Abstract

Speaker recognition has two words where the earlier word represents the one who speaks and the latter word means to recognize. Hence speaker recognition is a system that recognizer the speaker according to the trained audio. Speaker recognition is possible by converting the speech to graph called spectrogram and performing image classification of those graphs to obtain the varieties in sound. It is audio classification using image classification. Inception v3 architecture is used to train the images and as a classification model. Speaker recognition can be used for security purpose not only in digital but also embedded systems. It can be applied where voice detection is an issue like investigation bureaus.

Table of Content

Chapter 1: Introduction

Sounds produced by various instrument and voice of people are differentiable due to their pitch, frequency, intensity, amplitude, etc. Humans and animals can perform this task of differentiating using ears and nervous system, where those parameters of sound are calculated. In Machines, it is possible to separate the sound manually, through calculation of frequency, amplitude, pitch, quality etc, that requires complex vector or matrix calculations. The other way to differentiate various sounds is through a graph called **Spectrogram.** A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. Spectrogram not only tells about energy level, for example, 2 Hz vs 10 Hz, but also describes how energy levels vary over time. When applied to an audio signal, spectrograms are sometimes called sonographs, voiceprints, or voicegrams [1].

The project ‘Speaker Recognition’ uses the graph, spectrogram to classify human sounds. It classifies audio using image classification. The system takes input in the .mp3 format and classifies the sound as one of the trained sound. It calculates the match percentage between the trained audio and test audio.

Objectives

The speaker recognition system is built with the following objectives:

- To construct a securing mechanism for data and information using sound parameter in digital system.
- To develop a model that can be applied in lock and unlock phenomenon.
- To create a system that can help people with disabilities, recognize people by analyzing sound.
- To apply Artificial Intelligence for semester project.

Chapter 2: Agent

2.1 Agent Description

An agent perceives its environment through sensors and acts on the environment through actuators. Its behavior is described by its function that maps the percept to action. For Speaker Recognition, the agent is Computer System, basically the software. The software agent senses the input through microphone and then through files. The agent maps the input audio to classification function and results in match percentage of the audio.

PEAS measure for the agent are:

- **P (Performance):** The performance measure for agent would be: Match accuracy of the result, time taken to compute the result given input, etc.
- **E (Environment):** Environment where the agent performs is: Computer System. It can further be modified to work on android and ios sytems. Other environment where the software interacts is audio system.
- **A (Actuators):** Actuator defines the means through which the output occurs. For our system, it is Screen Display where the match percentage is displayed.
- **S (Sensors):** Sensor for the system should be a microphone. But since the system is not yet integrated with UI, input is taken in file format. So the sensor is keyboard and mouse using which the input is provided.

The agent used here is **Model - Based Reflex Agent** since it observes situations i.e. the audio input. Result of training data shows how the world revolves i.e. how the audio is separated from the other. Model – Based Reflex agent works for partially observable environment.

The agent here is **autonomous**, since it can perform tasks in pursuit of a goal with minimum of or direct supervision or direct control, but can interact with trained system to obtain output results. It may or may not have a user interface.

Agent comprises of architecture and program which is described in the chapter 4.

Agent = Architecture + Program

2.2 Agent Environment

Environment for the agent of Speaker Recognition system is the computer system, where it works and audio, what it deals with.

The environment of this system is **dynamic** because the test audio can change for various test. With dynamic input, the environment needs to be dynamic.

The environment of this system is **episodic** because the output does not depend on past input, its output or process. Once the audios are trained, the processing of input audio does not depend upon past input audio or their output. Hence, decision does not depend on previous decisions.

It is **continuous** because there are no distinct clearly defined states of the environment.

The environment of this system is **stochastic** because the next state of this environment is not completely determined by the current state and the action executed by the agent.

It is **partially observable** because the system does not know what input would be provided for test audio.

This environment is **single agent** because there is only one agent active in the environment.

Chapter 3: Problem Specification

Problem:

The purpose of this system is to recognize ones voice. This may be useful for security purpose like applock with voice sensor, digital support for handicapped people by making them know who they are talking to or whose voice they are listening.

So the problem is, given a large set of audio files, the system must be able to classify the new audio file as one of the voice trained earlier with larger set.

Goal:

Goal is obtained when the new test audio matches with any of the trained audio with maximum accuracy.

Constraints:

The system should not use other voices except for the trained ones to compare the test voice. And the system should classify the voice to at least one trained voice.

Specification defines the purpose of the system. The purpose of the system is to recognize the speaker. The tree types of specifications for the Speaker Recognition system are described below:

Ideal Specification: The system is wished to classify any humans' voice among those of the trained with 100% accuracy.

Design Specification: The specification that we actually use is that the system should classify human's voice with maximum accuracy.

Revealed Specification: The system actually classifies humans' voice but with few errors. Sometimes it classifies ones voice as others.

Chapter 4: Data Source

Inception V3 by Google is the 3rd version in a series of Deep Learning Convolutional Architectures. It was trained using a dataset of 1,000 classes from the original ImageNet dataset. Inception v3 is pre-trained, widely-used image recognition model that has been shown to attain greater than 78.1% accuracy on the ImageNet dataset. [3].

The Inception v3 model takes weeks to train on a monster computer with 8 Tesla K40 GPUs and probably costing $30,000 so it is impossible to train it on an ordinary PC. Pre-trained Inception model was downloaded and used to classify images. The Inception v3 model has nearly 25 million parameters and uses 5 billion multiply-add operations for classifying a single image.

ImageNet dataset:

ImageNet, is a dataset of over 15 millions labeled high-resolution images with around 22,000 categories. ILSVRC uses a subset of ImageNet of around 1000 images in each of 1000 categories. In all, there are roughly 1.2 million training images, 50,000 validation images and 100,000 testing images.

4.1. Data used

5 voices are used for training as well as testing.

I. PM KP Oli's speech in youtube as, PM KP oli speech
Link: https://www.youtube.com/watch?v=snsOPq0_HsA
It is 43 min 51 sec long.

II. Abdul Kalam's Speech in youtube as, Dr. APJ Abdul Kalam's Life Advice Will Change Your Future (MUST WATCH) Motivational Speech.
Link: https://www.youtube.com/watch?v=7fIL5s_Kq68&t=463s
It is 22 minute long.

III. Sashi Taroor's speech in dropbox,
Link:
https://www.dropbox.com/s/ycxi1fz3lzwagsd/tf_files.tar.gz?dl=0&file_subpath=%2F

tf_files%2Fdata_audio%2FTaroor%2Fshashi+tarur+about+Great+India..+great+speech+for+all+Indians+to+listen+and+know.mp3

The duration is 17 min 49 sec.

IV. Arijit Singh Songs from youtube as, Top 5 heart touching songs of arijit singh

Link: https://www.youtube.com/watch?v=8_rSsJXmLNk

It is 25 min 43 sec long.

V. The last audio is self recorded named as anje_story which is 5 min 5 sec long.

For testing, one spectrogram created during data preparation for all the audios is taken from each folder.

Chapter 5: Algorithm and Technical Descriptions

The project ‘Speaker Recognition’ uses the graph, spectrogram to classify human sounds. It classifies audio using image classification. User should input few recorded speech of different people in a .mp3 format. The .mp3 file is converted first to wav file since python libraries can better perform in wav format. Those wav files are then chunked in the interval of few seconds, basically 10 to 20. Then a spectrogram is created from each chunk of wav file. After training the graph using tensorflow, a bottleneck value is calculated which is then the comparison graph for the other testing sounds. Then the audio for testing is then compared with output value and generate the match percentage between various sounds.

5.1 Architecture: Inception v3

Google's inception model, trained on imagenet dataset, can classify 1,000 classes of objects and is open sourced by Google. . Inception v3 network stacks 11 inception modules where each module consists of pooling layers and convolutional filters with rectified linear units as activation function. [2]

Anyone can use the trained model or retrain its last layer for new classes or to build your own classification model. It gives accurate predictions in less time using deep convolution neural network. Inception can only be trained with images. It can only do image classification. So the audio needs to be converted into image. After generating the spectrogram, the inception model is trained with those graphs.

5.2 Algorithm in depth

Reading Audio Files

The first thing this system does is to read the audio file. The file must be in .mp3 format.

Data Preparation

Once the set of audio files is specified, they are stored in different folder with the same name as that of the audio respectively. For more effective result, the larger audio file is preferred and also the total time duration of all voice clips should be similar. Then, the script data_maker.py is executed which contains code to convert the .mp3 file to eventually a set of graphs.

The following libraries are used for reading of audio file and working with it:

- **sox**: SoX is a cross-platform (Windows, Linux, MacOS X, etc.) command line utility that can convert various formats of computer audio files in to other formats.
- **libsox-fmt-mp3:** This package contains the SoX MP2 and MP3 format library.
- **ffmpeg:** It is a cross-platform solution to record, convert and stream audio and video.
- **Python-tk:** It provides a robust and platform independent windowing toolkit.

This script does a series of tasks explained by the flowchart:

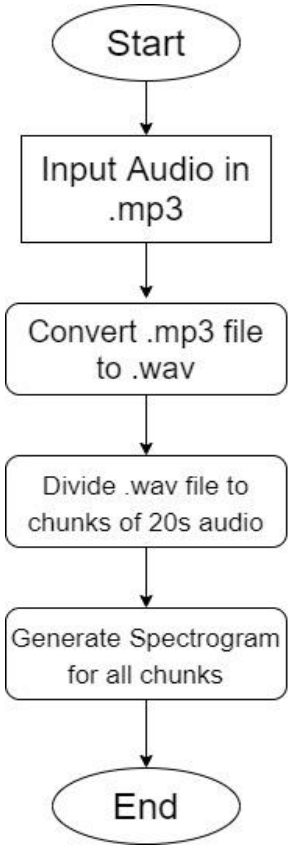

Start
Input Audio in .mp3
Convert .mp3 file to .wav
Divide .wav file to chunks of 20s audio
Generate Spectrogram for all chunks
End

Figure 1: Flowchart for Data Preparation

Training

Google inception model is used for training. Training occurs in file retrain.py where the series of activities occur:

1. Necessary directories that can be used during training are prepared.
2. Information about the inception model architecture is gathered.
3. Set up the pre-trained graph. Model is downloaded and extracted.
4. A list of all the images is created looking at the folder structure
5. Carries out distortions if required in the graph (for compression, height, width, pixels etc)
6. Then the image is decoded using tensorflow.
7. Bottleneck value is calculated.
8. A new layer that is to be trained is added.
9. The operation to evaluate the accuracy of result is evaluated on the basis of evaluation steps and predictions.
10. All the weights are set to their initial default value.
11. Training is run for few cycles around 500 to 5000.
    - 11.1 Batch bottleneck value is calculated, freshly if distortions were applied else from the cache disk.
    - 11.2 Bottleneck and ground truth value is fed to the graph, and a training step is run.
    - 11.3 Training accuracy and cross entropy is evaluated.
    - 11.4 Validation step is run and validation accuracy is calculated.
12. A final test evaluation on the images that were not trained is run.
13. A training graph is labeled with weight written as constant.

Testing

For testing label_image.py script is run. The spectrogram which is to be classified is sent as input in this section. Also the retrained graph (output graph) and the evaluation value obtained from training are also sent as input to this script. Graph is loaded into the system, and evaluated using tensorflow.

5.3 Technical Descriptions

Fast Fourier Transform (FFT)

Fast Fourier Transformation is a mathematical algorithm that calculates Discrete Fourier Transform (DFT) of a given sequence. FFT algorithm can convert this time-domain discrete signal into a frequency-domain. FFT is used in this system to generate spectrogram for the chunks of .wav file. FFT for any digital signal can be calculated as:

$$F(x) = \sum_{n=0}^{N-1} f(n) e^{-j2\pi(x\frac{n}{N})}$$

$$f(n) = \frac{1}{N}\sum_{n=0}^{N-1} F(x) e^{j2\pi(x\frac{n}{N})}$$

Spectrogram

Spectrograms may be created from a time-domain signal in one of two ways: approximated as a filterbank that results from a series of or calculated from the time signal using the Fourier transform. Spectrograms are basically two-dimensional graphs, with a third dimension represented by colors. Time runs from left (oldest) to right (youngest) along the horizontal axis. The vertical axis represents frequency, which can also be thought of as pitch or tone, with the lowest frequencies at the bottom and the highest frequencies at the top. The amplitude (or energy or “loudness”) of a particular frequency at a particular time is represented by the third dimension, color, with dark blues corresponding to low amplitudes and brighter colors up through red corresponding to progressively stronger (or louder) amplitudes.

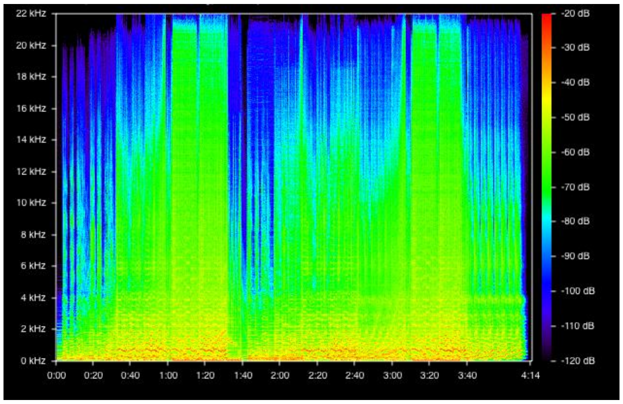


Figure 2: Spectrogram for speech

Chapter 6: Outcome Analysis

In order to use the system, first the training audio is fed. For that, 5 different speeches are taken; 4 from internet and 1 self spoken 20 minute audio. The audios are stored in respective folders name. Then on execution of data_maker.py, chunks of audio are received which are again converted to spectrogram as shown below:

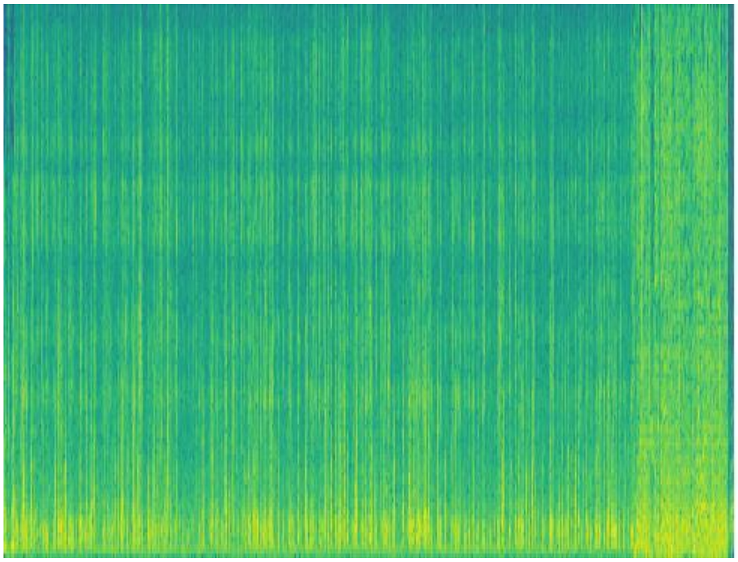
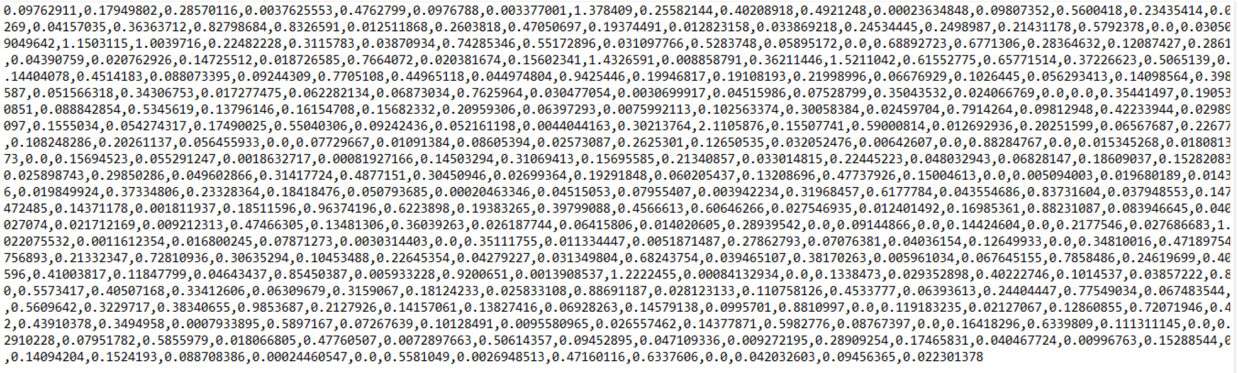


Figure 3: Spectrogram for audio of kp_oli

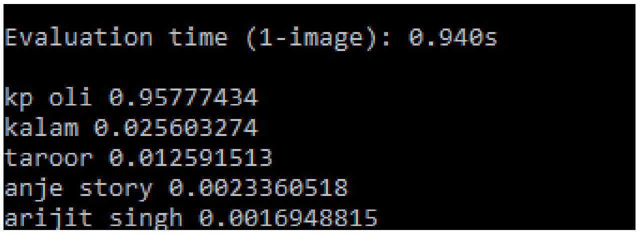
Then after training the images, output graph is received in .pb format along with bottleneck value for each audio.



```
0.09762911,0.17949802,0.28570116,0.0037625553,0.4762799,0.0976788,0.003377001,1.378409,0.25582144,0.40208918,0.4921248,0.00023634848,0.09807352,0.5600418,0.23435414,0.
269,0.04157035,0.36363712,0.82798684,0.8326591,0.012511868,0.2603818,0.47050697,0.19374491,0.012823158,0.033869218,0.24534445,0.2498987,0.21431178,0.5792378,0.0,0.0305
9049642,1.1503115,1.0039716,0.22482228,0.3115783,0.03870934,0.74285346,0.55172896,0.031097766,0.5283748,0.05895172,0.0,0.68892723,0.6771306,0.28364632,0.12087427,0.2861
,0.04390759,0.020762926,0.14725512,0.018726585,0.7664072,0.020381674,0.15602341,1.4326591,0.008858791,0.36211446,1.5211042,0.61552775,0.65771514,0.37226623,0.5065139,0.
.14404078,0.4514183,0.088073395,0.09244309,0.7705108,0.44965118,0.044974804,0.9425446,0.19946817,0.19108193,0.21998996,0.06676929,0.1026445,0.056293413,0.14098564,0.398
587,0.051566318,0.34306753,0.017277475,0.062282134,0.06873034,0.7625964,0.030477054,0.0030699917,0.04515986,0.07528799,0.35043532,0.024066769,0.0,0.0,0.35441497,0.19053
0851,0.088842854,0.5345619,0.13796146,0.16154708,0.15682332,0.20959306,0.06397293,0.0075992113,0.102563374,0.30058384,0.02459704,0.7914264,0.09812948,0.42233944,0.02989
097,0.1555034,0.054274317,0.17490025,0.55040306,0.09242436,0.052161198,0.0044044163,0.30213764,2.1105876,0.15507741,0.59000814,0.012692936,0.20251599,0.06567687,0.22677
,0.108248286,0.20261137,0.056455933,0.0,0.07729667,0.01091384,0.08605394,0.02573087,0.2625301,0.12650535,0.032052476,0.00642607,0.0,0.88284767,0.0,0.015345268,0.0180813
73,0.0,0.15694523,0.055291247,0.0018632717,0.00081927166,0.14503294,0.31069413,0.15695585,0.21340857,0.033014815,0.22445223,0.048032943,0.06828147,0.18609037,0.15282083
0.025898743,0.29850286,0.049602866,0.31417724,0.4877151,0.30450946,0.02699364,0.19291848,0.060205437,0.13208696,0.47737926,0.15004613,0.0,0.005094003,0.019680189,0.0143
6,0.019849924,0.37334806,0.23328364,0.18418476,0.050793685,0.00020463346,0.04515053,0.07955407,0.003942234,0.31968457,0.6177784,0.043554686,0.83731604,0.037948553,0.147
472485,0.14371178,0.001811937,0.18511596,0.96374196,0.6223898,0.19383265,0.39799088,0.4566613,0.60646266,0.027546935,0.012401492,0.16985361,0.88231087,0.083946645,0.040
027074,0.021712169,0.009212313,0.47466305,0.13481306,0.36039263,0.026187744,0.06415806,0.014020605,0.28939542,0.0,0.09144866,0.0,0.14424604,0.0,0.2177546,0.027686683,1.
022075532,0.0011612354,0.016800245,0.07871273,0.0030314403,0.0,0.35111755,0.011334447,0.0051871487,0.27862793,0.07076381,0.04036154,0.12649933,0.0,0.34810016,0.47189754
756893,0.21332347,0.72810936,0.30635294,0.10453488,0.22645354,0.04279227,0.031349804,0.68243754,0.039465107,0.38170263,0.005961034,0.067645155,0.7858486,0.24619699,0.40
596,0.41003817,0.11847799,0.04643437,0.85450387,0.005933228,0.9200651,0.0013908537,1.2222455,0.00084132934,0.0,0.1338473,0.029352898,0.40222746,0.1014537,0.03857222,0.8
0,0.5573417,0.40507168,0.33412606,0.06309679,0.3159067,0.18124233,0.025833108,0.88691187,0.028123133,0.110758126,0.4533777,0.06393613,0.24404447,0.77549034,0.067483544,
,0.5609642,0.3229717,0.38340655,0.9853687,0.2127926,0.14157061,0.13827416,0.06928263,0.14579138,0.0995701,0.8810997,0.0,0.119183235,0.02127067,0.12860855,0.72071946,0.4
2,0.43910378,0.3494958,0.0007933895,0.5897167,0.07267639,0.10128491,0.0095580965,0.026557462,0.14377871,0.5982776,0.08767397,0.0,0.16418296,0.6339809,0.111311145,0.0,0.
2910228,0.07951782,0.5855979,0.018066805,0.47760507,0.0072897663,0.50614357,0.09452895,0.047109336,0.009272195,0.28909254,0.17465831,0.040467724,0.00996763,0.15288544,0
,0.14094204,0.1524193,0.088708386,0.00024460547,0.0,0.5581049,0.0026948513,0.47160116,0.6337606,0.0,0.042032603,0.09456365,0.022301378
```

Figure 4: Bottleneck value for kp_oli audio file

Then for testing, a spectrograph for kp_oli's speech is sent as input. Then the result obtained is:

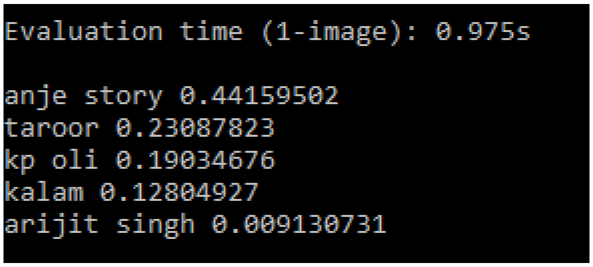


```
Evaluation time (1-image): 0.940s

kp oli 0.95777434
kalam 0.025603274
taroor 0.012591513
anje story 0.0023360518
arijit singh 0.0016948815
```

Figure 5: Output for kp_oli's speech testing

For anje story:

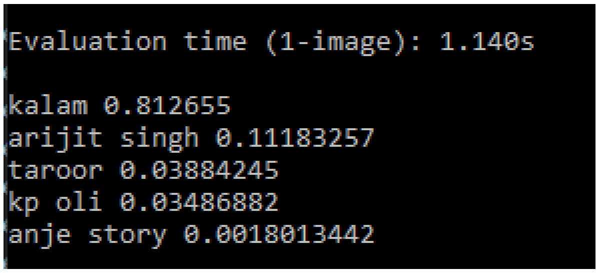


```
Evaluation time (1-image): 0.975s

anje story 0.44159502
taroor 0.23087823
kp oli 0.19034676
kalam 0.12804927
arijit singh 0.009130731
```

Figure 6: Output for anje_story's speech testing

For arijit singh:

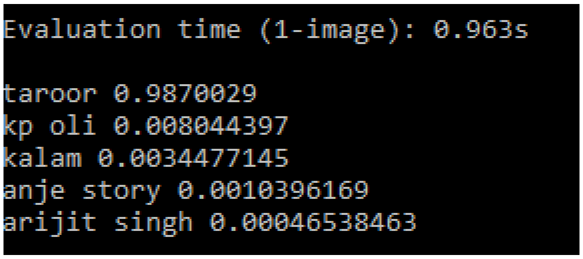


```
Evaluation time (1-image): 1.140s

kalam 0.812655
arijit singh 0.11183257
taroor 0.03884245
kp oli 0.03486882
anje story 0.0018013442
```

Figure 7: Output for arijit singh's speech testing

For taroor:

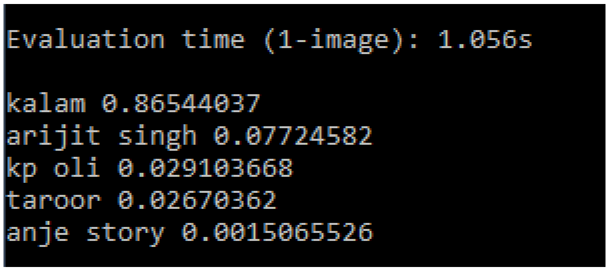


```
Evaluation time (1-image): 0.963s

taroor 0.9870029
kp oli 0.008044397
kalam 0.0034477145
anje story 0.0010396169
arijit singh 0.00046538463
```

Figure 8: Output for taroor's speech testing

For kalam:

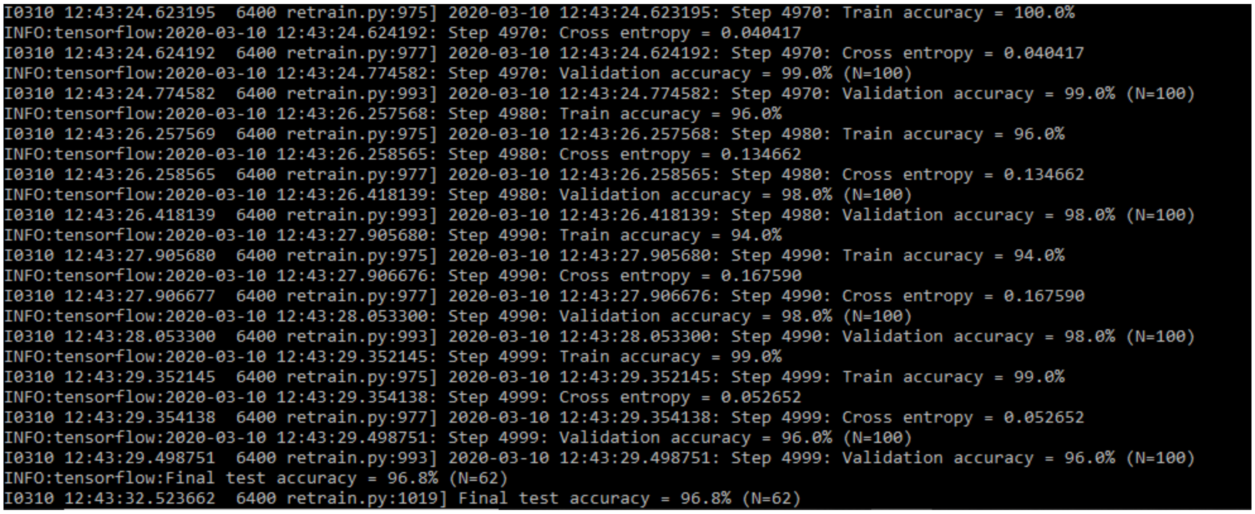


```
Evaluation time (1-image): 1.056s

kalam 0.86544037
arijit singh 0.07724582
kp oli 0.029103668
taroor 0.02670362
anje story 0.0015065526
```

Figure 9: Output for kalam's speech testing

The output pattern shows quite impressive result except for arijit singh’s input. This is the error due to less time duration of the audio, less training steps and complex data model. On 5 audio, 4 of the graphs were tested correctly. There was error for 1 spectrogram.



```
I0310 12:43:24.623195  6400 retrain.py:975] 2020-03-10 12:43:24.623195: Step 4970: Train accuracy = 100.0%
INFO:tensorflow:2020-03-10 12:43:24.624192: Step 4970: Cross entropy = 0.040417
I0310 12:43:24.624192  6400 retrain.py:977] 2020-03-10 12:43:24.624192: Step 4970: Cross entropy = 0.040417
INFO:tensorflow:2020-03-10 12:43:24.774582: Step 4970: Validation accuracy = 99.0% (N=100)
I0310 12:43:24.774582  6400 retrain.py:993] 2020-03-10 12:43:24.774582: Step 4970: Validation accuracy = 99.0% (N=100)
INFO:tensorflow:2020-03-10 12:43:26.257568: Step 4980: Train accuracy = 96.0%
I0310 12:43:26.257569  6400 retrain.py:975] 2020-03-10 12:43:26.257568: Step 4980: Train accuracy = 96.0%
INFO:tensorflow:2020-03-10 12:43:26.258565: Step 4980: Cross entropy = 0.134662
I0310 12:43:26.258565  6400 retrain.py:977] 2020-03-10 12:43:26.258565: Step 4980: Cross entropy = 0.134662
INFO:tensorflow:2020-03-10 12:43:26.418139: Step 4980: Validation accuracy = 98.0% (N=100)
I0310 12:43:26.418139  6400 retrain.py:993] 2020-03-10 12:43:26.418139: Step 4980: Validation accuracy = 98.0% (N=100)
INFO:tensorflow:2020-03-10 12:43:27.905680: Step 4990: Train accuracy = 94.0%
I0310 12:43:27.905680  6400 retrain.py:975] 2020-03-10 12:43:27.905680: Step 4990: Train accuracy = 94.0%
INFO:tensorflow:2020-03-10 12:43:27.906676: Step 4990: Cross entropy = 0.167590
I0310 12:43:27.906677  6400 retrain.py:977] 2020-03-10 12:43:27.906676: Step 4990: Cross entropy = 0.167590
INFO:tensorflow:2020-03-10 12:43:28.053300: Step 4990: Validation accuracy = 98.0% (N=100)
I0310 12:43:28.053300  6400 retrain.py:993] 2020-03-10 12:43:28.053300: Step 4990: Validation accuracy = 98.0% (N=100)
INFO:tensorflow:2020-03-10 12:43:29.352145: Step 4999: Train accuracy = 99.0%
I0310 12:43:29.352145  6400 retrain.py:975] 2020-03-10 12:43:29.352145: Step 4999: Train accuracy = 99.0%
INFO:tensorflow:2020-03-10 12:43:29.354138: Step 4999: Cross entropy = 0.052652
I0310 12:43:29.354138  6400 retrain.py:977] 2020-03-10 12:43:29.354138: Step 4999: Cross entropy = 0.052652
INFO:tensorflow:2020-03-10 12:43:29.498751: Step 4999: Validation accuracy = 96.0% (N=100)
I0310 12:43:29.498751  6400 retrain.py:993] 2020-03-10 12:43:29.498751: Step 4999: Validation accuracy = 96.0% (N=100)
INFO:tensorflow:Final test accuracy = 96.8% (N=62)
I0310 12:43:32.523662  6400 retrain.py:1019] Final test accuracy = 96.8% (N=62)
```

Figure 10: Accuracy during training

As shown in the figure above, after 5000 training epochs, training accuracy is 99 %, cross entropy is 0.052652, validation accuracy is 96 % and final test accuracy is 96 %. These figures are calculated using the model.

6.1 Limitations

There are few limitations in the project as listed below:

- Since already seen through the output, the accuracy is very high. It lacks training data and epochs.
- The system is not able to process large no. of audios.
- UI for the system is not developed yet.

6.2 Future Enhancement

The following attributes will be considered in the future:

- Increase of accuracy through maximum training, high quality data.
- Maintain system to handle large no. of audios.
- Develop interactive user interface for the system

6.3 Conclusion

Speaker Recognition system has use in various sectors involving from security to health. Just like fingerprints, speaker recognition can be used in those sectors which do not need very tight security since fingerprints are identical but voices are not. Speaker can be recognized through the mathematical relation among frequency, pitch, and other parameters of sound but that would lead to more complex system. Hence, use of inception v3 and image classification has made this task lot easier. The main aim of this project is to learn to implement artificial intelligence in digital system. The system represents machine learning which is a part of AI.

References

[1] Wikipedia, “Spectrogram”, https://en.wikipedia.org/wiki/Spectrogram, [Accessed: Mar. 9, 2020]

[2] ResearchGate, “Inception Modules”, https://www.researchgate.net/figure/Convolutional-neural-network-architecture-Inception-v3-used-in-this-study-Inception-v3_fig3_328775405, [Accessed: Mar 7, 2020]

[3] GoogleCloud, Advanced guide to inception v3 on Cloud TPU, https://cloud.google.com/tpu/docs/inception-v3-advanced, [Accessed: Mar 9, 2020]