# Speech to Face Generation using GAN's

A major project report submitted in partial fulfillment of the requirement for the award of degree of

**Bachelor of Technology**

in

**Computer Science & Engineering / Information Technology**

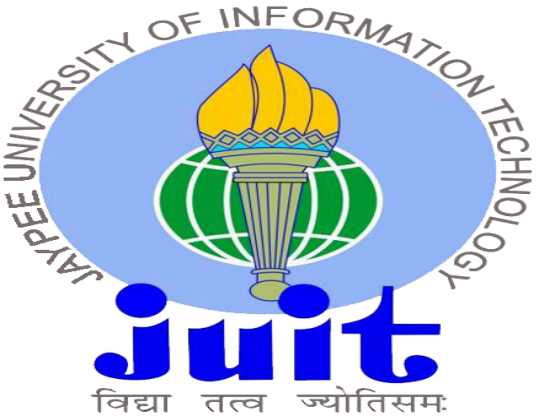
*Submitted by*

**PRIYANSH GARG (201348)**

**NIKHIL SHARMA (201408)**

*Under the guidance & supervision of*

**MR. RAMESH NARWAL AND MR. FAISAL FIRDOUS**




**Department of Computer Science & Engineering and Information Technology**

**Jaypee University of Information Technology, Waknaghat, Solan - 173234 (India)**

# CERTIFICATE

This is to certify that the work which is being presented in the project report titled "**Speech to Face Generation using GAN's** " in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by Priyansh Garg, 201348 and Nikhil Sharma, 201408 during the period from July 2023 December 2023 under the supervision of Mr. Faisal Firdous and Mr, Ramesh Narwal, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghar.

Priyansh Garg (201348)

Nikhil Sharma (201408)

The above statement made is correct to the best of my knowledge.

Mr. Faisal Firdous
Assistant Professor,
Computer Science & Engineering and Information Technology Jaypee University of Information Technology, Solan.

Mr. Ramesh Narwal
Assistant Professor,
Computer Science & Engineering and Information Technology Jaypee University of Information Technology, Solan.

# CANDIDATE'S DECLARATION

I hereby declare that the work presented in this report entitled **'Speech to Face Generation using GAN's'** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering / Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2023 to December 2023 under the supervision of **Mr. Faisal Firdous** (Assistant Professor, Department of Computer Science & Engineering and Information Technology) and **Mr. Ramesh Narwal** (Assistant Professor, Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature with Date)
Student Name:
Roll No.:

(Student Signature with Date)
Student Name:
Roll No.:

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature with Date)
Supervisor Name: Mr. Faisal Firdous
Designation: Assistant Professor
Department: CSE
Dated:

(Supervisor Signature with Date)
Supervisor Name: Mr. Ramesh Narwal
Designation: Assistant Professor
Department: CSE
Dated:

# ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible for us to complete the project work successfully.

I am really grateful and wish my profound indebtedness to Supervisor Mr Faisal Firdous and Mr. Ramesh Narwal Assistant Professor, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat. Deep knowledge & keen interest of my supervisor in the field of Machine Learning to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to Mr Faisal Firdous and Mr. Ramesh Narwal **,** Department of CSE, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Priyansh Garg (201348)
Nikhil Sharma (201408)

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| GAN | Generative Adversarial Neural Networks |
| CGAN | Conditional Generative Adversarial Neural Networks |
| SEGAN | Sound Enhancement Adversarial Neural Networks |
| VQ-VAE | Vector-quantized variational autoencoder |
| MSG-Style | Multi-Scale Gradient Style Generative Adversarial Neural Networks |
| AttGAN | Attribute GAN |
| FE-GAN | Facial expression GAN |
| FID | Fréchet Inception Distance |
| IS | Inception Score |
| MNIST | Modified National Institute of Standards and Technology |
| CIFAR-10 | Canadian Institute for Advanced Research - 10 classes |
| SVHN | Street View House Numbers |

# ABSTRACT

The use of GAN's to generate face images from speech has brought a new era of technological advancement for face generation. Speech to face generation allows the translation of spoken language into equivalent facial expressions. The intersection of speech and facial expressions holds great potential for enhancing human computer interactions and communication. Speech to face generation aims to generate realistic facial features with the help of given input speech. The generated images of facial expression must be synchronized with the input speech provided in terms of factors like lip movements, facial gestures, pitch of the sound , emotional expression.

In our work, we make use of a special dataset obtained from YouTubers to explore the novel use of voice Enhancement Generative Adversarial Networks (SEGANs) for the job of voice to face generation. Our goal is to achieve realism in facial expression recognition by using SEGANs. With the help of the SEGAN architecture, we have improved speech signals, resulting in audio representations that are easier to understand. These improved speech signals are then combined with matching facial photos from the YouTubers dataset to create synchronized facial emotions. This unique approach bears enormous implications for improving human-computer interaction and communication, while also pushing the bounds of machine learning approaches. We successfully generated the faces from the testing audios of our youtuber dataset. Our research gains authenticity via the use of real-world data from YouTubers, which guarantees that the variety and subtleties of human speech captured in internet content are faithfully reflected in the facial expressions we generate.

# CHAPTER-1 PROJECT INTRODUCTION

## 1.1. INTRODUCTION

Speech-to-face recognition is a game-changing technology that allows people to connect with computers in a more natural and subtle way by bridging the gap between spoken language and facial emotions. The use of Generative Adversarial Networks (GANs), complex machine learning models that combine the capabilities of a generator and a discriminator network, is the fundamental component of this breakthrough. This cooperative dynamic in GANs makes it possible to produce extremely realistic data, which is very useful for speech-to-face recognition.

The main goal of this project is to design a speech-to-face recognition system that is very effective in addition to being accurate. The goal is to develop a system that can precisely translate spoken information into comparable and visually appealing facial expressions by utilizing the capabilities of GANs. A device like this might completely change how people interact with computers by adding a new dimension of awareness and immersion.

This technology has a profoundly transforming effect not only on functionality but also on emotional intelligence and nonverbal communication. The goal is to create an environment where human-computer interactions replicate the richness of interpersonal communication by giving computer systems the capacity to comprehend and react to the subtleties of facial expressions that match to spoken words.

The realization that a successful speech-to-face recognition system has the potential to improve a variety of applications emphasizes this desire for accuracy and efficacy. Applications range widely and have a significant influence, from virtual assistants that can react not just to spoken words but also to the nuances of tone and mood to educational systems that may determine a learner's comprehension by examining their facial expressions during spoken conversations.

Essentially, a new era in human-computer interaction is being ushered in by the creation of an accurate and efficient speech-to-face recognition system with GANs. Its goal is to blur the lines between spoken and nonverbal communication in order to promote a paradigm of interaction that is more emotionally intelligent, intuitive, and responsive. With this

development, the goal is to build computer programs that can interpret our speech as well as the emotions and facial expressions that go along with it. This will enable people to interact with the digital world in a more meaningful and engaging way.

Because they can produce more realistic and subtle facial features than traditional diffused heads, Generative Adversarial Networks (GANs) are becoming the go-to option for speech-to-face recognition. Fundamentally adversarial in nature, GANs compete between a generator and a discriminator to generate samples that are more and more genuine. GANs are able to capture fine details in facial expressions and movements thanks to this adversarial training mechanism, which results in a more accurate representation of the wide range of human features connected to speech. GANs are excellent at learning from data distributions and generating coherent, high-fidelity facial images, in contrast to diffused heads, which might find it difficult to capture the complexity of facial images. Because GANs are dynamic and flexible, they are ideally suited for the challenging task of converting speech into realistic facial expressions. This allows for a more advanced and efficient solution in the field of speech-to-face recognition. Additionally, GANs add a creative and diverse element to their output, tackling the difficulty of producing a wide range of facial expressions that closely resemble the complexity of human communication. The process of adversarial training pushes the generator to keep getting better at simulating real-world facial reactions to speech inputs, which makes the generator's representation of emotions and intentions more flexible and realistic. The capacity of GANs to generalize and adapt to various speakers and linguistic subtleties is another benefit in speech-to-face recognition. Generally speaking, GANs are more robust to changes in accents and speech patterns because they automatically learn latent representations that capture the essence of facial expressions across different people. The deployment of such systems in real-world scenarios, where users may have different linguistic backgrounds and unique ways of expressing themselves, requires this flexibility.

There are many benefits to integrating speech-to-face recognition with GANs; it promotes improvements across multiple fields and greatly improves human-computer interactions. Here are a few of this cutting-edge technology's main benefits:

1. Natural and Intuitive Interaction: Technology makes interaction more natural and intuitive by allowing computers to recognize and react to facial expressions that match to spoken language. By simulating human communication, this improves the accessibility and usability of computer interfaces.

2. Enhanced Emotional Intelligence: The system gives robots a degree of emotional intelligence by recognizing and reacting to facial emotions. This has significant ramifications for applications that need a sophisticated comprehension of user emotions, including systems for mental health monitoring or virtual assistants that can respond with empathy.

3. Inclusive Communication: By recognizing facial expressions as an essential component of expression, speech-to-face recognition helps to promote inclusive communication. This opens up a new avenue for expression and conversation, which is especially helpful for people who have trouble speaking.

4. Better Virtual Meetings and Collaboration: Technology improves online communication, which benefits collaborative workspaces and virtual meetings. In addition to verbal communication, participants can communicate nonverbal clues as well, creating a more stimulating and productive collaborative atmosphere.

5. Advancements in Human-Computer Interfaces: Technological advancements have aided in the evolution of human-computer interfaces, resulting in more responsive and dynamic interactions. This is particularly important for applications such as gaming, where the addition of facial expressions can improve the immersion and realism of virtual worlds.

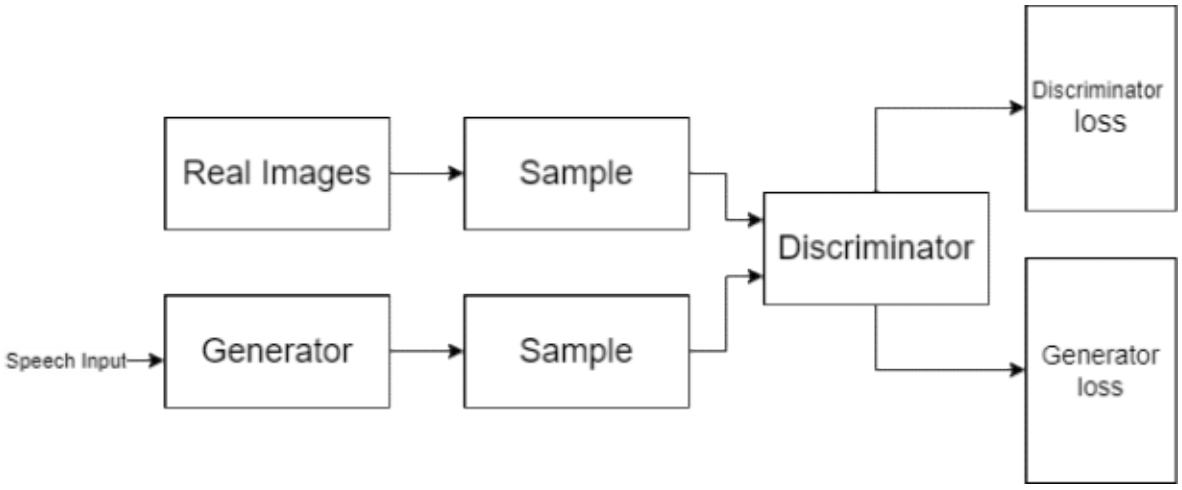



Fig. 1: GAN Architecture [7]

The structure of our work would be as shown. We propose a Generative Neural Networks (GAN) approach with the use of SEGAN (Speech Enhancement Generative Neural Networks).

Our project's foundation is the spoken content, which serves as the basis for our facial animation and expression production. Context and authenticity would be absent from the resulting facial features if the discourse was not clear and comprehensible. As a result, it is crucial that the input voice signal be precise in order to guarantee that the generated facial expressions correspond to the spoken words.

SEGAN, a state-of-the-art speech enhancement technology, is the foundation of our project. voice Enhancement Generative Adversarial Network, or SEGAN for short, is a highly developed neural network architecture created especially to take on the difficult task of enhancing voice signals. In contrast to conventional techniques that frequently suffer from distorted or noisy audio, SEGAN greatly improves speech clarity by utilizing the capabilities of Generative Adversarial Networks (GANs).
To function, SEGAN uses a two-network architecture that consists of a discriminator and a generator. The generator's job is to smooth out the speech signal input, which minimizes noise and improves overall clarity. As this is going on, the discriminator serves as a critic, assessing the output that has been produced and offering suggestions for improvement to the generator.
Unlike traditional speech augmentation methods, SEGAN can learn from data and change accordingly, resulting in continual performance improvement over time. Through the use of adversarial training, SEGAN is able to distinguish between clean speech and noisy signals with remarkable clarity and comprehensibility, resulting in audio output.

Moreover, SEGAN's architecture is precisely calibrated to comprehend the complex spectral and temporal properties of speech signals. SEGAN performs better than more broad GAN models, like Conditional Generative Adversarial Networks (CGANs), especially in the speech processing sector, thanks to its specialized focus.

To sum up, SEGAN is a revolutionary development in the field of audio signal processing, especially for speech improvement. It is a vital tool for our research since it can successfully remove noise from corrupted speech signals and improve overall clarity, which is necessary to produce convincing face expressions or animations.

The identification frame is used to maintain the speaker's identity when constructing an emotion or animating a face. It provides important information about the speaker's unique physical characteristics, which are then incorporated into the developed face features.

To create the series of frames that depict the movements or facial expressions corresponding to the enhanced speech signal, SEGAN is used instead of a diffusion model. The speaker's face's visual characteristics would be captured in the generated sequence.

A significant development in audio signal processing has been the Speech Enhancement Generative Adversarial Network (SEGAN), which tackles the difficult problem of speech enhancement. SEGAN demonstrates a unique ability to efficiently eliminate noise from distorted speech signals, greatly improving the output quality. SEGAN is better than Conditional Generative Adversarial Networks (CGAN) because of its unique architecture designed to handle the subtleties of speech signals. SEGAN performs better than the more generalized CGAN because its discriminator and generator networks are skilled at capturing and replicating the temporal and spectral aspects of speech. SEGAN outperforms CGAN in this particular domain and is a compelling option for applications that require high-quality speech enhancement due to its emphasis on the distinctive features of speech signals.

SEGAN is a major development in the field of audio signal processing, especially for speech improvement. Its distinct architecture makes it possible for it to effectively remove noise from distorted speech signals, significantly enhancing output quality. SEGAN performs better than other GAN models because of its specific focus on the temporal and spectral features of speech, as opposed to Conditional Generative Adversarial Networks (CGAN). SEGAN is a strong choice for applications needing high-quality speech enhancement because of its focus on capturing the unique characteristics of speech signals, especially when combined with face animation or expression generation.

In conclusion, GANs perform significantly better than diffused heads in speech-to-face recognition due to their ability to produce more varied and realistic facial features, as well as their ability to adapt to a wide range of linguistic contexts and capture the intricate relationship between speech and facial expressions. These characteristics put GANs in a strong and adaptable position for the development of highly precise and emotionally expressive HCI systems.

## 1.2. PROBLEM STATEMENT

The study of artificial intelligence has made substantial strides in a number of areas recently, including speech recognition and computer vision. The creation of lifelike human faces from

spoken language or voice signals is an intriguing example of how these two fields interact. In order to solve this problem, aural input, such as speech or spoken language, must be translated into corresponding face images that faithfully capture the speaker's emotional expressions, lip movements, and other non-verbal indicators. Applications for this difficult problem can be found in virtual communication, video game creation, animation, and other fields. Due to its capacity to generate crisp, clear images, Generative Adversarial Networks (GANs) have drawn interest as a potential solution to this issue.

Generative Adversarial Networks are used to create a reliable and effective method for generating facial images from spoken language. The basic objective is to build a model that can faithfully and consistently represent created faces while mapping audio input to facial expressions. The generated faces should be varied and expressive and match the emotional content and linguistic context of the input speech.

At the nexus of speech processing and computer vision, the Speech to Face Generation challenge with Generative Adversarial Networks offers an intriguing area for research and innovation. By making it possible to create more convincing and interesting virtual characters, this issue has the potential to completely change how people interact with computers. But it necessitates resolving issues with cross-modal mapping, emotional expression transfer, data variability, and other things. The suggested method seeks to construct a system that can faithfully translate spoken words into expressive and synchronized face images by utilizing the power of GANs and sophisticated machine learning algorithms.

To address the challenges mentioned above, a multi-stage approach can be adopted which can include Feature selection for accurate synchronization, Feature fusion and Alignment to align speech to visual expressions, Adversarial training to ensure faces are visually coherent and realistic, Evaluation Metrics to access quality and coherence of generated images and Fine-Tuning.

## 1.3. OBJECTIVES

- Design and implement a GAN architecture tailored for speech-to-face generation, capable of efficiently translating audio input into corresponding facial images.

- Collect a diverse dataset comprising paired audio and facial image samples, including various speakers, languages, accents, and emotional expressions, to facilitate training and evaluation.

- Compare the proposed model's performance against baseline models, such as traditional image synthesis techniques or simpler architectures, to highlight its advancements and improvements.

## 1.4. SIGNIFICANCE AND MOTIVATION OF THE PROJECT WORK

Speech to face generation using GAN represents a groundbreaking intersection of speech recognition and computer vision. By bridging these domains, we aim to unlock new possibilities in human face generation by speech.

Our project addresses a critical gap in technology - disconnection between audio and visual information. By developing a system capable of translating speech into realistic face expressions, we are breaking a new ground in bridging the audio visual gap.

The ability to synthesize realistic facial expressions and lip movement directly from spoken words. This innovation brings a human touch to machines which enhances the way we communicate with technology. Our motivation extends to making technology accessible to diverse communities, including those with speech related problems.

The project aspires to empower individuals by providing a more natural and inclusive model of communication.

Beyond the immediate applications, our project contributes to the evolution of human machine interaction. As we strive to create a system that understands the tone , pitch of sound to generate facial expressions. We vision a future where technology seamlessly integrates into our daily lives. Enhancing our interactions and experiences.

Our project has potential applications in therapeutic contexts in addition to mainstream communication. People who suffer from speech impairments, social anxiety, or communication disorders might gain a lot from a technology that encourages more subtle and expressive communication. This is consistent with our mission to develop technology that is not only cutting edge but also enhances people's lives by removing obstacles to communication and establishing a feeling of community.

Our project's motivational foundation is firmly anchored in the idea that human needs should drive technology, not the other way around. We hope to create a world where technology is seamlessly integrated into our daily lives, understanding and responding to our expressions, emotions, and intentions by bridging the gap between audio and visual information. According to this human-centric perspective, technology will be used in a way that feels intuitive and natural, improving our experiences in general. Our project is proof of the revolutionary potential of technology when it is fueled by a deep understanding of human communication dynamics, inclusivity, and empathy.

## 1.5. ORGANIZATION OF PROJECT REPORT

This detailed report is structured into six key chapters, each offering valuable insights into the Speech to face Generation using GAN.

**CHAPTER 1: INTRODUCTION**

The project's launchpad is this first chapter, which introduces the issue at hand, establishes specific goals, and explains the motivation behind the project. It offers a strong basis for the things that come next.

**CHAPTER 2: LITERATURE SURVEY**

In this chapter, we delve into existing knowledge, exploring reputable sources such as books and technical papers. The aim is to grasp the current landscape and identify gaps for our project to address.

**CHAPTER 3: SYSTEM DEVELOPMENT**

This chapter outlines the main points of the project, from requirements analysis to system design and implementation. We talk about the difficulties encountered during development and the tactical fixes used.

**CHAPTER 4: TESTING**

This section sheds light on the meticulous testing process, explaining the strategy and tools that we have used. We present test cases and outcomes, offering a clear picture of the system's reliability.

**CHAPTER 5: RESULTS AND EVALUATION**

This chapter, which focuses on results, analyzes findings and, if necessary, contrasts them with current solutions. It offers a thorough analysis of our project.

**CHAPTER 6: CONCLUSIONS AND FUTURE SCOPE**

Concluding our exploration, this chapter summarizes key findings, acknowledges limitations, and outlines potential future directions for research and development.

# CHAPTER-2 LITERATURE SURVEY

In 2018, a joint audio-visual model for isolating a single speech signal from a mixture of sounds such as other speakers and background noise was developed ,considering audio as the only input, a deep network-based model that incorporates both visual and auditory signals to solve this task The visual features are used to "focus" the audio on desired speakers in a scene and to improve the speech separation quality, was proposed by Ariel Ephrat et. al.[1].

In 2023, Hongwei yi et. al. addresses the problem of generating 3D holistic body motions from human speech. To achieve this, they first build a high-quality dataset of 3D holistic body meshes with synchronous speech and then defined a novel speech-to-motion generation framework in which the face, body, and hands are modeled separately and proposed a compositional vector-quantized variational autoencoder (VQ-VAE) for the body and hand motions [2].

In 2016 Scott Reed et. al. developed a novel deep architecture and GAN formulation to effectively bridge the advances in text and image modeling, translating visual concepts from characters to pixels [3].

In 2019, Duarte et.al. proposed that audio and visual signals are the most common modalities used by humans to identify other humans and sense their emotional state. Features extracted from these two signals are often highly correlated, allowing us to imagine the visual appearance of a person just by listening to their voice, or build some expectations about the tone or pitch of their voice just by looking at a picture of the speaker, so they proposed a conditional generative adversarial model shown in Fig 1. [4].

In 2020, Vijay et. al. proposed an end-to-end pipeline using Generative Adversarial Networks (GANs) for face construction based on speech-based descriptions, and iterative editing of the generated image to arrive at a close approximation of the expected face, a dialog-based interaction with the system where the user and system take turns providing descriptions and generating images respectively as MSG-Style GAN (Multi-Scale Gradient Style GAN) for face generation, and Attribute GAN (AttGAN) for facial attribute manipulation.[5].

In 2018, Speech-driven facial animation is the process which uses speech signals to automatically synthesize a talking character. The majority of work in this domain creates a mapping from audio features to visual features. This often requires post-processing using computer graphics techniques to produce realistic albeit subject dependent results and hence Konstantinos et. al. proposed a system for generating videos of a talking head, using a still

image of a person and an audio clip containing speech, that doesn't rely on any handcrafted intermediate features and proved that temporal GANs lead to more natural sequences than a static GAN-based approach.[6].

In 2023, Stypukowski et. al. presented an autoregressive diffusion model that requires only one identity image and audio sequence to generate a video of a realistic talking human head which was capable of hallucinating head movements, facial expressions, such as blinks, and preserving a given background. We evaluate our model on two different datasets, achieving state-of-the-art results on both of them.[7].

In 2022, Fang et. al. proposed a novel facial expression GAN (FE-GAN) which takes emotion and expressions into account in face generation. To achieve this goal, they used two auxiliary classifiers to learn more emotion and identity representations between different modalities, respectively .The triple loss is designed to make FE-GAN robust to voice variety and keep balance in two different modalities. The experimental results show that FE-GAN can not only outperform the previous models in terms of FID and IS values, but also generate more realistic face images compared with previous models.[8].

In 2019, Konstanitos et. al. again presented an end-to-end system that generates videos of a talking head, using only a still image of a person and an audio clip containing speech, without relying on handcrafted intermediate features since there temporal GAN uses 3 discriminators focused on achieving detailed frames, audio-visual synchronization, and realistic expressions.[9].

In 2021, Eskimez et.al. designed an end-to-end talking face generation system that takes a speech utterance, a single face image, and a categorical emotion label as input to render a talking face video synchronized with the speech and expressing the conditioned emotion which gave state of the art results. [10], and again in 2020 they proposed an end-to-end (no pre- or post-processing) system that can generate talking faces from arbitrarily long noisy speech and a mouth region mask to encourage the network to focus on mouth movements rather than speech irrelevant movements. In addition, we use generative adversarial network (GAN) training to improve the image quality and mouth-speech synchronization. [11].

In 2021, Shijing Si et.al. proposed a framework that captures the emotional expressions solely from speeches, and produces spontaneous facial motion in the video output. Compared to the baseline method where speeches are combined with a static image of the speaker, the results of the proposed framework were almost indistinguishable. User studies also show that the proposed method outperforms the existing algorithms in terms of emotion expression in the generated videos.[12].

In 2023, Chenpeng Du et. al. proposed a novel method called DAE-Talker that leverages data-driven latent representations obtained from a diffusion autoencoder (DAE). DAE contains an image encoder that encodes an image into a latent vector and a DDIM-based image decoder that reconstructs the image from it [13].

In 2016, Tim Salimans et. al. presented a variety of new architectural features and training procedures that we apply to the generative adversarial networks (GANs) framework to achieve state-of-the-art results in semi-supervised classification on MNIST, CIFAR-10 and SVHN.[14].

In 2017 Santiago et. al. proposed the use of generative adversarial networks for speech enhancement. In contrast to current techniques, we operate at the waveform level, training the model end-to-end, and incorporate 28 speakers and 40 different noise conditions into the same model, such that model parameters are shared across them and the enhanced samples confirm the viability of the proposed model, and both objective and subjective evaluations confirm the effectiveness of it. [15]

| **S. NO** | **Paper Title [Cite]** | **Journal/ Conference (Year)** | **Tools/ Techniques/ Dataset** | **Results** | **Limitations** |
|---|---|---|---|---|---|
| 1 | Generating Holistic 3D Human Motion from Speech [2] | IEEE Xplore (2023) | created a dataset of 3D holistic body meshes and synchronous speech recordings | 82.10% | loss of facial contours and details |
| 2 | Diffused Heads: Diffusion | arXiv (2023) | CREAMA and LRW | 75% | models suffer from long generation |

| | | | | | |
|---|---|---|---|---|---|
| | Models Beat GANs on Talking-Face Generation [7] | | | | times in comparison to other generative models. |
| 3 | Facial expression GAN for voice-driven face generation [8] | Springer (2021) | RAVDESS and eNTERFACE | 84.22% | moderate artifacts (e.g., the texture and color of face seem unnatural), loss of facial contours and details (e.g., tooth, hair and eyebrow region are obscure or missing), and minor semantic inconsistency |
| 4 | Dialog Driven Face Construction using GANs [5] | IEEE Xplore (2020) | CelebA | 73% | limited in context of the universe of features/loss of facial features |
| 5 | Realistic Speech-Driven Facial Animation with GANs [9] | Springer (2019) | GRID, TCD TIMIT, CREMA-D and and LRW datasets | 80% | only works for well-aligned frontal faces. Therefore, the natural progression of |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | this work will be to produce videos that simulate in the wild conditions. |
| 6 | Wav2Pix: Speech-conditioned Face Generation with Generative Adversial Neurals[4] | ICASSP (2019) | Youtubers Dataset | 76.81% | loss of facial shapes, inconsistency in facial hair, color etc. |
| 7 | End-to-End Speech-Driven Facial Animation with Temporal GANs [6] | arXiv (2018) | GRID and TCD TIMIT | 79.77% | Specific constraints need to be imposed in the latent space to generate consistent videos. |
| 8 | Generative Adversarial Text to Image Synthesis [3] | International conference on machine learning (2018) | MS COCO dataset | | Does not work for higher resolution images |

Table. 1: Literature Survey

## 2.1 OVERVIEW OF RELEVANT LITERATURE

**Ephrat, Ariel et al. (2018):**

- created a collaborative audio-visual model to separate a single spoken signal from a background noise.
- Integrated visual characteristics that highlight desired speakers in a scene to enhance the quality of speech separation.

**Yi Hongwei and others (2023):**

- discussed the process of creating 3D, comprehensive body motions using spoken language.
- produced a top-notch dataset of synchronous speech and 3D holistic body meshes.
- proposed a compositional vector-quantized variational autoencoder (VQ-VAE) for body and hand motions in a speech-to-motion generating framework.

**Reed, Scott et al. (2016):**

- created a revolutionary GAN formulation and deep architecture to bridge the gap between text and picture modeling advancements.
- converted ideas for a visual from letters to pixels.

**Duarte & associates (2019):**

- investigated the relationship between visual and aural cues for emotion detection and human identification.
- presented a conditional generative adversarial model that uses audio signals to produce visual appearances.

**Vijay et al.(2020):**

- suggested a whole process for creating faces using speech-based descriptions.
- employed Generative Adversarial Networks (GANs) for attribute modification and face generation, such as MSG-Style GAN and Attribute GAN.

**Konstantinos et al.(2018, 2023):**

- unveiled a method for speech-driven facial animation that does not require the use of manually created intermediary elements.
- presented an autoregressive diffusion model that requires little input to produce lifelike talking human heads.

**Fang et. al. (2022):**

- suggested a facial expression GAN (FE-GAN) takes expressions and emotions into account while creating faces.
- Triple loss and auxiliary classifiers were used to improve resilience to voice variability.

**Eskimez et al (2019, 2020, 2021):**

- Considering spoken utterance, face picture, and emotion labels, end-to-end talking face creation systems were designed.
- created a system that uses mouth motions to create talking faces from noisy speech.

**Shijing Si et al. (2021):**

- suggested a framework for video output that would only record emotional expressions from speeches in order to capture impromptu face movements.
- user research showed that it outperformed current algorithms in terms of expressing emotions in created videos.

**Du Chenpeng et al. (2023):**

- Presented DAE-Talker, an image reconstruction technique that uses data-driven latent representations from a diffusion autoencoder (DAE).

**Salimans et al. (2016):**

- New training techniques and architectural elements applied to GANs are presented, yielding state-of-the-art results in semi-supervised categorization.

**Santiago et al. (2017):**

- suggested using waveform-level generative adversarial networks for voice improvement.
- shared model parameters for various speakers and noise levels, proving efficacy in assessments that are both objective and subjective.

## 2.2 Key Gaps in the Literature

Some potential key gaps in the previous literatures where:

- The lack of large and diverse datasets linking speech to facial expressions is one of the main obstacles in speech-to-face generation. The richness and diversity of the training data greatly influences our models' efficacy. If we don't have a large enough sample size that includes different speakers, expressions, and language contexts, our models might not be able to generalize well.

- Investigating cutting-edge methods to record the complex interaction between audio and visual modalities is essential. Investigating cross-modal methods may allow us to have a better understanding of how speech affects facial expressions and vice versa. This could entail using multimodal neural networks or applying methods from related domains like audio-visual speech recognition.

- It is crucial to make sure that the generated faces appropriately reflect the speaker's identity and emotional context, in addition to exhibiting realism. Inconsistencies in facial emotions or features can compromise the generated content's validity. To preserve integrity to the original speaker, our models must be skilled at capturing and synthesizing minute details in speech as well as facial expressions.

- There exist considerable technical obstacles in the pursuit of real-time speech-to-face generation. Even while the results of our existing models are remarkable, it is still difficult to achieve real-time performance without sacrificing quality. We need to investigate ways to optimize current designs and create innovative frameworks that prioritize efficiency without compromising authenticity in order to meet this challenge.

This could mean using parallel computer architectures, optimizing computational procedures, or putting new, quickly generated algorithms into practice. In the end, developing speech-to-face generation's real-time capabilities opens the door to a wide range of applications in human-computer interaction, communication, and entertainment.

# CHAPTER-3 SYSTEM DEVELOPMENT

## 3.1. REQUIREMENTS AND ANALYSIS

**SOFTWARE RESOURCES**

- Python (Version: 2.7)
- PyTorch (Version: 2.0.0)
- OpenCV (Version: 4.8.0)
- Scipy (Version: 1.0.0)
- PyWavelets (Version: 0.5.2)

**HARDWARE RESOURCES**

- GPU
- RAM

**OTHERS**

- Recorded Speech

## 3.2. PROJECT DESIGN AND ARCHITECTURE

Project design and architecture is explained below and shown in Fig 5.

**PREPROCESSING:**

- Extract Audio Features: First, it is important to extract essential audio elements from the input voice signal before attempting to generate face animations or expressions from speech. In order to facilitate further processing and analysis, audio features act as the fundamental components that capture the essential aspects of the spoken signal. FFmpeg is a particularly strong and adaptable tool for extracting audio characteristics from spoken data. FFmpeg is a well-known command-line tool that can efficiently and adaptably work with multimedia files, including audio and video streams. We can successfully capture the acoustic characteristics of the speech stream by

extracting these and other pertinent audio elements, setting the stage for further processing operations.

- Normalize and Preprocess Audio Data: To guarantee consistency and strengthen the resilience of later processing techniques, the audio data must be preprocessed and normalized after the audio features have been extracted. To address concerns with fluctuating signal amplitudes, normalization is scaling the audio data to a specified range, usually between 0 and 1 or -1 and 1. By ensuring that every audio sample is represented on the same scale, processing will be more consistent and dependable. We can improve the audio data's quality and consistency by normalizing and preprocessing it, which will open the door for speech-to-face generation algorithms that are more precise and dependable. The efficacy and stability of the pipeline's later stages depend heavily on these first actions.

**GAN ARCHITECTURE:**

- Generator: The Generator is a neural network that takes a random noise or latent vector as input and produces a synthetic face image as output.

- Discriminator: The Discriminator is another neural network that evaluates the authenticity of an input image. [17]

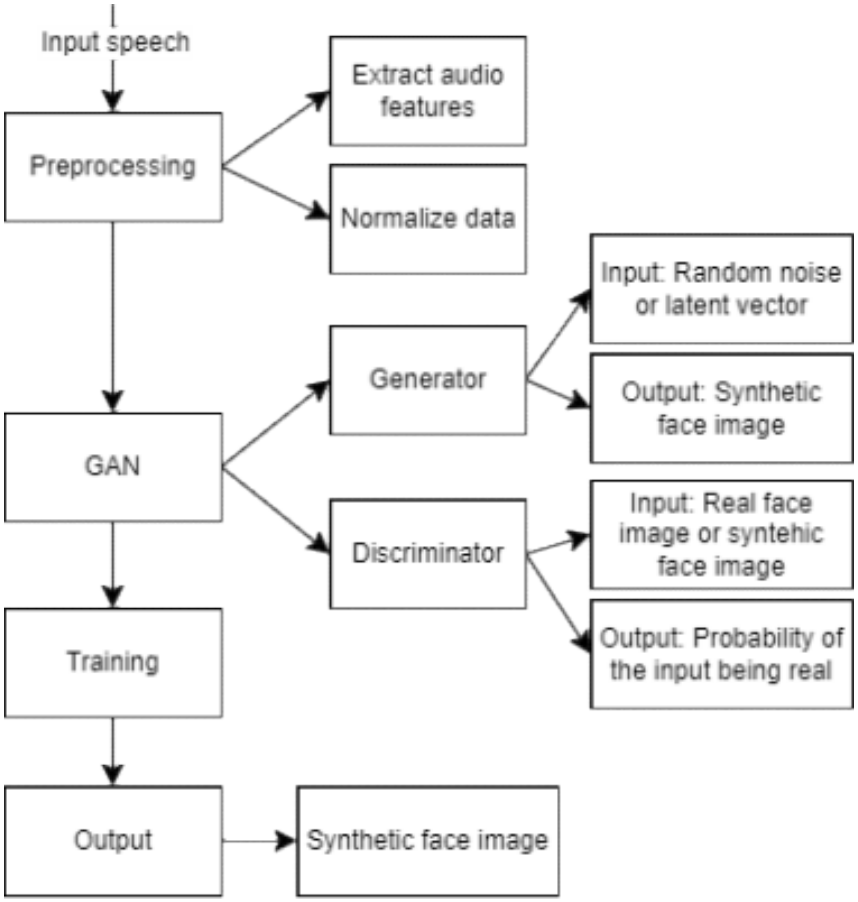



Fig. 2: Flowchart of working of our model

**TRAINING:**

- Initialize GAN Parameters
- Generate Synthetic Face Image
- Calculate Loss for Discriminator
- Update Discriminator Weights
- Calculate Loss for Generator
- Update Generator Weights

- Training continues for multiple iterations (201 in our case), with the Generator and Discriminator learning and improving their respective abilities until they reach convergence.

## 3.3. DATA PREPARATION

Our project's data preparation procedure entails a number of procedures meant to gather, preprocess, and arrange audio and video content from YouTube channels that are included in a.csv file. Below is a thorough breakdown of every step:

- Run get_data.py: This script is responsible for downloading YouTube videos specified in a .csv file. The YouTube URL, the YouTuber's name, and their gender are among the details included in the.csv file. The script generates a folder for every YouTuber listed in the.csv file when it runs. Two subfolders labeled "audio" and "video" are generated inside each folder. The audio and video files that were downloaded are located in these subfolders, respectively.
- Preprocess_audio_frames.py: Preprocessing is done on the audio data that is taken out of the downloaded videos by this script. It could include segmentation, normalization, and noise reduction, among other things. The audio files are saved back into their corresponding "audio" directories following preparation.
- Eliminate unnecessary audio files: In each "audio" folder, any files that do not conclude in "preprocessing_wav" are eliminated in order to guarantee that only pertinent audio samples are kept. This procedure lessens clutter in the dataset and aids in maintaining data cleanliness.
- face_detector.py: This script uses a pretrained Haar cascade classifier to identify and extract faces from the downloaded videos. The route to the pretrained classifier, a confidence level for bounding box identification, and the dataset path are all specified by the user. The identified countenances are trimmed and stored in the "video" subdirectories, which are located beneath the correspondingly downloaded videos.
- save paths in a pickle file: After the dataset is created, the script generate_pickle.py is used to save the paths for each image and audio frame in a pickle file. This phase makes it easier to access and retrieve data for use in later project phases, including model

training or evaluation. These procedures will help us collect, prepare, and arrange the audio and video material we'll need for our project in an organized and methodical manner. This will set the stage for further work like feature extraction, training, and assessment.

## 3.4 IMPLEMENTATION

**DATA COLLECTION AND PREPROCESSING:**

- We create our own dataset using the Youtube videos url, extracting audios and videos from them, removing all the noise during the preprocessing steps and then using the audio to generate faces.
- We create a csv while having the name, gender and url of the video as our columns.
- With the help of haarcascade frontal face recognition we extract all the frames where the face is visible.
- Then save the audio and the face images in different folders.
- We use the ffmpeg tool for the preprocessing of the audio file. [18]
- The audio file is downsampled to 16 bits.
- It converts the file from AAC to WAV format.
- Renames and saves the file as _preprocessed.wav.
- Below is the implementation of the above mentioned steps in Fig 5-6.

```python
import csv
import youtube_dl as yt
import argparse
import os
import shutil

parser = argparse.ArgumentParser()
parser.add_argument("Book1.csv",
                    help="CSV file where the url of the videos to download are.")
parser.add_argument("C:/Users/priya/OneDrive/Desktop/wav2pix-master/scripts/data_generator/Output",
                    help="Output folder of the dataset divided into /video and /audio.")
args = parser.parse_args()


def read_channels(file_path):

    parsed_channels = []
    with open(file_path) as csvfile:
        reader = csv.DictReader(csvfile)
        for row in reader:
            parsed_channels.append({"name": row["Name"],
                                    "gender": row["Gender"],
                                    "url": row["Channel-URL"]})
    return parsed_channels

def download(url_list):
    error_counter = 0
    count = 0

    for url in url_list:
        count += 1
        print ("Downloading videos and audios {}/{} with url [{}] from {}".format(count, len(url_list), url['url'], url['nam
```

Fig. 3: Creation of our dataset

```python
            video_options = {
                'format': "135, 140",
                'verbose': True,
                'continuedl': True,
                'ignoreerrors': True,
                'nooverwrites': True,
                'sleep_interval': 5,
                'playliststart': 1,
                'playlistend': 15,

            }

            with yt.YoutubeDL(video_options) as video_ydl:
                video_ydl.download([url['url']])


        except Exception:
            print ("Download error.")
            error_counter += 1


        # ffmpeg -i input_file -ss 00:00:15.00 -t 00:00:10.00 -c copy out.mp4
        workdir = os.listdir('./')
        for files in workdir:
            if files.endswith(".mp4"):
                shutil.move(files, video_out_path)
            elif files.endswith(".m4a"):
                shutil.move(files, audio_out_path)
    print ("Found {} errors".format(error_counter))


urls = read_channels(args.url_csv)
download(urls)
```

Fig. 4: Filtering of audio and faces

**MODEL:**

- A model called SEGAN, or Speech Enhancement Generative Adversarial Network, was created specifically for the aim of improving speech. Its main objective is to produce clean, high-quality speech signals from loud or distorted inputs.

- SEGAN's generator is in charge of converting distorted or noisy speech signals into improved and pure forms as shown in Fig 7.

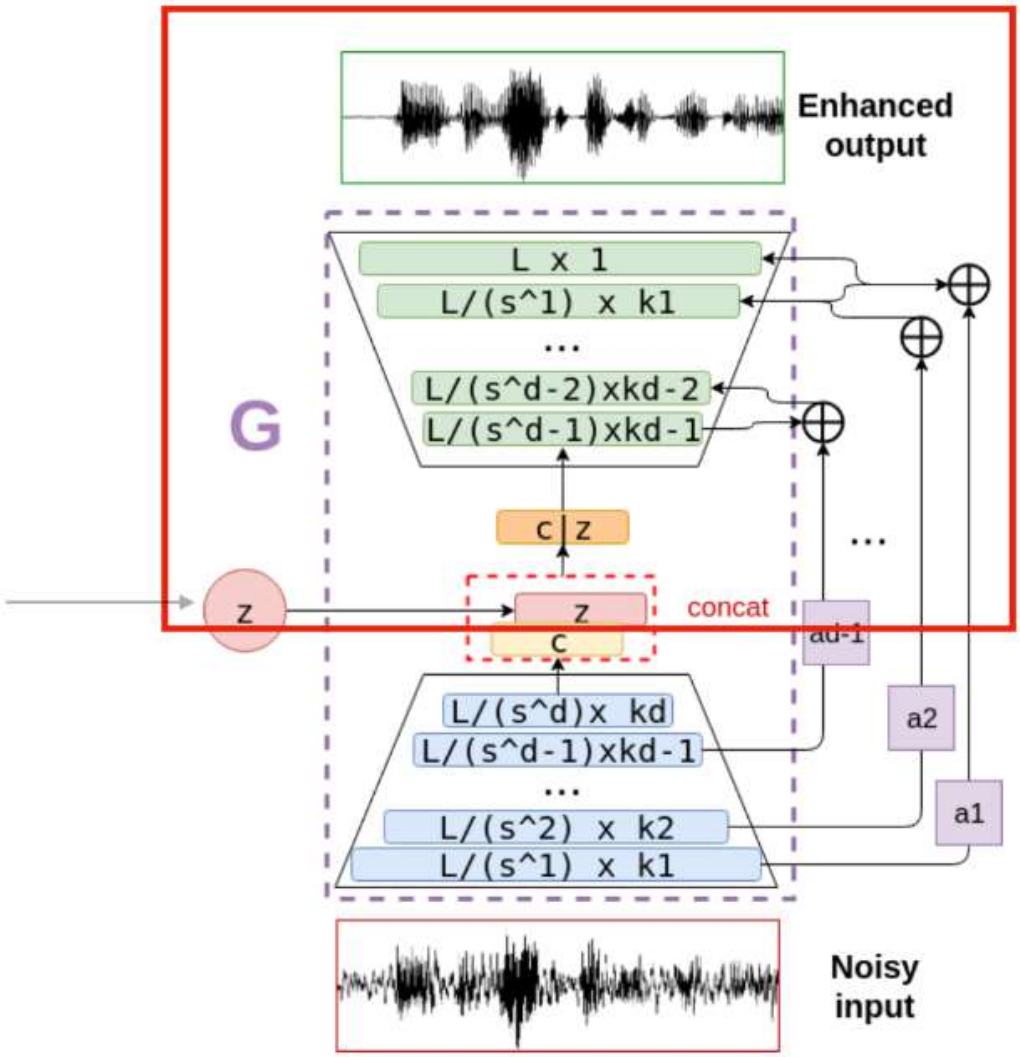



Fig. 5: SEGAN architecture

- It usually comprises a deep neural network that captures both short- and long-term dependencies in the audio input. Convolutional and recurrent layers are frequently used in its implementation.
- To aid in the training process and to make information flow between layers easier, skip connections can be employed.

- An important part of training the generator is the discriminator. It is in charge of differentiating between true, clean speech signals and augmented (produced) speech signals.
- Another neural network, the discriminator, is taught to feed back information to the generator so that it can produce more realistic and aesthetically beautiful speech.
- The generator is trained to reduce adversarial loss plus perceptual loss. In a perceptually relevant manner, perceptual loss quantifies the difference between created and genuine signals. The feedback from the discriminator is the source of adversarial loss.
- In order to correctly distinguish between produced and genuine signals, the discriminator must be trained. The degree to which it is able to discriminate between the two kinds of signals determines its loss.
- Typically, SEGAN is trained in an adversarial fashion, with the generator and discriminator attempting to outperform one another through iterative training. [16]
- The generator's goal is to produce more lifelike speech that deceives the discriminator.
- As a result, the discriminator gains proficiency in differentiating between generated and genuine speech.
- Below is shown the implementation of the above explained methods in Fig 8-11.

```
import torch
from torch import nn
from torch.autograd import Variable
from torch.utils.data import DataLoader
from models.discriminator import discriminator
from models.generator import generator
from scripts.utils import Utils, Logger, from_onehot_to_int
from scripts.dataset_builder import dataset_builder, Rescale
from PIL import Image
import os
import numpy as np

class Trainer(object):
    def __init__(self, vis_screen, save_path, l1_coef, l2_coef, pre_trained_gen,
                 pre_trained_disc, batch_size, num_workers, epochs, inference, softmax_coef, image_size, lr_D, lr_G, audio_seconds):

        # initializing the generator and discriminator modules.
        self.generator = generator(image_size, audio_seconds*16000).cuda()
        self.discriminator = discriminator(image_size).cuda()

        # if pre_trained_disc is true, load the already learned parameters. Else, initialize weights randomly
        if pre_trained_disc:
            self.discriminator.load_state_dict(torch.load(pre_trained_disc))
        else:
            self.discriminator.apply(Utils.weights_init)

        # the same as with the discriminator
        if pre_trained_gen:
            self.generator.load_state_dict(torch.load(pre_trained_gen))
        else:
            self.generator.apply(Utils.weights_init)
```

Fig. 6: Training our model

```
        # initializing other parameters
        self.inference = inference
        self.image_size = image_size
        self.batch_size = batch_size
        self.num_workers = num_workers
        self.beta1 = 0.5
        self.num_epochs = epochs
        self.l1_coef = l1_coef
        self.l2_coef = l2_coef
        self.softmax_coef = softmax_coef
        self.lr_D = lr_D
        self.lr_G = lr_G


        # building the data_loader
        self.dataset = dataset_builder(transform=Rescale(int(self.image_size)), inference = self.inference, audio_seconds = audio_seconds)
        self.data_loader = DataLoader(self.dataset, batch_size=self.batch_size, shuffle=True,
                                      num_workers=self.num_workers)

        # defining optimizers. Keeping all the parameters from the list of Module.parameters() for which requires_grad is TRUE
        self.optimD = torch.optim.Adam(filter(lambda p: p.requires_grad, self.discriminator.parameters()), lr=self.lr_D, betas=(self.beta1,
0.999))
        self.optimG = torch.optim.Adam(filter(lambda p: p.requires_grad, self.generator.parameters()), lr=self.lr_G, betas=(self.beta1,
0.999))

        # initializing a Logger in which we will create Log files and defining the directory name in which store checkpoints.
        self.logger = Logger(vis_screen, save_path)
        self.checkpoints_path = 'checkpoints'
        self.save_path = save_path

    def train(self):

        # initializing some loss functions that will be used
```

Fig. 7: Initializing the parameters for our model

In order to train a generative model that is specifically designed to produce visuals from audio inputs, the Trainer class is essential. The process of instantiating it involves setting up a number of parameters that are necessary for training, such as routes for storing results, visualization configurations, coefficients for different loss functions, pre-trained models (if any are available), and other hyperparameters like batch size, epochs, and learning rates. The class automatically decides at instantiation whether to do computations on the GPU or the CPU, an important optimization given the possible speedup provided by GPU acceleration. In order to prepare for the ensuing training procedure, it initializes instances of the discriminator and generator modules by either initializing them randomly or loading pre-trained weights.

In addition, the class creates a data loader with PyTorch's DataLoader module, which makes loading and prepping training data easier. To provide smooth and effective data processing during training, this data loader is customized with parameters including batch size, shuffle mode, and the number of worker processes.

The Adam optimizer is used to initialize the discriminator and generator optimizers, with special attention paid to optimizing parameters that have requires_grad=True. To ensure ideal convergence during training, learning rates and beta parameters are set in accordance with the configuration that is supplied.

Initializing a logger, which is in charge of monitoring and recording different training metrics and, if necessary, showing them on a designated screen, is a crucial component of the Trainer

class. To ensure that the model's current state can be permanently preserved and restored as needed during training, the class also defines the directory for storing checkpoints. To summarize, the Trainer class offers an all-inclusive structure for managing the generative model's training procedure, which includes initializing the model, loading data, configuring the optimizer, and configuring the logging system. Its adaptability and sturdy construction make it an essential part of the generative model's successful training for image synthesis from audio inputs.

```
class Saver(object):
    def __init__(self, model, save_path, max_ckpts=5, optimizer=None):
        self.model = model
        self.save_path = save_path
        self.ckpt_path = os.path.join(save_path, 'checkpoints')
        self.max_ckpts = max_ckpts
        self.optimizer = optimizer

    def save(self, model_name, step, best_val=False):
        save_path = self.save_path
        if not os.path.exists(save_path):
            os.makedirs(save_path)

        ckpt_path = self.ckpt_path
        if os.path.exists(ckpt_path):
            with open(ckpt_path, 'r') as ckpt_f:
                # read latest checkpoints
                ckpts = json.load(ckpt_f)
        else:
            ckpts = {'latest': [], 'current': []}

        model_path = '{}-{}.ckpt'.format(model_name, step)
        if best_val:
            model_path = 'best_' + model_path

        # get rid of oldest ckpt, with is the first one in list
        latest = ckpts['latest']
        if len(latest) > 0:
            todel = latest[0]
            if self.max_ckpts is not None:
                if len(latest) > self.max_ckpts:
                    try:
```

Fig. 8: Saving the checkpoints

```
def forward(self, x):
    h = x
    res_act = None
    for li, layer in enumerate(self.convs):
        if self.stride > 1 and li == 0:
            # add proper padding
            pad_tuple = ((self.kwidth//2)-1, self.kwidth//2)
        else:
            # symmetric padding
            p_ = ((self.kwidth - 1) * self.dilations[li]) // 2
            pad_tuple = (p_, p_)
        #print('Applying pad tupple: ', pad_tuple)
        if not (self.transpose and li == 0):
            h = F.pad(h, pad_tuple)
        #print('Layer {}'.format(li))
        #print('h padded: ', h.size())
        h = layer(h)
        h = self.acts[li](h)
        if li == 0:
            # keep the residual activation
            res_act = h
        #print('h min: ', h.min())
        #print('h max: ', h.max())
        #print('h conved size: ', h.size())
    # add the residual activation in the output of the module
    return h + res_act
```

Fig. 9: Forward layer of SEGAN

With a focus on Generative Adversarial Networks (GANs), a well-known deep learning framework for producing artificial data that resembles real-world samples, the code contains essential features required for creating and training neural network designs. One essential tool that makes sure that model checkpoints are maintained and managed during training is the Saver class. It facilitates smooth training from previously saved checkpoints, enhancing stability and dependability in model training by managing the loading and saving of both the optimizer's and the model's state dictionaries.

The Conv1DResBlock class is a crucial building block in architectural design that meets the specific requirements of 1D convolutional networks. Through the use of skip connections and a sequence of convolutional layers with different dilations, it makes it easier to describe complicated temporal dependencies, which improves the network's ability to recognize and

represent complex patterns in sequential data.

The foundation of the generator network is the GBlock class, which provides flexibility in creating individual blocks for specialized encoding or decoding jobs. Researchers and practitioners can design various generator architectures that are optimized for their specific applications with its provisions for convolutional or transposed convolutional layers, activation functions, batch normalization, dropout, and spectral normalization.

On the other hand, as the generator's opponent, the discriminator network is mostly shaped by the DiscBlock class. It provides the discriminator with the discriminatory power required to discern between real and artificial samples by encapsulating convolutional layers, activation functions, batch normalization, and dropout. This promotes adversarial competition and propels the iterative improvement of both networks.

These classes essentially serve as the foundation for a modular and extensible framework for building and training GANs, incorporating the concepts of robustness, scalability, and flexibility. The code promotes experimentation, creativity, and advancement in the field of generative modeling by encapsulating crucial capabilities within clearly defined and reusable components.

```
def train(self):

    # initializing some loss functions that will be used
    criterion = nn.MSELoss()
    l2_loss = nn.MSELoss()
    l1_loss = nn.L1Loss()

    print('Training...')
    for epoch in range(self.num_epochs):
        for sample in self.data_loader:

            # getting each key value of the sample in question (each sample is a dictionary)
            right_images = sample['face']
            onehot = sample['onehot']
            raw_wav = sample['audio']
            wrong_images = sample['wrong_face']
            id_labels = from_onehot_to_int(onehot) # list with the position of the youtuber which the audio in question belongs


            # defining the inputs as Variables and allocate them into the GPU
            right_images = Variable(right_images.float()).cuda()
            raw_wav = Variable(raw_wav.float()).cuda()
            wrong_images = Variable(wrong_images.float()).cuda()
            onehot = Variable(onehot.float()).cuda()
            id_labels = Variable(id_labels).cuda()


            # tensor of 64 (num of samples per batch) ones and zeros that will be used to compute D loss.
            real_labels = torch.ones(right_images.size(0))
            fake_labels = torch.zeros(right_images.size(0))

            # ======== One sided label smoothing ==========
            # Helps preventing the discriminator from overpowering the
```

Fig. 10: Implementation of the train function

The Generative Adversarial Network (GAN), a deep learning architecture well-known for producing lifelike synthetic data, has a primary training procedure that is encapsulated in the

train method. Several loss functions, such as Mean Squared Error (MSE) loss, L1 loss, and Cross-Entropy loss, are initialized in this approach. These loss functions are crucial in measuring the differences between generated and real samples during training. Each batch of data from the data loader is iterated over across several training epochs. The data loader contains face photos, one-hot encoded labels, raw audio waveforms, and mismatched face images with their identifiers. Iterative training of the generator networks and discriminator networks constitute the two halves of the training process.

The discriminator's loss is calculated by comparing the outputs of the discriminator for actual and fake images after the gradients have been cleared. This loss combines terminology for actual, fake, and mismatched photos, making it easier for the discriminator to distinguish between real and synthetic samples. On the other hand, the generator receives training after the discriminator. Gradients are cleaned, much like with the discriminator, and the generator loss—which includes several components like adversarial loss, feature matching loss, L1 distance loss, and softmax loss—is calculated. Together, these elements direct the generator's training process, which aims to produce synthetic images that are identical to actual ones.

The generator and discriminator networks' parameters are modified once losses and backpropagation are calculated. Simultaneously, scores for real, false, and mismatched images are logged together with various training metrics such as discriminator and generator losses. These metrics allow for real-time monitoring and evaluation by acting as vital indicators of model performance and training progress. Furthermore, model checkpoints of the discriminator and generator networks are periodically saved, guaranteeing that the state of the model parameters at any given time is preserved for possible use or examination in the future. To summarise, the train method represents the essence of GAN training by managing the process of optimisation, keeping an eye on training metrics, and assisting with model checkpointing to guarantee strong and efficient training of the discriminator and generator networks.

```
def predict(self):
    print('Starting inference...')

    starting_id = 0 # this would be the lower_bound id for the next image to be stored

    for id, sample in enumerate(self.data_loader):

        # id is the identifier of the batch. sample is a dictionary of 5 keys.
        right_images = sample['face']
        onehot = sample['onehot']
        raw_wav = sample['audio']
        paths = sample['audio_path']

        # retaining the right youtuber's name from the onehot vector and the id
        token = (onehot == 1).nonzero()[:, 1]
        ids = [path.split('_')[-1][:-4] for path in paths]

        txt = [self.dataset.youtubers[idx] + '_' + str(id) for idx,id in zip(token,ids)]

        if not os.path.exists('results/{0}'.format(self.save_path)):
            os.makedirs('results/{0}'.format(self.save_path))

        # storing raw_wav as a variable into the GPU
        raw_wav = Variable(raw_wav.float()).cuda()

        # feed the audio into the generator
        fake_images, _, _ = self.generator(raw_wav)

        for image, t in zip(fake_images, txt):
            im = image.data.mul_(127.5).add_(127.5).permute(1, 2, 0).cpu().numpy()
            rgb = np.empty((self.image_size, self.image_size, 3), dtype=np.float32)
            # bgr --> rgb
```

Fig. 11: Predicting the faces

When conducting inference with a pre-trained generative model—a typical practice in the context of Generative Adversarial Networks (GANs)—the predict approach is essential. When the method is initialized, it prints a message to indicate that it is ready to begin inferring and producing images. Iteratively processing batches of data supplied by the data loader, it extracts crucial data, including face images, labels encoded one-hot, and raw audio waveforms together with their respective routes. The technique uses the generator network to create synthetic images from the raw audio waveforms, an important step in producing new visual information from audio inputs.

The method performs post-processing steps to get the created images ready for storage after the generating step. This involves reformatting the color channels from the default BGR setup to RGB in accordance with industry standards for image representation, and transforming the tensor data into numpy arrays. By utilizing the Python Imaging Library (PIL), the technique transforms the arrays that have been processed into picture objects, which allow for easy storage and display. The created images are then permanently saved in the designated directory, where each image is individually recognized and labeled according to the accompanying audio input, allowing for traceability and simplifying additional analysis.

Essentially, the predict approach encapsulates key functions like data translation, image production, and storage management by orchestrating the complex process of converting unprocessed audio waves into visually appealing images. Its function goes beyond simple

image production; it is an essential part of the larger generative modeling pipeline that makes it possible to accurately and efficiently create synthetic visual content from acoustic inputs.

**FINE-TUNING:**

- Immediate Goal: After the first model training in speech-to-face generation is finished, fine-tuning takes place.
- Based on Evaluation Results: Assessing the model's performance and pinpointing areas for development serves as a guide for the fine-tuning procedure.
- Hyperparameter Adjustment: Increasing the model's accuracy in transforming speech into facial representations requires fine-tuning its hyperparameters. [19]
- Model Architecture Modifications: To address particular issues that arise during evaluation, changes to the model architecture may be necessary during fine-tuning.
- Targeted Improvements: The procedure attempts to address particular facets of the model's behavior that the assessment findings point out.
- Essential to Realism: In order to guarantee that the produced facial expressions correspond accurately with the spoken input, fine-tuning is essential.
- Adaptation to Input circumstances: The iterative process of fine-tuning makes the model more effective overall by enabling it to adapt to different speech input circumstances.

**DEPLOYMENT:**

- Our model would be deployed on a web app once tested and fine tuned with state-of-the-art accuracy. [20]
- User-Friendly Interface: The speech-to-face generation model will be simple for users to interact with through an intuitive and user-friendly interface on the web application, eliminating any technological complexity.
- Cross-Platform Accessibility: The online application is optimized to function on several platforms and gadgets, accommodating a diverse user base that includes computers, tablets, smartphones, and laptops.

## 3.5 KEY CHALLENGES

- Realistic Face Synthesis: Increasing the speech-to-face generation model's accuracy and precision requires producing images or face representations that closely match the speaker. Deep learning and sophisticated facial synthesis techniques can be used to do this. By training deep neural networks on large-scale facial picture datasets, these techniques produce individualized and remarkably realistic face images. The model can learn to capture the minute details of the speaker's facial features, such as shape, texture, and expression, by utilizing techniques like variational autoencoders (VAEs) and generative adversarial networks (GANs). This ensures that the generated face images closely resemble the speaker.

- Robustness: It is imperative to ensure that the system can adapt to a wide range of acoustic conditions for real-world applications where the input audios may be unclear or contain outside noise. The model can be trained on a variety of datasets with a wide range of acoustic circumstances, such as loud surroundings or low-quality recordings, to improve resilience. Furthermore, methods like data augmentation or noise reduction algorithms can be used to lessen the effects of outside noise and enhance the model's performance in difficult situations.

- Synchronization: Achieving a genuine and engaging user experience requires synchronizing the speech with the generated face animations. This entails precisely lining up the duration of lip movements, facial expressions, and emotional gestures with the related speech portions. To guarantee temporal coherence between the audio and visual modalities, sophisticated methods for audio-visual synchronization, such as dynamic time warping or attention processes, can be applied. The model can produce face animations that match the pitch and tone of speech by tightly integrating the speech processing and facial animation pipelines. This improves the created content's overall authenticity and plausibility.

- Variability: Having alternatives for producing various, contextually-appropriate face animations is crucial to producing visually captivating and emotive images. This entails making certain that the produced face images retain anatomical accuracy and spatial

coherence, as well as variety in facial expressions, movements, and stances. To allow the model to produce a variety of contextually relevant and animated faces, methods like style transfer and conditional GANs can be utilized. Furthermore, by carefully planning the model architecture and training objectives, distortions or artifacts in the generated images can be avoided, guaranteeing that each feature of the face is displayed naturally and realistically and is in the right place.

- Mode Collapse: Mode collapse is one of the difficulties in training generative models, especially GANs. When the generator only generates a small number of identical outputs rather than the entire diversity of the target distribution, mode collapse takes place. As a result, samples may be generated that are artificial or repetitious and do not accurately reflect the variability found in the training set. Careful architectural planning, regularization methods, and training approaches are needed to mitigate mode collapse in order to motivate the generator to examine the whole data distribution and produce a variety of realistic examples. The model can provide more accurate and expressive face synthesis results by addressing mode collapse, which improves its ability to capture the vast diversity of facial expressions and features seen in the training data.

# CHAPTER-4 TESTING

## 4.1 TESTING STRATEGY

Testing Strategy would comprise of the following:

**TEST DATASET:**

- an additional test dataset that wasn't employed in the training stage. To assess the generalization of the model, this dataset should include a wide variety of voice samples and accompanying facial photos.

**PERFORMANCE METRICS:**

- Metrics for measuring image quality include the Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSI). [23][24][25]
- Metrics pertaining to the correctness of face features (such as mouth movement and eye position) in the created pictures, if applicable.

$$
\begin{aligned}
PSNR &= 10 \cdot \log_{10} \left( \frac{MAX_I^2}{MSE} \right) \\
&= 20 \cdot \log_{10} \left( \frac{MAX_I}{\sqrt{MSE}} \right) \\
&= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE).
\end{aligned}
$$

Fig. 12. PSNR Formula

Peak Signal-to-Noise Ratio, or PSNR, is a commonly used metric in compression and image processing applications that is used to objectively assess the quality of created or reconstructed images. PSNR offers a numerical depiction of image integrity by comparing the greatest signal power to the power of noise introduced during production or reconstruction. The squared differences between corresponding pixels in the original and reconstructed images are used to calculate it. These differences are then averaged over all pixels and converted to decibels. Greater fidelity is shown by higher PSNR values, which

imply that the created or reconstructed image closely resembles the original with little distortion.

Even though PSNR is widely used, it has several drawbacks. It does not completely account for perceptual qualities, such as human visual perception, and instead concentrates on pixel-wise variations. Because of this, even photos with high PSNR values could have visible flaws or artifacts to the naked eye. Furthermore, PSNR ignores structural similarities between images and is sensitive to image scale. Thus, additional metrics such as the Structural Similarity Index (SSIM) or perceptual metrics derived from deep learning models may provide more thorough evaluations in situations when perceptual quality is critical. However, PSNR is still useful for tasks like lossy image reduction or image reconstruction from noisy data when pixel-level accuracy is critical. It is a helpful tool for evaluating the effects of compression methods on image fidelity or comparing the quality of various image processing algorithms due to its ease of use and clear interpretation. As a result, even with its acknowledged drawbacks, PSNR is still a crucial tool in the image processing toolbox, offering insightful information on the integrity of created or reconstructed images.

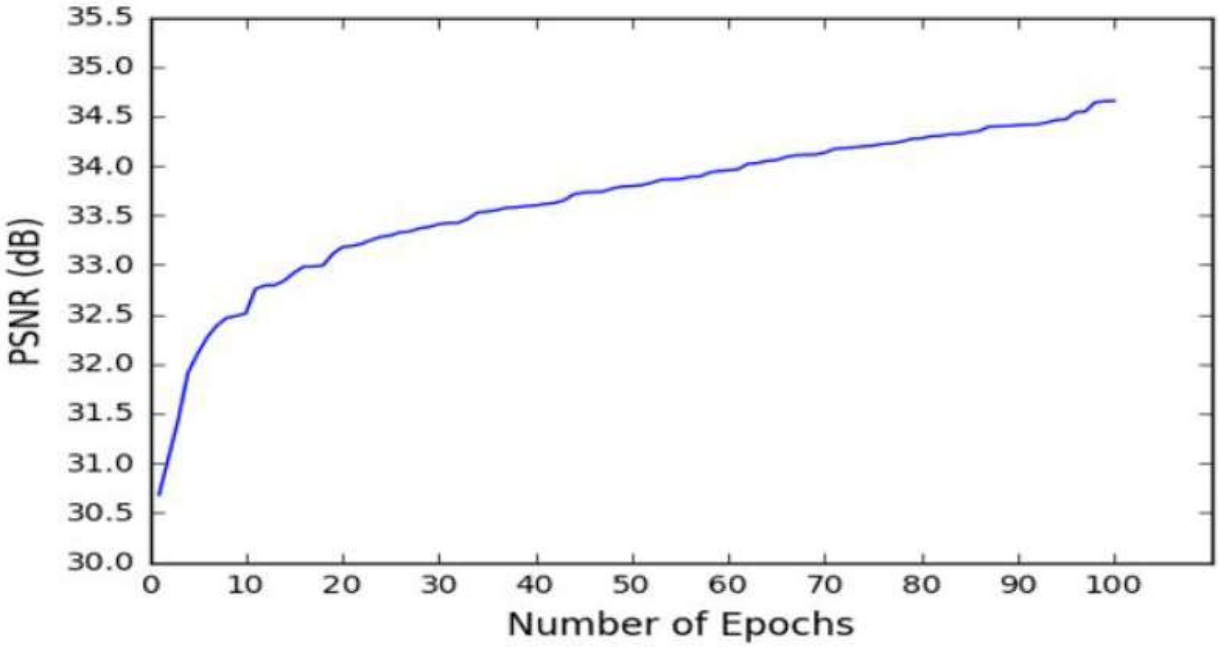



Fig. 13. PSNR of SEGAN

A thorough visual representation of how the quality of generated or reconstructed pictures varies during the training process is provided by the PSNR vs. Epoch graph. The PSNR

values, a measure of image fidelity, are plotted against the total number of training epochs in this graph. Every point on the graph represents the PSNR value determined during training at a particular period. Examining this graph offers important information about the generative model's performance and training dynamics. As training goes on, a rising trend in PSNR values shows constant improvement in image quality, but a plateauing trend indicates the model may have hit its maximum improvement potential. Suspended training is indicated by smooth, rising trends; anomalies or aberrant behavior needing further research are indicated by abrupt spikes or dips.

Finding the convergence points at which PSNR values stabilize also aids in figuring out the best training stop points to avoid overfitting. All things considered, the PSNR vs. Epoch graph is an essential diagnostic tool for tracking training progress and maximizing the performance of generative models.

**REAL-TIME INFERENCE:**

- Evaluate the model's performance in real-time circumstances, particularly if it is meant to be used in applications that need speedy face production.

**ERROR ANALYSIS:**

- Examine situations in which the model is unable to produce realistic-looking faces. Recognize the different kinds of faults (such as distorted expressions or missing facial features) and consider how they may be improved.

## 4.2 TEST CASES AND OUTCOMES

A speech-to-face generation model is tested by evaluating its performance in a range of scenarios in order to guarantee quality, robustness, and generalization. While assessing a speech-to-face generation model, take into account the following test cases:

**SPEECH LEVEL:**

- Metrics with an objective: Assess the generated speech quality using common metrics, like SNR (signal-to-noise ratio) and PESQ (perceptual evaluation of speech quality).

**SUBJECTIVE EVALUATION:**

- To get input on the generated speech's perceived quality, administer subjective listening tests to human evaluators.

**REALISTICNESS:**

- Evaluate the realism of facial animations by contrasting them with expressions found in real life. Make use of metrics such as human assessments or facial action unit analysis.

**EMOTIONAL EXPRESSIVENESS:**

- Evaluate how well the model can use facial expressions to represent various emotions in accordance with the emotional content of the speech input.

**CROSS-SPEAKER BROADCASTING:**

- Assess the model's capacity to produce realistic facial animations for speakers that aren't included in the training set in the speaker independence test. This aids in evaluating the capacity for generalization among various speakers.

**ROBUSTNESS TO BACKGROUND NOISE:**

- Assess the model's performance when exposed to varying intensities and kinds of background noise. For real-world applicability, this is essential.

**LATENCY:**

- Assess the model's processing speed and latency to make sure there are no appreciable delays when using it in real-time applications.

**VARIOUS SPEAKING STYLES:**

- Evaluate the model's ability to speak in a variety of contexts, including formal presentations, informal talks, and facial expressions.

# CHAPTER-5 RESULTS AND EVALUATION

## 5.1. RESULTS

A number of important factors must be taken into consideration when evaluating a speech-to-face recognition model that makes use of the SEGAN (Speech Enhancement Generative Adversarial Network) architecture, especially when considering real-time performance. Speech-to-Face Animation Model Pipeline: SEGAN is a crucial part of the pipeline because of its reputation for improving speech signal quality.

The quality of the speech generated by the SEGAN model is an important dimension to consider when assessing it. Quantitative measures that assess the fidelity and clarity of the enhanced speech include PESQ and SNR. Furthermore, human listeners' subjective assessments determine the perceived quality by picking up on subtleties that automated metrics might overlook. Human listeners should consistently receive high-quality speech outputs from a well-performing SEGAN model that are perceptually pleasing.
The below figure shows an example of the csv file required for the data creation , containing the Name, Gender and Channel URL from where the Audio of the youtuber and the video that is faced at different frames is extracted.
Alongside speech quality, the evaluation also includes the facial animation that is produced in reaction to the improved speech. Here, realism and emotive expressiveness are crucial. By using both automated facial action unit analysis and human evaluators, the generated facial expressions are compared to real-world human expressions in order to assess the realism. Emotional expressiveness is evaluated by comparing the facial animations to the speech's intended emotional content. This shows how well the model can represent complex emotions.

Since real-time processing is the main focus, the effectiveness and speed of the model are critical. One crucial metric is latency, which is the amount of time it takes to process and produce facial animations in response to speech input. A SEGAN-based model that works well in real-time applications should respond quickly and with minimal latency.

In summary, assessing a speech-to-face recognition model that uses the SEGAN architecture is a complex procedure. It entails closely examining the model's output in terms of speech quality, emotional expressiveness, realistic facial animation, audio-visual synchronization, and real-time processing effectiveness. Through a thorough evaluation of these dimensions, one can determine the model's advantages, pinpoint areas in need of development, and determine how well-suited it is for various real-world scenarios.

For the extraction of videos the Youtube_dl module is used whereas for the extraction of audio and faces the ffmpeg and face recognition using haar cascade is used respectively.

| Name | Gender | Channel-URL |
|---|---|---|
| CarryMina | Male | https://www.youtube.com/@CarryMinati |
| Scout | Male | https://www.youtube.com/@sc0utOP |

Fig. 14. CSV file created for downloading the dataset

The following shows how the output of our data generation takes place where a folder is created with the youtubers name inside of which two sub folders are created for their faces and audio respectively.

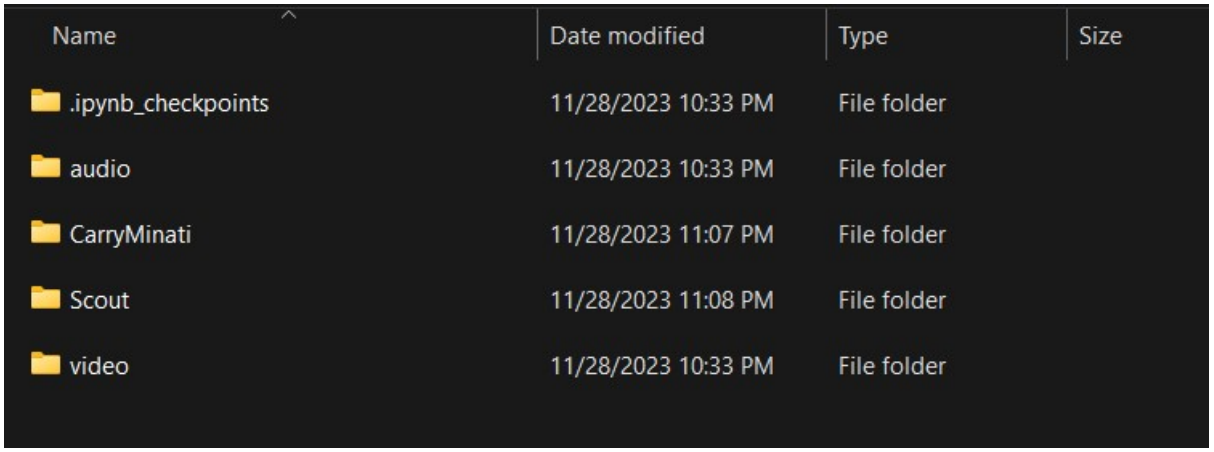



Fig. 15. Creation of folders containing dataset

Meticulous logging was used during the SEGAN (Speech Enhancement Generative Adversarial Network) training process to track the model's development and effectiveness. A range of important measures that are essential for assessing the effectiveness of the training procedure were included in the logged data. Among these measures were the values of the loss function, which showed how well the model was minimizing the discrepancy between the output it anticipated and the actual data. In order to evaluate the improvement in speech quality that the model produced, we also monitored measures like the improvement of the signal-to-noise ratio (SNR) and the ratings obtained from the Perceptual Evaluation of Speech Quality (PESQ). Through thorough examination of these logs, we were able to learn more about the convergence, stability, and general performance of the model, which helped us make well-informed decisions about changing training plans and hyperparameters.

```
Epoch: 0, d_loss= 0.055760, g_loss= 64.001129, D(X)= 0.926632, D(G(X))= 0.160045, wrong score: 0.043892
Epoch: 1, d_loss= 0.057926, g_loss= 63.796341, D(X)= 0.896822, D(G(X))= 0.160947, wrong score: 0.055384
Epoch: 2, d_loss= 0.077075, g_loss= 63.765766, D(X)= 0.907037, D(G(X))= 0.129011, wrong score: -0.008054
Epoch: 3, d_loss= 0.309811, g_loss= 61.452286, D(X)= 0.657787, D(G(X))= 0.475461, wrong score: 0.089087
Epoch: 4, d_loss= 0.227382, g_loss= 61.152405, D(X)= 0.843163, D(G(X))= 0.431370, wrong score: -0.027232
Epoch: 5, d_loss= 1.770741, g_loss= 62.963161, D(X)= 0.712333, D(G(X))= 1.104903, wrong score: -0.101106
Epoch: 6, d_loss= 1.025870, g_loss= 66.521713, D(X)= 0.408303, D(G(X))= 0.847737, wrong score: 0.129297
Epoch: 7, d_loss= 1.930231, g_loss= 63.691093, D(X)= 0.729312, D(G(X))= 1.319256, wrong score: 0.022362
Epoch: 8, d_loss= 15.920660, g_loss= 62.815128, D(X)= 2.725902, D(G(X))= 2.369642, wrong score: 2.580976
Epoch: 9, d_loss= 6.198458, g_loss= 61.954189, D(X)= 1.400263, D(G(X))= 2.287891, wrong score: 0.660507
Epoch: 10, d_loss= 1.263177, g_loss= 62.034813, D(X)= 0.747440, D(G(X))= 0.963219, wrong score: 0.090414
Epoch: 11, d_loss= 1.745187, g_loss= 60.240852, D(X)= 0.143689, D(G(X))= 0.874865, wrong score: -0.333523
Epoch: 12, d_loss= 1.909407, g_loss= 60.213707, D(X)= 0.377961, D(G(X))= 1.142021, wrong score: -0.243541
Epoch: 13, d_loss= 0.728737, g_loss= 57.716660, D(X)= 0.679110, D(G(X))= 0.738750, wrong score: 0.126716
Epoch: 14, d_loss= 2.672247, g_loss= 59.042160, D(X)= -0.118547, D(G(X))= 1.008113, wrong score: -0.602512
Epoch: 15, d_loss= 1.498632, g_loss= 59.110294, D(X)= 0.636049, D(G(X))= 1.067246, wrong score: 0.105891
Epoch: 16, d_loss= 0.901197, g_loss= 60.139008, D(X)= 0.918435, D(G(X))= 0.781849, wrong score: 0.329837
Epoch: 17, d_loss= 4.800796, g_loss= 58.505039, D(X)= 0.121582, D(G(X))= 1.834146, wrong score: -0.059421
Epoch: 18, d_loss= 1.987707, g_loss= 58.469753, D(X)= 0.782342, D(G(X))= 1.239910, wrong score: 0.087703
Epoch: 19, d_loss= 3.684190, g_loss= 58.249023, D(X)= 0.152806, D(G(X))= 1.117468, wrong score: -0.938156
Epoch: 20, d_loss= 2.038331, g_loss= 59.170319, D(X)= 0.948928, D(G(X))= 1.161451, wrong score: 0.108420
Epoch: 21, d_loss= 2.791769, g_loss= 57.553444, D(X)= 0.662643, D(G(X))= 1.397617, wrong score: 0.028475
Epoch: 22, d_loss= 1.612932, g_loss= 58.287037, D(X)= 0.593914, D(G(X))= 1.117608, wrong score: 0.022392
Epoch: 23, d_loss= 0.725739, g_loss= 58.282299, D(X)= 1.116192, D(G(X))= 0.557716, wrong score: -0.140275
Epoch: 24, d_loss= 1.915690, g_loss= 54.619354, D(X)= 0.661408, D(G(X))= 1.006487, wrong score: -0.044490
Epoch: 25, d_loss= 1.108053, g_loss= 57.725067, D(X)= 0.658789, D(G(X))= 0.857685, wrong score: -0.044859
Epoch: 26, d_loss= 1.268006, g_loss= 56.756210, D(X)= 0.085632, D(G(X))= 0.514517, wrong score: -0.039216
Epoch: 27, d_loss= 0.756410, g_loss= 57.014523, D(X)= 0.739413, D(G(X))= 0.747134, wrong score: 0.202692
Epoch: 28, d_loss= 0.591059, g_loss= 57.358467, D(X)= 0.897933, D(G(X))= 0.580169, wrong score: -0.145239
Epoch: 29, d_loss= 0.619169, g_loss= 56.988926, D(X)= 0.680125, D(G(X))= 0.493424, wrong score: 0.007526
Epoch: 30, d_loss= 0.481622, g_loss= 56.853569, D(X)= 0.694210, D(G(X))= 0.422460, wrong score: -0.269602
Epoch: 31, d_loss= 0.240885, g_loss= 56.958778, D(X)= 0.676620, D(G(X))= 0.292798, wrong score: -0.119488
Epoch: 32, d_loss= 0.285667, g_loss= 57.456665, D(X)= 0.991056, D(G(X))= 0.442384, wrong score: 0.014034
Epoch: 33, d_loss= 0.289083, g_loss= 56.633503, D(X)= 0.942221, D(G(X))= 0.485177, wrong score: 0.049960
Epoch: 34, d_loss= 0.180051, g_loss= 55.974991, D(X)= 0.901101, D(G(X))= 0.381720, wrong score: 0.007107
Epoch: 35, d_loss= 0.146973, g_loss= 55.566406, D(X)= 0.921171, D(G(X))= 0.322476, wrong score: 0.060977
Epoch: 36, d_loss= 0.132580, g_loss= 55.434799, D(X)= 0.858763, D(G(X))= 0.312841, wrong score: -0.047299
Epoch: 37, d_loss= 0.097533, g_loss= 55.363274, D(X)= 0.912691, D(G(X))= 0.244457, wrong score: 0.009028
Epoch: 38, d_loss= 0.063943, g_loss= 55.261738, D(X)= 0.878968, D(G(X))= 0.216504, wrong score: -0.037679
Epoch: 39, d_loss= 0.043793, g_loss= 54.543011, D(X)= 0.849327, D(G(X))= 0.141034, wrong score: -0.020480
```

Fig. 16. Log file containing epoch values  for SEGAN

Fig. 17. Good and bad results of SEGAN

There were times during the Generative Adversarial Network (GAN) training and evaluation process when there were notable disparities in the quality of the generated outputs. The given photos show instances of both subpar and acceptable results generated by the GAN.

The "bad results" graphics offer examples of situations in which the generated samples significantly differ from the intended output. In these cases, artifacts, blurriness, incoherence, and distortion are frequently seen problems. These flaws are a sign of issues encountered in the training process, like a lack of diversity in the data, a model with too little capacity, or hyperparameters that are not ideal. These problems point out places where our model design and training process need to be strengthened.

On the other hand, the pictures that show "good results" represent situations in which the GAN effectively extracts the desired features from the input data and generates outputs that are of excellent quality. These outcomes show better clarity, realism, and integrity than the poor examples. Effective regularization strategies, efficient training protocols, and careful hyperparameter tweaking may all have a role in the success of these results. These examples demonstrate the capabilities of our GAN model when correctly built and trained, and they also serve as benchmarks for desired performance.

All things considered, the contrast between the "bad" and "good" outcomes highlights how crucial thorough assessment and iterative improvement are to the creation of generative models. Through the identification and resolution of the issues that result in subpar results, we can work toward improving the stability and dependability of our GAN framework, which will ultimately increase its usefulness and relevance across a range of fields.

## 5.2. COMPARISON WITH EXISTING SOLUTIONS

Unlike previous approaches that primarily use Conditional Generative Adversarial Networks (CGANs) for tasks pertaining to speech synthesis, facial animation, and audio-visual processing, our method is unique in that it utilizes SEGAN (Speech Enhancement Generative Adversarial Network). With its specialized design for speech-related tasks, SEGAN provides exceptional performance in terms of improving speech input quality.

Although CGANs are flexible and commonly used for a wide range of generative tasks, SEGAN is particularly good at handling the special problems that speech signals present. Because of its specially designed architecture and training protocols for waveform-level operations, it is especially good at capturing and enhancing the subtleties of speech patterns. Our model performs exceptionally well in tasks pertaining to speech synthesis and enhancement because of its emphasis on speech-centric design. There models are compared in the given below Table 1.

We prioritize a model that is well-tuned to the subtleties of speech data by selecting SEGAN over CGAN, which offers improved capabilities in speech signal generation, synthesis, or processing. This tactical decision positions our method as a specialized and efficient solution for tasks where speech output fidelity and quality are critical.

For applications like interactive communication systems, live streaming, and virtual assistants, where low latency and instant responsiveness are critical, real-time processing is essential.

We are currently working on techniques to maximize the effectiveness and speed of our SEGAN-based model. To make sure that our model can function properly in real-time scenarios, this entails optimizing the architecture, streamlining computational procedures, and utilizing parallelization techniques. By doing this, we hope to close the gap that exists between the need for instantaneous results in various applications and the traditionally time-consuming nature of speech-related tasks.

The addition of real-time capabilities to our SEGAN-based model broadens its applicability to more dynamic use cases and improves its practical utility. Envision a situation where our model is able to produce realistic facial animations in real-time based on dynamic emotional cues, or instantly improve speech quality during a live conversation. In addition to being technically difficult, these developments have the potential to completely transform the way users interact with interactive systems.

Our commitment to real-time performance is an indication of our forward-thinking nature, guaranteeing that our model performs well in specific speech-related tasks and complies with the changing needs of contemporary applications. We see a smooth integration of facial animation and high-quality speech enhancement into real-time applications as we continue to develop and optimize our solution, providing users with a more responsive and natural interaction experience.

| **Feature** | **CGAN (Conditional GAN)** | **DCGAN (Deep Convolutional GAN) [22]** | **SEGAN (Speech Enhancement GAN)** |
|---|---|---|---|
| Primary Application | Image Generation | Image Generation | Speech Signal Enhancement |
| Input Conditioning | Conditioned on additional information (labels) | Unconditional (no explicit conditioning on input) | Unconditional (no explicit conditioning on input) |
| Objective | Generate data conditioned on specific labels | Generate high-quality images | Enhance the quality of noisy speech signals |
| Architecture | Typically uses a generator and discriminator | Uses convolutional neural networks | Includes an encoder, generator, and discriminator |
| Generator | Generates data based on input conditions | Generates high-quality images | Enhances noisy speech signals to produce cleaner audio |

| | | | |
|---|---|---|---|
| Discriminator | Distinguishes between real and generated data | Utilizes convolutional layers for image discrimination | Identifies whether the enhanced speech is real or generated |
| Training Data | Requires labeled data for conditioning | Unconditional training on images | Often trained on pairs of clean and noisy speech data |
| Loss Functions | Typically uses a combination of adversarial loss and conditional loss | Adversarial loss, possibly with additional losses | Combines adversarial loss, perceptual loss, and mean squared error |
| Evaluation Metrics | FID, IS, precision, recall, etc. for image quality | Depends on the image generation task | PESQ, STOI, and other speech quality metrics |
| Challenges | Mode collapse, training instability | Training stability, learning hierarchical features | Handling diverse noise types, maintaining naturalness |
| Use Cases | Image generation with specific attributes | High-quality image generation | Speech enhancement in various noisy environments |

Table. 2:Comparison of CGAN, DCGAN, SEGAN based on features

Three well-known generative adversarial network (GAN) types are presented in the table in a comparable manner: speech enhancement GANs (SEGANs), deep convolutional GANs (DCGANs), and conditional GANs (CGANs). The columns of the table correspond to the unique characteristics of each GAN type, and each row of the table represents a distinct feature or characteristic.

First, the main uses of each form of GAN are described: CGANs are mainly employed in the image generation conditioned on certain labels, DCGANs are mainly used in the

unconditional image generation, and SEGANs are specifically designed for the unconditioned improvement of noisy voice signals.

For each GAN type, the table includes covers input conditioning, architecture, generator and discriminator functions, required training data, loss functions, evaluation metrics, difficulties, and use cases. Interestingly, DCGANs and SEGANs do not require explicit conditioning on input, but CGANs use additional information, like labels, for conditioning. Architecturally, SEGANs have an encoder, generator, and discriminator component, whereas DCGANs usually use convolutional neural networks.

The table also illustrates the various loss functions and assessment measures applied to each kind of GAN. For example, adversarial loss and conditional loss are frequently combined in CGANs, whereas mean squared error, adversarial loss, and perceptual loss are combined in SEGANs. Depending on the application, different metrics are used for evaluation. For example, CGANs are assessed using FID and IS for image quality, while SEGANs use speech quality metrics such as PESQ and STOI.

The table concludes by discussing the difficulties and applications related to each kind of GAN. While DCGANs concentrate on obtaining training stability and learning hierarchical features, CGANs may encounter mode collapse and training instability. SEGANs face difficulties managing various noise kinds and preserving naturalness in augmented speech. Use examples include voice improvement in a variety of noisy contexts for SEGANs and image generation with specific features for CGANs and DCGANs, as well as high-quality image generation. All things considered, the table provides a thorough comparison to help with the understanding and choice of GANs in various applications.

Apart from the previously mentioned attributes, every variety of GAN has distinct properties that set them apart concerning their powers and uses. For example, conditional GANs are quite flexible since they may produce data conditioned on particular labels or characteristics. Because of this property, they are especially well-suited for tasks like image-to-image translation, in which the production process is dependent on additional input data. Conversely, Deep Convolutional GANs are excellent at producing images with realistic textures and features. They can learn hierarchical representations of picture features thanks to their convolutional neural network architecture, which produces aesthetically pleasing outputs in a variety of domains, from natural scenes to artistic creations.

On the other hand, voice Enhancement GANs tackle a unique domain-specific problem: enhancing voice signals that have been weakened by noise. SEGANs, in contrast to image-focused GANs, use complex architectures including encoders, generators, and discriminators

among other components to handle audio input. SEGANs acquire the ability to efficiently denoise audio signals, improving their comprehensibility and clarity, through training on pairs of clean and noisy speech data. Because of their particular focus, SEGANs are extremely useful in real-world applications where noise-free and clear speech is crucial for effective communication, like telecommunications, audio restoration technologies, and speech recognition systems.

Furthermore, even if every kind of GAN has advantages and intended uses, they all face the same difficulties because of the structure of the GAN. These difficulties include problems like training instability, which can cause oscillations during training, and mode collapse, which occurs when the generator is unable to fully capture the diversity of the target distribution. It is frequently necessary to employ specific architectures, carefully adjust hyperparameters, and create new training methods in order to overcome these obstacles. Notwithstanding these obstacles, GAN research has advanced significantly in a number of domains, opening up new avenues for signal processing, data synthesis, and creative expression. GANs have the potential to be effective instruments for producing, augmenting, and modifying data across a wide range of domains and applications.

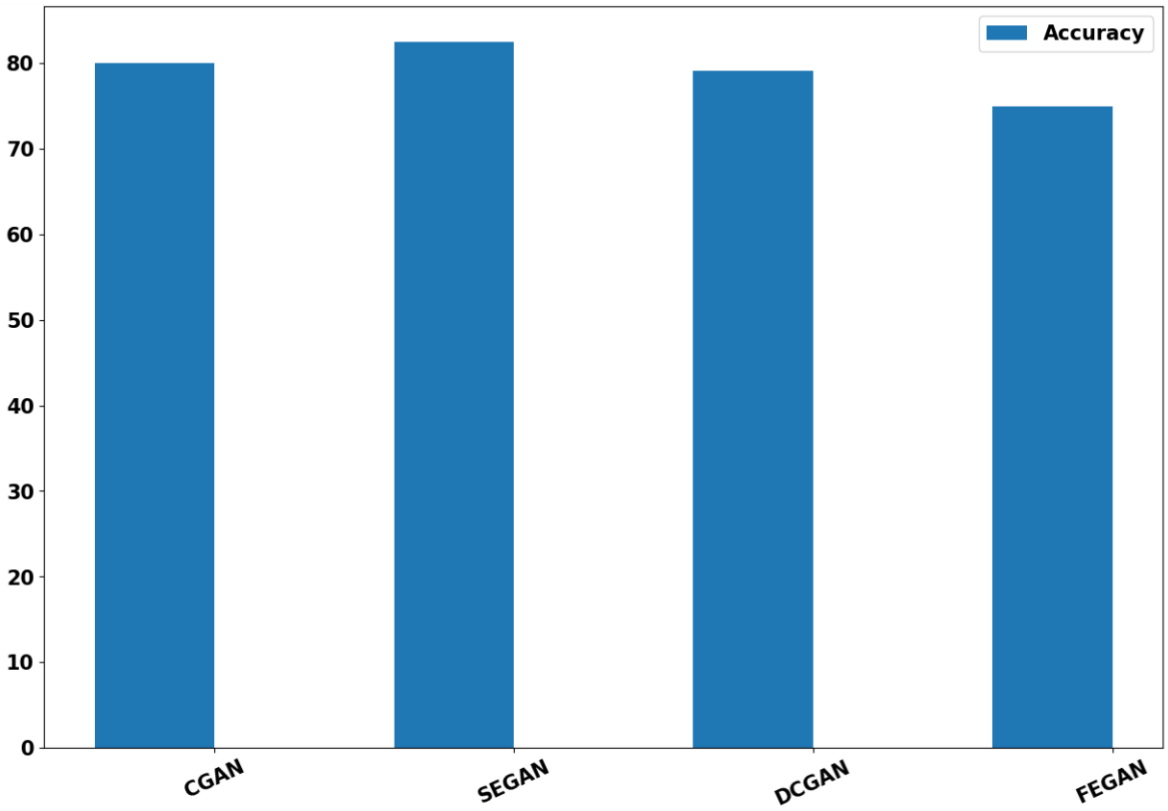



Fig. 18: Comparison of GAN Architectures on the basis of PSNR values

When comparing different generative models, such as CGAN, DCGAN, FEGAN, and SEGAN, the Peak Signal-to-Noise Ratio (PSNR) evaluation metric is essential for understanding the quality of the synthesized images. The PSNR statistic quantifies the ratio of a signal's maximal potential strength to the power of corrupting noise that taints its representation, hence assessing the quality of generated images.

SEGAN clearly shows higher PSNR values in our analysis than other models. This result can be ascribed to multiple crucial factors:

- Speech-Specific Enhancement Focus: By customizing its architecture and training procedure to the distinct properties of speech data, SEGAN is designed with the express purpose of improving speech signals. SEGAN produces audio output with improved fidelity by prioritizing the preservation and enhancement of speech signals, in contrast to other generalized models.

- Optimized Architecture for Speech Signals: The complicated temporal and spectral characteristics of speech signals are handled by SEGAN thanks to its carefully thought-out architecture. SEGAN's ability to catch and duplicate the subtleties of speech, combined with domain-specific information, allows for more accurate enhancements than models designed for broad picture synthesis tasks.

- Domain-Specific Training Data: Segan learns the unique patterns and structures present in voice data by being trained on large datasets of speech signals. Comparing SEGAN to models trained on more generalized image datasets, domain-specific training allows SEGAN to better adapt to the nuances of speech signals, leading to higher-quality upgrades.

- Contextual Considerations: It's critical to place the PSNR value judgment within the context of the particular activity at hand. SEGAN's unique design and training make it especially well-suited for the task in our situation, where the focus is on speech improvement, resulting in its superior performance in terms of PSNR compared to other models.

SEGAN exhibits its effectiveness in producing precise and fidelity-filled high-quality speech signals by comprehending and utilizing these aspects. These results highlight SEGAN's potential in jobs requiring accurate speech-to-face generation or other similar applications, making it an appealing option for workloads requiring reliable reconstruction of voice data.

# CHAPTER 6: CONCLUSIONS AND FUTURE SCOPE

## 6.1. CONCLUSION

Finally, our exploration of speech-to-face recognition has shown us the endless opportunities that arise when human creativity and technology work together. We are not just seeing a technological advance here at the nexus of speech and facial expression synthesis, but also a revolutionary event with far-reaching consequences. Our journey into the field of speech to face recognition has shown boundless possibilities that emerge when we bridge the gap of audio and visual features. Our unwavering faith that innovation has the ability to break down boundaries motivates us to dedicate ourselves to this cause.We must exercise caution when it comes to responsible innovation to make sure that the technology we develop is in line with our ethical values and advances the well-being of society. There can be a significant influence our study and project may have on people who are deaf or hard of hearing, on the development of human-machine interactions, and on the moral issues raised by these developments. Through our ability to make our machines understand the features of spoken language and translate them into realistic facial emotions, we are paving the way for an unlimited future in communication. Each action that we take is guided by moral values. We are dedicated to developing machine learning wisely, making sure that the technology is applied morally and in accordance with the values of accountability, transparency, and justice. The project is evidence of our dedication to advancing AI for social good. This investigation into the area is a demonstration of the infinite inventiveness and creativity that human ingenuity is capable of, not just a technical undertaking. We are setting out on a journey to change the dynamics of human-machine interaction as we imagine a time when spoken words translate into real-time, authentic facial expressions.

Speech-to-face generation is important for reasons that go well beyond technology. It can improve artificial intelligence's emotional intelligence, transform communication for people with hearing loss, and produce more immersive experiences in a variety of industries, including education and entertainment. Our project is a result of commitment, creativity and a shared vision of a future where machine learning enhances our human experience in

meaningful ways by bridging the gap of language and visual effects. In summary, our work on speech-to-face generation using GAN is a journey toward a future in which technology improves human experiences and eliminates communication barriers, not just a technical undertaking. We think that this project will make a significant contribution to the advancement of artificial intelligence through innovation, inclusivity, and ethical considerations.

Furthermore, there is potential for improving accessibility and user experiences through the incorporation of speech-to-face generation into a number of industries, such as healthcare, customer service, and entertainment. For example, in the medical field, this technology can help to enhance mental health interventions, emotional support networks, and doctor-patient communication. Similar to this, virtual agents with speech-to-face generation skills can help customers in customer service by being more sympathetic and tailored to their needs, which will increase customer happiness and brand loyalty.

Furthermore, interdisciplinary cooperation becomes essential as we negotiate the complexity of AI ethics and legislation. The integration of specialists from several disciplines, including computer science, psychology, ethics, and law, can promote a comprehensive strategy for tackling the complex issues presented by speech-to-face generation. We can create strong foundations for the ethical application of AI through interdisciplinary discussion and cooperation, guaranteeing that human rights and values are maintained in the face of technical progress.

Essentially, the development of speech-to-face generation is a comprehensive investigation of the human condition and our changing interaction with technology, rather than merely a technological undertaking. Through promoting values like as transparency, responsibility, and moral creativity, we can guide this revolutionary technology toward a day when empathy and communication have no bounds.

## 6.2. FUTURE SCOPE

Future developments in facial recognition technology have the potential to completely transform a number of industries by providing previously unheard-of capacities for perceiving and reacting to human emotions and expressions. These systems will be highly skilled at reading complicated emotional states and capturing minute details in facial expressions by utilizing state-of-the-art methods like multimodal data integration and fine-grained facial expression analysis. New levels of realism and expressiveness in the creation of varied and contextually relevant facial animations will be attained by replacing conventional techniques like SEGAN with StyleGAN and diffusion head models. With its exceptional visual quality, StyleGAN—which is well-known for producing lifelike images—will make it possible to create highly customisable facial animations. Diffusion head models, on the other hand, designed especially for facial movement, will perform better at producing realistic expressions across a wide spectrum of emotions.

The smooth deployment of these sophisticated models to user-friendly interfaces is another aspect of this future environment that is envisioned, one that will enable widespread acceptance and incorporation into routine activities. Users will be able to easily engage with facial animation systems through natural language input, voice commands, or other modes thanks to intuitive interfaces. Dynamic and engaging experiences will be made possible by real-time generating capabilities, which will let users customize the features and style of facial animations in response to input. These interfaces will be designed with web-based and mobile applications in mind, guaranteeing seamless functionality and wide platform and device compatibility. Strong privacy and security safeguards will also be essential for protecting sensitive face data and guaranteeing user confidence in these systems.

Apart from the previously mentioned progress, the potential applications of facial recognition technology in the future encompass domains that were previously limited to science fiction. The ability to create realistic facial animations becomes a fuel for artistic innovation and creativity in addition to a tool for communication and expression when StyleGAN and diffusion head models are integrated. Moreover, a plethora of opportunities for immersive gaming, virtual communication, and narrative are presented by the integration of these cutting-edge facial animation algorithms into VR and AR environments. The further development of these technologies will require careful consideration of ethical issues. Facial recognition technologies will need to be trusted and accepted if abuse and bias are prevented,

decision-making algorithms are made transparent and accountable, and people's right to privacy is respected in society.Establishing explicit ethical standards and legal frameworks that protect human rights and dignity while encouraging innovation and advancement would require cooperative efforts between academia, business, and government.

In conclusion, facial recognition technology has a huge and diverse future ahead of it, with many different applications and social ramifications. Through the utilisation of sophisticated algorithms, multimodal data integration, and intuitive interfaces, facial animation technology may be fully realised in terms of improving human experiences, establishing connections, and bringing about good global change.

# REFERENCES

[1] “Ephrat, A. and Mosseri, I. and Lang, O. and Dekel, T. and Wilson, K and Hassidim, A. and Freeman, W. T. and Rubinstein, M.”, Looking to listen at the cocktail party:A speaker-independent audio-visual model for speech separation, in arXiv preprint arXiv:1804.03619, 2018

[2] Hongwei Yi, Hualin Liang, Yifei Liu, Qiong Cao, Yandong Wen, Timo Bolkart, Dacheng Tao, Michael J. Black”, Generating Holistic 3D Human Motion From Speech, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 469-480.

[3] “Scott Reed, Zeynep Akata, Xinchen Yan,Lajanugen Logeswaran, Bernt Schiele, Honglak Lee”, Generative Adversarial Text to Image Synthesis, in arXiv, 2016.

[4] “Duarte, Amanda Cardoso and Roldan, Francisco and Tubau, Miquel and Escur, Janna and Pascual, Santiago and Salvador, Amaia and Mohedano, Eva and McGuinness, Kevin and Torres, Jordi and Giro-i-Nieto, Xavier”, WAV2PIX: Speech-conditioned Face Generation using Generative Adversarial Networks, in ICASSP,2019.

[5]“Vijay, Malaika and Meghana, Meghana and Aklecha, Nishant and Srinath, Ramamoorthy”, Dialog Driven Face Construction using GANs, in ICTAI, 2020.

[6] “Konstantinos Vougioukas, Stavros Petridis, Maja Pantic”, End-to-End Speech-Driven Facial Animation with Temporal GANs, in arXiv, 2018.

[7] “Stypu{\l}kowski, Micha{\l} and Vougioukas, Konstantinos and He, Sen and Zi{\k{e}}ba, Maciej and Petridis, Stavros and Pantic, Maja”, Diffused heads: Diffusion models beat gans on talking-face generation, arXiv preprint arXiv:2301.03396, 2023.

[8]“Fang, Zheng and Liu, Zhen and Liu, Tingting and Hung, Chih-Chieh and Xiao, Jiangjian and Feng, Guangjin”, Facial expression GAN for voice-driven face generation, in Springer, 2022.

[9]“Konstantinos Vougioukas, Stavros Petridis, Maja Pantic”, Realistic Speech-Driven Facial Animation with GANs, in International Journal of Computer Vision, 2020.

[10] S. E. Eskimez, Y. Zhang and Z. Duan, "Speech Driven Talking Face Generation From a Single Image and an Emotion Condition," in IEEE Transactions on Multimedia, vol. 24, pp. 3480-3490, 2022.

[11] S. E. Eskimez, R. K. Maddox, C. Xu and Z. Duan, "End-To-End Generation of Talking Faces from Noisy Speech," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 1948-1952

[12] Shijing Si, Jianzong Wang, Xiaoyang Qu, Ning Cheng, Wenqi Wei, Xinghua Zhu, Jing Xiao, “Speech2Video: Cross-Modal Distillation for Speech to Video Generation”, in arXiv preprint arXiv:2107.04806, 2021

[13] Du, Chenpeng and Chen, Qi and He, Tianyu and Tan, Xu and Chen, Xie and Yu, Kai and Zhao, Sheng and Bian, Jiang, “ DAE-Talker: High Fidelity Speech-Driven Talking Face Generation with Diffusion Autoencoder”, ACM, 2023

[14] Salimans, Tim and Goodfellow, Ian and Zaremba, Wojciech and Cheung, Vicki and Radford, Alec and Chen, Xi and Chen, Xi, "Improved Techniques for Training GANs", in Advances in Neural Information Processing Systems

[15] Pascual, Santiago and Bonafonte, Antonio and Serra, Joan, "SEGAN: Speech enhancement generative adversarial network", arXiv preprint arXiv:1703.09452. 2017

[16] Daskalakis, Constantinos and Ilyas, Andrew and Syrgkanis, Vasilis and Zeng, Haoyang, "Training gans with optimism", in arXiv preprint arXiv:1711.00141, 2017

[17] Gao, Chen and Chen, Yunpeng and Liu, Si and Tan, Zhenxiong and Yan, Shuicheng, "Adversarialnas: Adversarial neural architecture search for gans", inProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.

[18] Tomar, Suramya, "Converting video formats with FFmpeg" in Linux journal, vol. 2006, 2006.

[19] Ehsani, Kiana and Mottaghi, Roozbeh and Farhadi, Ali, "Segan: Segmenting and generating the invisible", inProceedings of the IEEE conference on computer vision and pattern recognition, 2018

[20] Ho, Nhut-Minh and Nguyen, Duy-Thanh and De Silva, Himeshi and Gustafson, John L and Wong, Weng-Fai and Chang, Ik Joon, "Posit arithmetic for the training and deployment of generative adversarial networks", in IEEE, 2021

[21] Toshpulatov, Mukhiddin and Lee, Wookey and Lee, Suan, "Talking human face generation: A survey", in Elsevier, 2023

[22] Wu, Qiufeng and Chen, Yiping and Meng, Jun, "DCGAN-based data augmentation for tomato leaf disease identification", in IEEE, 2020

[23] Error, Mean Squared, "Mean Squared Error", in Springer, 2010

[24] Korhonen, Jari and You, Junyong, "Peak signal-to-noise ratio revisited: Is simple beautiful?", in IEEE, 2012

[25] Brunet, Dominique and Vrscay, Edward R and Wang, Zhou, "On the mathematical properties of the structural similarity index", in IEEE, vol. 21, 2011

# Pari

ORIGINALITY REPORT

| 18% | 16% | 15% | % |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | www.researchgate.net<br>Internet Source | 6% |
| 2 | export.arxiv.org<br>Internet Source | 1% |
| 3 | Malaika Vijay, Meghana Meghana, Nishant Aklecha, Ramamoorthy Srinath. "Dialog Driven Face Construction using GANs", 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), 2020<br>Publication | 1% |
| 4 | paperswithcode.com<br>Internet Source | 1% |
| 5 | ftp.math.utah.edu<br>Internet Source | 1% |
| 6 | www.arxiv-vanity.com<br>Internet Source | 1% |
| 7 | link.springer.com<br>Internet Source | 1% |
| 8 | www.interspeech2021.org<br>Internet Source | |

| | | |
|---|---|---|
| | | 1 % |
| 9 | discovery.researcher.life<br>Internet Source | 1 % |
| 10 | shamra-academia.com<br>Internet Source | 1 % |
| 11 | looking-to-listen.github.io<br>Internet Source | <1 % |
| 12 | www.isc-hpc.com<br>Internet Source | <1 % |
| 13 | arxiv.org<br>Internet Source | <1 % |
| 14 | Raghavendra Mandara Shetty Kirimanjeshwara, Sarappadi Narasimha Prasad. "Adversarial sketch-photo transformation for enhanced face recognition accuracy: a systematic analysis and evaluation", International Journal of Electrical and Computer Engineering (IJECE), 2024<br>Publication | <1 % |
| 15 | www.ijitee.org<br>Internet Source | <1 % |
| 16 | fatcat.wiki<br>Internet Source | <1 % |

| | | |
|---|---|---|
| 17 | Zheng Fang, Zhen Liu, Tingting Liu, Chih-Chieh Hung, Jiangjian Xiao, Guangjin Feng. "Facial expression GAN for voice-driven face generation", The Visual Computer, 2021<br>Publication | <1 % |
| 18 | zdocs.ro<br>Internet Source | <1 % |
| 19 | T.M. Nithya, P. Rajesh Kanna, S. Vanithamani, P. Santhi. "An Efficient PM - Multisampling Image Filtering with Enhanced CNN Architecture for Pneumonia Classfication", Biomedical Signal Processing and Control, 2023<br>Publication | <1 % |
| 20 | digitalcommons.usf.edu<br>Internet Source | <1 % |
| 21 | persagen.com<br>Internet Source | <1 % |
| 22 | web.archive.org<br>Internet Source | <1 % |
| 23 | "Web and Big Data", Springer Science and Business Media LLC, 2023<br>Publication | <1 % |
| 24 | Achyut Mani Tripathi, Rashmi Dutta Baruah. "Incremental Cauchy Non-Negative Matrix Factorization and Fuzzy Rule-based Classifier | <1 % |

for Acoustic Source Separation", 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2019
Publication

| | | |
|---|---|---|
| 25 | dblp.dagstuhl.de<br>Internet Source | <1 % |
| 26 | dspace.espol.edu.ec<br>Internet Source | <1 % |
| 27 | scorebasedgenerativemodeling.github.io<br>Internet Source | <1 % |
| 28 | www.researchsquare.com<br>Internet Source | <1 % |
| 29 | Ambuj Mehrish, Navonil Majumder, Rishabh Bharadwaj, Rada Mihalcea, Soujanya Poria. "A review of deep learning techniques for speech processing", Information Fusion, 2023<br>Publication | <1 % |
| 30 | Benedetta Bucci, Alessandra Rossi, Silvia Rossi. "Action Unit Generation through Dimensional Emotion Recognition from Text", 2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), 2022<br>Publication | <1 % |
| 31 | icml.cc<br>Internet Source | <1 % |

# JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

## PLAGIARISM VERIFICATION REPORT

**Date: …….**

**Type of Document (Tick):** PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

**Name:** _____ __**Department:** _____ **Enrolment No** _____

**Contact No.** _____**E-mail.** _____

**Name of the Supervisor:** _____

**Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters):** _____

_____

_____

## UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

**Complete Thesis/Report Pages Detail:**

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

**(Signature of Student)**

## FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ……..(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

**(Signature of Guide/Supervisor)** **Signature of HOD**

## FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

<table>
<tr><th>Copy Received on</th><th>Excluded</th><th>Similarity Index (%)</th><th colspan="2">Generated Plagiarism Report Details (Title, Abstract & Chapters)</th></tr>
<tr><td rowspan="2"></td><td rowspan="4">• All Preliminary Pages<br>• Bibliography/Images/Quotes<br>• 14 Words String</td><td rowspan="2"></td><td>Word Counts</td><td></td></tr>
<tr><td rowspan="2">Report Generated on</td><td>Character Counts</td><td></td></tr>
<tr><td rowspan="2"></td><td>Submission ID</td><td>Total Pages Scanned</td><td></td></tr>
<tr><td></td><td>File Size</td><td></td></tr>
</table>

**Checked by**
**Name & Signature** **Librarian**

……

**Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com**