Dissertation on

"CHARACTER GENERATION"

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology
in
Computer Science & Engineering

UE20CS461A – Capstone Project Phase - 2

Submitted by:

Mitul Joby	**PES2UG20CS199**
Nihal Chengappa P.A	**PES2UG20CS224**
Nilesh Ravichandran	**PES2UG20CS225**
Pranav Rao Rebala	**PES2UG20CS248**

Under the guidance of

Prof. Nazmin Begum
Assistant Professor
PES University

June - Nov 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

FACULTY OF ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

'CHARACTER GENERATION'

is a bonafide work carried out by

Mitul Joby	**PES2UG20CS199**
Nihal Chengappa P.A	**PES2UG20CS224**
Nilesh Ravichandran	**PES2UG20CS225**
Pranav Rao Rebala	**PES2UG20CS248**

In partial fulfilment for the completion of seventh semester Capstone Project Phase - 2 (UE20CS461A) in the Program of Study -Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period June 2023 – Nov. 2023. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 7th semester academic requirements in respect of project work.

Signature Prof. Nazmin Begum Assistant Professor	Signature Dr. Sandesh B J Chairperson	Signature Dr. B K Keshavan Dean of Faculty

External Viva

Name of the Examiners	**Signature with Date**
1. _____	_____
2. _____	_____

DECLARATION

We hereby declare that the Capstone Project Phase - 2 entitled **"Character Generation"** has been carried out by us under the guidance of Prof. Nazmin Begum, Assistant Professor and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester June – Nov. 2023. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

PES2UG20CS199 Mitul Joby _____

PES2UG20CS224 Nihal Chengappa P A _____

PES2UG20CS225 Nilesh Ravichandran _____

PES2UG20CS248 Pranav Rao Rebala _____

ACKNOWLEDGEMENT

I would like to express my gratitude to Prof. Nazmin Begum, Department of Computer Science and Engineering, PES University, for her continuous guidance, assistance, and encouragement throughout the development of this UE20CS461A - Capstone Project Phase – 2.

I am grateful to the Capstone Project Coordinator, Dr. Sarasvathi V, Professor and Dr. Sudeepa Roy Dey, Associate Professor, for organizing, managing, and helping with the entire process.

I take this opportunity to thank Dr. Sandesh B J, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department. I would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

I am deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University and Prof. Nagarjuna Sadineni, Pro-Vice Chancellor, PES University, for providing to me various opportunities and enlightenment every step of the way. Finally, this project could not have been completed without the continual support and encouragement I have received from my family and friends.

ABSTRACT

In the evolving landscape of computer graphics, this project introduces a transformative method for creating 3D characters from textual descriptions. Dual Stable Diffusion pipelines generate 2D images corresponding to textual attributes for faces and bodies. These images are then converted into 3D models - the DECA model processes the facial image for detailed mesh and texture, hence generating a 3D face, while the PiFuHD model transforms the body image for accurate body mesh generation. This innovative approach ensures realistic and diverse character synthesis, matching specific features such as age, gender, and certain other physical traits. The effectiveness of the method is evident in its visual quality, hence setting a new standard in the realm of character design and generation.

The 3D character synthesis application is deployed on a React website, providing an intuitive and user-friendly interface. The React frontend enhances the user experience by offering a responsive and interactive platform for generating and customizing 3D characters. To facilitate the backend operations, a Flask backend is implemented, which receives requests from the user, communicates with the database and requests the model server thereby allowing users to experience a fluid and responsive character design process.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

There is no doubt that in recent years there has been an increase in demand for 3D characters that are not only highly realistic but also highly customizable. This increased demand is spreading across a variety of industries, including virtual reality, gaming, film, and simulation. The challenge at the forefront of computer graphics and artificial intelligence is to create 3D characters that not only fit predetermined attributes, but also have distinct personalities and subtle expressions. This article introduces an innovative 3D character generation method that focuses on compositing characters based on given attributes.

The innovative approach described in this article not only opens new possibilities for creative expression, but also promises practical applications in a variety of fields. The core of these innovative methods comes from the recognition of the inherent limitations of popular character generation technologies. These existing approaches are often based on predefined models or procedural algorithms and often struggle with the complex task of fully capturing the complex richness and diversity of facial attributes.

In response, the methodology presented in this report aims to break new ground in the field of 3D character generation. The overall goal is to provide content creators, developers, and designers with a robust toolset that goes beyond the limitations typically associated with traditional methods. At the heart of this toolset is the ability for users to create characters with an unprecedented level of precision, embodying specific subtle features such as distinct facial features and carefully defined body proportions. A transformative feature that allows you to create fine-tuned characters.

Although the presented project aims to address certain limitations within the current paradigm, it is just one of many potential advances in his ongoing 3D character generation development. Recognizing a project's specific contribution in the identified context allows its potential impact to be assessed without overstating its broader impact.

Fundamentally, this project contributes to the ongoing debate on character generation, aiming to strike a balance between innovation and a sound understanding of its role in the larger landscape of 3D graphics and artificial intelligence.

In addition to these features, the system also provides users with the ability to store and access rendered models through a secure sign-in/login mechanism. This additional feature greatly improves the user experience and provides a seamless workflow for managing and redisplaying previously created 3D representations.

In summary, this project presents an innovative approach that not only addresses the growing demand for realistic and customizable 3D characters but also has the potential to reshape the landscape of character generation in computer graphics and artificial intelligence.

CHAPTER 2

PROBLEM STATEMENT

The aim of this project is to develop a system that takes user-provided textual descriptions and turns them into realistic 3D facial representations. There is currently a gap in the field of creating 3D facial models from text descriptions. It takes a lot of effort and time to create 3D facial models requiring professional experience, hence the procedure has traditionally been difficult. To address these issues, the proposed system leverages generation of highly realistic 3D facial models from textual descriptions. The system provides a user-friendly interface that allows users to easily input text and customize the generated 3D model. Additionally, the system is designed to handle errors effectively and provide useful error messages to users in case of input errors or technical issues. Overall, the proposed system aims to provide a highly accurate, efficient, and user-friendly solution for generating realistic 3D facial models from textual descriptions, with potential applications in industries such as entertainment, virtual reality, and facial recognition.

CHAPTER 3

LITERATURE REVIEW

3.1. Tedigan: Text-guided diverse face image generation and manipulation[1]

Description

The objective of their model was to generate a 2d image from a given textual description. TediGAN model which is a novel framework developed by them using StyleGAN which is pretrained on FFHQ dataset.

Advantages

Able to generate high resolution at 1024. Compared with ManiGAN, this method achieves better FID, accuracy, and realism.

Limitation

Some unrelated attributes are unwantedly changed when you manipulate a given image according to a text description. Some attributes, such as hats, necklaces and earrings, are not well represented in its latent space.

3.2. A Realistic Image Generation of Face From Text Description Using the Fully Trained Generative Adversarial Networks[2]

Description

In this paper, the authors propose a fully trained generative adversarial network that simultaneously trains text encoders and image decoders to generate more accurate and efficient results. They also created a dataset by combining LFW, CelebA, and locally prepared data and labelled it according to defined classes. The authors conducted experiments that demonstrate that their proposed approach outperforms existing methods in generating high-quality face images according to input sentences. CelebA and LFW datasets are used.

Advantages

Optimization of joint learning (of text and image) is very high when compared to other models due to the presence of two discriminators. Realistic facial images could be generated. Two discriminators are proposed to utilize the strength of joint learning. Proposed model achieved a FSD score of 1.118 and FID score of 42.62 which is better than most models.

Limitation

Dataset used has only five categories so classification is done on a smaller number of factors. Accuracy is decent but can be improved.

3.3. Faces à la Carte: Text-to-Face Generation via Attribute Disentanglement[3]

Description

This paper proposes a Text-to-Face model that can generate high-resolution images of faces from free-form text descriptions with text-to-image consistency. The model can produce multiple diverse faces to cover a wide range of unspecified facial features, using fine-tuned multi-label classifiers and image encoders. CelebA dataset is used. StyleGAN2 is used as the generator to map the manipulated noise vectors into the disentangled feature space. A pre-trained BERT multi-label model is used for NLP.

Advantages

From both qualitative and quantitative evaluative comparisons, the generated images exhibit good image quality, text-to-image similarity, and image diversity. Has high image diversity.

Limitation

Model is not entirely robust- some are more consistent while some are less consistent; caused by insufficient accuracy of the text classifier and image encoder due simply to the lack of training data. Features in the latent space are not well disentangled.

3.4. Text to Face generation using Wasserstein stackGAN[4]

Description

The paper proposes a new model for generating high-resolution face images from textual descriptions. The model uses a stack architecture with two stages, and Wasserstein loss is used to stabilize training for both stages. Multiple text embeddings are provided for each real face dataset during training at different epochs. The proposed model outperforms models like AttnGAN and DF-GAN in terms of FID score and is comparable to TediGAN, the state-of-the-art architecture for text-to-face generation. The paper addresses the challenges of generating high-quality face images from textual descriptions and presents a promising solution to this problem.

Advantages

The proposed method involves using a Wasserstein stackGAN to generate the initial face image from the text description and then refining the image using a Wasserstein GAN. Wasserstein GANs have been shown to be more stable and effective in generating high quality images than traditional GANs. The proposed method is shown to produce more realistic images than previous methods, suggesting that it may be an improvement over existing text-to-face generation methods.

Limitation

The FID score of 109.63 may indicate that there is room for improvement, it should be interpreted in the context of the specific dataset and experimental setup used in the paper. The Wasserstein stackGAN model used in the proposed method is computationally expensive and requires significant computing resources for training and inference. This may limit its applicability in practical applications.

3.5. MeInGame: Create a Game Character Face from a Single Portrait[5]

Description

Given an input photo, a pre-trained shape reconstructor predicts the 3DMM and pose coefficients and a shape transfer module transforms the 3DMM shape to the game mesh while keeping their topology. Then, a coarse texture map is created by unwrapping the input image to UV space based on the game mesh. The texture is further refined by a set of encoder and decoder modules. We also introduce a lighting regressor to predict lighting coefficients from image features. Finally, the predicted shape, texture, together with the lighting coefficients, are fed to a differentiable renderer, and we force the rendered output similar to the input photo. Two discriminators are introduced to further improve the results.

Advantages

Able to generate a complete head model from a 2d image as opposed to just a 3D face mask. Does not lose the features of the original 2D image and maintains its nature.

Limitation

The same image under light of different colours will produce different models. When there are heavy occlusions (e.g., the hat), our method fails to produce faithful results since the renderer fails to model the shadow created by the objects outside the head mesh.

3.6. FaceScape: A Large-scale High Quality 3D Face Dataset and Detailed Riggable 3D Face Prediction[6]

Description

The authors present a large-scale detailed 3D face dataset, FaceScape, and propose a novel algorithm that is able to predict elaborate riggable 3D face models from a single image input. They captured the 3D face model using a dense 68-camera array under controlled illumination, which recovers the 3D face model with wrinkle and pore level

detailed shape. In addition to shape quality, the dataset provides a considerable number of scans for study. 938 people between the ages of 16 and 70 as subjects, and each subject is guided to perform 20 specified expressions, generating 18,760 high quality 3D face models.

There is a problem of recovering an animatable model from a single image. They demonstrate for the first time that a detailed and rigged 3D face model can be recovered from a single image. The rigged model exhibits expression-depended geometric details such as wrinkles. The pipeline to predict a detailed riggable 3D face from a single image consists of three stages: base model fitting, displacement map prediction, and dynamic details synthesis. They used 888 people in their dataset as training data with a total of 17760 displacement maps, leaving 50 people for testing.

Advantages

Main advantages of the FaceScape dataset is its large size and high quality, which enables researchers and developers to train and evaluate facial recognition, animation, and other applications. Additionally, the detailed annotations of facial landmarks and semantic regions enable more accurate and precise modelling and manipulation of facial expressions. It showed that their results recover 3D faces with photo-realistic details. The faces can be further rigged to other expressions, and the details in the new expressions are synthesized to make the rigged model plausible.

Limitation

One potential disadvantage of the dataset is that it may be biased towards certain demographic groups, as it was collected from a specific population in China. Additionally, the dataset may not capture the full range of facial diversity and expressions present in other populations, which may limit its generalizability to other contexts.

3.7. FENeRF: Face Editing in Neural Radiance Fields[7]

Description

There are two types of methods to generate portrait images: 2D GANs and 3D-aware GANs. 2D GANs produce high-quality images, but they are not consistent when viewed from different angles. 3D-aware GANs maintain consistency, but their images cannot be easily edited. To overcome these limitations, FENeRF is proposed, which is a 3D-aware generator that can produce consistent and editable portrait images. FENeRF uses two separate codes to generate facial details and texture in a 3D volume, which can be edited using GAN inversion. This 3D representation can be learned from simple image and mask pairs. It outperforms other methods in various face editing tasks. FENeRF: Face Editing in Neural Radiance Fields is a method for generating realistic and editable 3D portraits from monocular images. It uses a neural radiance field (NeRF) to represent the 3D geometry and appearance of a face, and a GAN inversion technique to edit the face via semantic masks. It can produce high-quality portraits with view consistency and local editability. It works by using two decoupled latent codes to generate facial semantics and texture in a 3D volume with shared geometry. It uses a neural radiance field (NeRF) to represent the 3D face as a continuous function that maps 3D coordinates and viewing directions to colours and densities. It also uses a GAN inversion technique to edit the face via semantic masks that control different facial attributes such as hair, eyes, nose, mouth, etc.

Advantages

It can produce view-consistent and locally editable portrait images. It can jointly render the boundary-aligned image and semantic mask. It can use widely available monocular image and semantic mask pairs as input. It can preserve high-frequency details in the generated image.

Limitation

It cannot handle occlusions or extreme poses well. It may produce artifacts or distortions when editing faces with large changes. It requires a large amount of training

data with high-quality face images and semantic masks. The generator cannot produce HD portrait images due to the computationally expensive ray casting and volume integration.

3.8. AvatarMe: Realistically Renderable 3D Facial Reconstruction "In-the-Wild"[8]

Description

The paper introduces AvatarMe, a new method for reconstructing high-resolution photorealistic 3D faces from single "in-the-wild" images. This method captures a large dataset of facial shape and reflectance and uses a state-of-the-art 3D texture and shape reconstruction method to refine its results while generating per-pixel diffuse and specular components required for realistic rendering. AvatarMe outperforms existing methods significantly and can produce authentic, 4K by 6K-resolution 3D faces from a single low-resolution image that bridges the uncanny valley.

Advantages

AvatarMe outperforms the existing arts by a significant margin and reconstructs authentic, 4K by 6K-resolution 3D faces from a single low-resolution image that, for the first time, bridges the uncanny valley. They capture faces of over 200 individuals of different ages and characteristics under 7 different expressions for more precision.

Limitation

Even though the dataset contains a relatively large number of subjects, it does not contain sufficient examples of subjects from certain ethnicities.

3.9. High-Resolution Image Synthesis with Latent Diffusion Models[9]

Description

Normally for image formation from text using a diffusion model by decomposing the image formation process into a sequential application of denoising autoencoders, diffusion models (DMs) achieve state-of-the-art synthesis results on image data and beyond.

However, this process takes a lot of time and inference is expensive. In this paper to enable Diffusion Model training on limited computational resources while retaining their quality and flexibility they have applied them in the latent space of powerful pretrained autoencoders. In contrast to previous work, training diffusion models on such a representation allows for the first time to reach a near-optimal point between complexity reduction and detail preservation, greatly boosting visual fidelity.

Advantages

By leaving the high-dimensional image space, they have obtained diffusion models which are computationally much more efficient because sampling is performed on a low-dimensional space. Moreover, they exploit the inductive bias of diffusion models inherited from their UNet architecture, which makes them particularly effective for data with spatial structure and therefore alleviates the need for aggressive, quality-reducing compression levels as required by previous approaches.

Limitation

While latent diffusion models significantly reduce computational requirements compared to pixel-based approaches, their sequential sampling process is still slower than that of GANs.

3.10. PIFuHD: Multi-Level Pixel-Aligned Implicit Function for High-Resolution 3D Human Digitization[10]

Description

PIFuHD is a complex deep learning technique which leverages 2D images to create high-resolution 3D digital models of human subjects. It is an improvement over the initial PIFu model, which was useful but had difficulty capturing minute parameters. In principle, PIFu generates a 3D model from a single 2D image by learning a mapping between 2D pixels in an image and their associated 3D positions. PIFuHD improves upon this model by adding multi-level feature maps and optimizing the implicit function to yield reconstructions that are more precise and comprehensive.

Advantages

- **Fine Detail Preservation:** PIFuHD excels at preserving minute details in 3D reconstructions, such as clothing folds, facial expressions, and other subtle surface features. To create high-fidelity and authentic digital representations of human subjects, this is essential.
- **High Resolution:** PIFuHD can handle reconstructions with a high degree of resolution. In applications where minute details are crucial, such as virtual reality, medical imaging, and visual effects for cinema and television, this is especially crucial.
- **Single-View Reconstruction:** In situations when acquiring numerous views of a topic may not be practical, the model's ability to generate 3D reconstructions from a single 2D image is a significant advantage.

Limitation

- **Computational Resources:** Training and using PIFuHD can be computationally intensive, requiring powerful hardware, especially for high-resolution reconstructions. This could be a limitation for researchers or practitioners with limited access to such resources.
- **Limited to Human Subjects:** PIFuHD is specialized for digitizing human subjects and may not perform optimally for objects or scenes that deviate significantly from the human form.
- **Data Dependency:** Like many deep learning approaches, PIFuHD relies on a large amount of annotated training data to achieve its high performance. Without sufficient data, the model's results may be less accurate or reliable.
- **Potential Artifacts:** In some cases, the model may introduce minor artifacts or distortions in the reconstructed 3D models, which could require additional post-processing or refinement.

3.11. Learning an animatable detailed 3D face model from in-the-wild images[11]

Description

DECA (Detailed Expression Capture and Animation) is a novel approach that aims to learn an animatable detail model from a single in-wild face image. With the aim to synthesize fine-scale facial geometry with mid-frequency geometric details, DECA is a focused expression capture and animation system that was trained on a diverse dataset consisting of around 2 million in-the-wild face photos. It uses a shape consistency loss for precise shape reconstruction and a low-dimensional detail latent space to improve robustness to noise and occlusions.

Advantages

- **State-of-the-Art Reconstruction:** Even in the absence of paired 3D training data, DECA delivers state-of-the-art performance in face shape reconstruction, outperforming current techniques on benchmarks like NoW and Feng et al.
- **Detailed Animation:** DECA brings animatable characteristics, enabling expressive facial animations that are unique to expression and identity. When compared to techniques that solely recreate static details, this represents a significant advancement.
- **Disentanglement of Details:** A new detail consistency loss aids in differentiating person-specific details from expression-dependent wrinkles, making reconstructions more accurate and lifelike.
- **Robustness to Noise and Occlusions:** The model's resilience to noise and occlusions, which are frequently seen in in-the-wild photos, is facilitated by the low-dimensional detail latent space.
- **Publicly Accessible:** DECA is made available to the research community for use in face recreation and the production of virtual avatars. It is released for research reasons.

Limitation

- **Data Collection Constraints:** Obtaining high-quality 4D scans can be challenging and may require specialized equipment, controlled environments, and skilled operators.
- **Computational Complexity:** Processing and analysing 4D scan data can be computationally intensive, particularly for real-time applications or high-resolution models.
- **Limited Generalization:** Models learned from 4D scans may be specific to the individuals and conditions present in the training data and may not generalize well to new faces or environments.
- **Artifact Handling:** 4D scanning may introduce artifacts or noise that need to be addressed during the modelling process.

CHAPTER 4

PROJECT REQUIREMENTS SPECIFICATION

4.1. Introduction

The project's main objective is to develop a sophisticated model for text-to-3D face generation. The model will be designed to convert textual inputs into 3D images of human faces. The text inputs will consist of various descriptive attributes such as age, gender, ethnicity, facial features, hair, and other relevant factors. The project aims to maintain individuality and diversity while generating the 3D images. This means that the generated images should not look like generic or stereotypical faces, but instead should represent the specific attributes provided in the text input accurately. The model's performance will be evaluated using different metrics, including accuracy and visual quality of the generated images. Accuracy measures how closely the generated images match the textual input. Visual quality, on the other hand, assesses how visually appealing and realistic the generated images look. The metrics used in the evaluation will help determine the model's effectiveness and enable fine-tuning to improve its performance.

4.1.1. Intended Audience and Reading Suggestions

The primary goal of this project is to develop a tool that can assist users in creating a 3D model based on textual descriptions. The tool will provide an intuitive interface where users can input textual descriptions of the 3D model they wish to create, including attributes such as size, shape, colour, texture, and other relevant factors. It is intended that the document may be read by developers, project managers, marketing staff, users, testers, and documentation writers. The remainder of the document contains all descriptions of product functions with functionality and basic Interface requirements. Readers are expected to read the stated references for a greater technical understanding.

4.1.2. Project Scope

The objective of the project is to create a system that can generate a realistic 3D facial model from a textual description quickly and easily without the need for manual modelling or scanning. This project has several benefits, including the ability to create personalized avatars for games and virtual reality applications and aid in forensic investigations. The goals of this project are to create a user-friendly interface that allows users to input textual descriptions and generate 3D facial models quickly and efficiently. The project's coverage will be limited to generating 3D facial models based on textual descriptions of faces. The project will not be able to generate accurate facial models based on incomplete or vague descriptions- more the descriptions, better will be the facial generation. The system may also be limited by the quality and quantity of the training data.

4.2. Product Perspective

The field of text-to-3D full body generation, with an emphasis on detailed facial features, represents a cutting-edge convergence of disciplines such as machine learning, deep learning, and computer graphics. The roots of this innovative approach can be traced to the early 2000s, a period marked by the burgeoning interest in leveraging machine learning to reconstruct 3D facial topologies from two-dimensional images. The real breakthrough came with the advent of sophisticated deep learning methodologies coupled with the proliferation of comprehensive facial datasets. This technological leap has ushered in new possibilities, particularly in the realm of full-body 3D generation while maintaining a sharp focus on crafting realistic facial details.

In practical terms, the technology addresses the burgeoning need for high-fidelity 3D facial and full-body models across a spectrum of applications. The entertainment sector, in particular, has witnessed a growing appetite for the creation of lifelike and customizable avatars in movies and video games. The potential of this technology extends beyond entertainment, promising significant implications for virtual reality experiences and enhancing the capabilities of forensic science. The project stands as a testament to the

remarkable strides made in artificial intelligence, highlighting the intricate interplay between technical sophistication and creative demand.

4.2.1. Product Features

The software allows users to input a textual description of a face or facial features, such as the shape of the face, high cheekbones or not, eye colour, etc.

The features of the product are as follows:

- Model generation
- Image rendering
- Exportable model
- Compatible with most modern web browsers and operating systems

4.2.2. User Classes and Characteristics

- **Game Developers:** To generate realistic 3D models of a human body for game characters.
- **Animators:** To generate movie characters based on an existing textual description like that of one you would find in a book.
- **Designers and Artists:** To generate a template which they can further improve upon.
- **Medical and research professionals:** These users may use text to 3D full body generation technology to create anatomically accurate 3D models of humans for medical simulations, facial reconstruction, or forensic investigations.
- **Social media and communication companies:** These users may use text to 3D full body generation to create avatars, emojis, or personalized stickers for their users.
- **Novice users:** These users may be new to 3D modelling or may be using a simplified or user-friendly interface. They may have limited technical knowledge or experience but may be able to provide simple descriptions or specifications for a model.

4.2.3. Operating Environment

Hardware Platform:

- **CPU and GPU:** The system may require a high-performance CPU and GPU to handle the computational tasks of natural language processing, 3D modelling, and rendering. The GPU may need to support parallel processing and specialized libraries, such as CUDA or OpenCL.
- **Memory and storage:** The system may need a sufficient amount of RAM and storage to load and process large 3D models, datasets, or training models. The memory and storage may need to be fast and reliable to minimize latency and errors.
- **Input and output devices:** The system requires input devices, such as a keyboard to receive text descriptions or commands from users. The system also requires output devices, such as a monitor, to display or export the generated 3D models.

Operating System:

Any operating system with a modern web browser

Software Components:

- **Machine learning frameworks:** The system may use frameworks such as TensorFlow, PyTorch, or Keras to train and deploy machine learning models that can generate 3D images from text inputs, based on datasets of existing faces and annotations.
- **3D modelling and rendering software:** The system may use software packages such as Blender to view 3D models of humans or libraries such as threejs to create and display animated 3D computer graphics in a web browser using WebGL.

4.2.4. General Constraints, Assumptions and Dependencies

- **Regulatory policies:** Depending on the location and context of the system's deployment, there might be certain regulatory policies that developers need to comply with.

- **Availability and quality of data:** The success of an AI model depends on the availability and quality of data used to train and test the model.
- **Hardware limitations:** The hardware requirements for running the model might be high but easily available nowadays. A good CPU, GPU with enough memory/storage.
- **Processing confidential data:** Must ensure processed images or descriptions of a certain user are not available for access to anyone else.
- **Criticality of application:** Since the model is intended to generate a face based on a description, it is critical to ensure the accuracy and consistency of the generation process. This could limit the choices available to developers in terms of the algorithm design, data processing, and other technical considerations.

4.2.5. Risks

- **Resource Requirements:** The system may require a significant amount of computing resources, such as processing power and memory, to generate 3D humans from text. This could lead to increased costs and potential performance issues, such as slow response times or system crashes.
- **Bias and Discrimination:** The system may unintentionally perpetuate biases and stereotypes, such as generating faces that are predominantly of a certain race, gender, or age group. This could lead to unintended harm and discrimination.
- **Data Privacy:** Generating 3D humans from text may require access to sensitive data, such as personal information or images, which could be at risk of unauthorized access or misuse. This could result in breaches of privacy and security, and damage to reputation.
- **Ethical Concerns:** The use of generated humans for malicious purposes, such as deep fakes or impersonation, could raise ethical concerns and potential legal ramifications.

4.3. Functional Requirements

- **Textual Input:** The system should allow users to input text descriptions of facial features, such as eye colour, face shape, nose size and shape, etc.
- **Validity tests on inputs:** The system should run validity tests on the inputs before processing them to make sure they are in the right format and contain all the required data.
- **Model Output:** The system should generate a 3D model based on the text input and allow users to view the model from different angles.
- **Model Saving and Exporting:** The system should allow users to save and export the 3D model as a file, which can be used for various purposes such as printing, animation, or virtual reality experiences.
- **Integration:** The system should be able to integrate with other software or hardware systems, such as virtual reality or augmented reality devices
- **Realism:** The system should aim to create realistic models that represent the given input.
- **User interface:** The system should have a user-friendly interface that allows users to easily input text and customize the 3D model.
- **Error handling:** The system should be able to handle errors and provide useful error messages to users in case of input errors or technical issues.
- **Relationship of outputs to inputs:** The outputs of the system, such as the 3D model and images of the model, should be directly related to the inputs provided by the user. The system should ensure that the 3D model accurately represents the facial features described.

4.4. External Interface Requirements

4.4.1. User Interfaces

- The system should have an intuitive and user-friendly interface that enables the user to give a textual description to generate a 3D full body image.

- The product should provide an interface which allows the user to easily and clearly view the full body generated.

4.4.2. Hardware Requirements

- **Processor:** A fast processor is recommended to handle the computation required for generating 3D models. A multicore processor can be especially helpful for parallel processing.
- **RAM:** Sufficient RAM is necessary to handle the memory-intensive nature of 3D modelling software.
- **Graphics Card:** A dedicated graphics card can improve performance and speed up the rendering process for 3D models. A high-end graphics card with dedicated memory is recommended for generating complex 3D models.
- **Storage:** Sufficient storage is necessary to store the input text and output 3D models.
- **Monitor:** A high-resolution monitor to accurately display the 3D models being constructed.
- **Networking:** High-speed internet connection is needed to download and upload large 3D models and associated files.

4.4.3. Software Requirements

Product Name: CharGen

Description: This project aims to develop a model for text-to-3D full body generation, with an emphasis on detailed facial features. The goal is to generate a 3D image from given textual descriptive inputs, such as age, gender, ethnicity, facial features, hair, etc. while trying to preserve individuality and diversity.

Version / Release Number: 1.0

Databases: The system uses a MongoDB database to store user's information, and the generated models.

Operating Systems: The system is to be compatible with most modern operating systems such as Windows, Linux, MacOS having a browser and good internet connection.

Tools and libraries: The system may use frameworks such as TensorFlow, PyTorch, or Keras to train and deploy machine learning models that can generate 3D images from text inputs, based on datasets of existing faces and annotations, 3D modelling and rendering software to view 3D models such as threejs to create and display animated 3D computer graphics in a web browser using WebGL.

4.4.4. Communication Interfaces

- **Internet Protocol (IP) network:** IP is essential for transmitting data between the client and server components of the project.
- **Hypertext Transfer Protocol (HTTP):** This is a protocol used for transmitting data over the internet. It will be used to transmit data between the client and server.
- **File Transfer Protocol (FTP):** This is a protocol used for transferring files over the internet. It could be used to transfer large 3D model files between different devices or servers.

4.5. Non-Functional Requirements

4.5.1. Performance Requirement

- **Performance:** The system should be able to generate 3D models quickly and efficiently, without any noticeable delays or lag. It should also be able to handle the text input and generate models in a timely manner.
- **Performance efficiency:** The system should be designed to use minimal system resources such as memory and processing power to conserve system resources.
- **Reliability:** The system should be reliable and effective in performance with minimal downtime and disruptions. It should be able to handle errors and recover from failures quickly and seamlessly.
- **Accuracy:** The 3D models generated by the software should be accurate and precise.

4.5.2. Safety Requirements

- **Data Privacy:** Text to 3D full body generation with a focus on facial features requires large datasets of facial images, which can raise privacy concerns. It is important to ensure that these datasets are collected and stored in compliance with data privacy laws and regulations.
- **Safety measure:** Implementing a comprehensive data privacy policy that outlines how personal data is collected, used, and stored.
- **Bias Considerations:** The system should be designed to avoid perpetuating biases or stereotypes, and to generate faces that accurately reflect the intended meaning of the text.
- **Safety measure:** Using diverse images to train our model and conducting regular bias assessments and implementing mitigation strategies.
- **Ethical Considerations:** The generation of humans for nefarious activities, like deep fakes or identity mimicry, may lead to moral dilemmas and possible legal consequences.

4.5.3. Security Requirements

- **Data safety:** The software should protect user data and prevent data loss or corruption.
- **Compliance:** The system should comply with relevant industry standards and regulations, such as data privacy laws or accessibility guidelines.
- **Security:** The software should have appropriate security measures to prevent unauthorized access.

4.6. Other Requirements

- **Maintainability:** The system should be designed in such a way that it can be easily maintained and updated in the future.
- **Usability:** The system should be easy to use and navigate, with clear and concise instructions on how to input the text and view the generated 3D models.
- **Compatibility:** The system should be compatible with various types of devices and operating systems.
- **Portability:** The system should be able to run on multiple platforms and devices, such as servers, desktops, laptops.

CHAPTER 5

SYSTEM DESIGN

5.1. Introduction

The project's main objective is to develop a sophisticated model for text-to-3D face generation. The model will be designed to convert textual inputs into 3D images of human faces. The text inputs will consist of various descriptive attributes such as age, gender, ethnicity, facial features, hair, and other relevant factors. The project aims to maintain individuality and diversity while generating the 3D images. This means that the generated images should not look like generic or stereotypical faces, but instead should represent the specific attributes provided in the text input accurately. The model's performance will be evaluated using different metrics, including accuracy and visual quality of the generated images. Accuracy measures how closely the generated images match the textual input. Visual quality, on the other hand, assesses how visually appealing and realistic the generated images look. The metrics used in the evaluation will help determine the model's effectiveness and enable fine-tuning to improve its performance. The system aims to create realistic models that accurately represent the facial features described by the user, using state-of-the-art machine learning models such as Stable Diffusion, DECA and PIFuHD trained on a large dataset of real-world faces. The system will also have a user-friendly interface with drop-down menus and text inputs to allow users to easily customize the 3D model. Error handling is an important consideration, and the system will be designed to handle errors and provide helpful error messages to users in case of input errors or technical issues. Finally, the outputs of the system, such as the 3D model and images of the model, will be directly related to the inputs provided by the user, ensuring accurate representation of facial features.

5.2. Current System

Currently, there is a gap in the field of generating 3D facial models from text descriptions, as no proven existing system with a reasonable level of accuracy has been publicly released or published to date. As such, there is a need to develop a new system that

can address this challenge and generate high-quality 3D facial models from textual input. This new system has the potential to revolutionize the way 3D facial models are generated, opening up new possibilities for applications in various fields such as animation, gaming, and virtual reality. By providing a more accurate and accessible means of generating 3D facial models, this system has the potential to significantly enhance the overall user experience and increase the efficiency of workflows in a variety of industries.

5.3. Design Considerations

5.3.1. Design Goals

- **Accuracy:** The system should accurately generate 3D facial models from the given text descriptions, with minimal errors and discrepancies. This can be achieved by using state-of-the-art deep learning models.
- **Usability:** The system should be user-friendly and easy to use, with clear and intuitive dropdowns and instructions for inputting text descriptions. The system should also be optimized for performance, with minimal delays and waiting times for generating the 3D models.
- **Scalability:** The system should be scalable and able to handle large volumes of text descriptions, without compromising on the accuracy or quality of the 3D models. This can be achieved by leveraging cloud computing platforms and distributed systems.
- **Compatibility:** The system should be compatible with different web browsers and operating systems, and designed to work seamlessly across different devices, such as desktops, laptops, tablets, and mobile phones.
- **Security and Privacy:** The system should be designed with security and privacy in mind, with measures in place to protect the user's data and prevent unauthorized access. This can include using secure communication protocols, data encryption, and access controls

5.3.2. Design Guidelines

- **Ensure data quality and diversity:** Using high-quality and diverse training datasets to ensure the accuracy and generalizability of the generated 3D facial models. This can include using data augmentation techniques to increase the diversity of the training data.
- **Clear and intuitive User Interface:** Designing a user interface that is clear, intuitive, and easy to use, with clear instructions and dropdowns for inputting text descriptions. The user interface should also provide visualizations and feedback to the users, to help them understand and make any necessary adjustments to the generated 3D facial models.
- **Prioritize performance and scalability:** Optimizing the performance and scalability of the project, with efficient algorithms and distributed computing techniques to handle large volumes of text descriptions and generate 3D models in real-time.
- **Ensure security and privacy:** Implementing security measures to protect user data and prevent unauthorized access, such as using secure communication protocols, data encryption, and access controls.

5.3.3. Constraints, Assumptions, Dependencies

5.3.3.1. Constraints

- **Hardware limitations:** The system may require significant processing power and memory to generate 3D models from text descriptions. This may limit the types of devices that can run the system effectively.
- **Time constraints**: The system may need to generate 3D models quickly to be useful for real-time applications. This may require optimization techniques to improve performance.
- **Compatibility constraints:** The system may need to work with various operating systems, browsers, and devices. This may require testing and optimization for different environments.

- **Data constraints:** The accuracy and quality of the 3D models may be dependent on the quality and quantity of data used to train the system. The system may need to access a large dataset of 3D models and text descriptions, which may require storage and processing considerations

5.3.3.2. Assumptions

- Users have a system with a modern web browser which is connected to the internet.
- The system will primarily be used for generating human face and character models and no other types of objects.

5.3.3.3. Dependencies

- The system depends on deep learning libraries such as PyTorch for model training and generation.
- The system depends on image processing libraries such as PIL and OpenCV for preprocessing text and images.
- On downloading the system depends on OBJ viewing software such as Blender for viewing the final model rendering.
- The system depends on web development technologies libraries such as ReactJS, Flask, threejs.

5.4. Architecture

For the generation of a three-dimensional model from text, it will be a several step process. First for the need of a two-dimensional intermediate image representation of the text description of the face and the body, two separate diffusion pipelines are used. The facial image generated by the stable diffusion pipeline is fed to the DECA model which generates a 3D mesh followed by texture mapping. This same image along with the body image generated based on another set of descriptions are fused and used to generate the appropriate body mesh using the PiFuHD model.

5.4.1. Text to 2D: Stable Diffusion

Stable diffusion refers to a class of generative models that use diffusion processes to transform noise into realistic images based on textual input. These models are called stable diffusion models because they use stable distributions to model the diffusion process. The basic idea behind stable diffusion models is to generate images by iteratively refining a noise signal through a series of diffusion steps, where each step involves adding a small amount of noise to the signal. The parameters of the diffusion process are learned from the textual input, allowing the model to generate images that are semantically consistent with the given text.

Components:

- **Encoder:** This neural network takes as input the original image, extract features from the image and map the features to the latent space.
- **Diffusion steps:** The diffusion steps consist of a series of reversible transformations that add Gaussian noise to the image.
- **Decoder:** The decoder takes as input the final noisy image and maps it back to the original image.
- **Text encoder:** This neural network takes as input the textual description and maps it to a lower-dimensional latent space representation.
- **Fusion layer:** The fusion layer combines the latent space representations from the encoder and the text encoder into a single latent space representation.
- **Generator:** This neural network takes as input the combined latent space representation and generates the corresponding 2D face image.

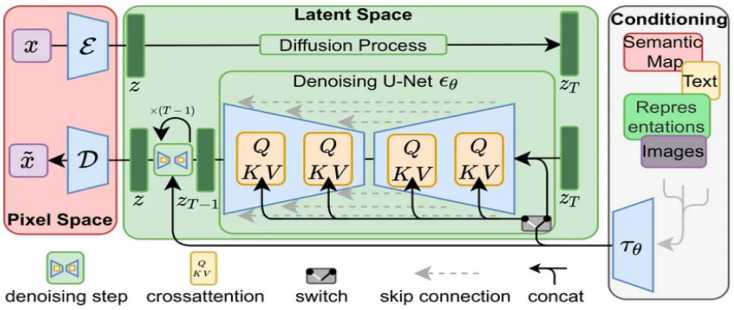

Figure 5.1: Stable Diffusion Model

5.4.2. 2D Face Image to Mesh: DECA

The DECA (Detailed Expression Capture and Animation) model is a comprehensive framework for 3D facial reconstruction and animation, with a strong emphasis on high-fidelity detail capture, including subtle expressions and skin textures. It is particularly suited for realistic character creation in applications such as visual effects, gaming, and virtual reality.

A key feature of DECA is its capability to generate detailed 3D facial meshes. This mesh generation is a crucial component, allowing for the precise modelling of facial structures and nuances. DECA leverages a deep learning approach, trained on thousands of high-resolution 3D facial scans that encompass a diverse range of ethnicities, ages, and expressions. This extensive training enables DECA to accurately reconstruct facial details and expressions from various inputs, such as single images or video frames.

DECA's mesh generation is sophisticated, capable of modelling complex facial geometries that include fine lines, wrinkles, and skin pores. This is achieved through its detailed shape and texture modelling techniques. The model also incorporates dynamic expression modelling, using a combination of expression blendshapes and a neural network. This setup predicts subtle expression variations, allowing for the recreation of nuanced facial movements.

Additionally, DECA provides advanced texture mapping capabilities, ensuring that the final rendered face has realistic skin textures and colours. This feature is essential for applications demanding realism, like creating virtual human avatars, etc. In comparison to models like FLAME, DECA distinguishes itself with its detailed expression capture, intricate texture mapping, and its mesh generation capabilities**.**

5.4.3. 2D Body Image to Mesh: PIFuHD

This model is used to generate the 3D body mesh for the character with the features selected by the user. PIFuHD is a neural network architecture for reconstructing a 3D human model from a 2D image which was developed by researchers from Facebook AI Research. The neural network accepts an image of a person with a resolution of 1024 ×

1024 as input. At the output, the approach gives a 3D model of the person. The architecture of the method consists of two levels of PIFu modules:

- A basic level that focuses on extracting global features from an image. This module is similar to PIFu;
- A refinement level that focuses on extracting local context information and adding precise detail to the 3D model

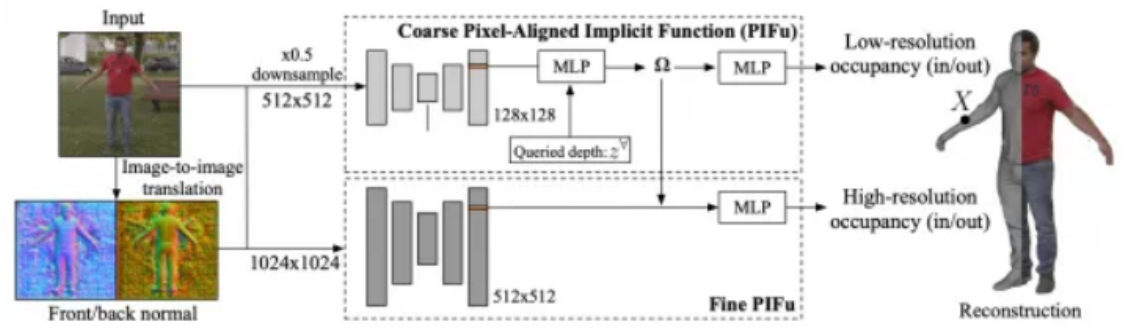

Figure 5.2: PiFuHD Model

5.5. High Level System Design

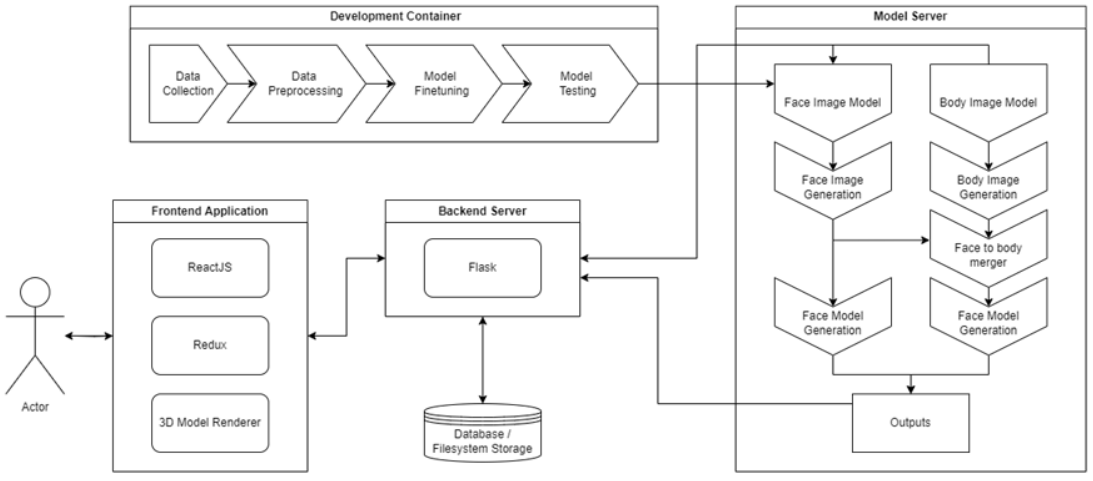

Figure 5.3: High Level Diagram

Modules

- **Development Container**
  - **Data Collection:** The first step is to collect data for training the diffusion model for which we used the CelebA dataset.
  - **Data Preprocessing:** Once the data is acquired, it needs to be pre-processed to make it suitable for fine tuning the diffusion model. This involved selecting appropriate images from the dataset with correct orientations, no accessories such as glasses, etc.
  - **Model Fine-Tuning:** Next step is to fine-tune the diffusion model on the pre-processed data. This is done using DreamBooth by using the selected images and its corresponding class names.
- **Model Server**
  - **Face Image Generation:** Using the fine-tuned diffusion model, the descriptions are passed to generate an image of the face.
  - **Body Image Generation:** Using a diffusion model, the descriptions are passed to generate an image of the body.
  - **FaceBodyTransferer:** As the face in the body image would not match to the users' specifications, the face is transferred from the face only image.
  - **3D Face Mesh:** The face image is used by the DECA model to create the 3D face model along with its texture.
  - **3D Body Mesh**. The body image is used in the PiFuHD model to create the 3D body mesh model.
  - **Output:** The outputs are the images, MTL and mesh (OBJ) files generated and zipped.
- **Backend Server**
  - **Flask Server:** This server will receive requests from the users and handle functions such as sign up, sign in, logout, viewing past generations, etc. by connection to the database. It is responsible for keeping track of sessions, etc. It acts as the bridge between the user's inputs and the model server.

- **Database:** The database is used to store user information, generated images and meshes/models.
- **Frontend Application:** This is the front-end of the system that interacts with the user. Users can register accounts and sign in. Users can make a description of the face they want to create using text dropdown fields with the attributes and then initiate the generation process by clicking a button and then view the final 3D models and even download them. The user has the ability to view their past generations also.

5.6. Design Descriptions

5.6.1. Master Class Diagram

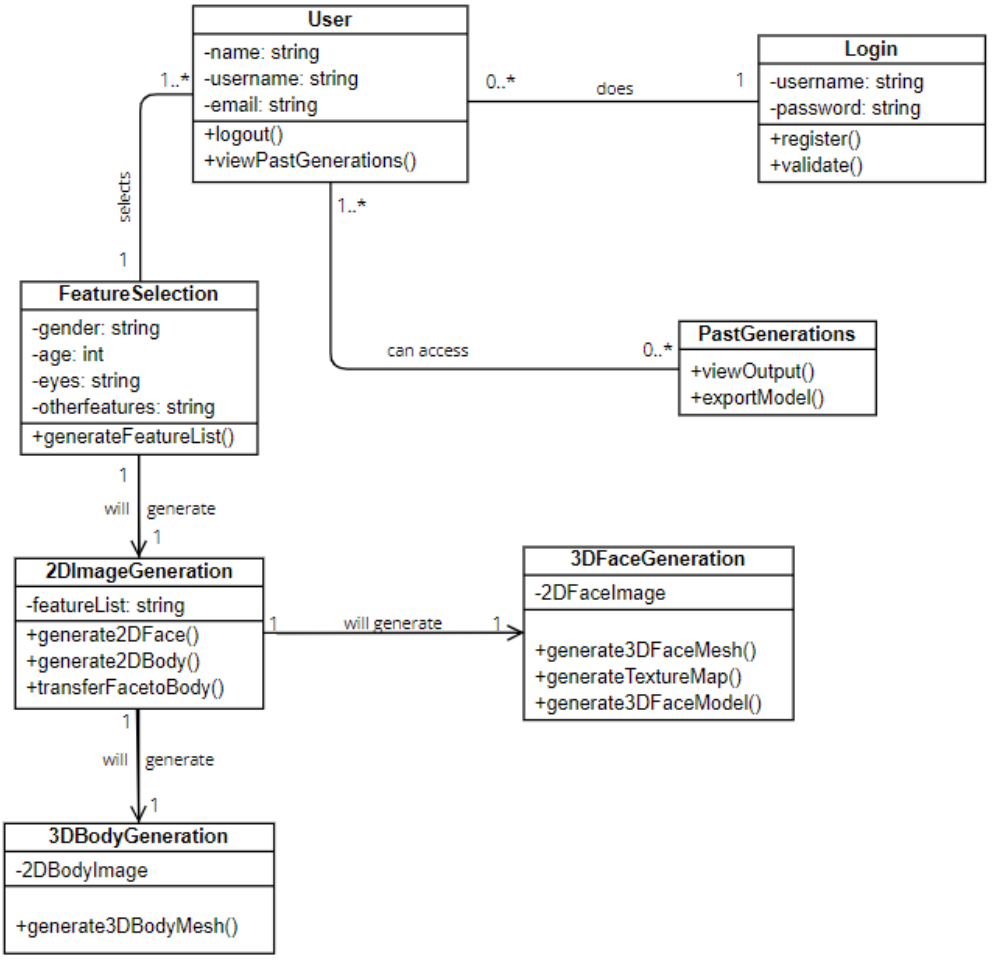

Figure 5.4: Master Class Diagram

- **User Class**
  - **Username:** Username entered by the user.
  - **email:** Email ID entered by the user.
  - **Name:** Name of the user.
  - **logout():** Method which allows users to log out of their account.
  - **viewPastGenerations():** Method which allows users to view their previous generations.
- **Login Class**
  - **username:** User enters their username.
  - **password:** User enters their password.
  - **register():** This method allows new users to register.
  - **validate():** This method validates the username and password provided.
- **Past Generations Class**
  - **viewOutput():** This method allows users to view any of their previous generations.
  - **exportModel():** This method allows users to export any of their previous generations.
- **Feature Selection Class**
  - **gender, age etc:** These are various attributes that must be selected by the user.
  - **generateFeatureList():** This method is used to take the selected features and generate prompts for face and body images which will be fed into the diffusion model.
- **2D Image Generation**
  - **featureList:** This is the prompt the diffusion pipeline takes to generate images.
  - **generate2DFace():** This method is used to generate a 2D face from the selected attributes.
  - **generate2DBody():** This method is used to generate a 2D body from the selected attributes.

- **transferFacetoBody():** This method is used to transfer the face from the generated face onto the generated body face so that they both have the same face.

- **3D Face Generation:**
  - **2DFaceImage:** This is the 2D face image generated by the diffusion model.
  - **generate3DFaceMesh():** This method is used to generate the 3D face mesh.
  - **generateTextureMap():** This method is used to generate the texture map for the face.
  - **generate3DFaceModel():** This method is used to generate the final 3D face model.
- **3D Body Generation:**
  - **2DBodyImage:** This is the body image generated by the diffusion model .
  - **generate3DBodyMesh():** This method is used to generate the body mesh of the body image using the PIFuHD model.

5.6.2. Use Case Diagram

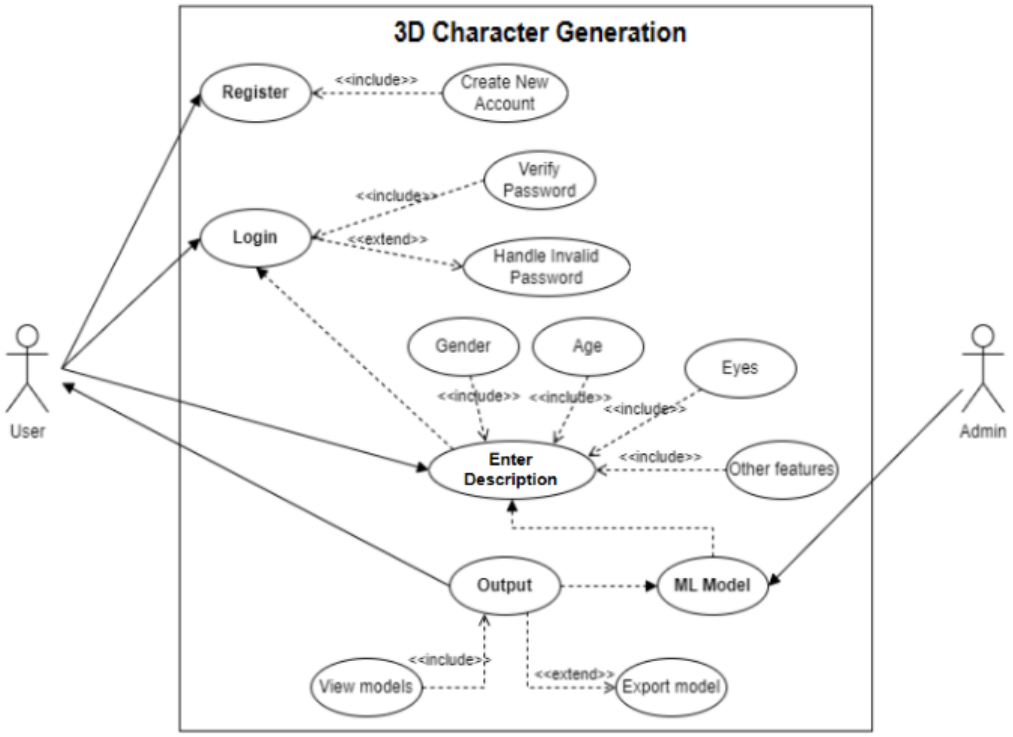

Figure 5.5: Use Case Diagram

- **Register:** New users must create new accounts and then log in.
- **Login:** Users that already have an account should log in to be able to access the page.
- **Enter Description:** Users can enter description by selecting their desired features of the face and body in order to generate the model.
- **ML Model:** The features are taken into account and an accurate realistic model is generated by using machine learning models such as Stable Diffusion, DECA and PIFuHD trained on a large dataset of real-world faces.
- **Output:** Once generated, the model is displayed onto the page. The 2D and 3D images of face and body are displayed side by side. Users can also view previously generated outputs and download them if required.

5.6.3. User Interface Diagram

Land Page: Land page gives an introduction to the project and information about the team.

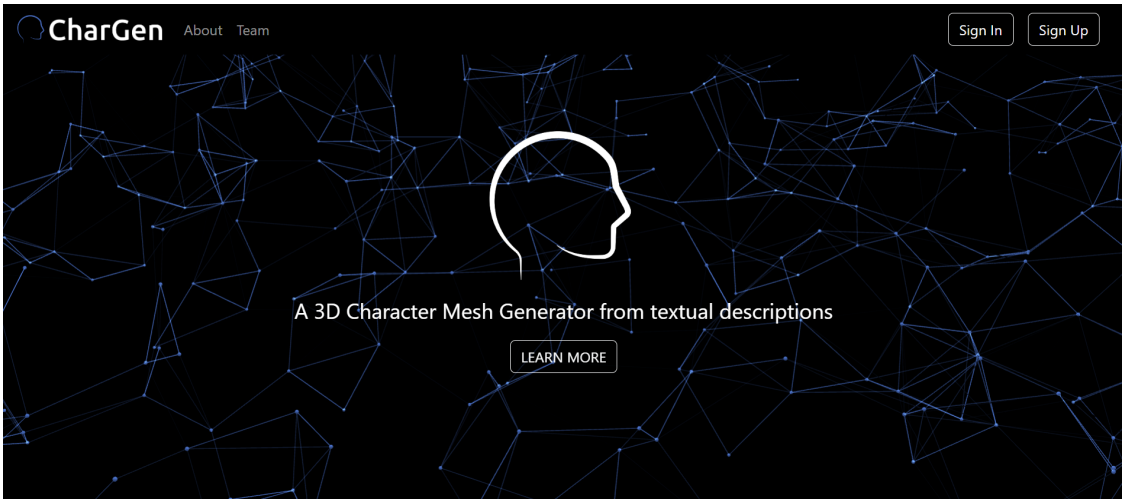

Figure 5.6: Land Page

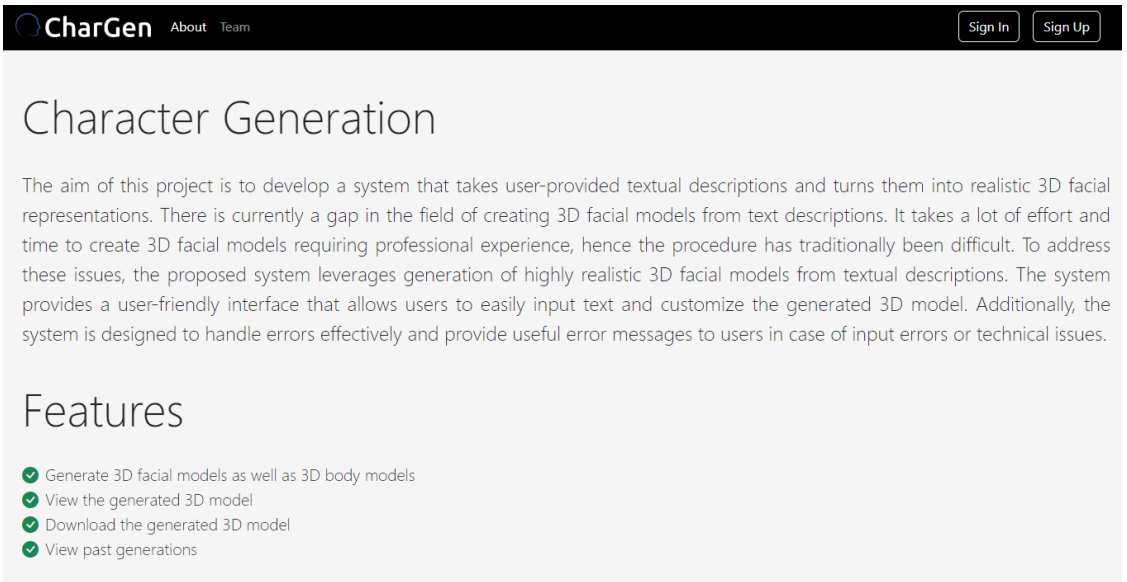

Figure 5.7: Introduction

Sign Up Page: New users are required to create an account using the sign up page. This will help them keep track of their previous generations.

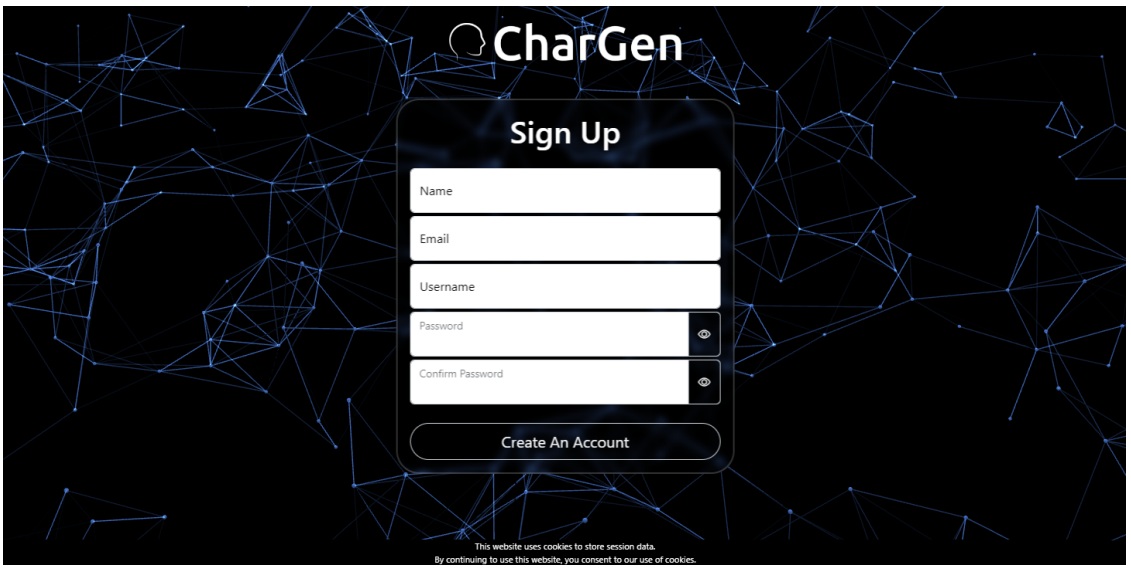

Figure 5.8: Sign Up Page

Sign In Page: Users can sign into their accounts using their usernames/emails and passwords.

Figure 5.9: Sign In Page

Home Page: Home page with options to generate new models, view past generations and sign out.

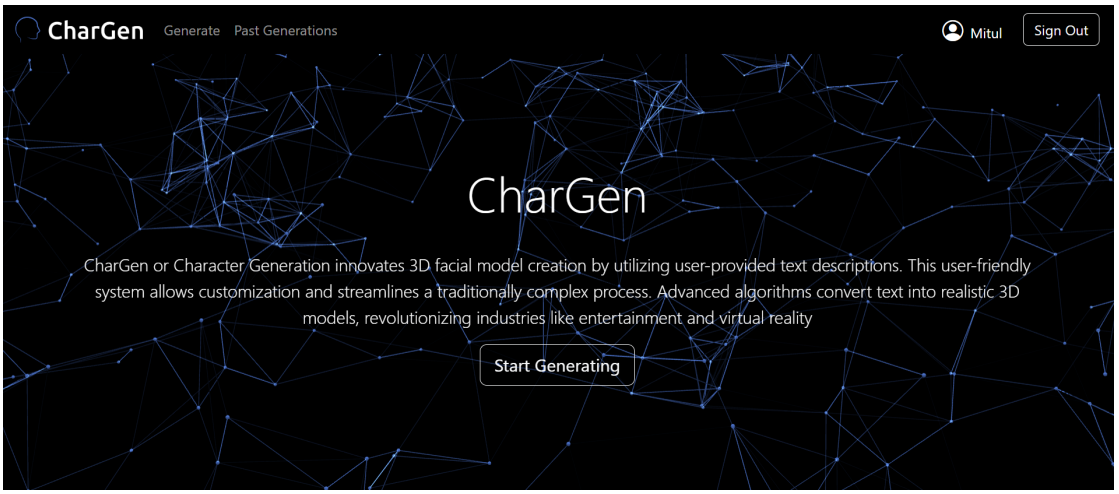

Figure 5.10: Home Page

Generation Page: Generation of new model page, where users can select different features of their choice. Default features are given in case a user does not select a certain feature as all features provided are essential for generation.

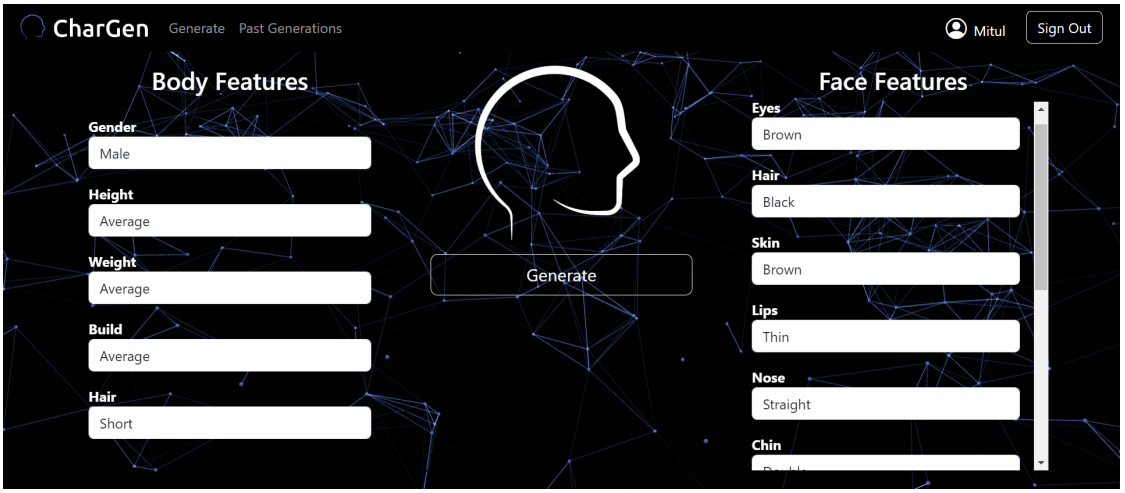

Figure 5.11: Generation Page

Here is an example of an image generation that was previously generated.

Description: Black Woman with Blonde Hair and Green Eyes

Face:

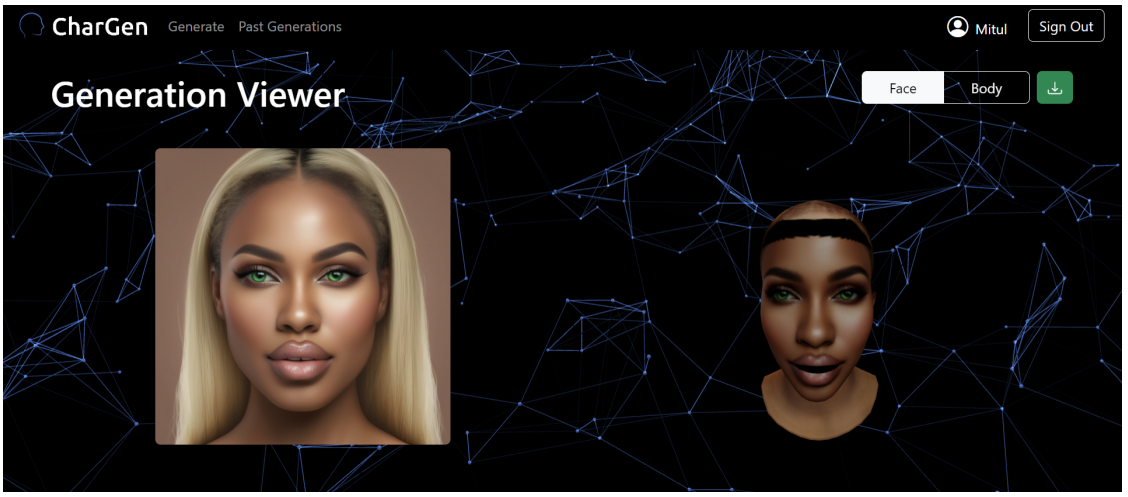

Figure 5.12: Output Page 1

Body:

Figure 5.13: Output Page 2

Past Generations Page: Past generations page, users can view their own previously generated models and have an option to download the same.

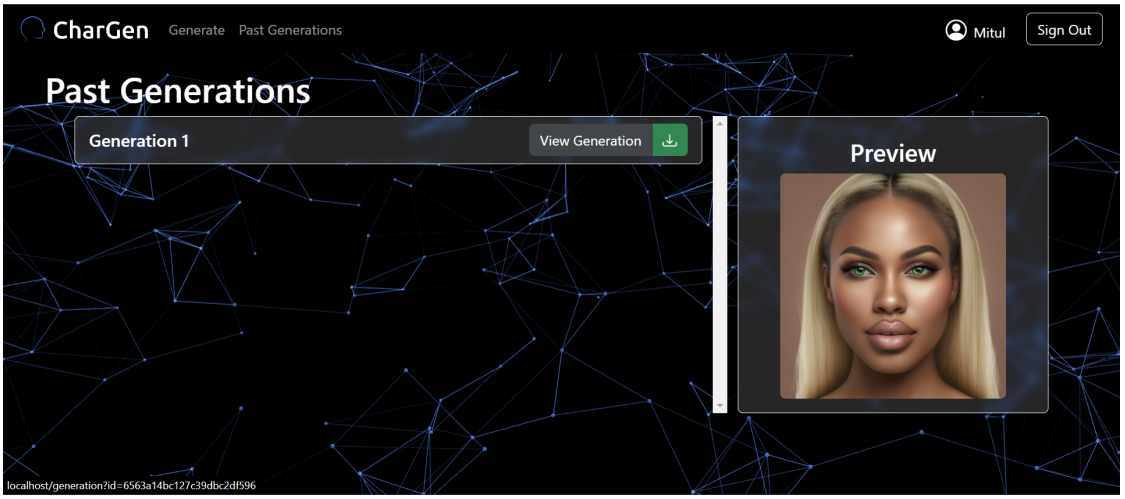

Figure 5.14: Past Generations Page

5.6.4. External Interface Diagram

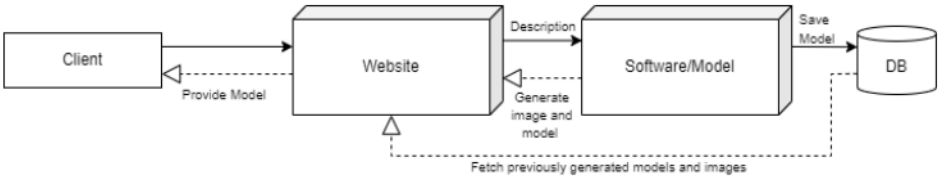

Figure 5.15: External Interface Diagram

- **Database Interface** - Interface with a database to store and retrieve data - MongoDB.
- **User Interface** - Interface for logging in, inputting key characteristics and viewing the created image. - Web based UI.
- **Backend Server Interface** - Interface for communicating with Website and Model and. Flask Server.
- **Model interface** - Server running model. PyTorch and Flask

5.6.5. Sequence Diagram

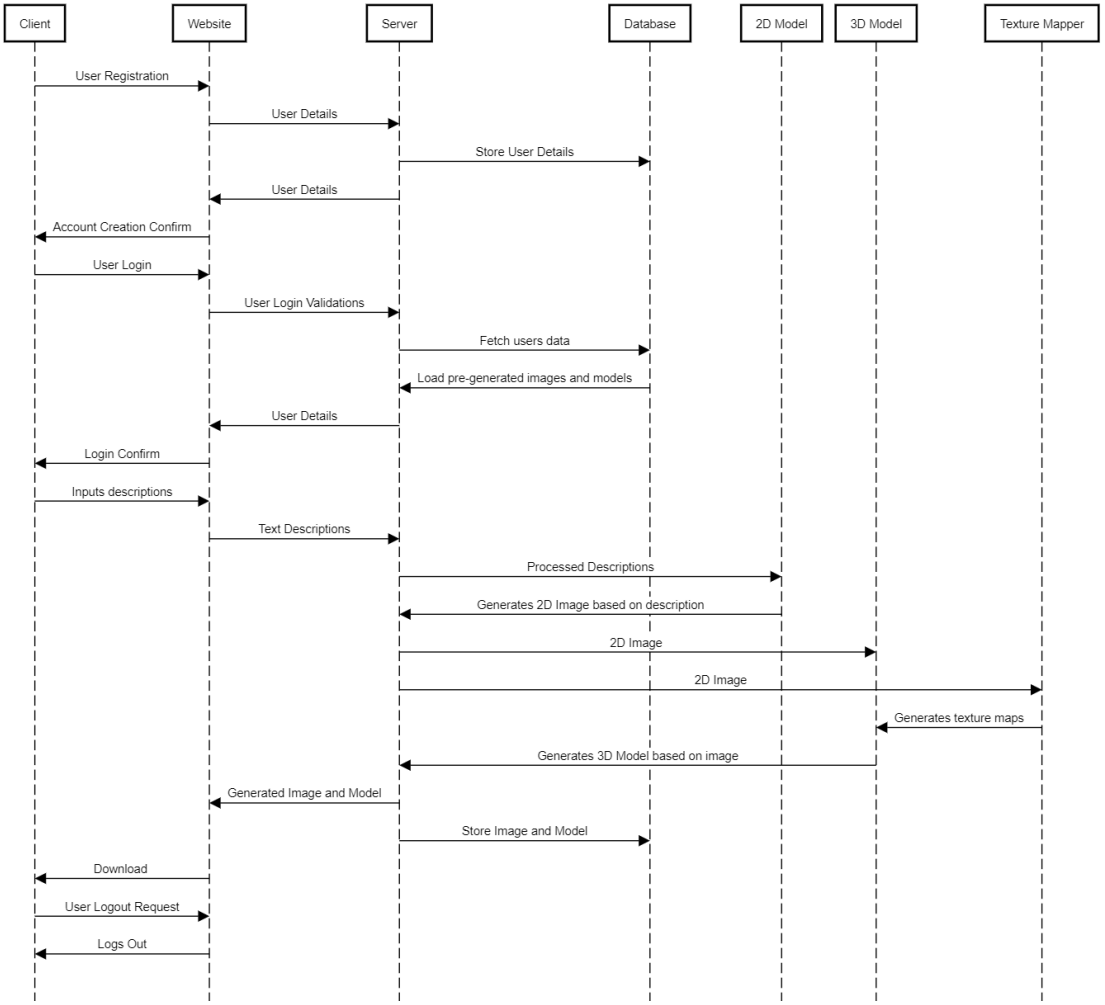

Figure 5.16: Sequence Diagram

This sequence diagram illustrates the interactions and order of events among various objects or components within a system. It provides a dynamic view of how different elements collaborate and communicate to accomplish a specific task or scenario. Objects are depicted as vertical lifelines, and messages exchanged between them are represented by arrows, showcasing the flow and timing of interactions. Activation boxes indicate when an object is actively processing information.

5.6.6. Packaging and Deployment Diagram

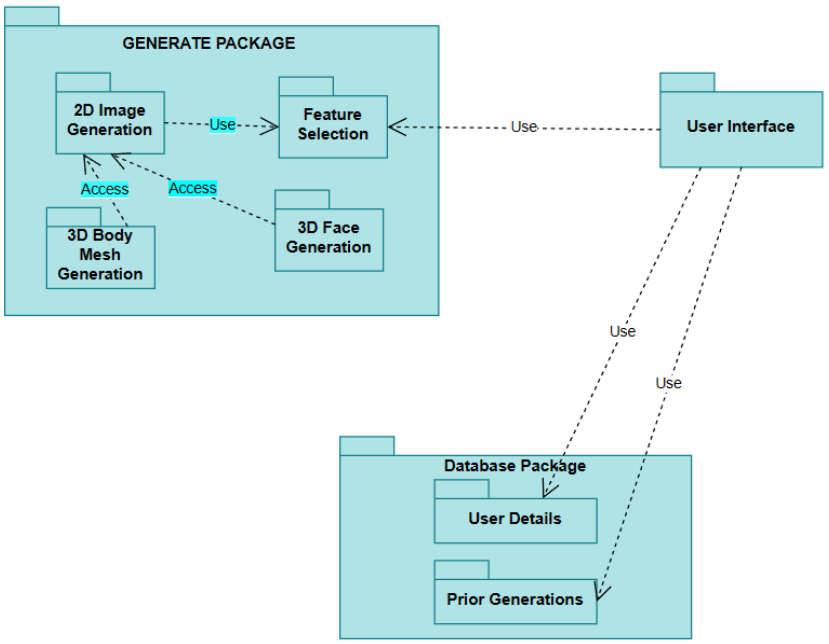

Figure 5.17: Packaging Diagram

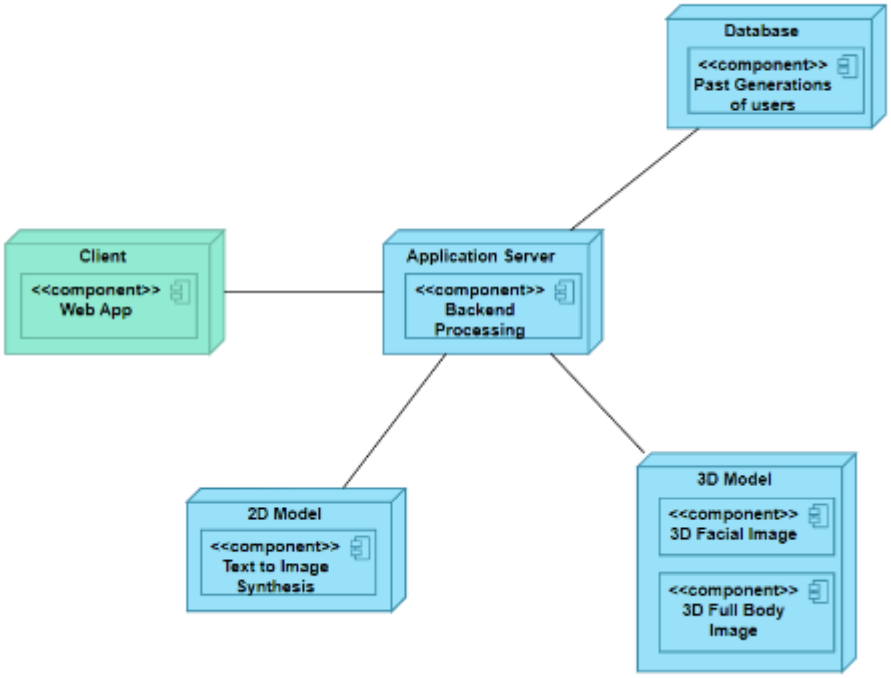

Figure 5.18: Deployment Diagram

5.6.7. Help

- **Technical Manual:** This handbook delves into the underlying algorithms and data structures of the app, catering to those who are interested in learning about its technical details. It gives a thorough grasp of the inner workings of the program by illuminating how it uses these components to generate navigation instructions.
- **User Manual:** The purpose of this tutorial is to walk users through the features of the application. It comes with comprehensive usage guidelines for the app, with an emphasis on setting settings like pollution weightage. For consumers looking for a flawless software experience, the user manual is a useful resource.
- **FAQ Section:** The Frequently Asked Questions (FAQ) section is designed to be a quick reference guide for typical questions and issues raised by users. This section functions as an easily navigable resource, offering prompt solutions to frequently encountered problems and augmenting the application's general usability.

5.7. Design Details

- **Novelty:** The system we plan to use is uniquely novel in terms of construction of 3D facial images based purely on textual descriptions. There are implementations of generating 3D models from textual descriptions of generic objects like balls, flowers etc. However, none were found at the time specialized for generating a 3D face and body from text descriptions.
- **Performance:** Performance is critical for the project, particularly in terms of training and deploying the deep learning model, as well as generating the 3D models in real-time in the web application.
- **Security:** Security is an important consideration, particularly when working with sensitive data or deploying the web application on a public network.
- **Reliability and Maintainability:** It is crucial for ensuring that the project can be maintained and updated over time, particularly in terms of ensuring that the deep learning model can be retrained with new data or improved algorithms.

- **Portability:** The system should be able to run on multiple platforms and devices, such as servers, desktops, laptops.
- **Reusability:** The system should be designed with reusability in mind, so that the algorithms and models can be reused for other applications beyond 3D facial model ling.
- **Application compatibility:** The system must be compatible with other applications that use 3D facial models, such as virtual reality or augmented reality applications.
- **Resource utilization:** The system must be designed to efficiently utilize computing resources, to minimize costs and maximize performance. This requires careful optimization of algorithms and data processing procedures to ensure that the system is running as efficiently as possible.

CHAPTER 6

PROPOSED METHODOLOGY

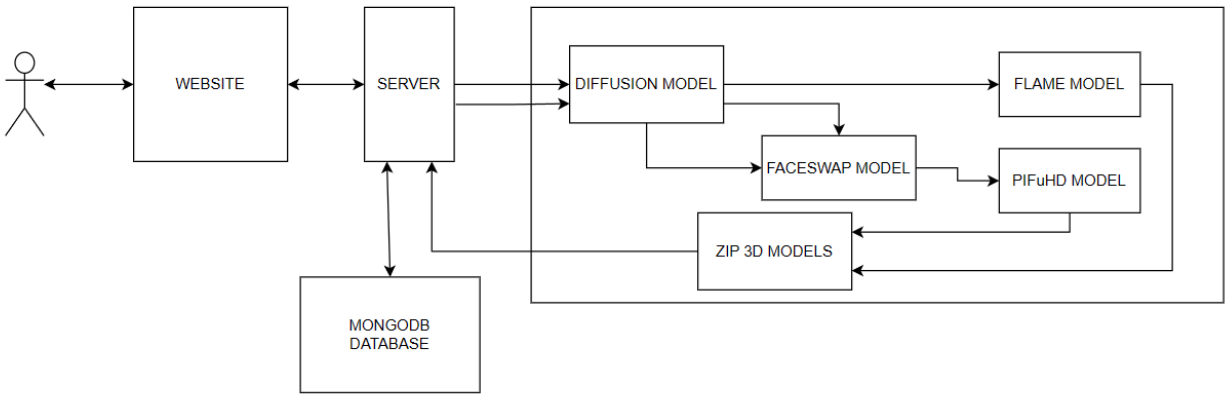

Figure 6.1: Proposed Methodology Diagram

6.1. Website

Frontend:

Develop a user-friendly interface using ReactJS where:

- There is a login portal where an existing user can login or a new user can register.
- The user has the option to generate a new 3D face and character mesh by selecting what characteristics they want their character to have.
- The user can view their past generations.

Backend:

Develop a user-friendly interface using React where:

- Flask is the backend framework to handle user's requests and connects to the database.
- It performs validations on user login/register.
- The server sends requests to the model server for generating the images and meshes.

6.2. Models

6.2.1. Text-to-2D Image Generation

- A stable diffusion model is to be finetuned on the Celeb -A dataset on several features as provided by the same.
- Then a pipeline is to be set up to this model which is fed two sets of prompts one relating to the facial attributes and another relating to the body attributes for which it accordingly generates a face and body image.
- Negative prompts/prompt-engineering is to be used to avoid any unfavourable generations.

6.2.2. Face Transferer

- InsightFace is to be used to transfer the face from the 2D generated face onto the generated body so that the 3D face and the face on the body mesh will be similar.
- Sometimes the face generated may not be able to be transferred due face dimension issues in which case the face is regenerated with the same prompt.

6.2.3. 3D Face Model Generation

- The DECA model is to be utilized to convert the 2D face image into a 3D face mesh.
- Texture mapping is to be performed to generate texture coordinates for the face mesh.
- This will yield a .mtl (Material Template Library) file and a .png file containing texture information and UV map respectively.

6.2.4. 3D Body Mesh Generation

- The PIFuHD model is to be used to generate a 3D mesh of the person's body based on the generated 2D body image.

6.3. Error Handling and User Feedback

- Error handling mechanisms shall be implemented to gracefully manage issues such as invalid prompts or model failures.
- Error messages shall be provided to guide users in case of unsuccessful model generation.

6.4. Testing and Optimization

6.4.1. Functional Testing

6.4.1.1. Unit Testing

- **Feature input:** The system should allow users to input text descriptions of facial and body features, such as eye colour, face shape, nose size and shape, fat/thin etc.
- **Model Output:**
    - **Diffusion:** The model should generate two 2D images based on the text input, one for body and another for face.
    - **FaceSwap:** The model should swap the face on the 3D body to match the 3D face
    - **DECA:** The model should generate a 3D face and produce a .mtl file and UV map for texture mapping
    - **PiFu:** The model should generate a 3D mesh.
- **Website:** Allow users to select features, sign in, register, give descriptions to generate models, and view previous generations.
- **Backend:** Should perform necessary validations for login/registrations, fetch previous generations for user to download/view and maintain a stable connection to the model through ngrok.
- **3D Model Saving and Exporting:** The system should allow users to save and export the 3D model as a file, which can be used for various purposes such as printing, animation, or virtual reality experiences.

6.4.1.2. Integration Testing

- The system should be able to integrate with other software or hardware systems and run on different operating systems.
- The different models, website and server should work together seamlessly.

6.4.2. Non – Functional Testing

- **Performance Requirement**
  - **Performance:** The system should be able to generate 3D models quickly and efficiently, without any noticeable delays or lag. It should also be able to handle the text input and generate models in a timely manner.
  - **Performance efficiency:** The system should be designed to use minimal system resources such as memory and processing power to conserve system resources.
- **Reliability:** The system should be reliable and effective in performance with minimal downtime and disruptions. It should be able to handle errors and recover from failures quickly and seamlessly.
- **Accuracy:** The 3D models generated by the software should be accurate and precise.

CHAPTER 7

IMPLEMENTATION AND PSEUDOCODE

7.1. User Class

7.1.1. User Class Description

This class is used to represent a registered user of the system and encapsulates their authentication and personal information.

7.1.2. Data members

Data Type	Data Name	Access Modifiers	Initial Value	Description
int	userID	private	Auto Generated	A unique identifier for the user.
string	username	private	Empty String	The username chosen by the user for login.
string	password	private	Empty String	The securely hashed password associated with the user's account.
string	email	private	Empty String	The user's email address for communication and account recovery.
string	name	private	Empty String	The user's name.
dateTime	joinedAt	private	NULL	Time account was created
array	generations	private	[ ]	Previous generations of the user

Table 7.1: Users Data Members Table

7.1.3. Methods

7.1.3.1. Register(newUser)

The following details shall be defined for the methods:

- **Purpose:** This method allows a new user to create an account in the system. It checks if the chosen username is available, securely hashes the password, and adds the user to the database upon successful registration.
- **Input:** newUser Object with details such as username, password, etc
- **Output:** None
- **Parameters:** newUser
- **Exceptions:** Appropriate error messages are returned
- **Pseudo-code:**

```
function Register(newUser):
    if UserExists(newUser.Username):
        return "Username already in use"

    newUser.Password = Hash(newUser.Password) // Securely hash the password
    Database.AddUser(newUser)
    return "Registration successful"
```

7.1.3.2. Login(username, password):

The following details shall be defined for the methods:

- **Purpose:** This method verifies the user's credentials and logs them into the system if the provided username and password match an existing user. Authenticates the user based on the provided username and password, allowing access to the system.
- **Input:** username, password
- **Output:** user
- **Parameters:** username, password
- **Exceptions:** Appropriate error messages are returned
- **Pseudo-code:**

```
function Login(username, password):
    if UserIsLoggedIn():
        return "User is already logged in"

    storedUser = Database.GetUserByUsername(username)
    if storedUser is null:
        return "User not found"

    if storedUser.Password == Hash(password):
        CurrentUser = storedUser
        return "Login successful"
    else:
        return "Invalid username or password"
```

7.1.3.3. Logout():

The following details shall be defined for the methods:

- **Purpose:** This method allows the currently logged-in user to log out of the system, effectively ending their session.
- **Input:** user
- **Output:** None
- **Parameters:** user
- **Exceptions:** Appropriate error messages are returned
- **Pseudo-code:**

```
function Logout():
    if UserIsLoggedIn():
        CurrentUser = null
        return "Logged out successfully"
    else:
        return "No user is currently logged in"
```

7.1.3.4. AccessGenerations():

The following details shall be defined for the methods:

- **Purpose:** This method allows users to access and retrieve 3D models they have created or saved in the system.
- **Input:** user
- **Output:** generations

- **Parameters:** user
- **Exceptions:** Appropriate error messages are returned
- **Pseudo-code:**

```
function AccessSaved3DModels():
    if UserIsLoggedIn():
        models = Database.GetModelsCreatedByUser(CurrentUser)
        return models
    else:
        return "User not logged in"
```

7.2. FeatureSelection Class

7.2.1. FeatureSelection Class Description

This class is used to handle input of various features and preprocess it as necessary to feed it into the stable diffusion pipeline to generate 2D images.

7.2.2. Data members

Data Type	Data Name	Access Modifiers	Initial Value	Description
string	Gender	Public	Male	Stores gender
string	Ethnicity	Public	Caucasian	Stores ethnicity
string	Eyebrows	Public	Arched	Stores eyebrow type
string	Hairstyle	Public	Sideburns	Stores Hairstyle
string	Skin tone	Public	Pale	Stores Skin tone
string	Eyes	Public	Baggy	Stores eye type
string	Nose	Public	Big	Stores Nose type
string	Hair Colour	Public	Black	Stores Hair colour
string	Cheekbones	Public	High	Stores cheekbones type
string	Age	Public	Young	Stores age range
string	Lips	Public	Small	Stores type of lips
string	Cheeks	Public	Rosy	Stores type of cheeks

Table 7.2: Features Data Members Table

7.2.3. Methods

7.2.3.1. generateFeatureList()

The following details shall be defined for the methods:

- **Purpose:** To generate a feature list from input features
- **Input:** Different features selected by the user
- **Output:** Final feature list to be fed to diffusion pipeline
- **Parameters:** features to be included
- **Exceptions:** Missing values handled by giving default feature values for each feature.
- **Pseudo-code:**

```
prompt = {"face" : "HD realistic front view face photo of a
" + features['body']['gender'] + " with ", "body" : "HD
realistic full-body image of a " +
features['body']['gender'] + " standing in an open space
with grey background facing towards the camera with"}

bodyFeatures = features.get('body')
faceFeatures = features.get('face')

for bodyFeatureName, bodyFeatureValue in
bodyFeatures.items():
     prompt['body'] += " " + bodyFeatureValue + " " +
bodyFeatureName + ","
     prompt['body'] = prompt['body'][:-1] + "."

for faceFeatureName, faceFeatureValue in
faceFeatures.items():
      prompt['face'] += " " + faceFeatureValue + " " +
faceFeatureName+ ","
     prompt['face'] = prompt['face'][:-1] + "."
```

7.3. 2D Image Generation Class

7.3.1. 2D Image Generation Class Description

The textual description of the face is taken into account and a 2D image is generated by use of the Stable Diffusion model. The generated 2D model of the face is then used further as input for the "2D-to-3D" conversion.

7.3.2. Data members

Data Type	Data Name	Access Modifiers	Initial Value	Description
list	FeatureList	Public	NULL	Stores features

Table 7.3: Methods Data Members Table

7.3.3. Methods

7.3.3.1. generateFaceImage()

The following details shall be defined for the methods:

- **Purpose:** To generate a 2D Face Image
- **Input:** Face Prompt
- **Output:** 2D Face Image.
- **Parameters:** prompt, imagePath, steps
- **Exceptions:** Missing values handled by giving default feature values for each feature.
- **Pseudo-code:**

```
def generateFaceImage(prompt, imagePath=None, steps=20):
    negative_prompt = "animated"
    image = facePipe(prompt,
negative_prompt=negative_prompt,
num_inference_steps=steps).images[0]
    if imagePath is not None:
      image.save(imagePath)
    torch.cuda.empty_cache()
    return image
```

7.3.3.2. generateBodyImage()

The following details shall be defined for the methods:

- **Purpose:** To generate a 2D Body Image
- **Input:** Body Prompt
- **Output:** 2D Body Image.
- **Parameters:** prompt, imagePath, steps
- **Exceptions:** Missing values handled by giving default feature values for each feature.
- **Pseudo-code:**

```
def generateBodyImage(prompt, imagePath=None, steps=20):
    negative_prompt = "animated"
    image = bodyPipe(prompt,
negative_prompt=negative_prompt,
num_inference_steps=steps).images[0]
    if imagePath is not None:
      image.save(imagePath)
    torch.cuda.empty_cache()
    return image
```

7.3.3.3. transferFaceToBody()

The following details shall be defined for the methods:

- **Purpose:** To swap the Face on the generate 2d Body image with the generated 2D face
- **Input:** Face Image, Body Image
- **Output:** 2D Body image with correct Face.
- **Parameters:** Face Image, Body Image.
- **Exceptions:** In case of incorrect Face generation image will be regenerated.
- **Pseudo-code:**

```
def transferFaceToBody(faceImage, bodyImage,
newImagePath=None):
    source_img = cv2.imread(faceImage) if (type(faceImage)
==     str)     else     cv2.cvtColor(np.array(faceImage),
cv2.COLOR_RGB2BGR)
    source_face = tranferApp.get(source_img)
    assert(len(source_face)==1)
    source_face = source_face[0]

    dest_img = cv2.imread(bodyImage) if (type(bodyImage) ==
str)         else         cv2.cvtColor(np.array(bodyImage),
cv2.COLOR_RGB2BGR)
    dest_face = tranferApp.get(dest_img)
    assert(len(dest_face)==1)
    dest_face = dest_face[0]

    dest_img = dest_img.copy()
    res = transferer.get(dest_img, dest_face, source_face,
paste_back=True)
                res    =    Image.fromarray(cv2.cvtColor(res,
cv2.COLOR_BGR2RGB))

    if newImagePath is not None:
        res.save(newImagePath)
    return res
```

7.4. 3DFaceGeneration Class

7.4.1. 3DFaceGeneration Class Description

The 2D image generated by the Stabilized Diffusion model is then fed into DECA, a 3D mesh generation model which uses the 2D image as input to create a detailed 3D facial mesh representation of the human subject along with the image as its texture.

7.4.2. Data members

Data Type	Data Name	Access Modifiers	Initial Value	Description
cv2 array	faceImage	Public	NULL	2D Image

Table 7.4: 3D Face Generation Data Members Table

7.4.3. Methods

7.4.3.1. generateFaceMesh()

The following details shall be defined for the methods:

- **Purpose:** To generate 3D Mesh for the 2D Face Image.
- **Input:** 2D Face Image
- **Output:** 3D Face Mesh OBJ, Texture files: MTL and PNG
- **Parameters:** faceImage
- **Exceptions:** Incorrect face image will lead to regeneration.
- **Pseudo-code:**

```
def generateFaceMesh(faceImage):
    loadImage()
    detectFaceAndMakeBoundingBoxes()
    cropImage()
    runDECA()
    saveFiles()
```

7.5. 3DBodyGeneration Class

7.5.1. 3DBodyGeneration Class Description

The 3D Body image generated by the Stabe Diffusion model is then fed into PIFuHD, a 3D mesh generation model which uses the 2D image as input to create a detailed 3D mesh representation of the human subject. This mesh represents the shape and structure of the human in a 3D space.

7.5.2. Data members

Data Type	Data Name	Access Modifiers	Initial Value	Description
cv2 array	bodyImage	Public	NULL	2D Image

Table 7.5: 3D Body Generation Data Members Table

7.5.3. Methods

7.5.3.1. generateBodyMesh()

The following details shall be defined for the methods:

- **Purpose:** To generate 3D Mesh for the 2D Body Image.
- **Input:** 2D Body Image
- **Output:** 3D Body Mesh
- **Parameters:** bodyImage
- **Exceptions:** Incorrect body image will lead to regeneration.
- **Pseudo-code:**

```
def generateBodyMesh(bodyImage):
    torch.cuda.empty_cache()
    rect = getRect(net.cuda(), [bodyImage], 512)
    bodyMesh = runPifuHD(bodyImage, rect)
    return bodyMesh
```

CHAPTER 8

RESULTS AND DISCUSSION

The project's objective was to create a three-dimensional representation of a human figure, emphasizing facial detail. This was accomplished by first generating a two-dimensional facial image based on detailed descriptions of facial features such as the eyes, nose, ethnicity, lips, cheeks, etc and also generating a 2D body image with the selected features. Utilizing stable diffusion, two distinct processes/pipelines were developed:

- The first process takes the 2D facial image and applies it to the DECA model to create a meshed and textured 3D facial representation.
- The second process involves face-swapping the 2D facial image onto a full-body human figure, also generated through stable diffusion. This face-swapped full-body image is then used as input for the PIFuHD (Pixel-aligned Implicit Function) model to create a 3D full-body human figure. This model includes a mesh but lacks texture.

In addition to the technical achievements outlined above, the project has also resulted in the development of a fully functional interactive website. Users can access the website to generate personalized 3D human figures, specifying details such as facial features, ethnicity, and full-body characteristics. The website also includes a robust backend that stores user-generated data, allowing them to save their created figures and view past generations of images. Users can download their 3D human figures as well, enabling them to utilize the figures in various digital applications. This integrated website further enhances the utility of the project's results, providing a user-friendly platform for creating human figures.

Following the description of the project's advancements, an illustrative example of the process is presented:

For example:

Input: Black Woman, Young, Blonde Hair, Green Eyes, High Cheekbones, Young, Thick Lips, Big Nose

Here the unspecified features will be set to default/initial values. The provided feature list is used to generate 2D images using the stable diffusion model: one of that of the face and one of that of the body.

Pipeline 1:

The 2D facial image is processed through the DECA model, resulting in the 3D image of the person's face.

Output

Figure 8.1: 2D Face

Figure 8.2: 3D Face

Pipeline 2:

The same facial image is input into the InsightFace model for face-swapping. This model transposes the 2D generated face onto the generated body, ensuring a match between the 3D face and the face on the body mesh. The combined image is then fed into the PIFuHD model, which produces the output of a mesh-rendered 3D full-body image.

Output

Figure 8.3: 2D Body

Figure 8.4: 3D Body

The conclusion drawn from this project is the successful generation of final outputs that can be employed effectively as digital representations. These outputs have the potential to be utilized as avatars in virtual environments, as well as characters within video games and various other digital applications, signifying the latest progression within the project.

CHAPTER 9

CONCLUSION AND FUTURE WORK

The project has successfully achieved its objective of creating a three-dimensional representation of a human figure, with a particular emphasis on facial detail. The innovative approach, combining stable diffusion techniques with advanced models like DECA for facial representation and PIFuHD for full-body imaging, has resulted in highly detailed digital representations. The project's outcomes are not only technically proficient but also versatile, opening up possibilities for usage in diverse domains such as virtual reality, video games, and digital avatars, among others.

The integrated website platform enhances user interaction by allowing storage, retrieval, and downloading of generated images, effectively bridging the gap between sophisticated 3D modelling technology and end-user accessibility. This ease of use is a significant step towards democratizing high-end 3D model generation for various applications, from entertainment to perhaps even educational and professional simulations in the future.

Looking ahead, some avenues present themselves for future development. The addition of texture mapping to the full-body 3D images generated by the PIFuHD model could enhance the realism and visual appeal of the full body characters. Further, extending the capabilities of the DECA model to encompass more features like hair would create a more holistic representation of the human head.

REFERENCES

[1] Lin, J., Yuan, Y., & Zou, Z. (2021, May). Meingame: Create a game character face from a single portrait. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 1, pp. 311-319).

[2] Xia, W., Yang, Y., Xue, J. H., & Wu, B. (2021). Tedigan: Text-guided diverse face image generation and manipulation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 2256-2265).

[3] M. Z. Khan et al., "A Realistic Image Generation of Face From Text Description Using the Fully Trained Generative Adversarial Networks," in IEEE Access, vol. 9, pp. 1250-1260, 2021, doi: 10.1109/ACCESS.2020.3015656.

[4] T. Wang, T. Zhang and B. Lovell, "Faces à la Carte: Text-to-Face Generation via Attribute Disentanglement," 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2021, pp. 3379-3387, doi: 10.1109/WACV48630.2021.00342

[5] Kushwaha, A., Chanakya, P., & Singh, K. P. (2022, December). Text to Face generation using Wasserstein stackGAN. In 2022 IEEE 9th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON) (pp. 1-7). IEEE.

[6] Lattas, A., Moschoglou, S., Gecer, B., Ploumpis, S., Triantafyllou, V., Ghosh, A., & Zafeiriou, S. (2020). AvatarMe: Realistically Renderable 3D Facial Reconstruction" in-the-wild". In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 760-769)

[7] Yang, H., Zhu, H., Wang, Y., Huang, M., Shen, Q., Yang, R., & Cao, X. (2020). FaceScape: A LargeScale High Quality 3D Face Dataset and Detailed Riggable 3D Face

Prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 601-610).

[8] Sun, J., Wang, X., Zhang, Y., Li, X., Zhang, Q., Liu, Y., & Wang, J. (2022). Fenerf: Face editing in neural radiance fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7672-7682).

[9] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10684-10695).

[10] Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., & Aberman, K. (2023). Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 22500-22510).

[11] Saito, S., Simon, T., Saragih, J., & Joo, H. (2020). Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 84-93).

[12] Li, T., Bolkart, T., Black, M. J., Li, H., & Romero, J. (2017). Learning a model of facial shape and expression from 4D scans. ACM Trans. Graph., 36(6), 194-1.

[13] Feng, Y., Feng, H., Black, M. J., & Bolkart, T. (2021). Learning an animatable detailed 3D face model from in-the-wild images. ACM Transactions on Graphics (ToG), 40(4), 1-13.

[14] A. Rauniyar, A. Raj, A. Kumar, A. K. Kandu, A. Singh and A. Gupta, "Text to Image Generator with Latent Diffusion Models," 2023 International Conference on Computational Intelligence, Communication Technology and Networking (CICTN), Ghaziabad, India, 2023, pp. 144-148, doi: 10.1109/CICTN57981.2023.10140348.

[15] M. F. Sutedy and N. N. Qomariyah, "Text to Image Latent Diffusion Model with Dreambooth Fine Tuning for Automobile Image Generation," 2022 5th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia, 2022, pp. 440-445, doi: 10.1109/ISRITI56927.2022.10052908.

[16] Z. Canfes, M. F. Atasoy, A. Dirik and P. Yanardag, "Text and Image Guided 3D Avatar Generation and Manipulation," 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2023, pp. 4410-4420, doi: 10.1109/WACV56688.2023.00440.

[17] M. Rohith, L. Pallavi, K. Shirisha, M. Sanjay and V. S. Priya, "Image Generation Based on Text Using BERT And GAN Model," 2023 International Conference on Computational Intelligence, Communication Technology and Networking (CICTN), Ghaziabad, India, 2023, pp. 214-218, doi: 10.1109/CICTN57981.2023.10141495.

[18] H. Lu, H. Tunanyan, K. Wang, S. Navasardyan, Z. Wang and H. Shi, "Specialist Diffusion: Plug-and-Play Sample-Efficient Fine-Tuning of Text-to-Image Diffusion Models to Learn Any Unseen Style," 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 2023, pp. 14267-14276, doi: 10.1109/CVPR52729.2023.01371.

[19] F. Wang, J. Zou and Z. Wan, "Human 3D model generation method based on PIFu improvement strategy," 2023 4th International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE), Nanjing, China, 2023, pp. 182-187, doi: 10.1109/ICBASE59196.2023.10303193.

[20] O. Aleksandrova, Y. Bashkov and A. Pohosian, "Approach for Creating a 3D Model of a Face from its 2D Image," 2022 IEEE 4th International Conference on Advanced Trends in Information Theory (ATIT), Kyiv, Ukraine, 2022, pp. 173-176, doi: 10.1109/ATIT58178.2022.10024231.

APPENDIX A

ACRONYMS AND ABBREVIATION

Celeb A	CelebFaces Attributes Dataset
DECA	Detailed Expression Capture and Animation
FLAME	Faces Learned with an Articulated Model and Expressions
NERF	Neural Radiance Fields
NLP	Natural Language Processing
WebGL	Web Graphics Library
PIFuHD	Pixel-Aligned Implicit Function Human Digitization
FLAME	Faces Learned with an Articulated Model and Expressions
GAN	Generative Adversarial Networks
CGAN	Conditional Generative Adversarial Networks
3DMM	3Dimensional Morphable Model
FFHQ	Flickr-Faces-HQ
FID	Fréchet Inception Distance
LFW	Labeled Faces in the Wild
Celeb A	CelebFaces Attributes Dataset
DECA	Detailed Expression Capture and Animation
FLAME	Faces Learned with an Articulated Model and Expressions
NERF	Neural Radiance Fields
NLP	Natural Language Processing
WebGL	Web Graphics Library
PIFuHD	Pixel-Aligned Implicit Function Human Digitization

APPENDIX B

FIGURES

ANNEXURE I