \frac{(\text{audio length} - \text{Size Of FFT})}{\text{Lenght of frame}} + 1
I \leftarrow 1
while No_Of_Frame do
Frame[i] \leftarrow \text{audio_signal} [(i \times \text{Frame length}) : (i \times \text{Lenght of Frame} + \text{Size Of FFT})]
i++ end while

Step 2. Filter window’s audio signal. Convert the mel to frequency domain
Filtered_window \leftarrow \text{apply Hann function}
windowed_audio \leftarrow \text{Frame} \times \text{Filtered_window}

Step 3. T_Filtered_window \leftarrow \text{Transpose} \text{windowed_audio}

Step 4. FFT_of_audio \leftarrow \text{apply fft on} \ T\_\text{Filtered_window}

Step 5. FFT_of_audio_\text{T} \leftarrow \text{transpose} (\text{FFT of audio})

Step 6. power_of_audio \leftarrow (\text{FFT of audio}_T)^2 /* calculate signal power */

Step 7. Calculate Mel's Spectrum
Determine the filter points
Convert audio from the frequency domain to Mel
Max_Mel \leftarrow 2595 \times \log_{10} \times \left(1 + \frac{\text{max_freq}}{700}\right)
Min_Mel \leftarrow 2595 \times \log_{10} \times \left(1 + \frac{\text{min_freq}}{700}\right)
mel_spectrum \leftarrow \text{get no_of_mel_filters between Max_Mel and Min_Mel}

Step 8. Determine the filter points

Step 9. Convert audio from the frequency domain to Mel

Step 10. Mel to Frequency conversion
m\_to\_f \leftarrow 700 \times \left\{\exp \left(\frac{\text{mel spectrum}}{2595}\right) - 1\right\}

Step 11. Create filter bank
Filter\_Points, \text{Mel Frequency} \leftarrow \frac{(\text{Size Of FFT} + 1)}{(\text{Sample rate} \times m\_\text{to}\_f)}

Step 12. j \leftarrow 0
while j < \text{length (Filter\_Points)} do
\text{filter} [j, j+1] \leftarrow \text{Filter\_Points}[j+1] - \text{Filter\_Points}[j]
\text{filter} [j, j+1: j+2] \leftarrow \text{Filter\_Points}[j+2] - \text{Filter\_Points}[j+1]
end while

Step 13. To control the noise, normalize the filter. Since the filter width is large, noise will increase as the mel to frequency increased.
filter \leftarrow \text{filter} \times \frac{\text{TriangularMelWeight}}{\text{Mel Bands Width}}
Step 14. filtered_audio ← \text{filter\times transpose(audio\_power)}

Step 15. Convert Mel to Cepstral using the Discrete Cosine Transformation (DCT).

\begin{align*}
\text{Start} &\leftarrow 1 \\
\text{stop} &\leftarrow 2 \times \text{no\_of\_mel\_filters} \\
\text{step} &\leftarrow 2 \\
\text{sample} &\leftarrow \text{array(start, stop, step)} \times \left( \frac{\pi}{2 \times \text{no\_of\_mel\_filters}} \right) \\
\text{while} \ i < \text{DCT\_filters} \text{do} \\
\text{dctFilters}[i] &\leftarrow \cos(i \times \text{sample}) \times \sqrt{\frac{2}{\text{no\_of\_mel\_filters}}} \\
\text{i} &\leftarrow i + 1 \\
\text{end while}
\end{align*}

Step 16. \text{Cepstral\_coeff} ← \text{dctFilters} \times (10 \times \log_{10}(\text{filtered\_audio}))

Step 17. Compute second derivatives of \text{Cepstral\_coeff} to get temporal dynamics of the signal

\begin{align*}
\text{for} \ i \ \text{in range}\ (-\text{Fr} \ \text{to} \ \text{Fr}) \ \text{do} \\
V_{\text{MF}}[i] &\leftarrow W[i] \times \text{Cepstral\_coeff}[i + i] \\
\text{end for} \\
t &\leftarrow t^{th} \text{time frame} \\
W[i] &\leftarrow i^{th} \text{weight} \\
\text{Fr} &\leftarrow \text{successive Frames}
\end{align*}

### 2.3.3 LPC

The LPC is a linear predictive filter that uses a linear combination of previous samples to calculate the value of the next sample. LPC assumes that the glottis at the end of the tube produces a speech signal that is distinguished by its strength and frequency. LPC analyses the speech signal by estimating the formants, removing their effects from the signal, and estimating the concentration and frequency of the residual left behind. The method of extracting the formants is known as inverse filtering, and the signal that remains is known as residue.

The LPC coefficients at lag La is given by:

\begin{equation}
A(La) = \sum_{i=1}^{n} \text{audio}(i) \times \text{audio}(i - La)
\end{equation}

This is further expressed as
Equation (5) is Yule-Walker equation.

**Algorithm 2: Feature Extraction using LPC**

**Setup:**
Define and initialize required variables

**Start:**
Step 1. Frame $\rightarrow$ frame the audio  
Step 2. wind $\leftarrow$ hamming(frame)  
Step 3. Compute autocorrelation  
$A \leftarrow \text{fft}((\text{wind, power (2, exponent_of_next_high_power_of 2(2 x size(wind,1)-1)})$  
$\quad r \leftarrow \text{ifft}(A^2)$  
$(m, n) \leftarrow \text{size(wind)}$

Step 4. Compute biased correlation  
$r \leftarrow r/m$

Step 5. Lpc $\leftarrow$ LPC (r, N) //compute LPC using eq (2.1) and eq (2.2)
Here N is order
We have computed LPC to 10th order. The algorithm will return 10 main features.

### 2.3.4 Calculate Pitch

After filtering the voice, the next step is to calculate the pitch using autocorrelation. The calculations are shown using Eq. 6.

$$Pitch = \frac{\text{Sampling Frequency}}{\max_l + \text{index}}$$  \hspace{1cm} (6)

Here, $\max_l$ is the maximum lag; index is the index of the maximum peak.
2.3.5 Prepare Data

\[ V_{dataset} = \sum_{i=1}^{n} \{ V_{MF}[i], Lpc[i], Pitch[i] \} \]  (7)

\[ label = \sum_{i=1}^{n} \{ L[i] \} \]  (8)

We use Algorithms 1, 2 and 2 to measure Pitch and MFCC (VMF) Features for each sound format. A numeric mark is assigned to each record in the dataset, ranging from 1 to the number of files.

2.3.6 Design Deep Learning Model

\[ Cat_m(\{p\}) \]  (9)

Here, \( Cat_m \): Catboost regressor model, \( p \): parameters.
Model creates a structure with the following parameters and values:
p: {
    objective function = regression
    type of boosting = goss
    maximum depth of sampling = 10
    number of features selected = 100%
    minimum number of data in the leaf = 20
    number of boosting loops = 150
    minimum number of loops = 25 // solves validation issues
    rate of learning = 0.1 // minimize training loss
    verbosity = True // to maintain log
}

We increased the number of iterations to a high amount, used the overfitting detector parameters, and turned on the use best model choices to avoid overfitting or underfitting before modifying other parameters. Only the first best iterations are included in the final model. We resolved the issue of overfitting by keeping the minimum number of data in a leaf to 20.

2.3.7 Fit and Predict

\[ V_c = \text{Cat}_m \text{fit}(V_{\text{dataset}}, \text{label}) \quad (10) \]

The model in fn (9) is trained using Eq. (10). The model trains and classifies on \( V_{\text{dataset}} \) and label. The output is stored in \( V_c \).

\[ S_v = \text{Cat}_m \text{predict}(N_v) \quad (11) \]

Here, \( S_v \) is the sentiment; \( N_v \) is the new voice having sentiment.

Once the model trains and classifies the training data, the next step is to predicted sentiment from a new voice. The trained model from Eq. (10) is used to predict the sentiment using Eq. (11). We took a new voice \( (N_v) \) with random sentiment and ran the trained model over it. The predicted sentiment is stored in \( S_v \). We evaluate the accuracy based on the correct or incorrect sentiment predicted. \( N_v \) is recorded in the same environment, and we pre-processed it using Eq. (2) before passing it to the model.

3 Results

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

Figure 2 is a frequency plot of a Bollywood movie dialogue. The plot is before applying any filter. The result of algorithm 2 is plotted in Fig. 3 on the original voice.

Effect of Eq. (2) if reproduced in Fig. 4. The recorded dialogues do not have much leading of trailing spaces. There is in between silence which is removed from the original
signal. Pitch of the filtered voice is shown in Fig. 5. The evident change in the frequency of the filtered voice affects the pitch. There is no loss of voice after filtering. It only plays as if one has pressed the fast forward button.

To calculate MFCC we used Algorithm 1. The voice is windowed first and then converted to frequency domain using hann filter. Steps 6–16 in Algorithm 1 computes the filter points and then mel bands are computed as shown Figs. 6 and 7, respectively. We have considered 10 mel bands.

The cepstral coefficients are commonly referred to as static features as they only contain information from a single frame, the second derivative of MFCC is computed in step 17 of Algorithm 1. A spectrogram plot of the same is shown in Fig. 8. The second order derivative contains details such as speech acceleration, which is a critical function for emotion recognition. Figure 9 is a plot showing LPC estimate against filtered voice. LPC is calculated using algorithm 2.

### 3.1 Calculate Robustness of the Model

We calculated True Negative (TN), True Positive (TP), False Negative (FN), and False Positive (FP) to determine accuracy. The following equations are used to measure precision, recall, accuracy, and F1 score:

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{12}
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{13}
\]

\[
\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{14}
\]

TP denote the number of correctly predicted sentiment. FP is the number of sentiments detected by the model, but they are not actually sentiments. FN is the number of sentiments that do not match. TN is the number of incorrect sentiments detected; the output is a sentiment but not the actual sentiment. The tabulation is done using manual approach. The model is fed new sentiments at random, and the evaluated sentiment is registered (Figure 10).

Table 3 uses Catboost to train and predict. 80% of the data is used for training and 20% is used for testing and validation. Tables 4 and 5 highlights the reason for choosing Catboost. The results helped us to select Catboost for classification and prediction. Further tests were conducted using CNN, LSTM, Xboost, and LightGBM.

Results in Tables 3 and 4 indicates that CatBoost is a better choice with MFCC, LPC, and Pitch. CatBoost extracts its own features from the data.

The average accuracy of different emotions from Table 6 is 95.05%. The accuracy of emotions from individual languages is computed in Table 7. The average of the languages come to 95.05082%. Pre-processing, selective feature selection and fine tuning of CatBoost parameters helps the classifier to classify and predict the emotions accurately.
4 Conclusion

ERIL is a step towards addressing the Indian languages. The supplied dataset is limited due to the small number of speakers. A tiny dataset is difficult to train and classify. In a cocktail party-like environment, ERIL can extract and identify the voices. The accuracy of the results is 95.05 percent, which is very promising for Indian languages having smaller datasets. ERIL can be used to create an Indian language speech bot. Even in a noisy atmosphere, it will work. ERIL would assist native language speakers in effectively communicating with computers. It could assist businesses in comprehending the speakers’ emotions.
and providing appropriate replies. ERIL can assist the online education providers to understand the emotional connect of the students with the subject. It could help them to design better strategies and syllabus for the students.

We plan to add more languages and noises in the future. In a nosy setting, we’d like to try out different native speakers speaking at the same time. To transform the signal to the frequency domain, we’d like to try out FFT and wavelets. Even if they overlap, the future model should be able to grasp more Indian languages. It could aid corporations in developing devices that recognize patients’ emotions. To treat patients, medical science can use emotion recognition.
Fig. 6 Plot of the filter points

Fig. 7 Plot of Mel bands calculated

Fig. 8 Spectrogram plot of the second derivative of MFCC
Fig. 9  LPC plot with filtered voice

Fig. 10  Based on Table 6 a plot is drawn of contribution of each language in the accuracy
Table 3  Accuracy comparison with different feature extractions

| Features                  | Accuracy (%) |
|---------------------------|--------------|
| LPC and pitch             | 91.33        |
| LPCC and pitch            | 90.22        |
| i-vector and pitch        | 93.32        |
| v-vector and pitch        | 92           |
| MFCC, LPC and pitch       | 95           |

Table 4  Evaluation of Happiness using different algorithms

| Algorithm | Precision       | Recall | Accuracy     |
|-----------|-----------------|--------|--------------|
| CNN       | 0.810810811     | 1      | 0.883333     |
| LSTM      | 0.833333333     | 1      | 0.9          |
| XBoost    | 0.857142857     | 1      | 0.916667     |
| LightGBM  | 0.833333333     | 1      | 0.9          |
| CatBoost  | 0.909090909     | 1      | 0.95         |

Table 5  Evaluation of Sadness

| Algorithm | Precision       | Recall | Accuracy     |
|-----------|-----------------|--------|--------------|
| CNN       | 0.821918        | 1      | 0.891667     |
| LSTM      | 0.84507         | 1      | 0.908333     |
| XBoost    | 0.869565        | 1      | 0.925        |
| LightGBM  | 0.84507         | 1      | 0.908333     |
| CatBoost  | 0.923077        | 1      | 0.958333     |

Table 6  Evaluation of Different Emotions

| Emotion    | Precision       | Recall | Accuracy     |
|------------|-----------------|--------|--------------|
| Neutral    | 0.902256        | 1      | 0.945833     |
| Fear       | 0.91047         | 1      | 0.950833     |
| Anger      | 0.9113          | 1      | 0.951333     |
| Sadness    | 0.903614        | 1      | 0.946667     |
| Happiness  | 0.923077        | 1      | 0.958333     |
| Surprise   | 0.909091        | 1      | 0.95         |
**Author contributions**  As per our study, ERIL is the first algorithm to address the problem of extracting emotions from 8 Indian languages with one algorithm. Equation 1 designed by us able to remove the blanks and silence from the audio files. We wrote the code in Matlab to accomplish the task. We selected MFFCC and LPC features based on a threshold value. We ignored other features. Both the methods are based on autocorrelation method. The time complexity of a conventional method is $O(N^2)$ ERIL has a complexity of $O(N \log N)$. The parameters of Catboost are fine tuned to avoid an over-fitting issue. The dataset has some recordings of dialogues in a studio and rest are cut from the movies. A code is written to extract only the audio from the movies.

**Funding**  There is no funding of the project.

**Data availability**  Our research is in the growing stage once it is complete the data and the code will be available in the public domain as soon as our research is complete and thesis is awarded.

**Declarations**

**Conflict of interest**  The authors declare that they have no conflict of interest.

**References**

1. Shrawankar, U., & Thakare V. (2010). Feature extraction for a speech recognition system in noisy environment: A study. In: 2010 second international conference on computer engineering and applications (pp. 358–361). Bali, Indonesia.

2. Chaloupka, J., Červa, P., Silovský, J., Žďánský, J. & Nouza, J. (2012). Modification of the speech feature extraction module for the improvement of the system for automatic lectures transcription. In: Proceedings ELMAR-2012 (pp. 223–226). Zadar, Croatia.

3. Chadha, A. N., Zaveri, M. A., & Sarvaiya, J. N. (2016). Optimal feature extraction and selection techniques for speech processing: A review. In: International conference on communication and signal processing (ICCSP) (pp. 1669–1673). Melmaruvathur, India.

| Table 7  Average contribution of each language |
|---|---|---|---|
| Emotion | Precision | Recall | Accuracy |
| Hindi | 0.920245 | 1 | 0.956667 |
| Gujarati | 0.909642 | 1 | 0.950333 |
| Marathi | 0.907716 | 1 | 0.949167 |
| Punjabi | 0.908265 | 1 | 0.9495 |
| Bengali | 0.90799 | 1 | 0.94933 |
| Tamil | 0.909366 | 1 | 0.950167 |
| Oriya | 0.90854 | 1 | 0.949667 |
| Telugu | 0.908816 | 1 | 0.949833 |
| Kannada | 0.90945 | 1 | 0.950169 |
| Assamese | 0.90962 | 1 | 0.950246 |
4. Letaifa, L. B., Torres, M. I., & Justo, R. (2020). Adding dimensional features for emotion recognition on speech. In: 2020 5th international conference on advanced technologies for signal and image processing (ATSIIP) (pp. 1–6). Sousse, Tunisia.
5. Strengers, Y., & Kennedy, J. (2020). 1 Meet the smart wife. In: The smart wife: Why Siri, Alexa, and other smart home devices need a feminist reboot (pp.1–22). MIT Press.
6. Lakomkin, E., Zamani, M. A., Weber, C., Magg, S. & Wernter, S. (2019). Incorporating end-to-end speech recognition models for sentiment analysis. In: International conference on robotics and automation (ICRA) (pp. 7976–7982). Montreal, QC, Canada.
7. Sajjad, M. & Kwon, S. (2020). Clustering-Based Speech Emotion Recognition by Incorporating Learned Features and Deep BiLSTM. In: IEEE access (Vol. 8, pp. 79861–79875).
8. Garg, K. & Jain, G. (2016) A comparative study of noise reduction techniques for automatic speech recognition systems. In: International conference on advances in computing, communications and informatics (ICACCI) (pp. 2098–2103). Jaipur, India.
9. Sárosi, G., Mozsáry, M., Mihajlik, P., & Fegyó, T. (2011) Comparison of feature extraction methods for speech recognition in noise-free and in traffic noise environment. In: 6th conference on technology and human-computer dialogue (SpeD) (pp. 1–8). Brasov, Romania.
10. Ali, S. A., & Rashid, N. K. A. (2018). Some commonly used speech feature extraction algorithms. From Natural to Artificial Intelligence-Algorithms and Applications. https://doi.org/10.5772/intecopen.80419,Dec.
11. Gill, A. S. (2016). A review on feature extraction techniques for speech processing. International Journal of Engineering and Computer Science, 5(10), 18551–18556.
12. Itakura, F. (1975). Line spectrum representation of linear predictor coefficients of speech signals. The Journal of the Acoustical Society of America., 57(S1), S35–S35.
13. Aarti, B., & Kopparapu, S. K. (2018). Spoken Indian language identification: A review of features and databases. Sadhanä, 43(4), 53. https://doi.org/10.1007/s12046-018-0841-y
14. Luo, Y., et al. (2020). EEG-based emotion classification using spiking neural networks. IEEE Access, 8, 46007–46016. https://doi.org/10.1109/ACCESS.2020.2978163
15. Mohammad Amini, M., & Matrouf, D. (2021). Data augmentation versus noise compensation for x-vector speaker recognition systems in noisy environments. In: 28th European signal processing conference (EUSIPCO) (pp. 1–5). Amsterdam, Netherlands.
16. Wu, J., Hua, Y., Yang, S., Qin, H., & Qin, H. (2019). Speech enhancement using generative adversarial network by distilling knowledge from statistical method. Applied Sciences, 9(16), 3396. https://doi.org/10.3390/app9163396
17. Cummins, N., Amiriparian, S., Hagerer, G., Batliner, A., Steidl, S., & Schuller, B. W. (2017). An image-based deep spectrum feature representation for the recognition of emotional speech. In: Proceedings of the 25th ACM multimedia conference (MM) (pp. 478–484).
18. Karim, F., Majumdar, S., & Darabi, H. (2019). Insights into LSTM fully convolutional networks for time series classification. IEEE Access, 7, 67718–67725.
19. Wang, D., & Chen, J. (2018). Supervised speech separation based on deep learning: An Overview. IEEE/ACM Transactions Audio, Speech, and Language Processing, 26(10), 1702–1726. https://doi.org/10.1109/TASLP.2018.2842159,Oct.
20. Pulugundla, B. et al. (2018). BUT system for low resource Indian Language ASR. In: Interspeech (pp. 3182–3186).
21. Gogoi, S., & Bhattacharjee, U. (2017). Vocal tract length normalization and sub-band spectral subtraction based robust Assamese vowel recognition system. In: International conference on computing methodologies and communication (ICCMC) (pp. 32–35). Erode.
22. Wang, J., Zhang, J., Honda, K., Wei, J., & Dang, J. (2016). Audio-visual speech recognition integrating 3D lip information obtained from the Kinect. Multimedia Systems, 22(3), 315–323. https://doi.org/10.1007/s00530-015-0499-9,Jun.
23. Sirsa, H., & Redford, M. A. (2013). The effects of native language on Indian English sounds and timing patterns. Journal of Phonetics, 41(6), 393–406. https://doi.org/10.1016/j.wocn.2013.07.004
24. Singh, J., & Kaur, K. (2019). Speech enhancement for Punjabi Language using deep neural network. In: International conference on signal processing and communication (ICSC) (pp. 202–204). Noida, India.
25. Reddy, M. G., et al. (2015). Automatic pitch accent contour transcription for Indian languages. In: International conference on computer, communication and control (IC4) (pp. 1–6). Indore, India.
26. Polasi, P. K., Krishna, S. R., & K. (2016). Combining the evidences of temporal and spectral enhancement techniques for improving the performance of Indian language identification system in the presence of background noise. International Journal of Speech Technology, 19(1), 75–85. https://doi.org/10.1007/s10772-015-9326-0
27. Patil, A., More, P., & Sasikumar, M. (2019). Incorporating finer acoustic phonetic features in lexicon for Hindi language speech recognition. *Journal of Information and Optimization Sciences*, 40(8), 1731–1739. https://doi.org/10.1080/02522667.2019.1703266

28. Nath, S., Chakraborty, J., & Sarmah, P. (2018). *Machine identification of spoken Indian languages*. pp. 6.

29. Mullah, H. U., Pyrtuh, F., & Singh, L. J. (2015). Development of an HMM-based speech synthesis system for Indian English language. In: *2015 international symposium on advanced computing and communication (ISACC)* (pp. 124–127). Silchar, India.

30. Londhe, N. D., Ahirwal, M. K., & Lodha, P. (2016). Machine learning paradigms for speech recognition of an Indian dialect. In *International conference on communication and signal processing (ICCSPI)*, (pp. 0780–0786). Melmaruvathur, Tamilnadu, India.

31. Khan, M. K. S., & Al-Khatib, W. G. (2006). Machine-learning based classification of speech and music. *Multimedia Systems, 12*(1), 55–67. https://doi.org/10.1007/s00530-006-0034-0

32. Hao, X., Wen, S., Su, X., Liu, Y., Gao, G., & Li, X. (2020). Sub-band knowledge distillation framework for speech enhancement. In: *Interspeech* (pp. 2687–2691).

33. Yang C., Xie L., Su C., & Yuille A. L. (2019). Snapshot distillation: Teacher-student optimization in one generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2854–2863).

34. Desai Vijayendra, A., & Thakar, V. K. (2016). Neural network based Gujarati speech recognition for dataset collected by in-ear microphone. *Procedia Computer Science*, 93, 668–675. https://doi.org/10.1016/j.procs.2016.07.259

35. Billa, J. (2018). ISI ASR system for the low resource speech recognition challenge for Indian languages. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, 2018*, 3207–3211.

36. Fang, M., Zhao, H., Song, X., Wang, X., & Huang, S. (2019). Using bidirectional LSTM with BERT for Chinese punctuation prediction. In: *IEEE International Conference on Signal, Information and Data Processing (ICSIDP)* (pp. 1–5). Chongqing, China.

37. Kierszbaum, S., & Lapasset, L. (2020). Applying distilled BERT for question answering on ASRS reports. In: *New trends in civil aviation (NTCA)* (pp. 33–38). Prague, Czech Republic.

38. Luo, S. -B., Kuo, C. -C., & Chen, K. -Y. (2020). Spoken multiple-choice question answering using multi-turn audio-extractor BERT. In: *2020 Asia-Pacific signal and information processing association annual summit and conference (APSIPA ASC)* (pp. 386–392). Auckland, New Zealand.

39. Sowmya, V. & Rajeswari, A. (2020). Speech emotion recognition for Tamil language speakers (pp. 125–136).

40. Polasi, P. K., & Sri RamaKrishna, K. (2016). Combining the evidences of temporal and spectral enhancement techniques for improving the performance of Indian language identification system in the presence of background noise. *International Journal of Speech Technology, 19*(1), 75–85. https://doi.org/10.1007/s10772-015-9326-0

41. Das, A., Guha, S., Singh, P. K., Ahmadian, A., Senu, N., & Sarkar, R. (2020). A hybrid meta-heuristic feature selection method for identification of Indian spoken languages from audio signals. *IEEE Access, 8*, 79861–79875. https://doi.org/10.1109/ACCESS.2020.2990405

42. Gadhe, R. P., Babasaheb, D., Deshmukh, R. R., & Babasaheb, D. (2015). Emotion recognition from isolated Marathi speech using energy and formants. *International Journal of Computer Applications.*, 125(2), 22–24.

43. Gadhe, R. P., Babasaheb, D., Deshmukh, R. R., & Babasaheb, D. (2015). Emotion recognition from isolated Marathi speech using energy and formants. *International Journal of Computer Applications.*, 125(2), 22–24.

44. Chandran, A., Pravena, D., & Govind, D. (2017). Development of speech emotion recognition system using deep belief networks in Malayalam language. In: *International conference on advances in computing, communications and informatics (ICACCI)* (pp. 676–680). https://doi.org/10.1109/ICACCI.2017.8125919
Publisher's Note  Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Pramod Mehra is a research scholar of Uttarakhand Technical University, Dehradun, Uttarakhand, India who works as an Assistant Professor CSE Department at Central Institute of Petrochemicals Engineering & Technology (CIPET), Lucknow, Uttar Pradesh, India. He has 08 years of experience in teaching. His research aims are to recognize the emotions from speech. He has Presented related paper in National and International Conferences and has published paper in peer reviewed international journals.

Parag Jain is Director of Roorkee Institute of Technology Roorkee, Uttarakhand, India. Being Director, he got INDIA STEM SUMMIT & AWARDS’19 FOR Outstanding Initiative of the Year in Robotics BY All India Council for Robotics & Automation at IIT Delhi was also Winner of Texas Instruments Innovation Challenge India Design Contest 2015 and received an Award for Innovative Technological Research & Dedicated Principal Award in ITU GLOBAL SUMMIT & AWARD, 19 by International Technological University, SAN JOSE, USA. He has presented related paper in National and International Conferences and has published paper in peer reviewed international journals. He has five patents also.