Automatic Speech Recognition for Chhattisgarhi

Project report submitted for

5th Semester Minor Project

in

Department of Computer Science and Engineering

By,

Aman Kumar Seth (16100009)

Mayank Kumar Giri (16101032)

Himanshu Singh (16100030)

Department of Computer Science and Engineering

Dr. Shyama Prasad Mukherjee

International Institute of Information Technology, Naya Raipur

(A Joint Initiative of Govt. of Chhattisgarh and NTPC)

Email: iiitnr@iiitnr.ac.in, Tel: (0771) 2474040, Web: www.iiitnr.ac.in

CERTIFICATE

This is to certify that the project titled "Small Vocabulary Chhattisgarhi Speech Recognition" by Mayank Kumar Giri, Himanshu Singh and Aman Kumar Seth has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

_____

Dr Ankit Chaudhary

Assistant Professor

Department of Computer Science and Engineering

Dr. SPM IIIT-NR

December, 2018

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

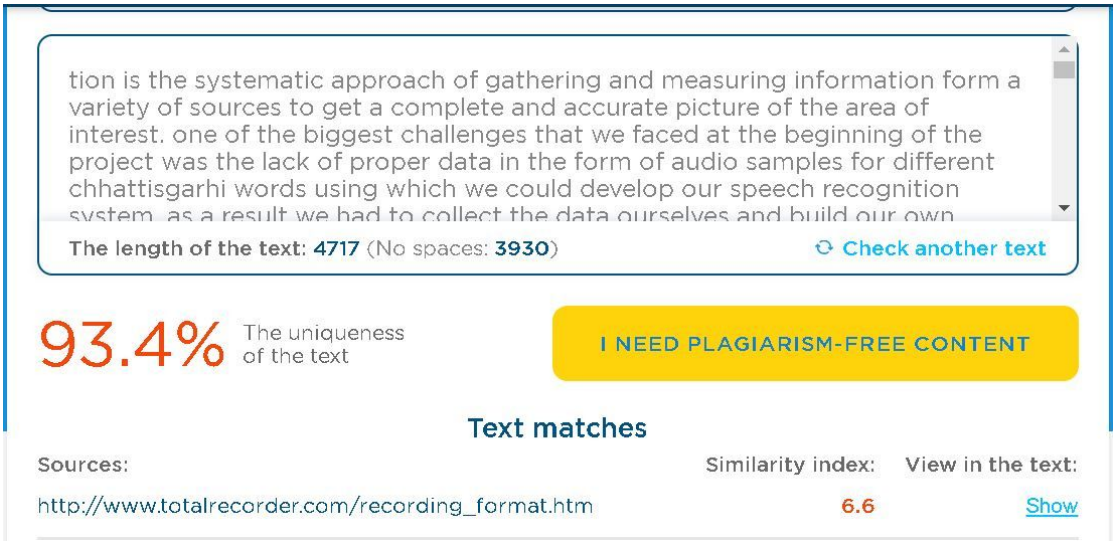
Date : _____ (Signature of Author)

Mayank Kumar Giri

Himanshu Singh

Aman Kumar Seth

PLAGIARISM REPORT



Source : https://edubirdie.com/plagiarism-checker

Approval

This project report entitled "Small Vocabulary Chhattisgarhi Speech Recognition" by Mayank Kumar Giri, Himanshu Singh and Aman Kumar Seth is approved for Vth Semester Minor Project.

_____

Dr Anurag Singh

_____

Dr Santosh Kumar

_____

Dr Ankit Chaudhary

Date: _____ Place: _____

Table of Contents

Table of Contents

ABSTRACT

In our task titled "Automatic Speech Recognition for Chhattisgarhi" we endeavor towards building and idealizing an Automatic Speech Recognition System (ASR) for Chhattisgarhi ,which is the neighborhood lingo of Chhattisgarh. In this we previously gathered the examples for the different words in our vocabulary. At first we began with 5 words. Yet, at that point stretched out it to 58 words. Every one of the accounts were taken from our phones in a uniform .wav format in order to keep the loss of highlights. Once the dataset was made, we started applying different calculations to order the different information signals into their comparing target classes. At first we connected the Dynamic TIme Warping however then changed to different systems as it was extremely moderate yet created a decent exactness.

The Other methodologies incorporated the usage of different classifiers like SVM, choice trees, arbitrary backwoods and neural systems. The principle issue with these classifiers was the little measure of information that we had. Every one of these classifiers required gigabytes of datasets in order to deliver exact outcomes. In any case, even in the wake of gathering 2300 examples the measure of our dataset just expanded to 60MB.

The vast majority of the classifiers created a precision of around 40-60 % and that is the elbow purpose of exactness that can be accomplished from the dataset in hand.The best outcomes were acquired by utilizing the irregular timberland which delivered a precision of around 65% now and again.

When the model was settled, a windows application was made utilizing the Tkinter library in python. It is the accepted library for formation of GUI in python. The application has some fundamental highlights. It can perceive hindi and english discourse in rapid utilizing the google discourse API. Aside from that it can perform chhattisgarhi discourse acknowledgment utilizing the model that we have made . The speed of acknowledgment is somewhat ease back when contrasted with the google discourse API however the exactness and accuracy was very high.

List of All Tables

List of All Figures

1. Introduction

In our project titled “Small Vocabulary Chhattisgarhi Speech Recognition” we strive towards building and perfecting an Automatic Speech Recognition System (ASR) for Chhattisgarhi ,which is the local dialect of Chhattisgarh. In this we first collected the samples for the various words in our vocabulary. Initially we started with 5 words . But then extended it to 58 words. All the recordings were taken from our cell phones in a uniform .wav format so as to prevent the loss of features. Once the dataset was created, we began applying various algorithms to classify the various input signals into their corresponding target classes. Initially we applied the Dynamic TIme Warping but then switched to other techniques as it was really slow but produced a good accuracy.
The Other approaches included the implementation of various classifiers like SVM, decision trees, random forests and neural networks .

The main problem with these classifiers was the little amount of data that we had. All these classifiers required gigabytes of datasets so as to produce accurate results. But even after collecting 2300 samples the size of our dataset only increased to 60MB.

Most of the classifiers produced an accuracy of around 40-60 % and that is the elbow point of accuracy that can be achieved from the dataset in hand.The best results were obtained by using the random forest which produced an accuracy of around 65% in some cases.

Once the model was finalized, a windows application was created using the Tkinter library in python. It is the de facto library for creation of GUI in python. The app has some basic features. It can recognize hindi and english speech in high speed using the google speech API. Apart from that it can perform chhattisgarhi speech recognition using the model that we have created . The speed of recognition is a bit slow when compared to the google speech API but the accuracy and precision was quite high.

2. Data Collection

Data collection is the systematic approach of gathering and measuring information form a variety of sources to get a complete and accurate picture of the area of interest. One of the biggest challenges that we faced at the beginning of the project was the lack of proper data in the form of audio samples for different chhattisgarhi words using which we could develop our speech recognition system. As a result we had to collect the data ourselves and build our own database consisting of neatly trimmed audio samples for around 60 chhattisgarhi words that were later used for training our machine learning models.

2.1 Data Specifications

All the audio samples were recorded in the '.wav' format at a sampling rate of 16 kHz with the audio format set to "mono" .

The reason for using the '.wav' format is that it is one of the simplest audio format available at present . It is an uncompressed format unlike Mp3 , thus the recordings are reproduced without any loss in the audio quality. This is a crucial feature as the quality of the training samples directly affects the quality of the results that are produced by the system. Moreover these files are fairly simple to process and edit.

The sampling rate is the rate at which an audio signal is sampled and digitized.In general, the higher the sampling rate, the more information preserving capacity it has. The bandwidth is a property of a signal which represents the bounds of the frequency upto which the signal retains information. We used the 16 kHz sampling rate as it contains more information than a signal sampled at 8kHz and if needed it can be downgraded to 8 kHz if there ever arises a need. The disadvantage of using a higher sampling rate is that their processing requires a longer times and thus makes the system slow and in some cases it may also result in the additional noise.

The 'mono' audio format was used as most of the libraries that perform the operations related to digital signal processing are compatible with the mono format . The stereo format is used for creating a cinematic experience by giving the perception of depth in sound. For audio classification the samples do not need this depth enhanced sound but only need a sample in which the voice of the speaker is amplified. .

2.2 Initial Approach

Initially , we selected 10 volunteers and took recordings of 20 chhattisgarhi words . The recordings were taken manually by personally approaching each volunteer . The voice samples were recorded on our cell phones in the proper specifications . An individual recording was made for each word and thus these recordings has to be trimmed further to remove the silences at the beginning and the end. This approach

turned out to be quite inefficient as the time taken for collecting just 800 samples was quite large.

2.3 The Smart Approach

We then completely automated the process of data collection . Instead of taking the recordings of individual words, the samples were taken in the form of well phrased chhattisgarhi sentences . These recordings were automatically fragmented into individual word recordings by detecting the silences in between the utterance of two consecutive words. As a result , these individual recordings were neatly trimmed and comparatively smaller in size. They did not contain any silence in the beginning or at the end. They were then automatically renamed according to a uniform nomenclature of the form '*word_speaker_srno*' and stored in a separate folder.

2.4 Database Architecture

The basic database architecture of the system consists of a main database folder which contains two sub-folders named training and testing . The training folder consists of individual folders for each of the word in our vocabulary . Each of these folders contains the audio samples collected from the 20 volunteers . The advantage of this way of storage is that the class labels are directly extracted from the names of these folders and in the future if the number of words has to be increased , one simply has to add a new folder for that word along with the audio samples and its corresponding label will automatically be added without writing any additional piece of code.The same structure has been followed for the testing folder as well.

3. Feature Extraction

Feature extraction is the process of transforming the input data or signal into a set of features which can represent the data well. Audio feature extraction are a number of Digital Signal Processing techniques using which the audio signals can be converted into numerical feature sets which can then be used for training our machine learning models. To extract the useful and important features from the audio samples, we used the librosa library. The following features were extracted for the audio samples:

- MFCC
- Mel Spectrogram
- Chroma - stft
- Tonnetz
- Spectral - Contrast

3.1 MFCCs

MFCC stands for mel-frequency cepstral coefficients. These are the most commonly used features in automatic speech recognition. MFCCs are derived from a type of cepstral representation of the audio clip . A cepstrum is the result of taking the inverse fourier transform of the logarithm of the estimated spectrum of the signal. The power cepstrum in particular finds applications in the analysis of human speech. The main difference between the MFC and cepstrum is that in MFC , equal spacing of the frequency bands takes place which approximates the human auditory system's response more closely. The MFCC function of librosa return a 1D matrix of size 40.

3.2 Mel Spectrogram

A spectrogram is the visual representation of the spectrum of frequencies of sound or other signal as they vary with time. These are also known as voicegrams, sonographs or voiceprints. It is a visual method of depicting the strength of the signal or loudness of a signal over time at various frequencies that are present in a particular waveform. When we use a nonlinear mel scale of frequency, we obtain the Mel Spectrogram. A mel scale is a scale of pitches judged by listeners to be equal in distance from one another.

3.3 Chroma STFT

The term chromagram or chroma feature closely relates to the 12 different classes of pitch . These are also referred to as pitch class profiles and a powerful tool for the analysis of music whose pitches can be meaningfully categorized.One main property of chromagram is that they capture the melodic and harmonic characteristics of music

while being robust and agile to changes in instrumentation and timbre. In this we usually calculate a chromagram from the waveform of the power spectrum.

3.4 Spectral Contrast

Octave based Spectral Contrast considers the spectral peaks, spectral valleys and their difference in each sub-band. In simple terms, it roughly represents the relative distribution of the harmonic and non-harmonic components in the spectrum. Other features like MFCC , take the average of the spectral distribution in each subband and are thus prone to lose valuable spectral information.

3.5 Tonnetz

This is also feature which detects the changes in the harmonic content of the musical audio signals . A peak in the detection function represents that a transition was made from one harmonically stable region to another. It has been observed that the algorithm can successfully detect harmonic changes such as chord boundaries in the polyphonic audio recordings.

3.6 Characteristics of voice

The three main characteristics of the human voice that can be used to uniquely identify it are as follows:

1. Loudness: It is the magnitude of the change in the air pressure. As mentioned earlier, the mel spectrogram is used to represent the loudness of an audio signal.
2. Pitch: It is the frequency that tells us the number of times a pressure pattern is repeated per unit time.The Chroma stft , Spectral contrast and tonnetz all tell us about the pitch of the audio samples.
3. Timbre: It is the general term for the distinguishable characteristics of a tone. It is a quality of sound that makes voices sound different from each other. It is mainly determined by the harmonic content of a sound and the dynamic characteristics. Tonnetz gives us an idea about the timbre of the voice.

4. Algorithms and Approaches

During the entire duration of the project a number of approaches were adopted to create a model that accurately and precisely recognized chhattisgarhi speech . Some approaches yielded better results at the cost of computation time and efficiency while the other produced faster results at the cost of accuracy.

4.1 Recognition using DTW

DTW stands for Dynamic Time Warping. In time series analysis , DTW is one of the algorithms for computing the similarity between two temporal sequences , which may or may not vary in speed. It is widely used in the processing of video, audio and graphical data . Indeed , any data that can be converted into a linear sequence can be analyzed with DTW . The most well known application of DTW is its use in automatic speech recognition to deal with speakers speaking at different speeds. Other domains where it has been widely used is speaker recognition ,signature recognition and partial shape matching.

In general , DTW is a technique that computes an optimal match between two given temporal sequences with certain restriction and rules :

- Every index from the first sequence must be matched with one or more indices from the other sequence, and vice versa
- The first index from the first sequence must be matched with the first index from the other sequence (but it does not have to be its only match)
- The last index from the first sequence must be matched with the last index from the other sequence (but it does not have to be its only match)

The optimal match is the one that satisfies all the restrictions and has the least cost. Here the cost is computed as the sum of absolute differences.

In our approach we calculated the MFCCs of all the audio samples in the training folder and stored them in a 2D square matrix of order equal to the number of samples in the training folder. The MFCCs were normalized to further improve their quality. The DTW function calculated the closest match of the input audio signal by comparing it with all the entries of the 2D MFCC matrix. It was a kind of 1-Nearest Neighbour as the input sample was classified as the the word whose training sample produced the least DTW value when compared to the MFCC of the input signal [2] .

This approach produced good results in terms of accuracy and precision but the computation time was very high as the comparisons were done in **O(n^2)**. This technique has now turned archaic and thus is not used further.

4.2 Recognition using Classification

The initial approach was used as the initial audio samples were quite small in number(800) . But when the size of our dataset increased to a significant number (2300+) we switched to the machine learning approach in which various models were trained on the feature sets and then the test samples were classified into one of the 58 target classes, one for each word.

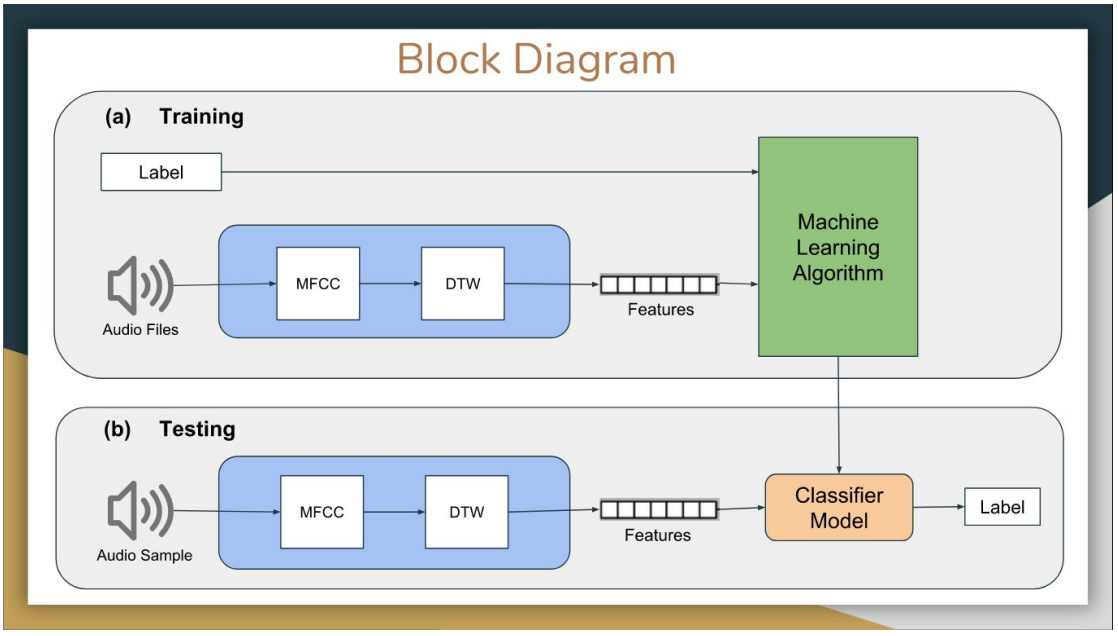


fig 1 : Block Diagram

4.2.1 Support Vector Machine (SVM)

In machine learning, support vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest

training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier

SVMs have a number of applications like text and hypertext categorization , classification of images and they are also widely used in a lot of biological applications [6]. The main disadvantage is that SVM works best on linearly separable data in which the clusters are far apart from each other and thus it did yield good results for our dataset.

4.2.2 Random Forest

Since, decision trees performed very poorly producing less than 30% accuracy (as demonstrated below), we went for an Ensemble method, i.e., Random Forest.

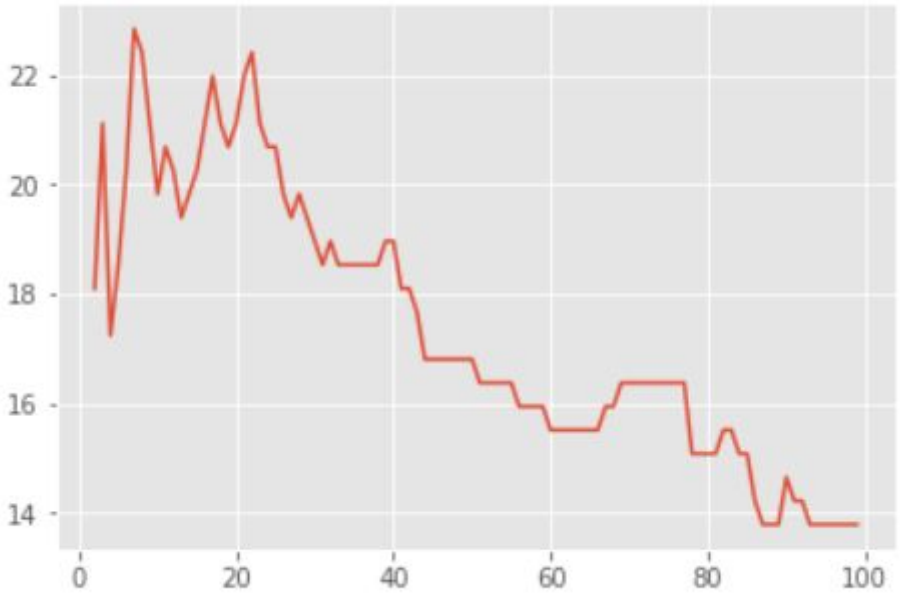


fig 2 : Decision Tree Accuracy vs Minimum samples required for a split

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

Random Forest is a supervised learning algorithm. Like you can already see from it's name, it creates a forest and makes it somehow random. The „forest" it builds, is an ensemble of Decision Trees, most of the time trained with the "bagging" method. The

general idea of the bagging method is that a combination of learning models increases the overall result.One big advantage of random forest is, that it can be used for both classification and regression problems, which form the majority of current machine learning systems.Random Forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.Therefore, in Random Forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random, by additionally using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does)[7].

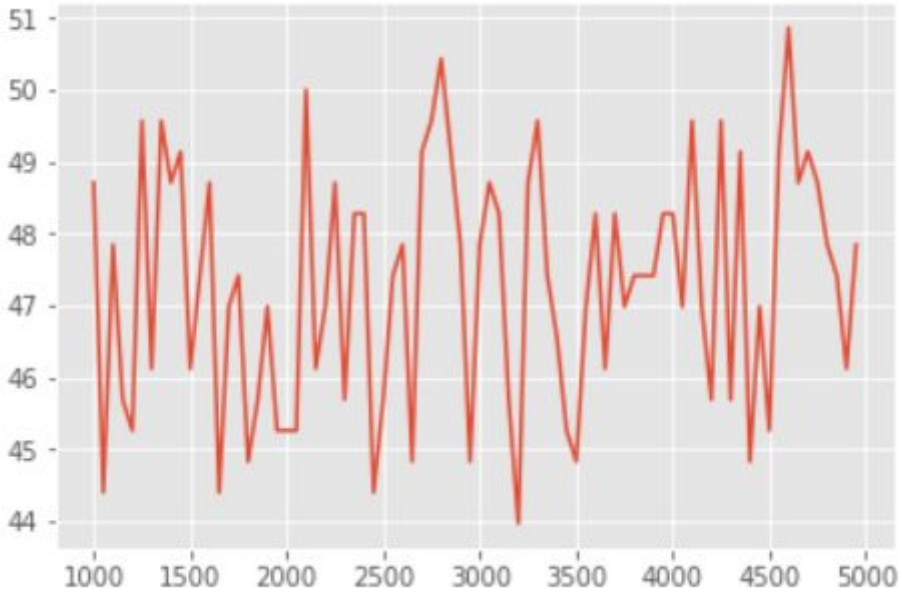


fig 3 Random Forest Accuracy vs Number of Decision trees used

In our dataset the random forest produced satisfactory results as the maximum approach that was achieved was 71%. This accuracy could have increased further by increasing the size of the training dataset.

4.2.3 Artificial Neural Network (ANN)

Artificial neural networks (ANN) or connectionist systems are computing systems vaguely inspired by the biological neural networks that constitute animal brains. The neural network itself is not an algorithm, but rather a framework for many different machine learning algorithms to work together and process complex data inputs.Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules. For example, in image recognition, they

might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the results to identify cats in other images. They do this without any prior knowledge about cats, for example, that they have fur, tails, whiskers and cat-like faces. Instead, they automatically generate identifying characteristics from the learning material that they process. An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it.

In common ANN implementations, the signal at a connection between artificial neurons is a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs. The connections between artificial neurons are called 'edges'. Artificial neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Artificial neurons may have a threshold such that the signal is only sent if the aggregate signal crosses that threshold. Typically, artificial neurons are aggregated into layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times.

The original goal of the ANN approach was to solve problems in the same way that a human brain would[8].

The basic requirement of this approach was the availability of a huge dataset (minimum 30,000 audio samples) therefore this approach did not produce good results and was thus not included in the final model. We only had 2300 samples and even after bootstrapping the minimum threshold could not be satisfied.

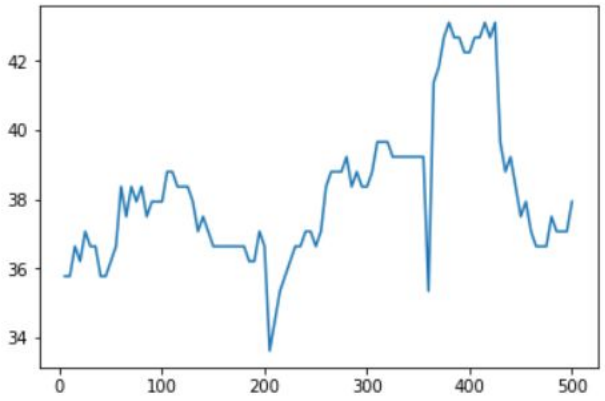


Fig 4 . ANN Accuracy vs Number of Epochs

5 . Graphical User Interface (GUI)

A number of machine learning models like decision trees , random forest , artificial neural network etc. were trained on the training samples and the results were analyzed to compute the accuracy and precision . Based on these observations the most accurate and precise model was used in the windows application that we created. The app was created using the Tkinter library in python.

Tkinter is the python binding to the TK GUI toolkit and is regarded as python's de facto standard GUI . GUI stands for graphical user interface. It is a type of user interface that allows a person to interact with our speech recognition system through icons that are graphical in nature and visual indicators like buttons , secondary notations etc. instead of a text based user interface.

Our windows app has the following features :

- **Automatic speech recognition using the google speech API**

  In this we take voice input from a microphone and it is automatically converted to text . It is available for both english and hindi text. Since it used the google cloud services consisting of a really large dataset , the accuracy is really high and the computation time is really small.

- **Chhattisgarhi speech recognition**

  This feature is based on the machine learning model that we implemented . It does testing on the test folder that is stored in the system. The input can be given in the form of a file from the test folder or in the form of a voice signal from a microphone. Additionally, all the files in the test folder can also be tested at once to compute the accuracy and precision of the model implemented.

All the trained models as stored as pickles in the app folder and thus this app can be shared on other systems as well.

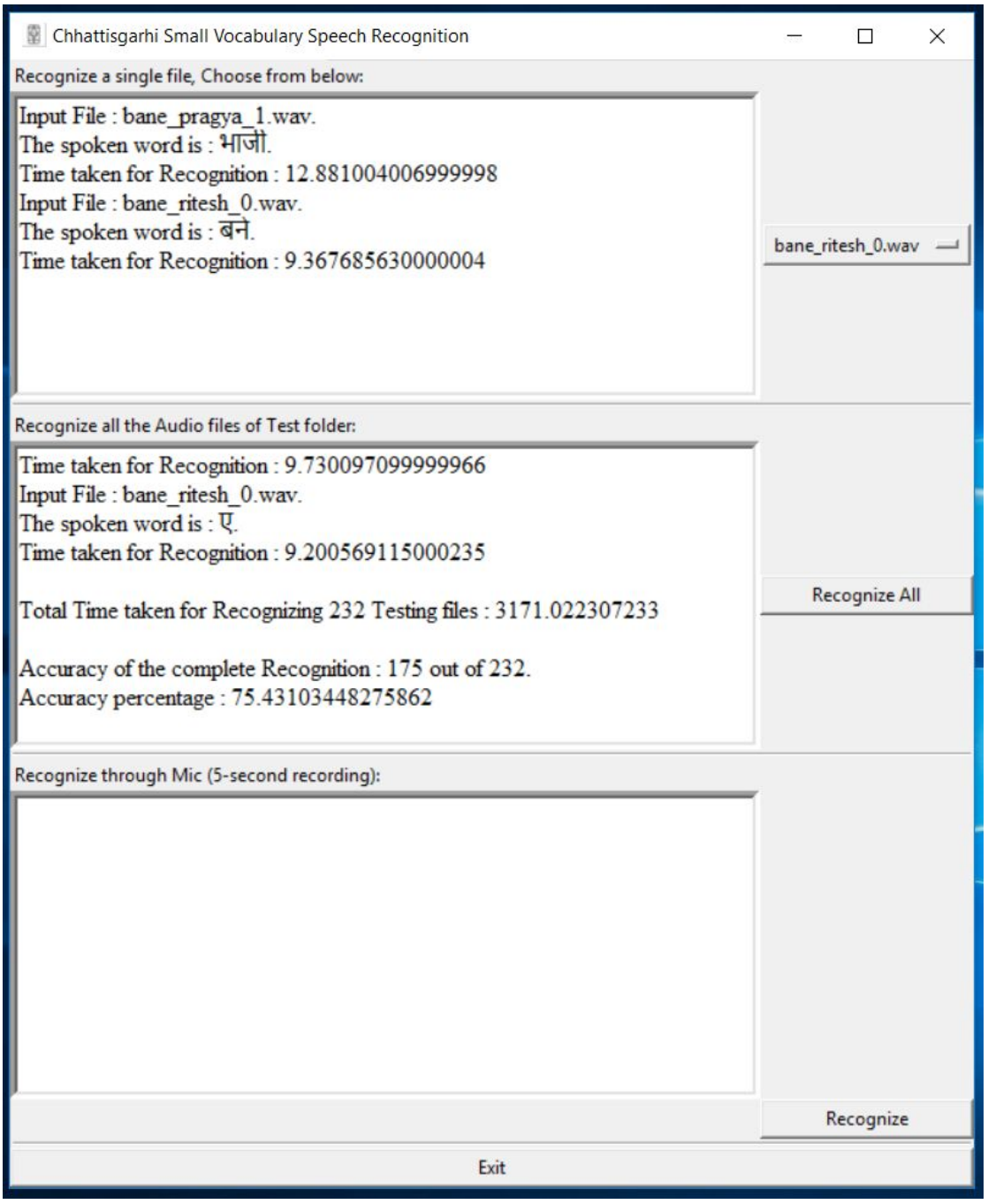


Fig 5 Windows Application

6. Results

The table gives a comparison of the accuracy obtained by the various models.

Model name	Accuracy
DTW	**75.43%**
SVM	**48%**
Decision Tree	**22%**
Random Forest	**49.56 %**
Naive bayes classifier	**26.72%**
ANN	**43.10%**

Table 1

Precisions

DTW

[80, 28, -1, 100, 66, 100, 75, 100, 100, 100, 80, 66, 100, 66, 100, 50, 100, 100, 100, 50, 27, 100, 100, 80, 40, 100, 100, 50, 100, 100, 100, 100, 100, 50, 100, 42, 100, 100, 80, 75, 100, 57, 50, 75, 100, 100, 66, 100, 66, 100, 50, 100, 100, 100, 66, 80, 100, -1]

SVM

[33.34, 40.0, 16.67, 66.67, 50.0, 50.0, 100.0, 100.0, 40.0, 20.0, 100.0, 33.34, 66.67, 0.0, 33.34, -1, 50.0, 100.0, 100.0, 50.0, 33.34, 100.0, -1, 33.34, 50.0, 57.14, 75.0, 50.0, 50.0, 100.0, 66.67, 50.0, 100.0, 33.34, 25.0, 33.34, 40.0, 50.0, 33.34, 50.0, 33.34, 33.34, 33.34, 37.5, 75.0, -1, 0.0, 37.5, 33.34, 33.34, 100.0, 66.67, 100.0, 100.0, 50.0, 100.0, 100.0, 0.0]

Decision Tree

[-1, 16.67, 0.0, 50.0, 25.0, 50.0, 0.0, 12.5, 50.0, 7.14, 100.0, 14.29, 14.29, 33.34, 25.0, 0.0, -1, 60.0, 50.0, 0.0, 0.0, 0.0, 0.0, 50.0, 50.0, 16.67, 0.0, 25.0, 16.67, 0.0, 11.11, 0.0, 0.0, 20.0, 40.0, 0.0, 0.0, 14.29, 0.0, 0.0, 16.66, 33.34, 0.0, 50.0, -1, 0.0, 11.11, 0.0, 0.0, 100.0, 25.0, 0.0, 0.0, 50.0, 44.44, 75.0, 0.0]

Naive Bayes

[25.0, 0.0, 100.0, 50.0, 0.0, -1, -1, 50.0, 75.0, 14.29, 0.0, 0.0, 33.34, 16.67, 42.86, 14.23, 75.0, 23.08, -1, 16.67, -1, 16.67, -1, -1, 100.0, 40.0, 100.0, 40.0, 33.34, 0.0, -1, 0.0, 100.0, -1, 12.5, -1, 42.86, 0.0, 0.0, 0.0, 21.43, 37.5, 0.0, 33.34, 0.0, 25.0, 37.5, 20.0, -1, 22.23, 0.0, 0.0, 0.0, 0.0, 100.0, 50.0, 100.0, 12.5]

RF

[100.0, -1, -1, 66.67, 100.0, 100.0, 100.0, 100.0, 100.0, 42.85, 50.0, 50.0, 40.0, 66.67, 26.67, 100.0, 16.67, 57.14285714285714, 100.0, 57.14, 33.34, 75.0, 66.67, 40.0, 100.0, 60.0, 66.67, 33.34, 100.0, 66.67, 50.0, 75.0, 44.45, -1, 100.0, -1, 33.34, -1, 0.0, 66.67, 37.5, 50.0, -1, 33.34, 57.14, -1, 0.0, 0.0, 0.0, 57.14, 40.0, 100.0, 20.0, 50.0, 66.67, 100.0, 100.0, 33.34]

ANN

[-1, 25.0, 0.0, 37.5, 60.0, 100.0, 33.34, 100.0, 28.57, 100.0, 40.0, 33.34, 0.0, 50.0, -1, 20.0, 50.0, 33.34, 60.0, 66.67, -1, 11.11, 100.0, 50.0, 50.0, 37.5, 40.0, 40.0, 66.67, -1, 100.0, 0.0, 10.52, -1, 50.0, -1, 16.67, 25.0, 0.0, 0.0, 66.67, 30.76, 60.0, 25.0, 25.0, -1, 0.0, 50.0, 100.0, 40.0, 100.0, 100.0, 42.85, 100.0, 0.0, 60.0, 80.0, -1]

REFERENCES

Research Papers

1. Chadawan Ittichaichareo,Siwat Suksri and Thaweesak Yingthawornsuk, “Speech Recognition using MFCC”
2. Bhadragiri Jagan Mohan, Ramesh Babu , “Speech Recognition using MFCC and DTW”
3. Fang Qiao , Jahanzeb Sherwani , "Small-Vocabulary Speech Recognition for Resource-Scarce Languages"
4. D. Bansal, N. Nair, R. Singh, and B. Raj. “A joint decoding algorithm for multiple-example-based addition of words to a pronunciation lexicon.” InProc. ICASSP, 2009.
5. C. Cortes and V.N. Vapnik , “Support vector networks”,Machine Learning, vol.20, pp. 1-25, 1995.

URL

6. https://en.wikipedia.org/wiki/Support_vector_machine
7. https://en.wikipedia.org/wiki/Random_forest
8. https://en.wikipedia.org/wiki/Artificial_neural_network
9. http://aqibsaeed.github.io/2016-09-03-urban-sound-classification-part-1/