Automated Notes Maker from Audio Recordings

A

MAJOR PROJECT REPORT

Submitted by

Lakshay Chawla	Mehul Rekhi	Aditya Mehta
C.S.E Department	C.S.E Department	C.S.E Department
MAIT, Rohini	MAIT, Rohini	MAIT, Rohini
40814802718	35814802718	35114802718
lakshaychawla13@gmail.com	rekhi.mehul2000@gmail.com	delhi.adityamehta@gmail.com

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Under the Guidance

of

Mr. Ajay Tiwari

Assistant Professor, CSE

Department of Computer Science and Engineering

Maharaja Agrasen Institute of Technology,

PSP area, Sector – 22, Rohini, New Delhi – 110085

(Affiliated to Guru Gobind Singh Indraprastha, New Delhi)

(JUNE 2022)

MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering

CERTIFICATE

This is to Certified that this MAJOR project report "Automated Notes Maker from Audio Recordings" is submitted by "LakshayChawla (40814802718) , Aditya Mehta (35114802718) and Mehul Rekhi (35814802718) who carried out the project work under my supervision.

I approve this MAJOR project for submission.

Prof. Namita Gupta
(HoD, CSE)

Mr. Ajay Tiwari
(Assistant Professor, CSE)
(Project Guide)

ABSTRACT

"Voice is the basic, common, and efficient form of a communication method for people to interact with each other."

Today speech technologies are commonly available for a limited but interesting range of tasks. These technologies enable machines to respond correctly and reliably to human voices and provide useful and valuable services. As communicating with a computer is faster using voice rather than using a keyboard, people will prefer such a system.

Communication among human beings is dominated by spoken language, therefore it is natural for people to expect voice interfaces with computers. This can be accomplished by developing a voice recognition system - speech-to-text which allows the computer to translate voice requests and dictation into text. Voice recognition system - speech-to-text is the process of converting an acoustic signal which is captured using a microphone to a set of words. The recorded data can be used for document preparation.

In this project we aim to develop a computer program based on **Deep Learning Neural Networks** to make the computer understand speech commands and convert it into text.

ACKNOWLEDGEMENT

It gives me immense pleasure to express my deepest sense of gratitude and sincere thanks to my respected guide Mr. Ajay Tiwari (Assistant Professor , CSE) MAIT Delhi, for their valuable guidance, encouragement and help for completing this work. Their useful suggestions for this whole work and co-operative behavior are sincerely acknowledged.

I also wish to express my indebtedness to my parents as well as my family member whose blessings and support always helped me to face the challenges ahead.

Place: Delhi

Date:

Lakshay Chawla (40814802718)
Aditya Mehta (35114802718)
Mehul Rekhi (35814802718)

Table of Contents

List of Figures

List of Tables

INTRODUCTION

(fig.1)

Voice is the basic, common and efficient form of communication method for people to interact with each other. Today speech technologies are commonly available for a limited but interesting range of task. These technologies enable machines to respond correctly and reliably to human voices and provide useful and valuable services. As communicating with computer is faster using voice rather than using keyboard, so people will prefer such system. Communication among the human being is dominated by spoken language, therefore it is natural for people to expect voice interfaces with computer.

This can be accomplished by developing voice recognition system: speech-to-text which allows computer to translate voice request and dictation into text. Voice recognition system: speech-to-text is the process of converting an acoustic signal which is captured using a microphone to a set of words. The recorded data can be used for document preparation.

Classification of speech recognition system:

Speech recognition system can be classified in several different types by describing the type of speech utterance, type of speaker model and type of vocability that they have the ability to recognize.
The challenges are briefly explained below:

A. Types of speech utterance Speech recognition are classified according to what type of utterance they have ability to recognize.

   They are classified as:

   1) Isolated word: Isolated word recognizer usually requires each spoken word to have quiet (lack of an audio signal) on bot h side of the sample window. It accepts single word at a time.

   2) Connected word: It is similar to isolated word, but it allows separate utterances to „run-together“ which contains a minimum pause in between them.

   3) Continuous Speech: it allows the users to speak naturally and in parallel the computer will determine the content.

4) Spontaneous Speech: It is the type of speech which is natural sounding and is not rehearsed.

B. Types of speaker model Speech recognition system is broadly into two main categories based on speaker models namely speaker dependent and speaker independent.

1) Speaker dependent models: These systems are designed for a specific speaker. They are easier to develop and more accurate but they are not so flexible.

2) Speaker independent models: These systems are designed for variety of speaker. These systems are difficult to develop and less accurate but they are very much flexible.

C. Types of vocabulary The vocabulary size of speech recognition system affects the processing requirements, accuracy and complexity of the system. In voice recognition system: speech-to-text the types of vocabularies can be classified as follows:

1) Small vocabulary: single letter.
2) Medium vocabulary: two or three letter words.
3) Large vocabulary: more letter words

LITERATURE SURVEY

(fig.2)

Kuldip K. Paliwal and et al in the year 2004 had discussed that without being affected by their popularity for front end parameter in speech recognition, the cepstral coefficients which had been obtained from linear prediction analysis is sensitive to noise. Here, the use of spectral sub band centroids had been discussed by them for robust speech recognition. They discussed that performance of recognition can be achieved if the centroids are selected properly as in comparison with MFCC. to construct a dynamic centroid feature vector a procedure had been proposed which essentially includes the information of transitional spectral information [1].

Esfandier Zavarehei and et al in the year 2005, studied that a time-frequency estimator for enhancement of noisy speech signal in DFT domain is introduced. It is based on low order auto regressive process which is used for modelling. The time-varying trajectory of DFT

component in speech which has been formed in Kalman filter state equation. For restarting Kalman filter, a method has been formed to make alteration on the onsets of speech. The performance of this method was compared with parametric spectral subtraction and MMSE estimator for the increment of noisy speech. The resultant of the proposed method is that residual noise is reduced and quality of speech in improved using Kalman filters [2].

Kavita Sharma and Prateek Hakar in the year 2012 has represented recognition of speech in a broader solution. It refers to the technology that will recognize the speech without being targeted at single speaker. Variability in speech pattern, in speech recognition is the main problem. Speaker characteristics which include accent, noise and co-articulation are the most challenging sources in the variation of speech. In speech recognition system, the function of basilar membrane is copied in the front-end of the filter bank. To obtain better recognition results it is believed that the band subdivision is closer to the human perception. In speech recognition system the filter which is constructed for speech recognition is estimated of noise and clean speech [10].

Chadawan Ittichaichareon and Patiyuth Pramkeaw in the year 2012 had discussed that signal processing toolbox has been used in order to implement the low pass filter with finite impulse response. Computational implementation and analytical design of finite impulse response filter has been successfully accomplished by performing the performance evaluation at signal to noise ratio level. The results are improved in terms of recognition when low pass filters are used as compared to those process which involves speech signal without filtering [3].

Geeta Nijhawan, Poonam Pandit and Shivanker Dev Dhingra in the year 2013 had discussed the techniques of dynamic time warping and mel scale frequency cepstral coefficient in the isolated speech recognition. Different features of the spoken word had been extracted from the input speech. A sample of 5 speakers has been collected and each had spoken 10 digits. A database is made on this basis. Then feature has been extracted using MFCC.DTW is used for effectively dealing with various speaking speed. It is used for similarity measurement between two sequence which varies in speed and time [5].

Table of comparison -

(Table -1: Summarization of various methods applied for Speech-To-Text and Text-To- Speech conversion)

S No.	Techniques Used	Description
1.	TTS,STT Conversions and IVR	[1] They suggested that for STT conversion the audio message should first be recorded and then be converted to text form and for TTS conversion the text should be translated tothe audio and then play the audio message to the user. The proposed idea of an email system based on voice, makes use of 3 modules, namely: STT conversion, TTS conversion and IVR
2.	TTS,STT Conversions and IVR	[2] The proposed system focuses on providing user friendly platform to it users. The system implements Interactive-Voice-Response technology. In this system a pre-recorded voice will indicate the user to do some functions to avail some services.
3.	STT, TTS, Face recognition	[3] The paper proposes to develop a system that enables visually impaired, blind and people to use email facility as efficiently as some normal user. The dependency of the system on mouse or keyboard is almost diminished and it work on STT and TTS processes. Face Recognition is also used for authenticating the user identity.
4.	MFCC and HMM	[4] Proposed a STT system replacing traditional MFCC with HMM. The conventional MFCC approach was less efficient in extracting the features from the speech signals hence a new approach was suggested using HMM. The features passed to the HMM network resulted in better feature recognition from the input audio in contrast with the MFCC method. HMM exhibited vast improvement in the quality of feature extraction from the audio resulting in better computational time and accuracy for a Speech-To-Text conversion system.
5.	Automatic Speech Recognitio n, HMM model and human machine interface	The paper studied the deployment of STT by HMM and suggested to develop a machine interface system that depends on voice. The system could be deployed for helping 2 types of users: • People with disability who cannot access their email through use of mouse and keyboard, this category of users will be benefitted by the usage of a Speech-to-Text conversion system. • People who do not understand English or are not efficient in English and feel goodto communicate in their native language i.e. English, Punjabi, Hindi.
6.	Pattern Recognition, Neural Network, Artificial intelligence	They suggested a number of speech representation and classification methods. A number of feature extraction techniques were also deployed by them along with database evaluation and performance. The analyzed the various concerns related to Automatic-Speech- Recognition and proposed methods to resolve them. The various methods to speech recognition addressed by them are: the AI Approach, the pattern recognition Approach and acoustic phonetic approach.
7.	ANN and HMM	[6] Suggested rate of STT conversion can be made better using various techniques together and better-quality of text can be obtained. The objective is to develop a continuous STT system that has a much wider vocabulary and is speaker independent that can detect voice of different speakers with precision. For developing such a system, a combination of ANN and HMM will be used highly.

Comparison between different models –

(Table -2: The various models for Speech-To-Text Conversion)

METHOD	ADVANTAGE	DISADVANTAGE
Linear Predictive Coding (LPC	• LPC is a Static approach used for feature extraction. • The concept of LPC is that it can take the voice sample as linear combination combining past voice samples. • The voice signal is fragmented into N frames and then these framed windows are converted into text.	• Uses fixed resolution spectral analysis along witha subjective frequency scale.
Mel-Frequency Cestrum Co-efficient(MFCC)	• MFCC is another approach based on extracting features of signal by using filter bank. • The technique applies steps like Framing, Windowing and Discrete Fourier Transform for STT conversion.	• The problem with MFCC that it requires Normalization as values in MFCC are not very efficient in existence of surroundings or additivenoises.
Dynamic Time Wrapping	• The DTW algorithm is used to find the analogy in two-time series events that vary in speed by using dynamic programming. • Its purpose is to iterate the pair of sequence of feature vectors and finding a feasible match between them.	• The problem arises in selecting the reference template for comparing the time series events.
Hidden Markov Model	• HMM is a statistical model used for STT conversion. • HMM exhibits its own structure and self -learning which makes them very useful for STT conversion.	• In this method, the voice signal is seen as a static signal or short-term time static signal. • HMM is serial.
Neural Network	• Neural network is also a statistical model, represented as a graph. • Neural networks make use of connection functions values and connection strengths for the state transactions.	• Here in neural network model, ANN are parallel.
Hybrid Approach	• The proposed hybrid approach is used for Speech to Text conversion because speech frequencies are in parallel, whereas syllable series and words are in serial. • This shows that both the methods are useful indifferent context. Both HMM and Neural Networks techniques are implemented together. • As Neural networks show good performance in studying the probability from parallel voice input and Markov models can use the phoneme observation probabilities that neural networks provide to produce the possible phoneme sequence or word.	

(Table -3: The various approaches for Text-To- Speech conversion)

METHOD	ADVANTAGE	DISADVANTAGE
Rule Based Machine Translation (RBMT)	• RBMT makes use of syntactic and semantic analysis for conversion of text to speech. • The system is collection of Grammatical rules. • It performs a lookup of each word present in the input text consisting the Grammar and Dictionary base of the particular language to perform a TTS conversion.	• The RBMT is inefficient for big systems.
Statistical Machine Translation (SMT)	• SMT is probabilistic technique using Bayes Theorem that assigns each sentence in the input with a probability. • The more the value of probability the more is the efficiency in conversion of that sentence into the speech format.	• The disadvantage of this approach is the high cost involvement and it doesn't work well enough for different languages.
Hidden Markov Model(HMM	• HMM is a probabilistic technique similar to SMT but given better accuracy for TTS conversion. • HMM can be deployed for both voice recognition systems and also text-to-speech synthesis systems to generate an audio signal from text input. • The advantage of adopting HMM is, it is an automatically trained network.	

By analyzing the various methods for STT and TTS we have examined that HMM provides maximum efficiency for STT and TTS conversion. Also an optimal amount of efficiency is provided by neural network for STT. Hence we have proposed the Hybrid approach for STT conversion that deploys HMM and Neural network and for TTS, the HMM model provides the highest accuracy in comparison to others.

Requirement Analysis

1. PYTHON –

(fig.3)

Python is the language used for machine learning and model training. The main functioning of the form is dependent on this part.

2. Librosa –

(fig.4)

Librosa is a Python package for music and audio analysis. Librosa is basically used when we work with audio data like in music generation (using LSTM's), Automatic Speech Recognition. It provides the building blocks necessary to create the music information retrieval systems.

3. Tensorflow –

(fig.5)

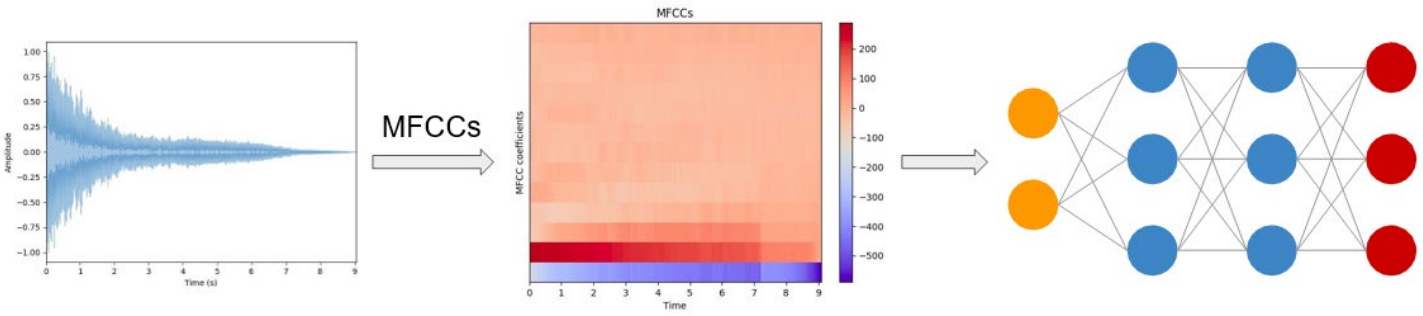
TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.

4. Tensor board –

(fig.6)

Tensor Board is a tool for providing the measurements and visualizations needed during the machine learning workflow. It enables tracking experiment metrics like loss and accuracy, visualizing the model graph, projecting embeddings to a lower dimensional space, and much more.

SYSTEM DESIGN / ARCHITECTURE



(fig.7)

A. Raw data

We have a dataset of recordings under various labels which can be used to train a neural network.

To be specific, we have the following 31 labels:

i. bed
ii. bird
iii. cat
iv. dog
v. down
vi. eight
vii. five
viii. four
ix. go
x. happy
xi. house
xii. left
xiii. marvin
xiv. nine
xv. no
xvi. off
xvii. on
xviii. one
xix. right
xx. seven
xxi. sheila
xxii. six
xxiii. stop
xxiv. three
xxv. tree
xxvi. two
xxvii. up
xxviii. wow
xxix. yes
xxx. zero
xxxi. _background_noise_

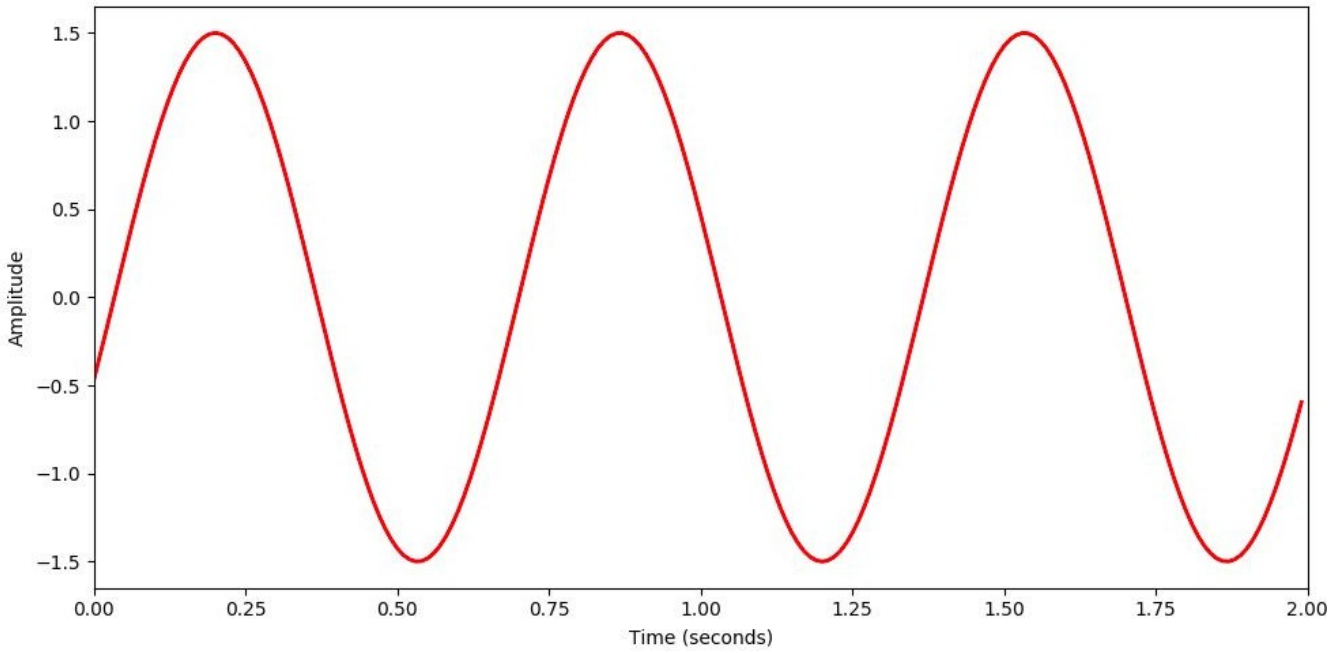
B. Pre-processing and Feature Engineering

Understanding how to process audio

To start with processing, we need to first understand what audio signal is.

- Produced by the vibration of an object
- Vibrations determine oscillation of air molecules
- Alternation of air pressure causes a wave

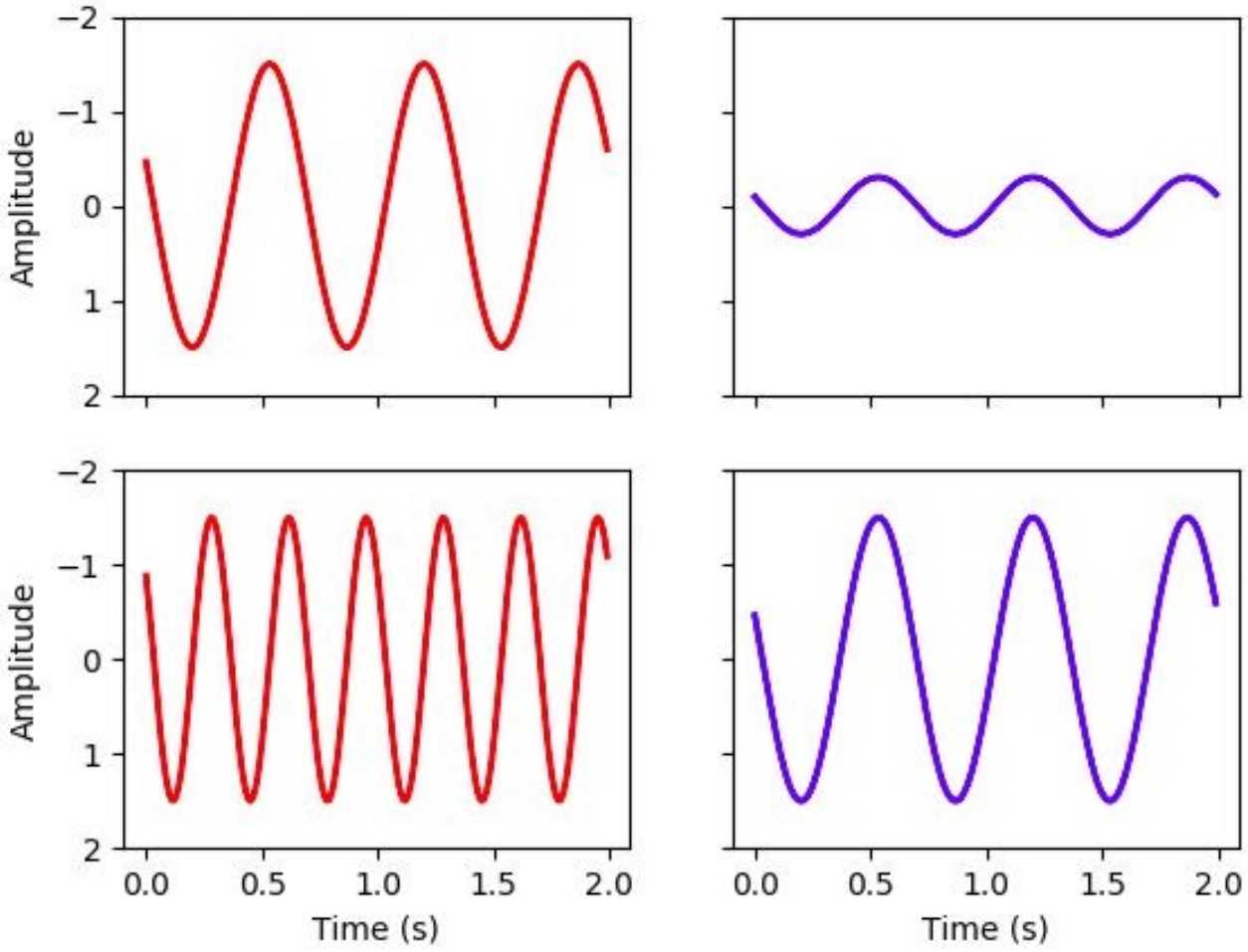
Waveform



(fig.8)

$$y(t) = A\sin(2\pi ft + \varphi)$$

Frequency/pitch and amplitude/loudness



(fig.9)

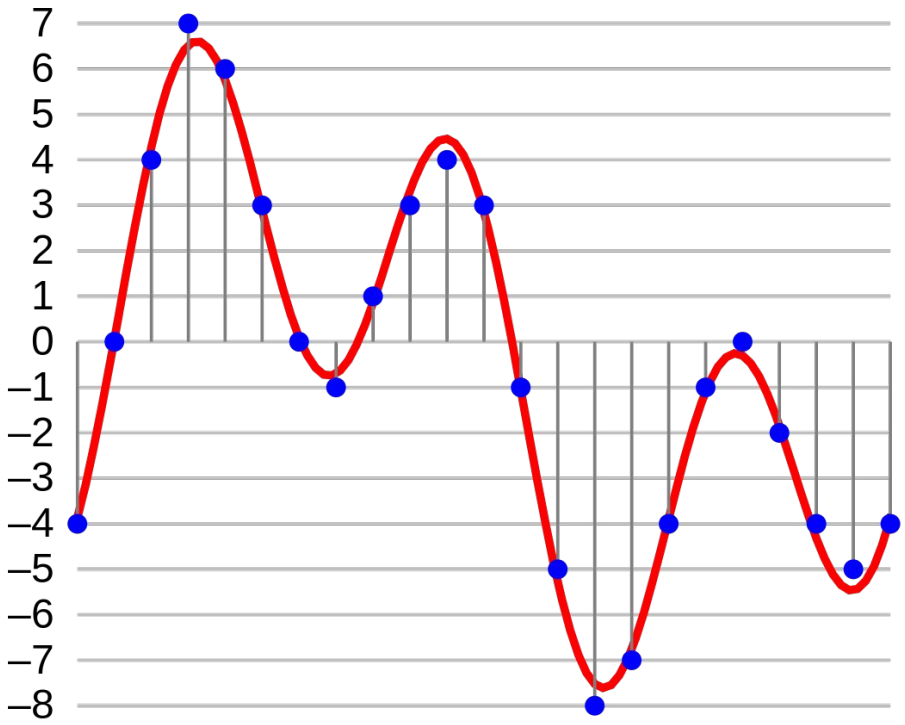
Higher Frequency -> Higher Pitch

Larger Amplitude -> Louder

Frequency and Amplitude determines pitch and loudness of the sound respectively.

We need to get essence of amplitude as well as frequency to be able to process sound in a meaningful way.

Analog to Digital Conversion

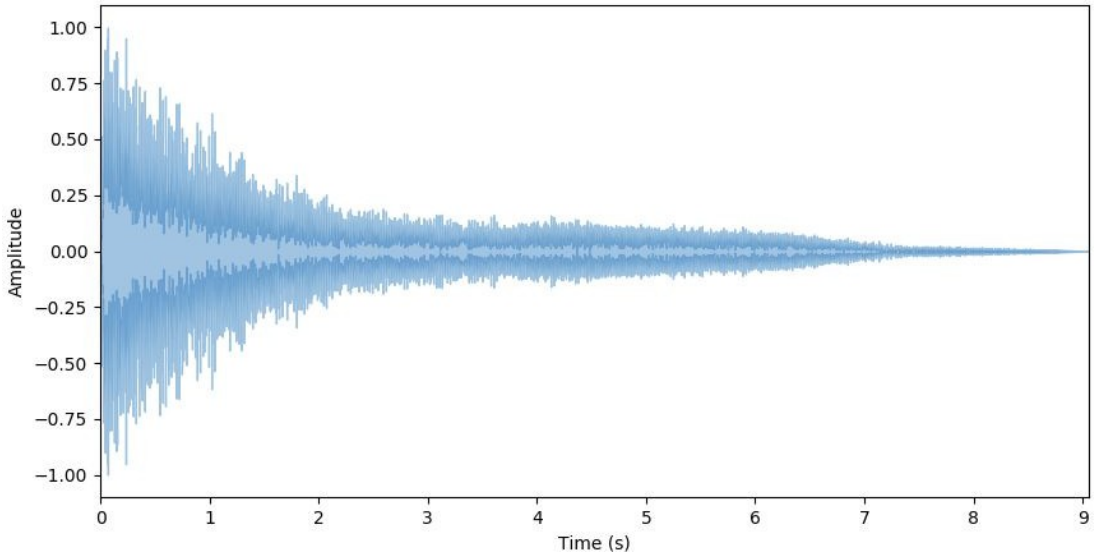


(fig.10)

When audio signal that is inherently analog has to be processed, we need to convert it to Digital signal first.
To convert it to digital, the signal is sampled at uniform time intervals. The number of times it is sampled each second is known as Sample Rate.

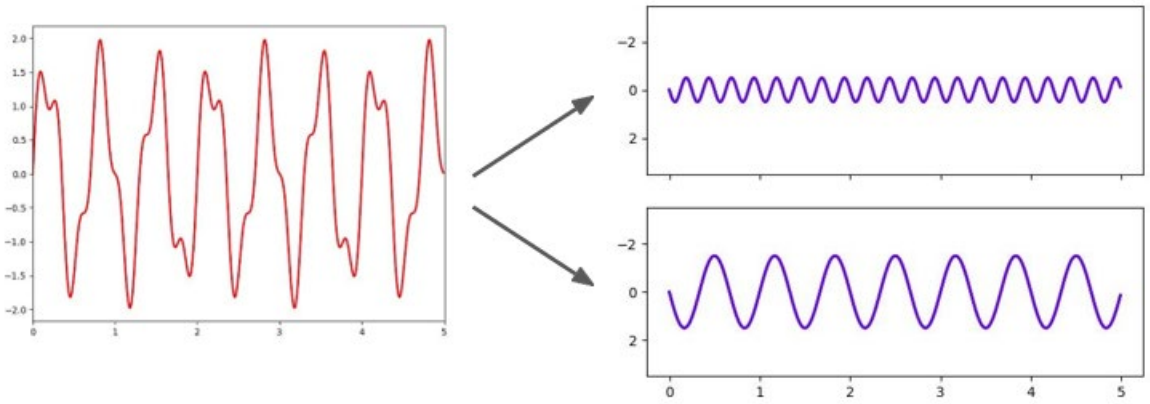
Conversion of amplitude of the signal takes place with limited number of bits. The number of bits used for amplitude quantization is known as Bit Depth.
A typical CD album has a Sample Rate = 44,100 Hz and Bit Depth = 16.

A real world piano key sound wave



(fig.11)

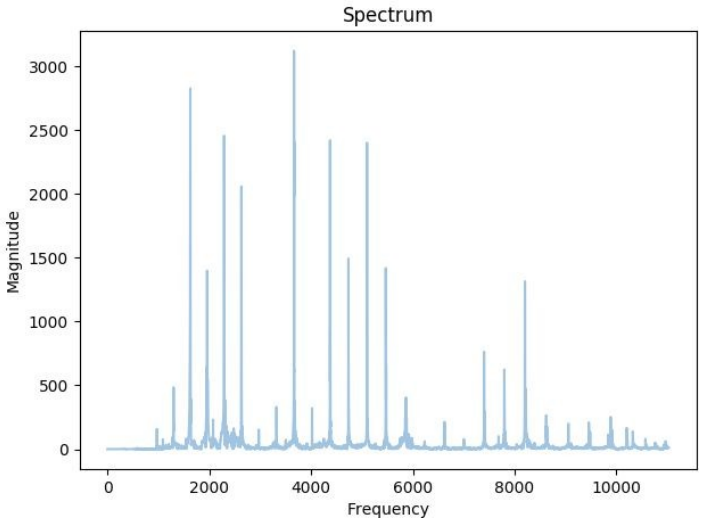
As we can see the sound wave on its own looks pretty messy and it is really difficult to make out different frequencies. To get frequency picture of the sound wave we need to perform a fourier transform.
Fourier transform decomposes complex periodic sound into sum of sine waves oscillating at different frequencies.



(fig.12)

$$s = A_1 \sin(2\pi f_1 t + \varphi_1) + A_2 \sin(2\pi f_2 t + \varphi_2)$$

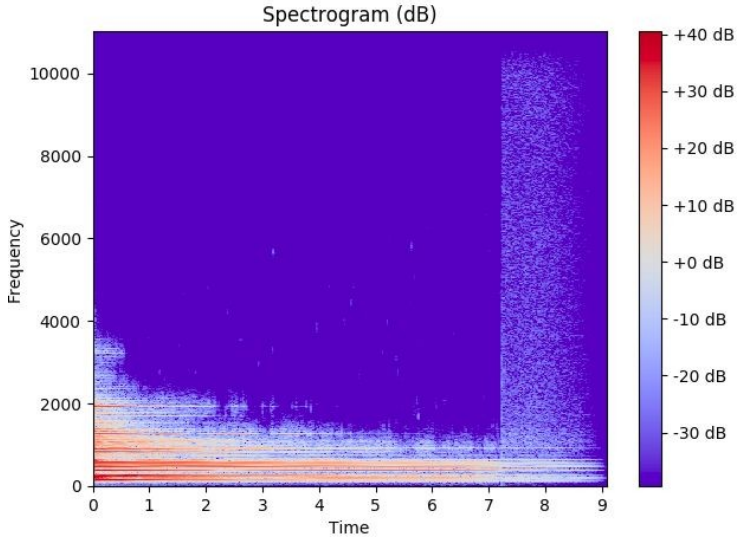
After fourier transform, the sound wave gets changed from time domain to frequency domain.



(fig.13)

STFT – SHORT TIME FOURIER TRANFORM

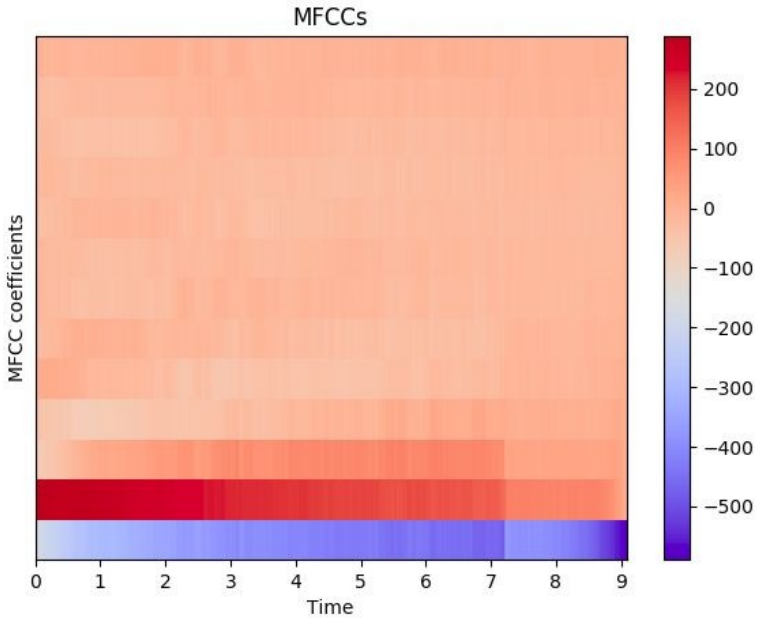
- Computes several FFT at different intervals
- Preserves time information
- Fixed frame size (e.g., 2048 samples)
- Gives a *spectrogram* (time + frequency + magnitude)



(fig.14)

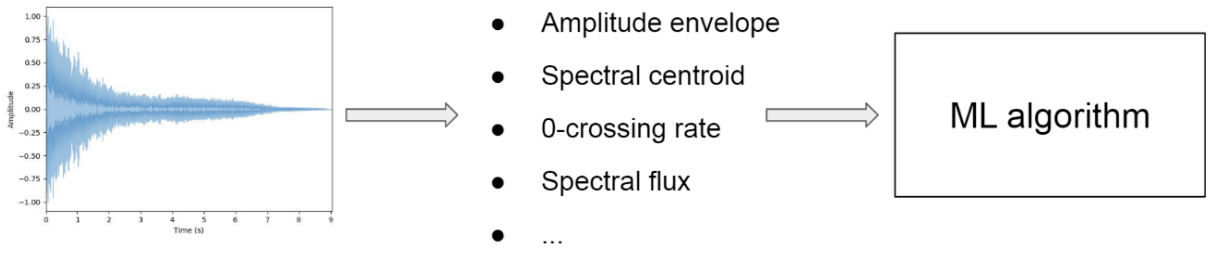
MFCC – MEL FREQUENCY CEPSTRAL COEFFICIENTS

- Capture timbral/textural aspects of sound
- Frequency domain feature
- Approximate human auditory system
- 13 to 40 coefficients
- Calculated at each frame



(fig.15)

Traditional ML pre-processing pipeline for audio data



(fig.16)

- Feature engineering
- Perform STFT
- Extract time + frequency domain features

Tradition ML relied heavily on feature engineering. Initially, we needed to extract features from the soundwave like amplitude envelope or spectral centroid. So, we start from a waveform and go back to extract features from FFTs and STFTs to perform algorithms like KNN, Logistic Regression or SVMs.
Deep Learning helps us skip the feature engineering part as the neural network learns the features on itself and it enables us to use more advanced visualization (MFCCs).

```
train_audio_path = '../MAJOR PROJECT/tensorflow-speech-recognition-challenge/train/audio/'
samples, sample_rate = librosa.load(train_audio_path+'yes/0a7c2a8d_nohash_0.wav', sr = 16000)
fig = plt.figure(figsize=(14, 8))
ax1 = fig.add_subplot(211)
ax1.set_title('Raw wave of ' + '../input/train/audio/yes/0a7c2a8d_nohash_0.wav')
ax1.set_xlabel('time')
ax1.set_ylabel('Amplitude')
ax1.plot(np.linspace(0, sample_rate/len(samples), sample_rate), samples)
```

```
#find count of each label and plot bar graph
no_of_recordings=[]
for label in labels:
    waves = [f for f in os.listdir(train_audio_path + '/'+ label) if f.endswith('.wav')]
    no_of_recordings.append(len(waves))

#plot
plt.figure(figsize=(30,5))
index = np.arange(len(labels))
plt.bar(index, no_of_recordings)
plt.xlabel('Commands', fontsize=12)
plt.ylabel('No of recordings', fontsize=12)
plt.xticks(index, labels, fontsize=15, rotation=60)
plt.title('No. of recordings for each command')
plt.show()
```

```
duration_of_recordings=[]
for label in model_labels:
    waves = [f for f in os.listdir(train_audio_path + '/'+ label) if f.endswith('.wav')]
    for wav in waves:
        sample_rate, samples = wavfile.read(train_audio_path + '/' + label + '/' + wav)
        duration_of_recordings.append(float(len(samples)/sample_rate))

plt.hist(np.array(duration_of_recordings))
```

```
train_audio_path = '../MAJOR PROJECT/tensorflow-speech-recognition-challenge/train/audio'

all_wave = []
all_label = []
for label in model_labels:
    print(label)
    waves = [f for f in os.listdir(train_audio_path + '/'+ label) if f.endswith('.wav')]
    for wav in waves:
        samples, sample_rate = librosa.load(train_audio_path + '/' + label + '/' + wav, sr = 16000)
        samples = librosa.resample(samples, sample_rate, 8000)
        if(len(samples)== 8000) :
            all_wave.append(samples)
            all_label.append(label)
```

(fig.17)

C. Neural Networks

A neural network works similarly to the human brain's neural network. A "neuron" in a neural network is a mathematical function that collects and classifies information according to a specific architecture. The network bears a strong resemblance to statistical methods such as curve fitting and regression analysis.

A neural network contains layers of interconnected nodes. Each node is a known as perceptron and is similar to a multiple linear regression. The perceptron feeds the signal produced by a multiple linear regression into an activation function that may be nonlinear.

Multi-Layered Perceptron

In a multi-layered perceptron (MLP), perceptron's are arranged in interconnected layers. The input layer collects input patterns. The output layer has classifications or output signals to which input patterns may map. For instance, the patterns may comprise a list of quantities for technical indicators about a security; potential outputs could be "buy," "hold" or "sell."

Hidden layers fine-tune the input weightings until the neural network's margin of error is minimal. It is hypothesized that hidden layers extrapolate salient features in the input data that have predictive power regarding the outputs. This describes feature extraction, which accomplishes a utility similar to statistical techniques such as principal component analysis.

Application of Neural Networks

Neural networks are broadly used, with applications for financial operations, enterprise planning, trading, business analytics, and product maintenance. Neural networks have also gained widespread adoption in business applications such as forecasting and marketing research solutions, fraud detection, and risk assessment.

What Are the Components of a Neural Network?

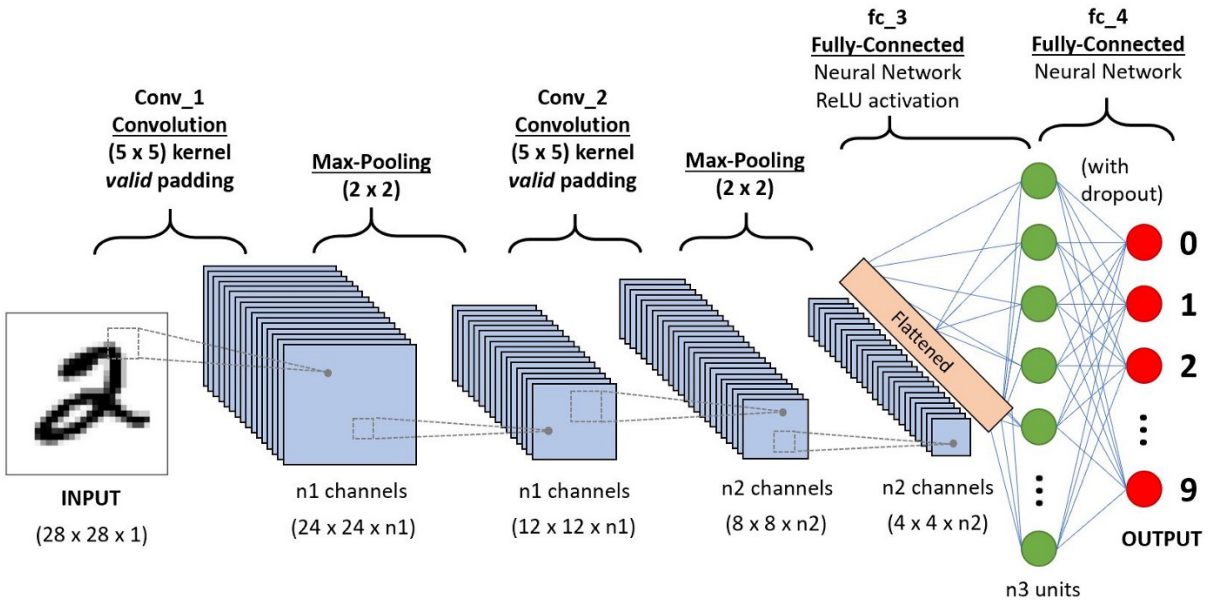
There are three main components: an input later, a processing layer, and an output layer. The inputs may be weighted based on various criteria. Within the processing layer, which is hidden from view, there are nodes and connections between these nodes, meant to be analogous to the neurons and synapses in an animal brain.

Convolutional Neural Network.

Introduction

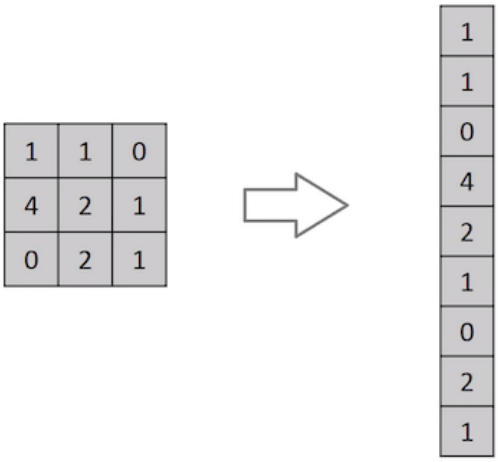
A CNN sequence to classify handwritten digits
A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.



(fig.19)

Why ConvNets over Feed-Forward Neural Nets?



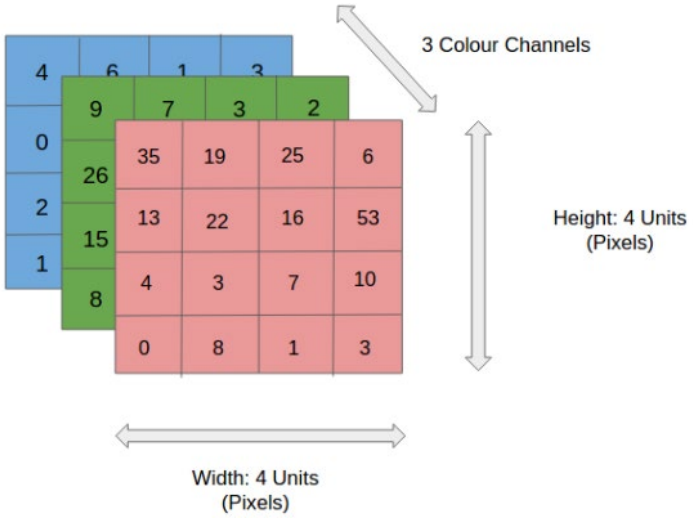
(fig.20)

An image is nothing but a matrix of pixel values, right? So why not just flatten the image (e.g. 3x3 image matrix into a 9x1 vector) and feed it to a Multi-Level Perceptron for classification purposes? Uh.. not really.

In cases of extremely basic binary images, the method might show an average precision score while performing prediction of classes but would have little to no accuracy when it comes to complex images having pixel dependencies throughout.

A ConvNet is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

Input Image

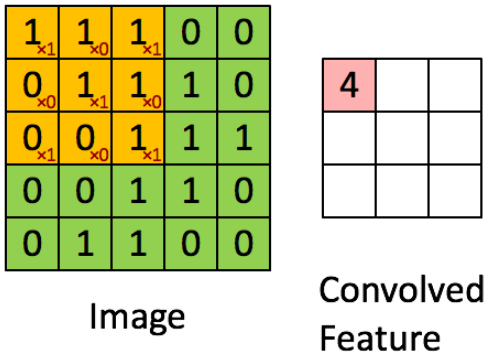


(fig.21)

In the figure, we have an RGB image which has been separated by its three color planes — Red, Green, and Blue. There are a number of such color spaces in which images exist — Grayscale, RGB, HSV, CMYK, etc.

You can imagine how computationally intensive things would get once the images reach dimensions, say 8K (7680×4320). The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction. This is important when we are to design an architecture which is not only good at learning features but also is scalable to massive datasets.

Convolution Layer — The Kernel

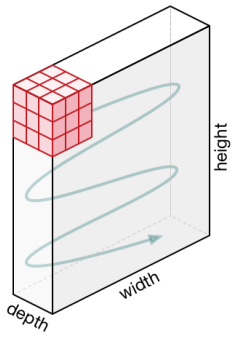


(fig.22)

Image Dimensions = 5 (Height) x 5 (Breadth) x 1 (Number of channels, eg. RGB)

In the above demonstration, the green section resembles our 5x5x1 input image, I. The element involved in carrying out the convolution operation in the first part of a Convolutional Layer is called the Kernel/Filter, K, represented in the color yellow. We have selected K as a 3x3x1 matrix.

Kernel/Filter, K =
1 0 1
0 1 0
1 0 1
The Kernel shifts 9 times because of Stride Length = 1 (Non-Strided), every time performing a matrix multiplication operation between K and the portion P of the image over which the kernel is hovering.



(fig.23)

The filter moves to the right with a certain Stride Value till it parses the complete width. Moving on, it hops down to the beginning (left) of the image with the same Stride Value and repeats the process until the entire image is traversed.

Convolution operation on a MxNx3 image matrix with a 3x3x3 Kernel
In the case of images with multiple channels (e.g. RGB), the Kernel has the same depth as that of the input image. Matrix Multiplication is performed between Kn and In stack ([K1, I1]; [K2, I2]; [K3, I3]) and all the results are summed with the bias to give us a squashed one-depth channel Convoluted Feature Output.

There are two types of results to the operation — one in which the convolved feature is reduced in dimensionality as compared to the input, and the other in which the dimensionality is either increased or remains the same.

When we augment the 5x5x1 image into a 6x6x1 image and then apply the 3x3x1 kernel over it, we find that the convolved matrix turns out to be of dimensions 5x5x1. Hence the name — Same Padding.

On the other hand, if we perform the same operation without padding, we are presented with a matrix which has dimensions of the Kernel (3x3x1) itself — Valid Padding.

The following repository houses many such GIFs which would help you get a better understanding of how Padding and Stride Length work together to achieve results relevant to our needs.

Pooling Layer
Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model.

There are two types of Pooling: Max Pooling returns the maximum value from the portion of the image covered by the Kernel.Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.

Max Pooling also performs as a Noise Suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling.

Types of Pooling
The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-levels details even further, but at the cost of more computational power.

D. Result Analysis

While testing for just a single word conversion, we got accuracy up to 83.2 %.

Dataset	Number of Samples
Training set	17049
Validation Set	4263

(fig.25)

Dataset	Accuracy
Training set	89.24%
Validation Set	83.2%

(fig.26)

Statement	Time taken by project	Time Taken by google API
One dog on the tree house	0:00:00.001013	0:00:00.010130
Stop, go right	0:00:00.000993	0:00:00.014895

(fig.27)

E. Final Deployment

After the single word conversions, we implemented a piece of code that breaks down any audio to 1 second chunks and predicts on them. This helps us to avoid using NLPs that complicate the project and are too computationally expensive.

When comparing to Google's api, our times are significantly better.

Although our accuracy is a little lower, we can easily make a pass on that because most of the errors are in small words which are classified as Background noise and never misclassified.

Conclusion

The aim of the project was accomplished. Without using any of the pre-existing speech to text conversion tools, we were able to convert the recordings to text with a decent amount of accuracy and faster than Google's own solution.

For the future scope of the project, we can make the input stream of the project live through a mic instead of recordings to make this project a live captioning project and implement more languages.

References

[1] Jingdong Chen, Member, Yiteng (Arden) Huang, Qi Li, Kuldip K. Paliwal, "Recognition of Noisy Speech using Dynamic Spectral Subband Centroids" IEEE SSIGNAL PROCESSING LETTERS, Vol. 11, Number 2, February 2004.

[2] Hakan Erdogan, Ruhi Sarikaya, Yuqing Gao, "Using semantic analysis to improve speech recognition performance" Computer Speech and Language, ELSEVIER 2005.

[3] Chadawan Ittichaichareon, Patiyuth Pramkeaw, "Improving MFCC-based Speech Classification with FIR Filter" International Conference on Computer Graphics, Simulation and Modelling (ICGSM"2012) July 28-29, 2012 Pattaya(Thailand).

[4] Bhupinder Singh, Neha Kapur, Puneet Kaur "Sppech Recognition with Hidden Markov Model:A Review" International Journal of Advanced Research in Computer and Software Engineering, Vol. 2, Issue 3, March 2012.

[5] Shivanker Dev Dhingra, Geeta Nijhawan, Poonam Pandit, "Isolated Speech Recognition using MFCC and DTW" International Journal of Advance Research in Electrical, Electronics and Instrumentation Engineering, Vol.2, Issue 8, August 2013.

[6] Ibrahim Patel, Dr. Y. Srinivas Rao, "Speech Recognition using HMM with MFCC-an analysis using Frequency Spectral Decomposition Technique" Signal and Image Processing:An International Journal(SIPIJ), Vol.1, Number.2, December 2010.

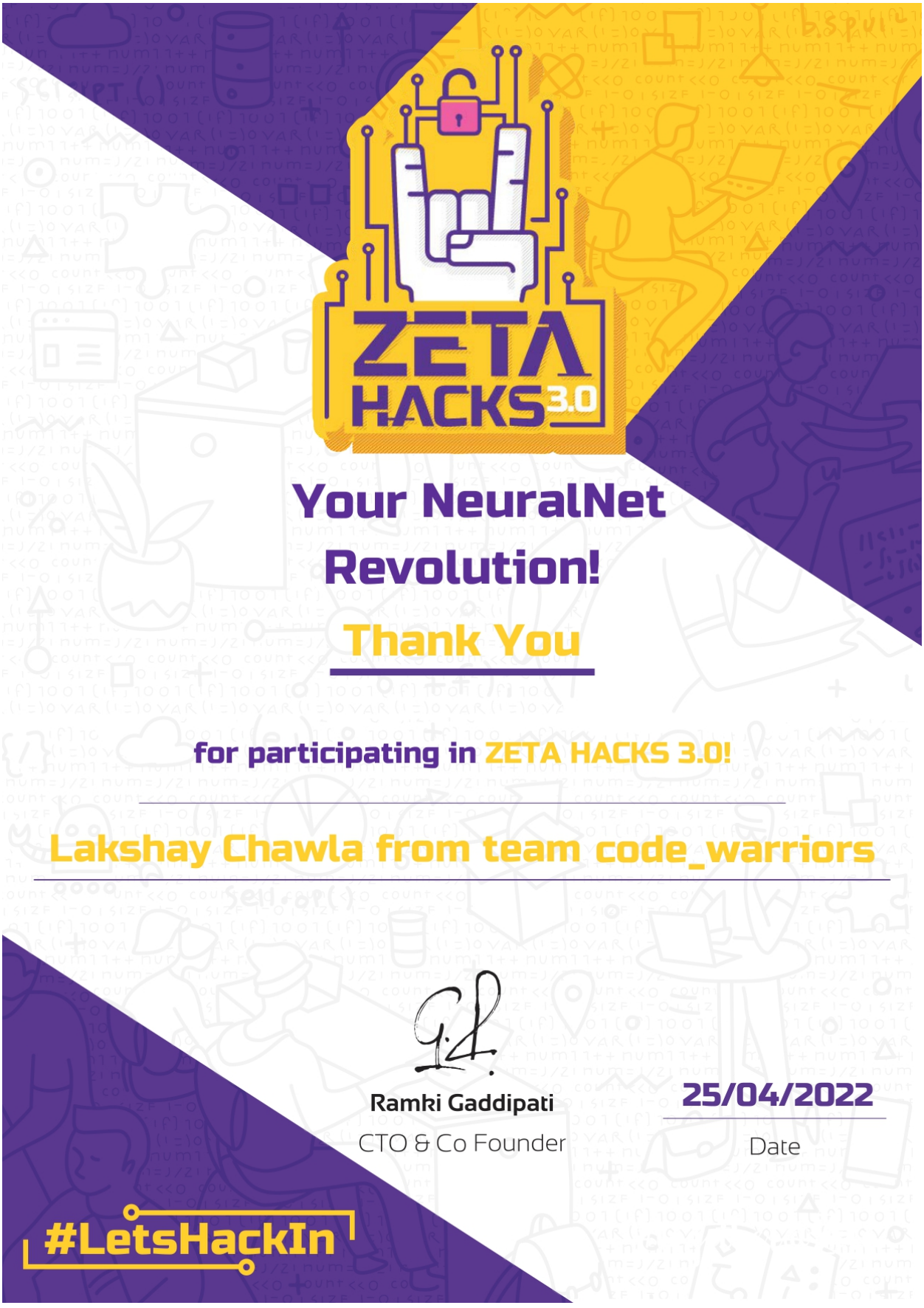
[7] Om Prakash Prabhakar, Navneet Kumar Sahu,"A Survey on Voice Command Recognition Technique" International Journal of Advanced Research in Computer and Software Engineering, Vol 3,Issue 5,May 2013.

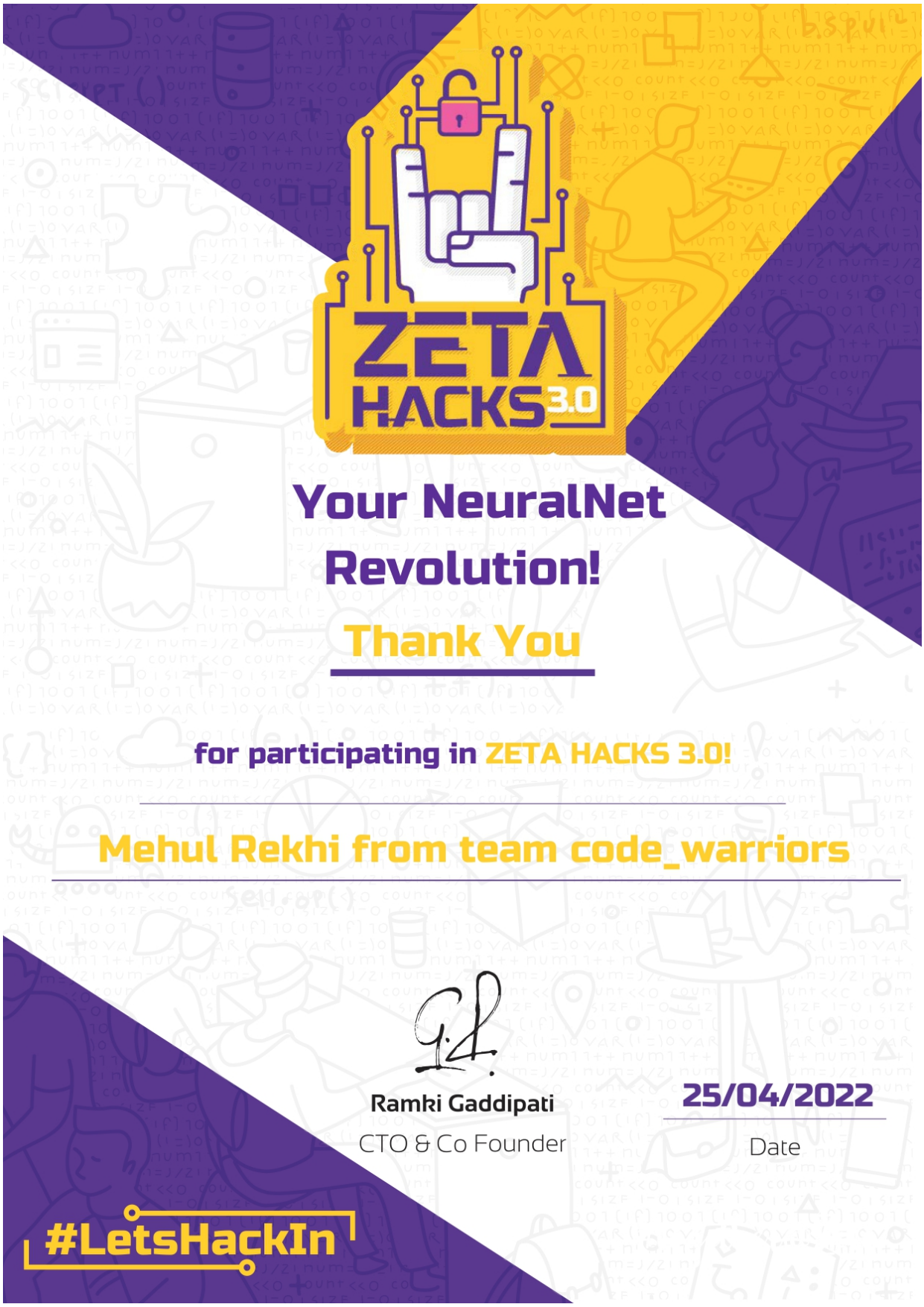
[8] M A Anusuya, "Speech recognition by Machine", International Journal of Computer Science and Information security, Vol. 6, number 3,2009.

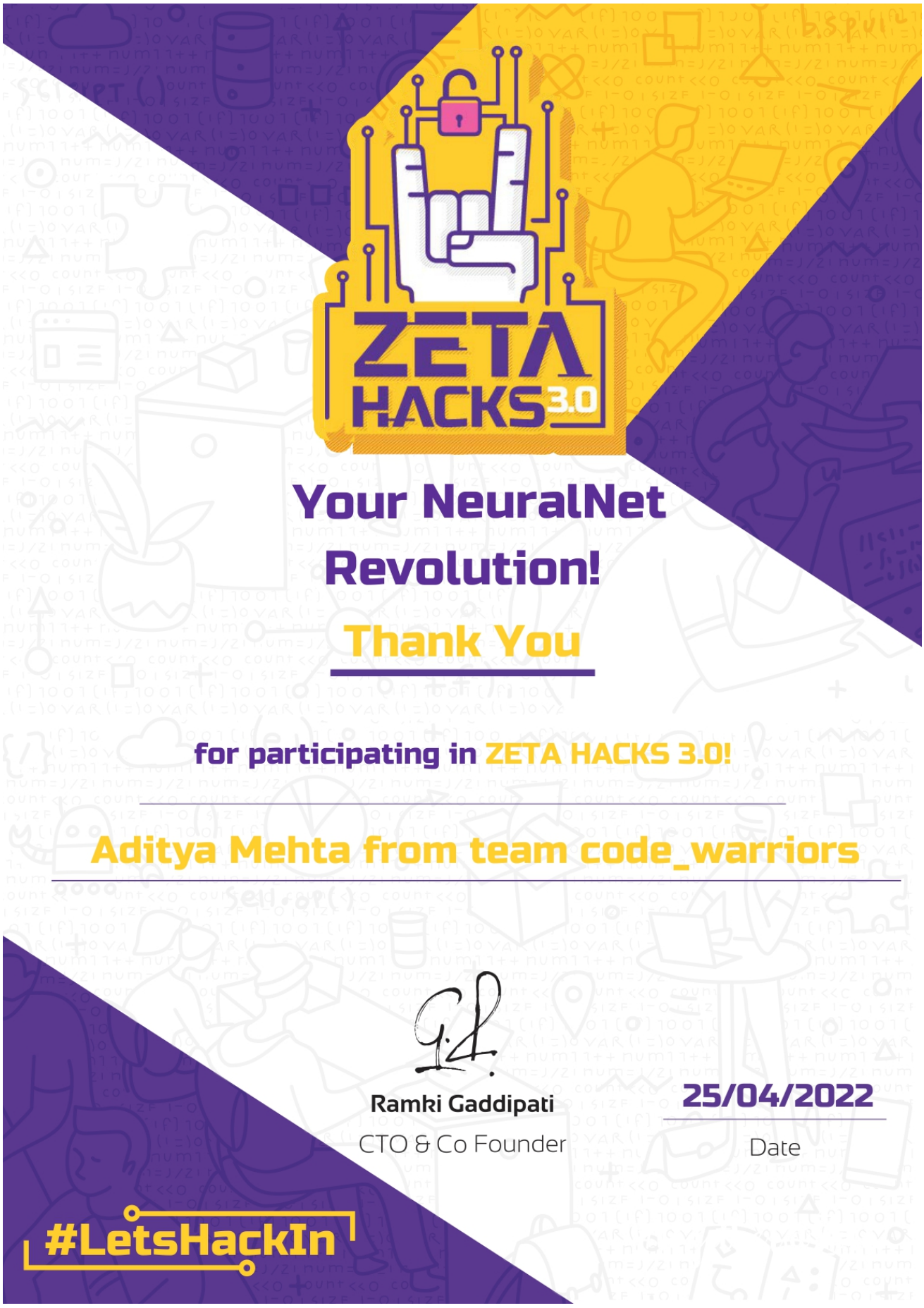
[9] Sikha Gupta, Jafreezal Jaafar, Wan Fatimah wan Ahmad, Arpit Bansal, "Feature Extraction Using MFCC" Signal & Image Processing:An International Journal, Vol 4, No. 4, August 2013.

[10] Kavita Sharma, Prateek Hakar "Speech Denoising Using Different Types of Filters" International journal of Engineering Research and Applications Vol. 2, Issue 1, Jan-Feb 2012

Proof of Hackathon Participation




ZETA HACKS 3.0
Your NeuralNet Revolution!
Thank You
for participating in ZETA HACKS 3.0!
Mehul Rekhi from team code_warriors
Ramki Gaddipati
CTO & Co Founder
25/04/2022
Date
#LetsHackIn


ZETA HACKS 3.0
Your NeuralNet Revolution!
Thank You
for participating in ZETA HACKS 3.0!
Aditya Mehta from team code_warriors
Ramki Gaddipati
CTO & Co Founder
25/04/2022
Date
#LetsHackIn