# Welcome to the IBM Developer DDC: Data & AI

## Speech Synthesis by Using Advanced Machine Learning Techniques for Easy Readabllity of Dyslexic Children

**Speaker:** Geeta Atkar

Assistant professor, G H Raisoni College of Engineering & Management, Pune, India

# Welcome to the IBM Developer DDC: Data & AI

## Speech Synthesis by Using Advanced Machine Learning Techniques for Easy Readabllity of Dyslexic Children

**Speaker:** Geeta Atkar

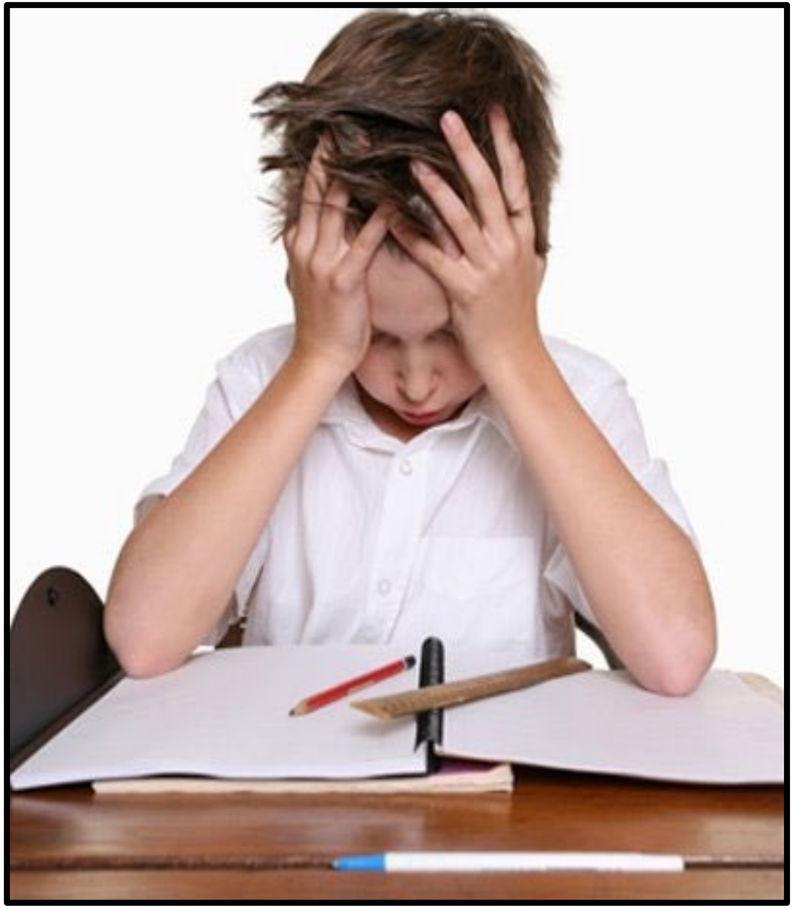
**Agenda:**

1. Dyslexia in Children
2. Dataset
3. Generative Adversarial Networks(GAN)
   1. WaveGAN
   2. MelGAN
4. Results
5. Conclusion

# Dyslexia in Children

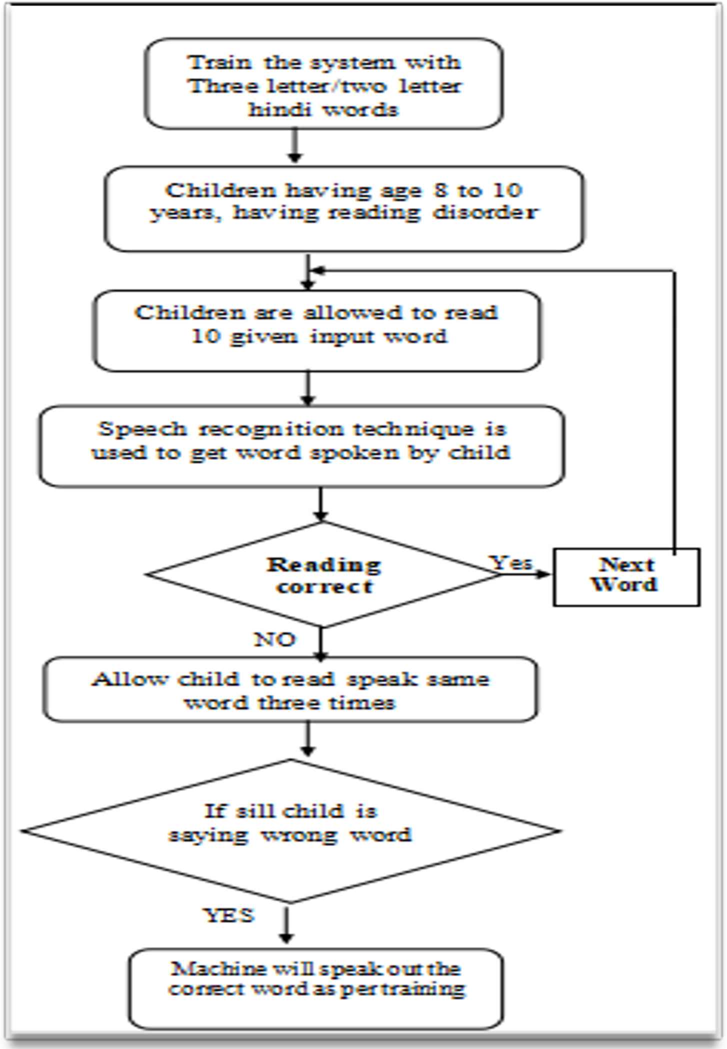
**Dys+Lexic**

- Learning Disability
- Easy Readability
- Children of age 6 to 10 Years
- Speech Synthesis



# General Architecture

1. Take two and three letters words as an input
2. By using Speech synthesis Techniques, generate multiple speech of every single word.
3. Train the system with all those words
4. Child is allowed to read the word displayed on screen
5. Once Children Reads word, speech recignition technique is used to get word spoken by children
6. If he/she reads correct word, give him/her next word
7. If he reads wrongly, allow child to speak same words three times .
8. If still he/she is reading wrong word, mine will speak out the word


Train the system with Three letter/two letter hindi words
Children having age 8 to 10 years, having reading disorder
Children are allowed to read 10 given input word
Speech recognition technique is used to get word spoken by child
Reading correct
Yes
Next Word
NO
Allow child to read speak same word three times
If sill child is saying wrong word
YES
Machine will speak out the correct word as pertraining

# Speech Synthesis DataSet

The dataset used for testing WaveGAN is SC09, or Speech Commands Zero through Nine.

The dataset used is a subset of the actual dataset built by Google in 2017

The dataset consists of 65,000 clips of one-second-long duration. Each clip contains 30 different words spoken by thousands of different people.

.

The clips were recorded in realistic environments with the help of phones and laptop

# Simple Generative Adversarial Network

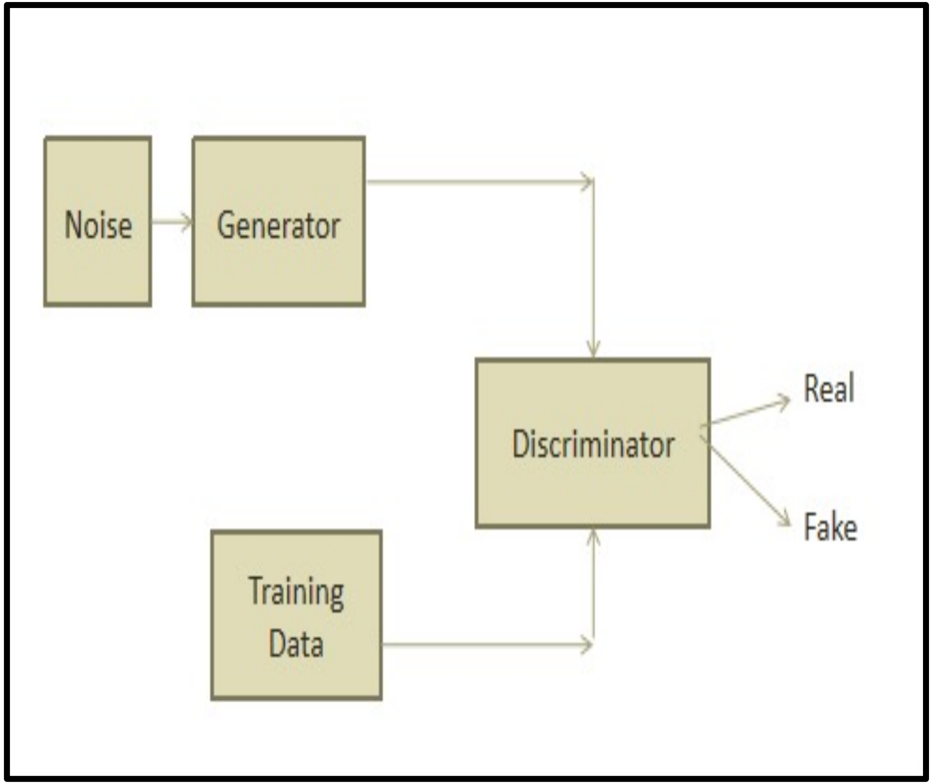
Step 1: Train Discriminator by using training data
.

Step 2: Take random inputs from training data and introduce to noise

Step 3: Generator takes random noise and tries to reconstruct input

Step 4: Discriminator takes input from two sources real data and generator  and classifies Real and Fake.

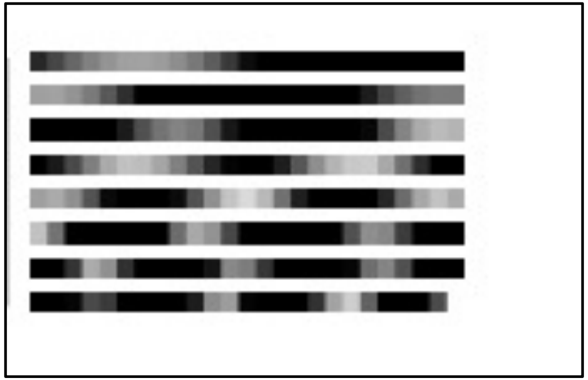
Step 5:Error is computed



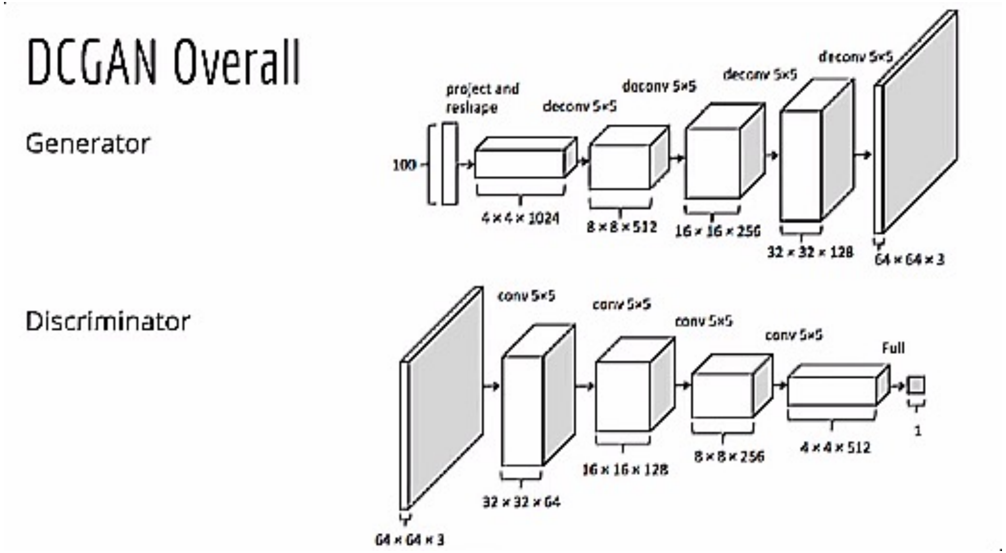
**Simple GAN**

# Case Option 1 (Wave GAN)

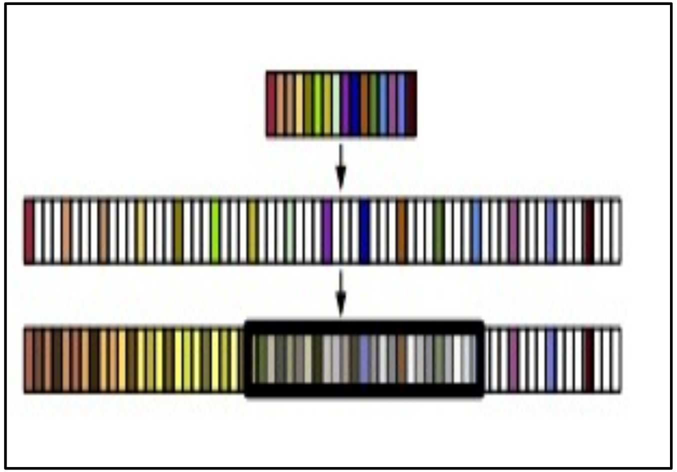
- To overcome issues of DCGAN, WaveGAN is used with following changes:
- Audio slices from speech having fundamental Structure
- WaveGAN uses one-dimensional filters of length 25
- Larger upsampling factor ie 4
- Overlapping Frequencies Issue: Phase Shuffling is used in WaveGAN



**1 D filters**



**General DC GAN Architecture**

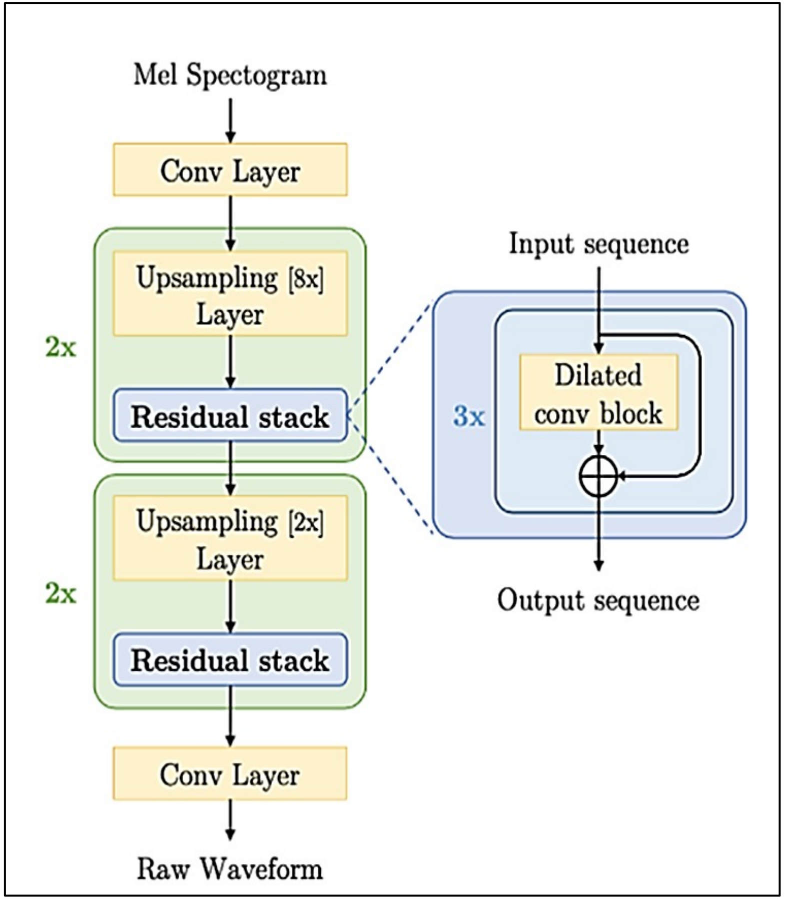


**Phase Shuffling**

# Case Option 2 (Mel GAN)

## MegGAN Generator Architecture

1. MelGAN Generator uses three network structure Architecture
2. Generator has convolutional layer and two upsampling layers
3. Each convolution layer is followed by residuel blocks which has dilated convolutions
4. Generator used Mel-Spectrogram as an input instead of random noise
5. Leaky Relu function is used as an activation function

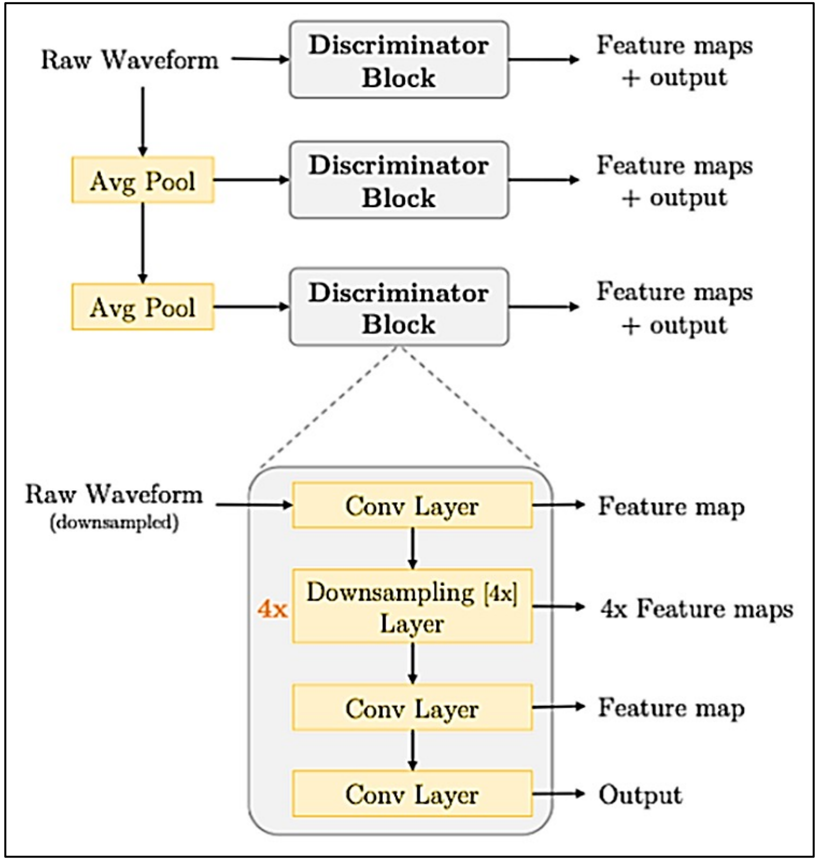


**MelGAN Generator**

# Case Option 2 (Mel GAN)

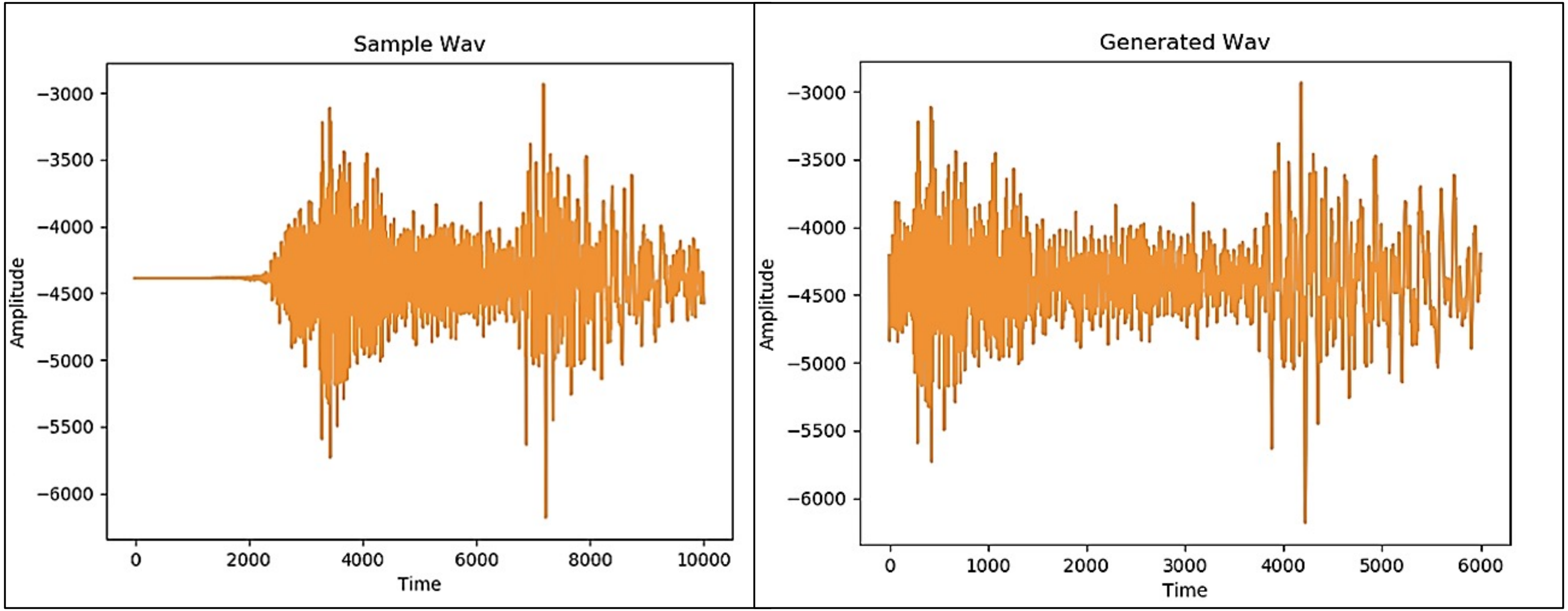
## MelGAN Discrimonator Architecture

1. MelGAN Discriminator has three Discriminator Blocks D1,D2, D3
2. D1 takes raw audio waveform, after average pooling, it is sent to second and third Discriminator.
3. D2,D3 operates on audio, which is downsampled by factor of 2 and 4
4. Each discriminator has convolutional layer, 4 downsampling layer and feature map layer
5. It uses leaky Relu activation Function



**MelGAN Discriminator**

# Original and Generated WaveForm by MELGAN

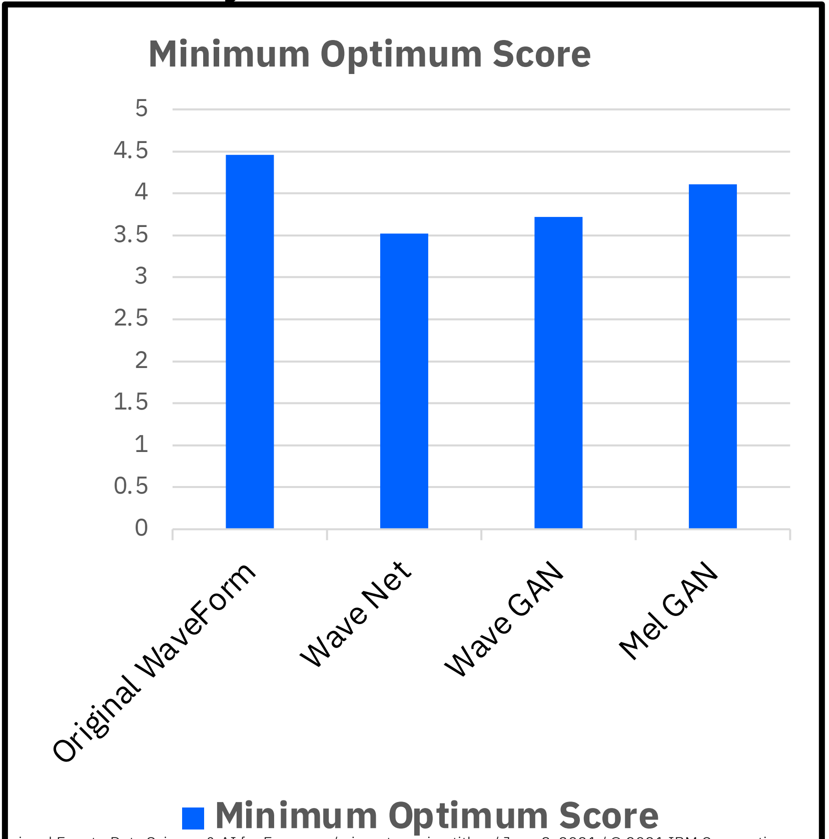


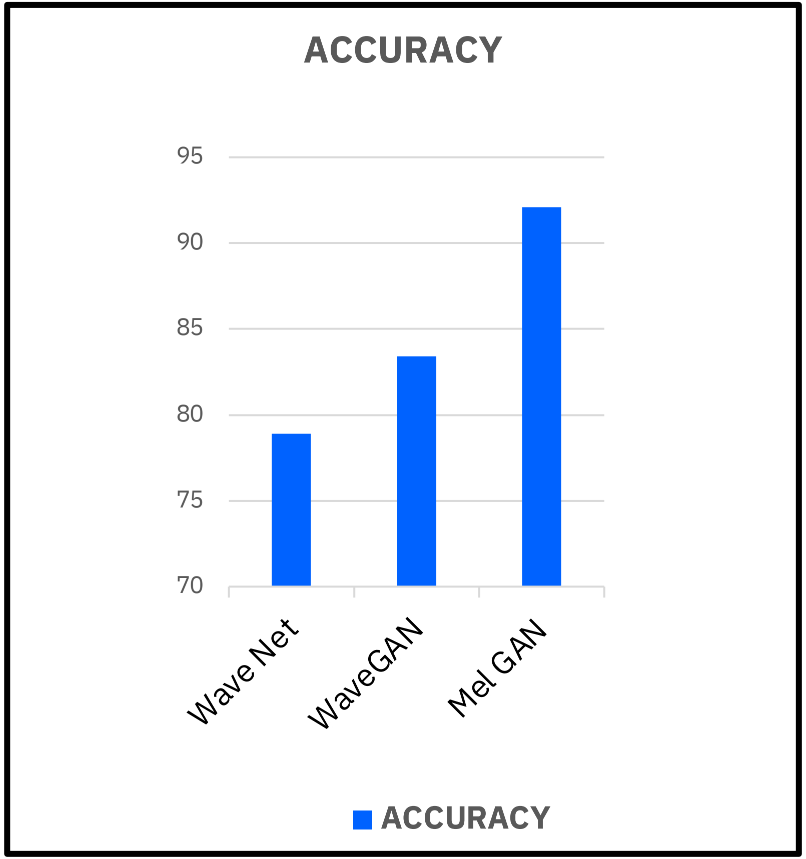
# Comparative Study (Minimum Optimum Score & Accuracy)

Original MOS Score is 4.46

| MODEL | MOS | ACCURACY (%) |
|---|---|---|
| Wave Net | 3.52 | 78.9 |
| WaveGAN | 3.72 | 83.4 |
| MelGAN | 4.11 | 92.1 |

# Comparative Study (Minimum Optimum Score & Accuracy)





# Programming Perspective : How to use IBM Cloud for GAN?

- Create an account with IBM Cloud.
- Install the IBM Cloud CLI.
- Log in to your IBM Cloud account using CLI.
- Set up the IBM Cloud Target Org and Space.
- Clone the GitHub repository.
- Create a GAN configuration file.
- Edit the manifest file and ProcFile.
- Push the app to a new Python runtime in IBM Cloud

# Speech Synthesis steps by GAN

- Create a JSON config file that defines the architecture choices of the GAN model to be trained.
- Send the JSON config file  to a **Python-Flask server** in IBM Cloud through a **REST API call**
- The Flask API decodes the JSON config file and creates a GAN model definition.
- The Flask API then converts the GAN model definition into an **error-free PyTorch code.**
- The GAN model in PyTorch is then trained using the given input fashion audio data set.
- The trained model generates new fashion audio that are not in the input data set but look similar to them.
- The newly generated audio can be collected from the Python runtime in IBM Cloud.

# Conclusion

Each model has its own advantages but this system required high fidelity with less resource intensiveness, hence, MelGAN was chosen.

WaveNet gives accuracy as 78.9%, WaveGAN gives accuracy as 84.3%

.

MelGAN shows a lot of future promise for audio generation. Primarily for being fast and being able to generate long audio samples compared to other models.

On comparing the actual waveform with the waveform of the generated audio, the similarity score was found to be 92.1%.

# Thank you.

Geeta Atkar

- Assistant professor, G H Raisoni College of Engineering & Management, India
- Udemy Instructor on www.udemy.com
- Researcher at Vellore Institute of Tcehnology, Chennai,India
- Blogger at www.gatechnix.com

—

git.121@gmail.com
+91-9822342320
India

Use this cutout for a speaker video thumbnail. Delete if not using