

Visual Intelligence in Conversational Solutions for Voice Intelligence Security System (VOISS)

Dr. Ilayaraja N, Mirrudubashini S, Logesh Kumar S, Sitaraman Ramachandrala

{nir.mca@psgtech.ac.in1, mirrudubashini.sakthi@gmail.com2, logeshkumarsuresh@gmail.com3, sitaram.rnv@gmail.com4}

Assistant Professor, Department of Computer Applications, PSG College of Technology, Coimbatore, India1, PG Student, Department of Computer Applications, PSG College of Technology, Coimbatore, India2, PG Student, Department of Computer Applications, PSG College of Technology, Coimbatore, India3, Senior Director, Data Science, Bangaluru4

Abstract: Authenticating human beings accurately by a voice Intelligence in Conversational Solutions. VISS is used to improve accuracy level of speech recognition. As people become increasingly comfortable with biometrics, voice authentication is finding wider application across industries, including healthcare, banking, and education. The voice biometrics market is set to grow at an explosive CAGR of 19.4% between 2017 and 2021. The human vocal has been extracted from the video and modelling with the Audio features. The first step of the authentication process is to detect and extract voice features of corresponding voice frames. Finally both the features of voice and video of corresponding lip-movements will be aligned and jointly model to characterize people and authenticate. Speech recognition enables a machining system to convert incoming speech signals into commands by identifying and comprehending them. The primary goal of speech recognition is to improve language communication between humans and machines, making it an excellent human-machine interface technology. Speech recognition technology development encompasses all of the fundamental principles, methods, and classifications of this technology.

Keywords: Voice Recognition Algorithm, Mel-Frequency Cepstral Coefficients (MFCC), Sampling Frequency, Vector Quantization, K-means Clustering, Audio data, Features Extractions

1 Introduction

Voice Speech Recognition is an area that has high potential in solving the challenges in speech processing. It has received more attention in the last decade for its potential use in applications such as Human-Computer Interaction (HCI), Audio-Visual Speech Recognition, sign language recognition, video surveillance etc. Voice Intelligence security systems will provide the high level accuracy of speech recognition with visual features.

Purpose of the Project

The aim of this project is to design and develop an efficient system that classifies the high accuracy level of speech recognition from the visual data. The system uses the custom video dataset with proper lighting and is recorded without background noise. This dataset will be pre-processed to grab the sequence of voice datasets and extract the audio points for the

corresponding voice features. These audio points are used to get the pitch of the words and are used to classify the audio vector points. These vector points are used for vector quantization and codebook generation. The symbols or codes in the codebook have been considered for words spoken by the speaker and the accuracy level has been determined.

Scope of the Project

The scope of this project is to identify spoken words from visual data only without the corresponding acoustic signals. It is useful in situations in which conventional audio processing is ineffective like very noisy environments or impossible like the unavailability of audio signals. It is used to improve the accuracy level of Audio data and to extract the information via voice features. Its main aim is to recognize spoken words based on voice or information. The process starts by extracting the sequence of frequency from the voice dataset. Based on the voice features the vectors will be detected, integrated, and tested with video features. As People become increasingly comfortable with biometrics, Voice authentication is finding wider applications across the industries, including healthcare, banking, and education.

Limitations

The limitation of this project is that it uses only the audio dataset to predict the spoken word by the person. The voice features should be recorded without background noise (recorded in a silent place) and it might result in better accuracy but when we record with background noise, the voice of the person and the words he/she has spoken were not able to get the accuracy level. And also by processing the voice features along with video features might produce better results.

2. Literature Survey

The literature survey provides an overview of current knowledge, allowing us to identify relevant theories, methods, and gaps in existing research that are well studied in [1], [2], [3], [9], [10].

2.1 Existing Model

BhadragiriJagan Mohan and Ramesh Babu N [14] have recognized the voice speech using Mel- Frequency Cepstral Coefficient(MFCC) and Dynamic Time Warping (DTW). Speech recognition has a wide range of applications in security systems, healthcare, telephony, military, and handicapped equipment. Speech is a constantly changing signal. As a result, a suitable digital processing algorithm must be chosen for the automatic speech recognition system. To make decisions for recognition, the features are analyzed. This paper describes the implementation of a speech recognition system in the MATLAB environment. Mel-Frequency Cepstral Coefficients (MFCC) [7], [13], [14] and Dynamic Time Warping (DTW) [14] are feature extraction and pattern matching algorithms, respectively. The results are obtained through a one-time training phase and continuous testing phases.

2.2 Dynamic Time Warping

Speech recognition algorithms are broadly classified as speaker-dependent or speaker-independent [1]. The speaker-dependent system aims to create a camera to capture individuals' unique voiceprints [1], [10], [11]. The speaker-independent system identifies the word uttered by the speaker. It is further subdivided into isolated word detection and continuous speech recognition. [1] Single words separated by pauses are used for isolated word detection. Because the system does not need to learn a fluidic sequence of dictionary words, this is a simpler method than continuous speech recognition. In security systems, continuous speech recognition can be used to validate the user's password. The user's words are processed by the speech recognition system, which generates features based on them. When a user speaks, it is routed to the speech engine, which processes it and converts it to the digital domain. MFCC is used to extract features from digitized speech samples. Once the desired number of features has been obtained, they can be routed through the feature matching stage, where DTW is used to compare saved templates and recorded speech. The entire system is written in MATLAB, and the speech samples are recorded using a Windows sound card [1], [2], [3], [4], [5] and [6]. The model was evaluated using accuracy, sensitivity, and specificity as the metrics. Using Probability distribution, 98.5% of accuracy was achieved and 85% of Voice recognition of a person was identified using MFCC and DTW [14].

3 Project Description

The model perspective and functions of the system which also provide the general constraints and assumptions and dependencies are discussed in this chapter.

3.1 Model Perspective

This project is intended for Voice Intelligence Security System (VOISS) and the end deliverable of this project is a Voice Speech Recognition Model. This model serves as a major interface for all the functionalities required. This System consists of seven modules, namely

- Audio Frame Extraction
- Mel-frequency Cepstral Coefficient
- K means Clustering
- Vector Quantization (Dimensionality reduction)
- Probability Distribution
- Code Book
- Integrating with Video Model

Initially, the voice speech is recorded for every speaker and it is stored as Mp3 files (wav format). The files are grabbed frame by frame and applied as input to the voice detection module. The voice frame for each person is segmented from the entire audio file and applied for extracting MFCC points [7], [9]. Here, audio features for each audio frame is detected and using K means clustering the points for the audio features are extracted and formed the code Book /Symbol book for corresponding audio file of a person. Vector points have been determined based on the audio points extracted from the symbol book. The dimensionalities of the features are reduced using Vector Quantization. Finally based on the points, the probability distribution has been applied and the accuracy of the person has been found using the test dataset and identifying the spoken word.

3.2 GENERAL CONSTRAINTS

- Voice should be recorded with audio high clarity and with no background noise.
- This Intelligence Model is a desktop based Web Application.

3.3 ASSUMPTIONS AND DEPENDENCIES

The system has been developed considering the following assumptions and dependencies.

- The user must have camera and microphone to record the video
- The user must have a Web Browser.
- The web application requires a stable network connection.

4. Extraction Of Audio Features

The Audio Features extraction [9] and [12] done using the custom dataset creation and the process of extracting are discussed in this section.

4.1 Dataset Creation

The Dataset for our project is a customized dataset in which we have created our own video files and based on this, have extracted the audio features for the corresponding video file. The format for the video file is .mp4 format. We have created two video files for a person, one for training the dataset and the other for testing the dataset. The duration of the training dataset is 60 secs (1 min) and for testing, the dataset is 10 secs, based on these two video files of a person we have extracted the corresponding train audio files and test audio files. Figure 1 contains the custom Dataset in which it contains the Train Dataset and Test Dataset for a person in a separate video file.

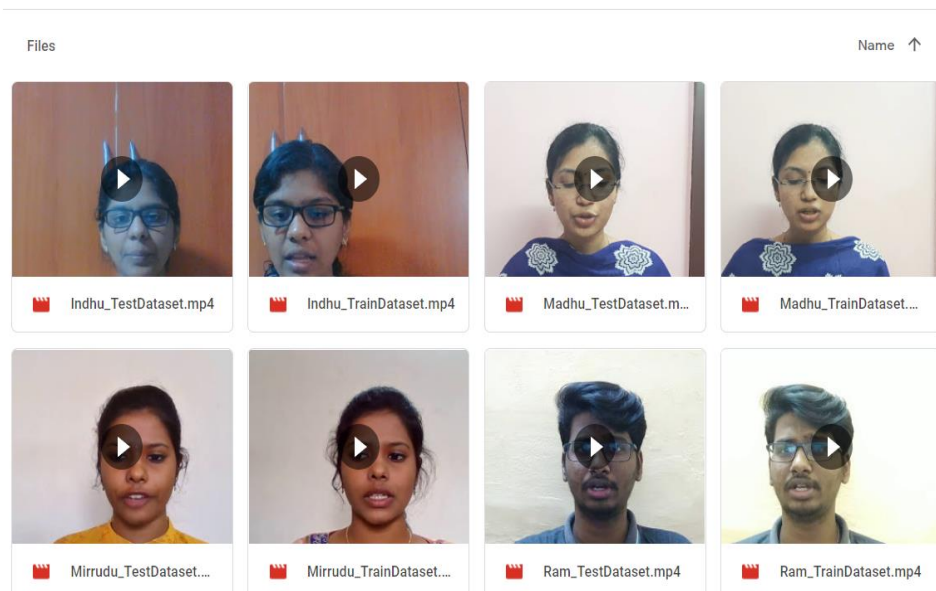


Figure 1 Custom Dataset

4.2 AUDIO FEATURES EXTRACTION

From the Video File, the corresponding audio files have been extracted using the moviepy library in python. The file format of the audio file is .wav format. Figure 2 contains the audio dataset that has been extracted with 60 sec (1-min). Figure 3 contains the audio test dataset that has been extracted within 10 sec.

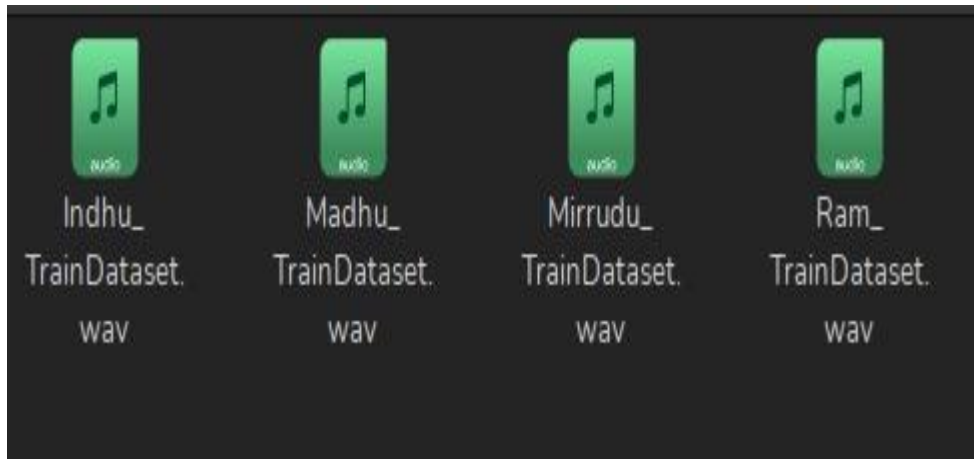


Figure 2 Train Audio Dataset



Figure 3 Test Audio Dataset

Working of Moviepy Library

Moviepy is a Python Library which it helps to read and write all the most common audio and video format of any type (mp3, mp4, GIFs, etc). First step is to install the moviepy library, and next step is to define which video files should be extracted to get the audio files, based on the video files the audio files has been extracted in the any one of the format (mp3, wav, etc).

4.3 Audio Processing

The audio processing can be done in three ways they are:

- Time Domain Features
- Frequency Domain Features
- Time - Frequency Domain Features

4.3.1 Time Domain Features

In the Time domain Features they have corresponding to the time features in the x- axis. Figure 4 shows the temporal features (time domain features) that are simple to extract and have an easy physical interpretation, such as signal energy, zero crossing rate, maximum amplitude, minimum energy, and so on. These characteristics can be used to identify the notes, pitch, rhythm, and melody.

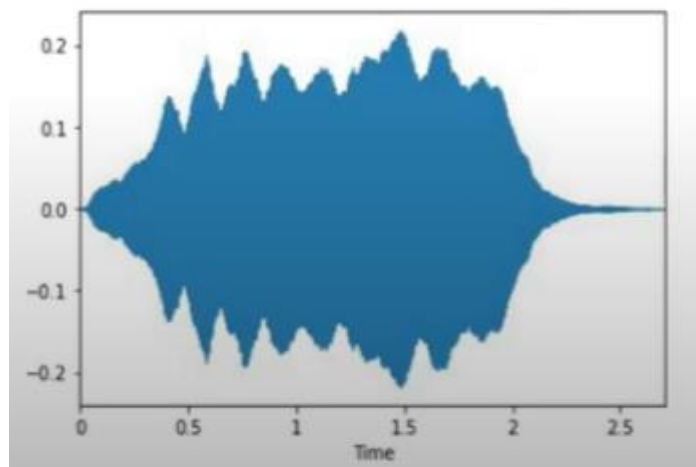


Figure 4 Time Domain Features

4.3.2 Frequency Domain Features

In the Frequency Domain Frequency they have the corresponding frequency domain in the x -axis.

Figure 5 Image enhancement in the frequency domain is straightforward. To produce the enhanced image, we simply compute the Fourier transform of the image to be enhanced, multiply the result by a filter (rather than convolving in the spatial domain), and take the inverse transform.

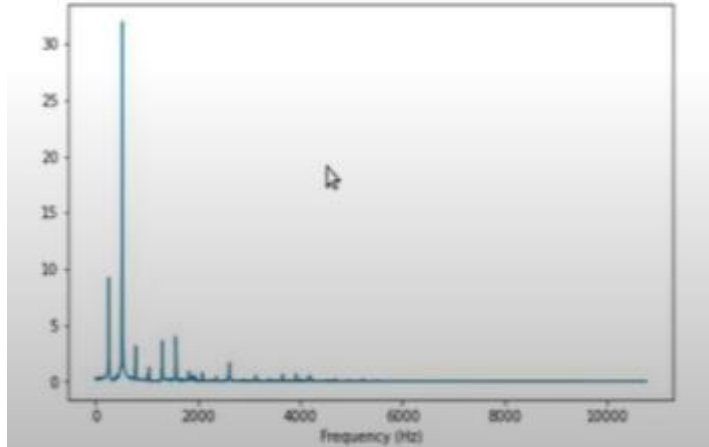


Figure 5 Frequency Domain Features

4.3.3 Time - Frequency Domain Features

The time - frequency domain features is the combination of both time domain features and the frequency domain features. **Figure 6** the frequency domain is in the y-axis and the unit is Hz and the time domain is in the X- axis and the unit is sec.

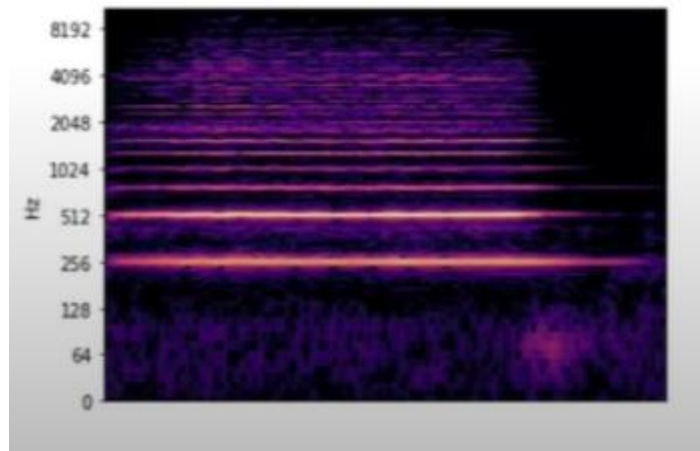


Figure 6 Time-Frequency Domain Features

5. Point Extraction Of Audio Features

The Audio Features extracting the points are done using sampling frequency and the overlapping of the audio frames are discussed in this chapter.

5.1 Sampling Frequency

The number of samples per second (or other unit) taken from a continuous signal to produce a discrete or digital signal is referred to as the sampling rate or sampling frequency. Frequencies are calculated in hertz (Hz), or cycles per second, for time-domain signals such as sound waveforms (and other audio-visual content types). The Nyquist–Shannon sampling theorem (also known as the Nyquist principle) states that perfect sampling results in perfect sampling.

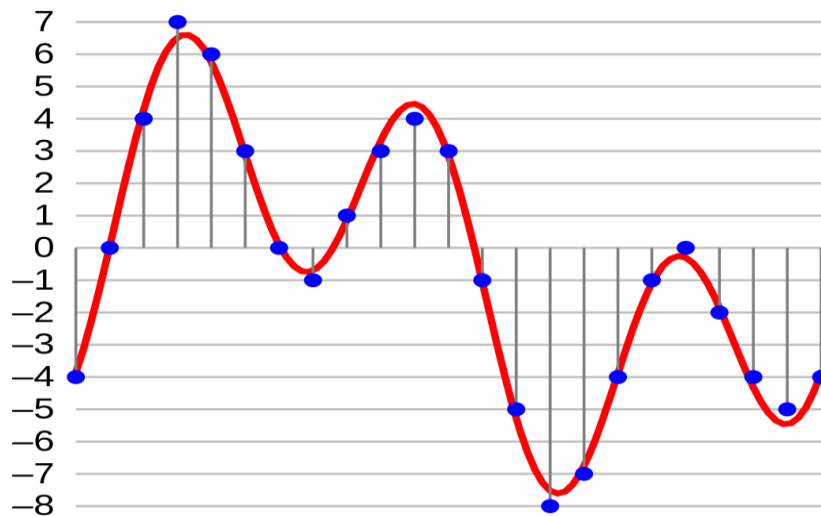


Figure 7 Sampling for audio points

The number of samples per second (or other unit) taken from a continuous signal to create a discrete or digital signal is referred to as sampling rate or sampling frequency in **Figure 7**. An audio frame is a series of audio samples from one or more audio sources for a particular period of time. The most common sampling rates are the aforementioned 8kHz (most commonly used for telephone communications), 44.1kHz (most commonly used for music CDs), and 48kHz (most common for audio tracks in movies).

5.2 Overlapping of Audio Features

The audio features for the corresponding audio frames should be overlapped to achieve the sequence/continuation of the audio with the corresponding video frame.

For 1 sec video there are 30 fps (Frames Per Sec), and the sample rate for the audio is 22050 KHZ

5.2.1 Matching Audio Data With Video Data

In the audio features, the audio frames have been generated for a duration of 20 sec for each audio dataset. For 1 video frame will have 33.3 (1000/30) this is the duration of 1 frame considered for the video dataset. In the audio frame 1 sec (1000 ms) we have 22050 samples each, 1 audio frame is measured as 20 ms we get 441 samples each and the shifting of each audio frame take 20 ms for each shift for the frame and will have 220 samples in the

audio frame and the corresponding video dataset, have to take 1 video frame (22050/30) will give 753 samples .

5.3 Visualizing the Cepstrum

To understand the working of the cepstrum, there are few steps to follow as given in Figure 8 .

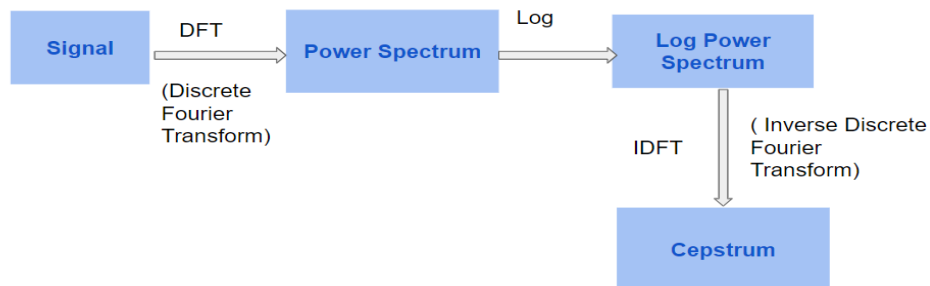


Figure 8 Steps for cepstrum

5.3.1 Signal

A signal is a change in a quantity over time. The quantity that varies in audio is air pressure. The rate at which we sample the data can vary, but it is most commonly 44.1kHz, or 44,100 samples per second. We have captured a waveform for the signal, which can be interpreted, modified, and analyzed using computer software.

5.3.2 Discrete Fourier Transform

It transforms a finite sequence of equally spaced samples of a function into a same-length sequence of equally spaced samples of the discrete-time Fourier transform (DTFT), which is a complex-valued frequency function.

5.3.3 Power Spectrum

It describes the signal's power distribution into frequency components. Any physical signal can be decomposed into a number of discrete frequencies or a spectrum of frequencies spanning a continuous range using Fourier analysis.

5.3.4 Cepstrum

The goal of cepstral is to separate speech into its source and system components without any prior knowledge of the source and/or system. In the time domain, the speech sequence must be deconvolved into the excitation and vocal tract components.

5.4 MEL-FREQUENCY CEPSTRAL COEFFICIENTS (MFCC)

To generate the Mel-Frequency Cepstral Coefficients Extraction there are some steps to be followed discussed in Figure 9.

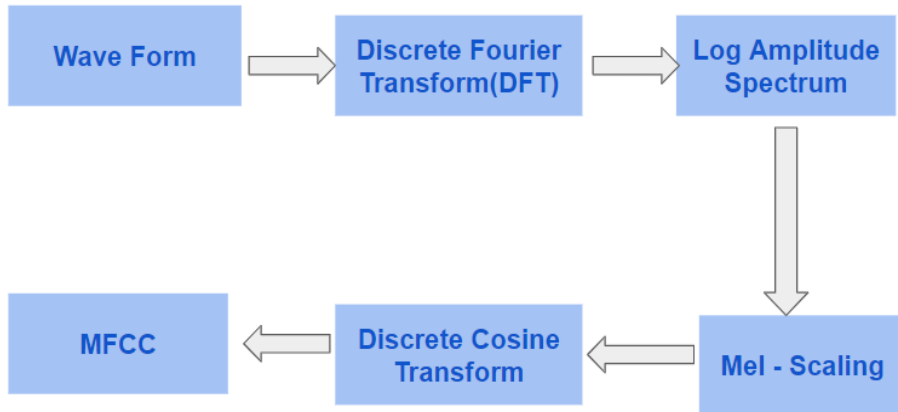


Figure 9 Generating MFCC

5.4.1 Mel Scale

Humans detect differences in lower frequencies better than higher frequencies. Equal distances in pitch sounded equally distant to the listener if equal distances in pitch were equal. It adjusts the frequency to match what the human ear can hear more closely (humans are better at identifying small changes in speech at lower frequencies).

5.4.2 Mel-frequency Cepstral Coefficients(MFCC) Point Extraction

The MFCC is achieved using the librosa in-build package in python in which it takes the audio file dataset and converts it into points. Divide the audio signal into 20–40ms frames. Audio signals do not change much on short time scales, but if the frames are longer, the audio signals change too much throughout the frame, so a frame size of 20–40ms works. These are the audio vector points after being extracted from the logarithmic scale of audio frequency.

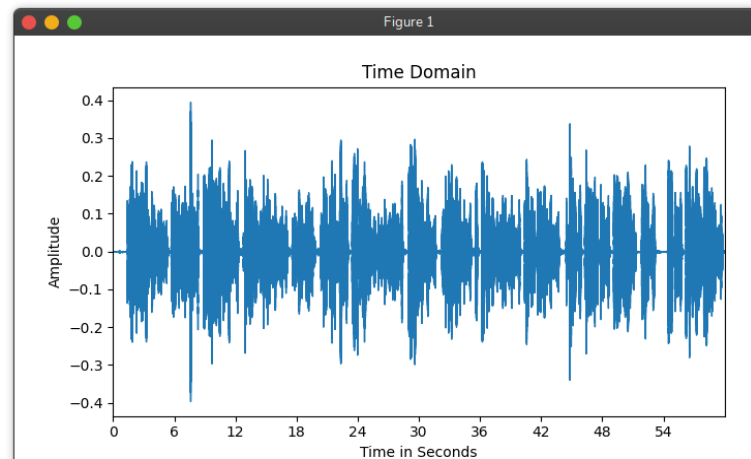


Figure 10 Audio in WaveForm

Figure 10 shows waveform is a representation of an audio signal or recording in the form of an image. It depicts amplitude changes over time. The signal's amplitude is measured vertically on the y-axis, and time is measured horizontally on the x-axis (horizontally). Figure 11 shows an audio frequency, also known as an audible frequency, is a periodic vibration with a frequency in the human hearing range. The hertz is the standard SI unit of frequency. Pitch is mostly determined by the property of sound.

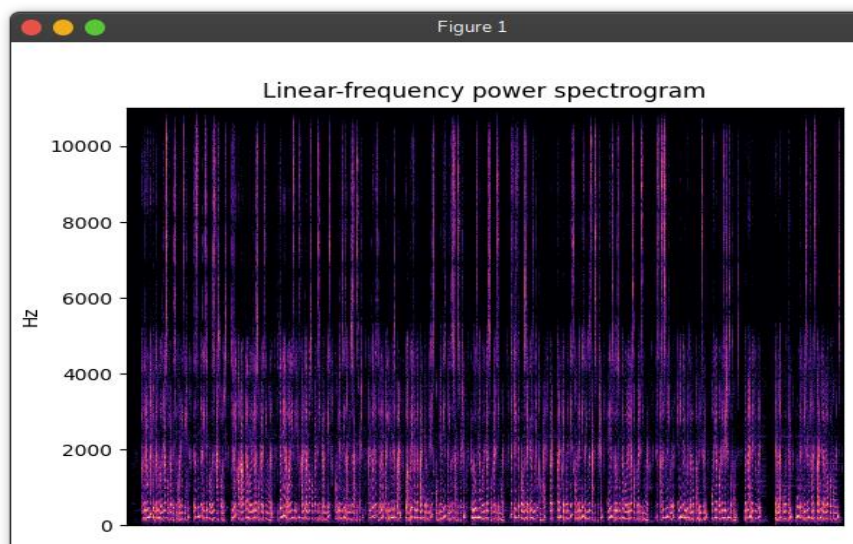


Figure 11 Audio in Frequency Form

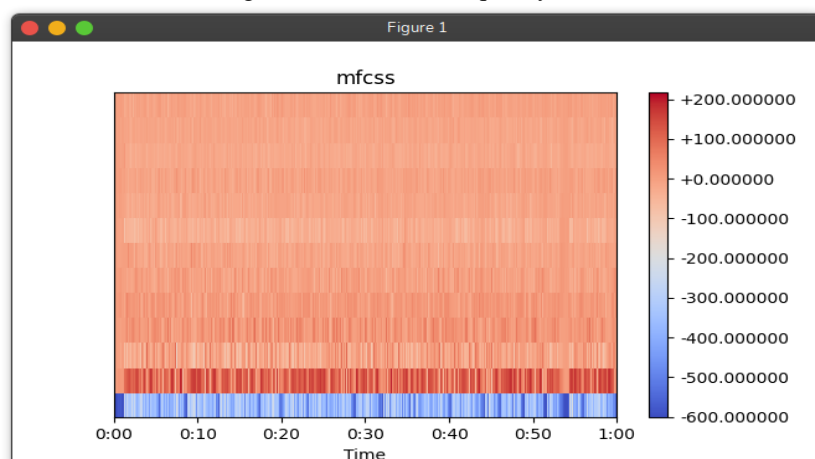


Figure 12 Mel-Spectrum

Figure 12 shows the mel-frequency cepstrum (MFC) in sound processing is a representation of a sound's short-term power spectrum based on a linear cosine transform of a log power spectrum on a nonlinear mel frequency scale. Mel-frequency cepstral coefficients (MFCCs) are the coefficients that comprise an MFC.

Consider the Fourier transform of (a windowed excerpt of) a signal. Using triangular overlapping windows, map the powers of the obtained spectrum onto the mel scale. Take the logs of the powers at each of the mel frequencies. Take the discrete cosine transform of the list of mel log powers, as if it were a signal. The MFCCs are the amplitudes of the resulting spectrum.

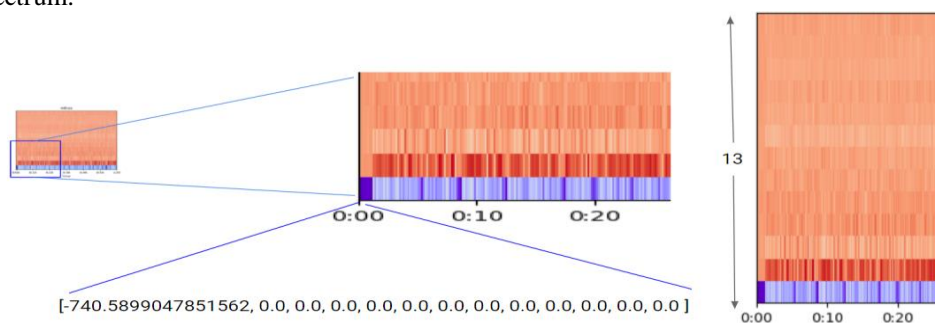


Figure 13 Output of Mel -spectrum

Figure 13 shows output for each audio frame which has 13 variables for each frame and the corresponding are the vector points for a frame. These are the vector points for the audio after being extracted from the logarithmic scale of audio frequency.

6. Generating Code Book

The dataset preprocessing techniques used to develop the model for codebook generation using the Clustering methods and probability distribution are discussed in this section.

6.1 K-MEANS Clustering

A number of individual neural networks are trained, and then the k-means clustering algorithm is used to select a subset of the trained individuals' weights and thresholds for improving diversity. Following that, individuals from the nearest clustering center are chosen to make up the membership's initial weights and thresholds of the ensemble learning. K-means clustering has been performed for vector points. Clusters have been generated with the size of 256.

The scatterplot has been drawn for the centroids with the x-axis. Figure 14 shows the Snapshot of K-means clustering.

```

kmeans = KMeans(n_clusters=256,)
kmeans.fit(singleData)

centroids = kmeans.cluster_centers_
labels = kmeans.labels_

data['names'].append(names[i])
data['centroids'].append(centroids.tolist())
data['labels'].append(labels.tolist())

```

Figure 14 Snapshot of K-means clustering

6.2 Codebook Generation

An image is divided into 4 x 4 pixel blocks during codebook generation. The blocks are transformed into K-dimensional vectors. These vectors are referred to as training vectors, and the collection of training vectors is referred to as the training set of size N vectors. The equation is used to calculate N. Where N is the number of clustering to be formed.

```

kmeans = KMeans(n_clusters=256)
kmeans.fit(allData)
labels = kmeans.labels_

```

Based on the K-means clustering, the labels have been determined. Labels denote the symbols in the codebook. This symbol is the representation of visual data. The Snapshot of codebook generation and Figure 15 depicts Snapshot of audio codebook symbols is generated.

```

246, 246, 246, 246, 246,
29, 76, 76, 128, 128,
51, 51, 89, 51, 187, 27,
140, 140, 1, 15, 15, 250,
51, 15, 140, 187, 140,
51, 140, 187, 7, 7, 89,

```

Figure 15 Snapshot of video codebook symbols

6.3 Probability Distribution

The Probability distribution is used to integrate the audio and video vector points. The Conditional Probability distribution has been determined for the audio symbol codebook and

video symbol codebook. Conditional probability is the probability of one event occurring with some relationship to one or more other events.

For every occurrence of the video symbol, the corresponding co-occurrence of the audio symbol was determined and this probability has been used to predict the accuracy level of the developed model. The formula of conditional probability is

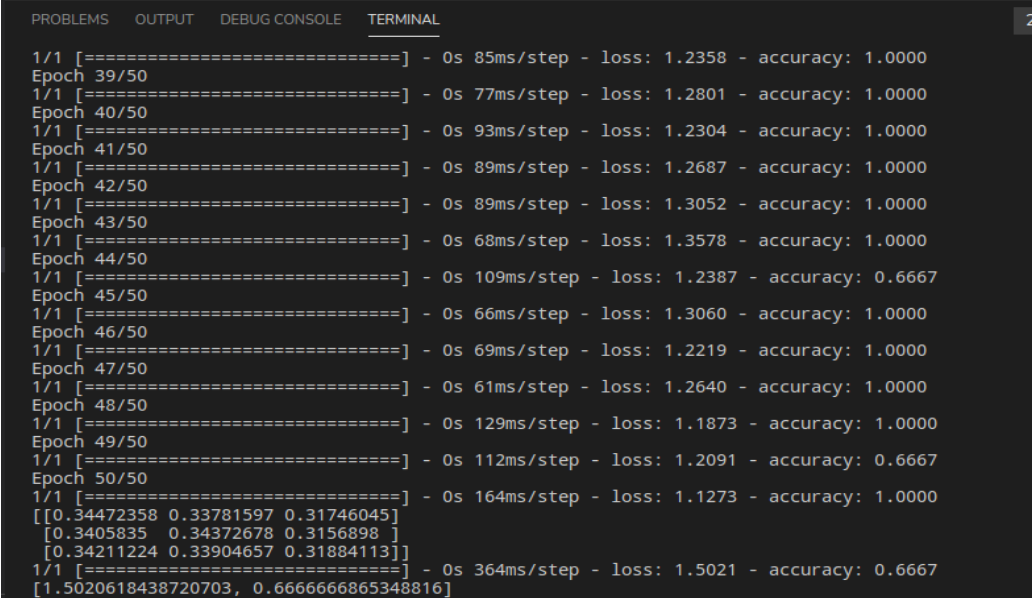
$$P(B|A) = P(A \cap B) / P(A)$$

Where $P(A \cap B)$ – the joint probability of events A and B (i.e) the probability that both events A and B occur and $P(B)$ – the probability of event B.

7 Experimental Results and Discussion

The prediction and experimental results and discussion of the implemented model, which takes input from the user and determines the accuracy level and checks whether the user is authenticated are discussed in this section.

7.1 Prediction



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2.
1/1 [=====] - 0s 85ms/step - loss: 1.2358 - accuracy: 1.0000
Epoch 39/50
1/1 [=====] - 0s 77ms/step - loss: 1.2801 - accuracy: 1.0000
Epoch 40/50
1/1 [=====] - 0s 93ms/step - loss: 1.2304 - accuracy: 1.0000
Epoch 41/50
1/1 [=====] - 0s 89ms/step - loss: 1.2687 - accuracy: 1.0000
Epoch 42/50
1/1 [=====] - 0s 89ms/step - loss: 1.3052 - accuracy: 1.0000
Epoch 43/50
1/1 [=====] - 0s 68ms/step - loss: 1.3578 - accuracy: 1.0000
Epoch 44/50
1/1 [=====] - 0s 109ms/step - loss: 1.2387 - accuracy: 0.6667
Epoch 45/50
1/1 [=====] - 0s 66ms/step - loss: 1.3060 - accuracy: 1.0000
Epoch 46/50
1/1 [=====] - 0s 69ms/step - loss: 1.2219 - accuracy: 1.0000
Epoch 47/50
1/1 [=====] - 0s 61ms/step - loss: 1.2640 - accuracy: 1.0000
Epoch 48/50
1/1 [=====] - 0s 129ms/step - loss: 1.1873 - accuracy: 1.0000
Epoch 49/50
1/1 [=====] - 0s 112ms/step - loss: 1.2091 - accuracy: 0.6667
Epoch 50/50
1/1 [=====] - 0s 164ms/step - loss: 1.1273 - accuracy: 1.0000
[[0.34472358 0.33781597 0.31746045]
 [0.3405835 0.34372678 0.3156898 ]
 [0.34211224 0.33904657 0.31884113]]
1/1 [=====] - 0s 364ms/step - loss: 1.5021 - accuracy: 0.6667
[[1.5020618438720703, 0.66666666865348816]
```

Figure 16 Snapshot of Testing the Model

The model has been developed and trained using a train dataset and the model has been tested using a test dataset. Figure 16 shows the Snapshot of Testing the model and loading the test dataset in Tensorflow.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Epoch 44/50
1/1 [=====] - 0s 109ms/step - loss: 1.2387 - accuracy: 0.6667
Epoch 45/50
1/1 [=====] - 0s 66ms/step - loss: 1.3060 - accuracy: 1.0000
Epoch 46/50
1/1 [=====] - 0s 69ms/step - loss: 1.2219 - accuracy: 1.0000
Epoch 47/50
1/1 [=====] - 0s 61ms/step - loss: 1.2640 - accuracy: 1.0000
Epoch 48/50
1/1 [=====] - 0s 129ms/step - loss: 1.1873 - accuracy: 1.0000
Epoch 49/50
1/1 [=====] - 0s 112ms/step - loss: 1.2091 - accuracy: 0.6667
Epoch 50/50
1/1 [=====] - 0s 164ms/step - loss: 1.1273 - accuracy: 1.0000
[[0.34472358 0.33781597 0.31746045]
 [0.3405835 0.34372678 0.3156898 ]
 [0.34211224 0.33904657 0.31884113]]
1/1 [=====] - 0s 364ms/step - loss: 1.5021 - accuracy: 0.6667
[1.5020618438720703, 0.66666666865348816]
```

Figure 17 Snapshot of Predicting the Accuracy

The accuracy of the model resulted in 66.67% accuracy. Figure 17 shows the Snapshot of Predicting the Accuracy.

7.2 Live Dashboard

This subsection has the snapshots of the dashboard which was designed and developed using a python-based framework, Flask. Figure 18 Snapshot of the dashboard before uploading the dataset. The user can choose a video file for predicting the accuracy level and to determine whether the user is authorized. Figure 19 shows a Snapshot of uploading the dataset from a local machine. After fetching the video file, the user is able to play the video and When the user clicks the predict button, the accuracy level has been displayed.



Figure 18 Snapshot of the dashboard before uploading the dataset.

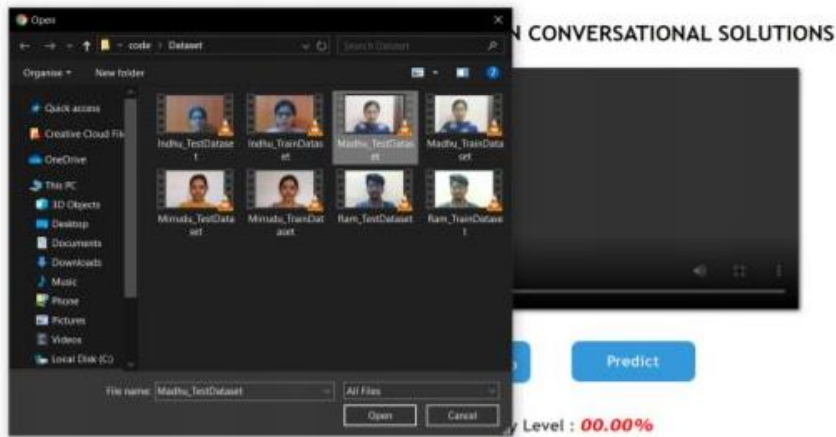


Figure 19 Snapshot of fetching the dataset.

VISUAL INTELLIGENCE IN CONVERSATIONAL SOLUTIONS

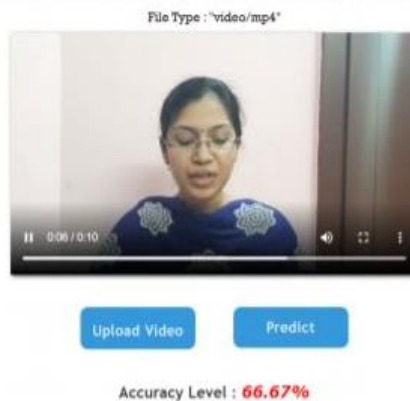


Figure 20 Snapshot of the dashboard after prediction

Figure 20 shows the experiments are done to predict the accuracy of the models and to determine if the user is authenticated or not. The snapshots of the dashboard that takes the user input as the video file and predicts the accuracy and user authentication.

Conclusion

Custom Video dataset has been recorded without any background noise and Audio Dataset has been extracted. These datasets have the sequence of audio files to extract the audio features for each person, both train dataset and test dataset. The preprocessed input of the model involves audio features of a person that are used to classify the vector points. Vector Quantization and K means classification are implemented and take the preprocessed datasets as input and generate the codebook of the size 256. The symbols or codes in the codebook will be compared and models with maximum probability have been chosen and considered for

words spoken by the speaker. The models were trained with custom datasets which produced an overall accuracy of about 67%. The Voice Intelligence security system is envisaged that this model can be developed for real time applications. As People become increasingly comfortable with biometrics, Voice authentication is finding wider application across the industries, including healthcare, banking, and education.

Acknowledgement

We thank industry mentor Dr. Sitaram Ramachandrupa, Senior Director, Data Science, [24]7ai, Bengaluru, who provided insight and expertise that greatly assisted the research of this paper. We would like to thank and show our gratitude to Mr Shanmugam Nagarajan, Co-Founder, [24]7ai, for sharing his pearls of wisdom with us during the collaborative students' project work.

References

- [1] US Patent Application for Authenticating A User By Correlating Speech and Corresponding Lip Shape Patent Application - SITARAM RAMACHANDRULA , HARIHARAN RAVISHANKAR
- [2] Mohan, Bhadrageetha. "Speech recognition using MFCC and DTW." 2014 International Conference on Advances in Electrical Engineering (ICAEE). IEEE, 2014.
- [3] Wang, Fang, and Q. J. Zhang. "An improved K-means clustering algorithm and application to combine multi-codebook/MLP neural network speech recognition." In Proceedings 1995 Canadian Conference on Electrical and Computer Engineering, vol. 2, pp. 999-1002. IEEE, 1995.
- [4] Furui, Sadaoki. "Vector-quantization-based speech recognition and speaker recognition techniques." In Conference Record of the Twenty-Fifth Asilomar Conference on Signals, Systems & Computers, pp. 954-955. IEEE Computer Society, 1991.
- [5] Chadawan Ittichaichareon, Siwat Suksri and Thaweesak Yingthawornsuk "Speech Recognition using MFCC" International Conference on Computer Graphics, Simulation and Modeling (ICGSM'2012) July 28-29, 2012 Pattaya (Thailand)
- [6] Murali Krishnan, Chris P. Neophytou and Glenn Prescott "WAVELET TRANSFORM SPEECH RECOGNITION USING VECTOR QUANTIZATION, DYNAMIC TIME WARPING AND ARTIFICIAL NEURAL NETWORKS" Center for Excellence in Computer Aided Systems Engineering and Telecommunications & Information Sciences Laboratory 2291 Irving Hill Drive, Lawrence, KS 66045
- [7] Sitaram Ramachandrupa , Hariharan Ravishankar, US Patent Application for Authenticating A User By Correlating Speech and Corresponding Lip Shape Patent Application.
- [8] Sujatha Paramasivam, Radhakrishnan Murugesanadar, Published 2018, An Optimized Model for Visual Speech Recognition Using HMM, Int. Arab J. Inf. Technol.
- [9] Ara V. Nefian, Luhong Liang, Xiaobo Pi, Liu Xiaoxiang, Crusoe Mao and Kevin Murphy, Published 2002, A COUPLED HMM FOR AUDIO-VISUAL SPEECH RECOGNITION, IEEE.
- [10] Morade, S. S., & Patnaik, B. S. (2013). Automatic lip tracking and extraction of lip geometric features for lip reading. International Journal of Machine Learning and Computing, 3(2), 168.
- [11] Nainan, S., & Kulkarni, V. (2019). Lip tracking using deformable models and geometric approaches. In Information and Communication Technology for Intelligent Systems (pp. 655-663). Springer, Singapore.
- [12] Raghuvver, L. V. S., & Deora, D. (2017). Lip Localization and Visual Speech Recognition with Optical Flow in Hindi. International Journal of Computer Sciences and Engineering (JCSE), 5(5), 209-212.
- [13] Nainan, S., Kulkarni, V., & Srivastava, A. (2017, March). Computer Vision Based Real Time Lip Tracking for Person Authentication. In International Conference on Information and Communication Technology for Intelligent Systems (pp. 608-615). Springer, Cham.

- [14] BhadragiriJagan Mohan and Ramesh Babu N., "Speech recognition using MFCC and DTW," *2014 International Conference on Advances in Electrical Engineering (ICAEE)*, 2014, pp. 1-4, doi: 10.1109/ICAEE.2014.6838564.