3D face reconstruction from a front image by pose extension in latent space

by

Zhao Zhang

Thesis submitted to the University of Ottawa
in partial fulfillment of the requirement for the
Master of Applied Science
in
Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Engineering
University of Ottawa

Abstract

Numerous techniques for 3D face reconstruction from a single image exist, making use of large facial databases. However, they commonly encounter quality issues due to the absence of information from alternate perspectives. For example, 3D reconstruction with a single front view input data has limited realism, particularly for profile views. We have observed that multiple-view 3D face reconstruction yields higher-quality models compared to single-view reconstruction. Based on this observation, we propose a novel pipeline that combines several deep-learning methods to enhance the quality of reconstruction from a single frontal view.

Our method requires only a single image (front view) as input, yet it generates multiple realistic facial viewpoints using various deep-learning networks. These viewpoints are utilized to create a 3D facial model, significantly enhancing the 3D face quality. Traditional image-space editing has limitations in manipulating content and styles while preserving high quality. However, editing in the latent space, which is the space after encoding or before decoding in a neural network, offers greater capabilities for manipulating a given photo.

Motivated by the ability of neural networks to generate 2D images from an extensive database and recognizing that multi-view 3D face reconstruction outperforms single-view approaches, we propose a new pipeline. This pipeline involves latent space manipulation by first finding a latent vector corresponding to a given image using the Generative Adversarial Network (GAN) inversion method. We then search for nearby latent vectors to synthesize multiple pose images from the provided input image, aiming to enhance 3D face reconstruction.

The generated images are then fed into Diffusion models, another image synthesis network, to generate their respective profile views. The Diffusion model is known for producing more realistic large-angle variations of a given object than GAN models do. Subsequently, all these images (multi-view images) are fed into an Autoencoder, a neural network designed for 3D face model predictions, to derive the 3D structure of the face. Finally, the texture of the 3D face model is combined to enhance its realism, and certain areas of the 3D shape are refined to correct any unrealistic aspects.

Our experimental results validate the effectiveness and efficiency of our method in reconstructing highly accurate 3D models of human faces from a single input (front view input) image. The reconstructed models retain high visual fidelity to the original image, even without the need for a 3D database.

Acknowledgements

I would like to express my deep appreciation to Dr. WonSook Lee for her unwavering support during my master's studies. Her constructive criticism and invaluable feedback have played a pivotal role in advancing my research. Furthermore, Prof. Lee has been a consistent source of encouragement in both my academic and personal life. Especially during challenging times, I faced personal difficulties, and Prof. Lee provided me with significant assistance and motivation. I am incredibly fortunate to have had such an exceptional advisor and mentor for my master's studies.

Last but not least, I extend my gratitude to my family for their unconditional support.

Table of Contents

List of Tables

List of Figures

Abbreviations

3DMM Three-Dimensional Morphable Models 27

AE Autoencoder 10

BFM Basel Face Model 46

CelebA Celeb Faces Attributes 44

CGANs Conditional Generative Adversarial Networks 12

Comsats Commission on Science and Technology for Sustainable Development in the South 45

CPU Central processing unit 43

DCGANs Deep Convolutional Generative Adversarial Networks 12

DreamBooth Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation 22, 34

GAN Generative Adversarial Network 12

GPU Graphics Processing Unit 43

HS HyperStyle GAN 17

image2Style GAN How to Embed Images Into the StyleGAN Latent Space 16

InterfaceGAN Interpreting the Disentangled Face Representation Learned by GANs 20

LR Low Resolution 16

MAD Mean Angle Deviation 2, 3, 31, 60, 61

mGANprior Multi-code GAN prior 16

ML Machine Learning 7

NN Neural Network 7

OpenPose Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields 23, 34

PDE partial differential equation 21

PGGAN Progressive Growing GAN 12

PULSE Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models 16

RAM Random Access Memory 43

SIDE Scale-invariant Depth Error 2, 3, 30, 60, 61, 71

SMPL Skinned Multi-Person Linear model 28

StyleGAN A Style-Based Generator Architecture for Generative Adversarial Network 13, 15

Chapter 1

Introduction

1.1 Motivation and Objective

3D face reconstruction is a process that involves building a three-dimensional model of a human face from a two-dimensional image. One of the common approaches to 3D face reconstruction involves capturing multiple images or video frames from different angles and using these to reconstruct a 3D model. However, in many practical scenarios, only a single image is available, which makes the task considerably more challenging. This is due to the inherent ambiguity of the task: a single 2D image does not contain direct information about the depth or the third dimension.

The problem of 3D face reconstruction from a single image, therefore, requires solving this ambiguity, and various methods have been proposed to do so. Some approaches rely on a database of pre-existing 3D face models and try to find the model that best fits the given image. Other methods employ learning-based techniques, where a machine learning model, typically a deep neural network, is trained to predict the 3D shape of a face from a 2D image. However, all these methods struggle to generate a lifelike 3D face solely from a single image, owing to the absence of depth or third-dimensional information.

From our observations, we arrived at two key findings: First, multi-view image reconstruction for 3D faces yields better results than single-view reconstruction. Second, neural networks that generate 2D images are more advanced and reliable than those designed to create 3D faces using 2D images.

Therefore, driven by the goal of enhancing the realism of 3D face reconstruction and inspired by our two key observations, we designed a pipeline. This approach creates a

large-angle variation of a given image in the 2D domain using GAN networks [15] and Diffusion models [50]. We then utilize these multi-view images to reconstruct the 3D face, aiming to achieve a more realistic representation.

We plan to employ Scale-invariant Depth Error (SIDE) and Mean Angle Deviation (MAD) as evaluation criteria to assess the similarity of the reconstructed 3D face with the ground truth. Moreover, we also intend to incorporate a human perspective in the evaluation process to gauge the realism of the 3D face.

1.2 Proposed Pipeline

GAN inversion [67] is the process of finding the corresponding latent vector in a GAN's latent space [36]for a given real or target image. This allows for the manipulation and understanding of images within the latent space.

A latent space, also known as a latent feature space or embedding space, is a lower-dimensional representation of a set of items where similar items are located closer to each other. It is constructed by extracting latent variables that capture resemblances between the objects. The dimensionality of the latent space can be either higher or lower than the original feature space, making it an example of dimensionality reduction and data compression. [36]

In our quest to enhance the quality of 3D face reconstruction from a single image (front view input), we have experimented with multiple GAN-based techniques to explore if 2D GAN methods can assist in generating superior quality images in varying poses, thereby boosting the 3D face quality and realism. Accordingly, we propose the integration of HyperStyle GAN (a type of GAN inversion method) [5] into our pipeline for generating images in different poses, a method that has demonstrated high fidelity to the ground truth.

Despite the proven effectiveness of the HyperStyle GAN [5] in generating diverse pose images, it struggles to produce satisfactory profile views. In our quest for improved profile views, we explored various deep-learning methodologies and found that Diffusion models [50] yield superior results. As a result, we've incorporated Diffusion models [50] into our process to produce enhanced profile views, using the outputs of the HyperStyle GAN [5] as inputs.

Additionally, for the generation of the 3D neural network, we employ autoencoders, a type of deep learning model, to predict the 3D face. This approach was chosen due to the

model's ability to be trained with a multitude of datasets, thus enabling it to learn and recognize profile views as well as other viewpoints.

In the pipeline we propose, we first leverage HyperStyle GAN [5] together with Interface GAN [56], a latent space modification tool used for changing the image's style, to create 14 unique pose images. These images are then introduced into the Diffusion models to predict profile views. Subsequently, the generated images serve as input for the autoencoders [66], which deduce the 3D facial structure.

We have designed the pipeline for this architecture and delineated the method for deriving profile views using the Diffusion model [50]. Additionally, we have trained the autoencoders with four datasets, all of which include profile views. Subsequently, we integrate 3D face texture maps and rectify the forehead region of the face using our unique approach. Finally, we contrast the depth map with the ground truth to calculate the SIDE and MAD. These metrics enable us to assess the similarity of the reconstructed 3D face with the actual 3D face. Furthermore, we use the Mean Opinion Score to evaluate the reconstructed face's realism compared to the input image.

The pipeline is in Figure 1.1:

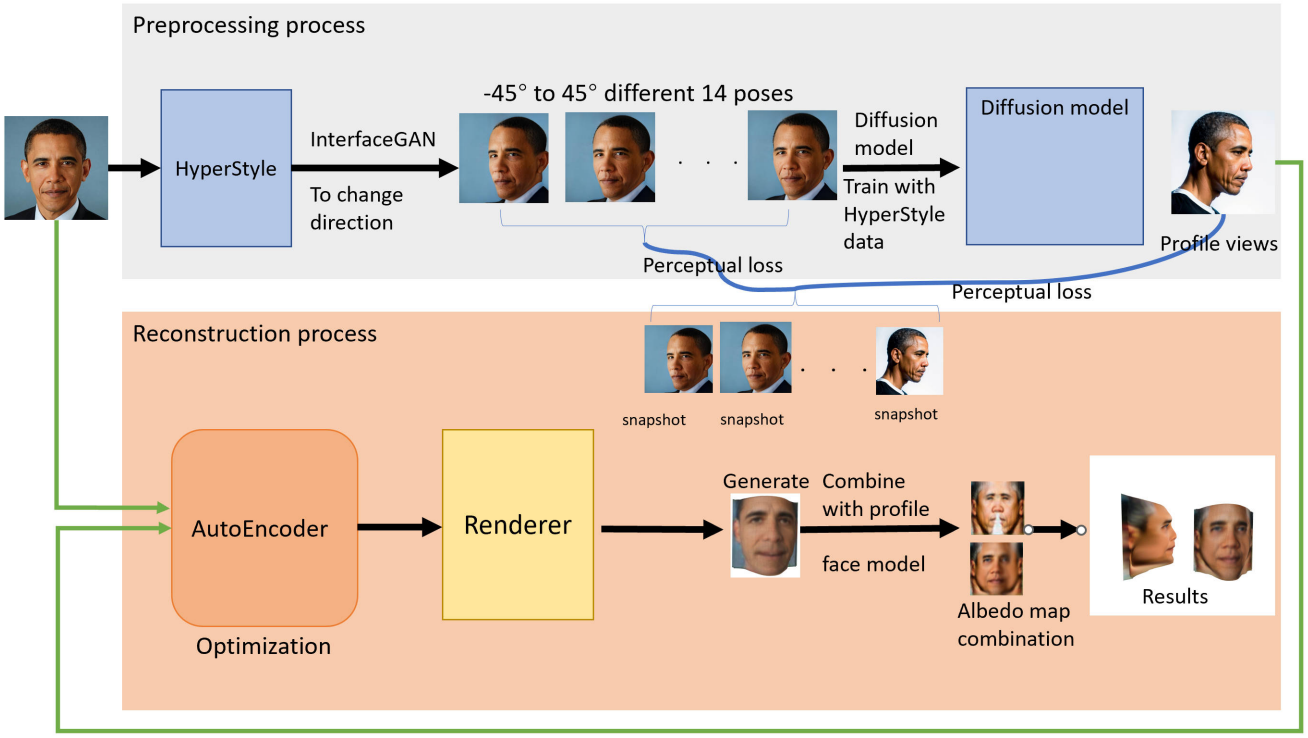


Figure 1.1: Our proposed framework is summarized as follows: Initially, an input image undergoes preprocessing using the HyperStyle GAN to produce poses from -45° to 45°. Subsequently, it traverses through the Diffusion model to generate 90° pose images before being subjected to the Autoencoders, resulting in the creation of a 3D facial representation. Finally, we employ a profile view combination to produce the ultimate 3D facial outputs.

1.3 Main Contributions

This thesis introduces an innovative architecture based on HyperStyle GAN [5], which is combined with a Diffusion method for reconstructing a 3D human face using the Neural 3D Mesh Renderer [33]. The main contributions of this thesis can be outlined as follows:

- With our single front view image, our proposed pipeline enable multi view inputs to create 3D face model by using deep learning network.
- We utilize GAN inversion and GAN manipulation techniques in the domain of single-view 3D modelling.
- We proposed Diffusion methods with our HyperStyle GAN results and Openpose method to synthesize profile view images. This method has not been employed in prior work to generate profile view images. (The Diffusion model cannot generate profile views from a single input image.)
- We proposed to use those synthesized images (profile views and other views) to feed into autoencoders to train to predict 3D model features.
- We proposed to optimize our 3D models with albedo map combination and depth map correction techniques.

Our approach incorporates innovative ideas to address the challenges of 3D face reconstruction from a single image (front view). We leverage the power of HyperStyle GAN to generate diverse pose images, which are then utilized in a diffusion model to synthesize realistic profile views. Additionally, we combine the results from HyperStyle GAN [5] and the Diffusion model [50] to reconstruct the 3D model using autoencoders [66]. This combination of techniques enables us to overcome the limitations of a single image and generate accurate and detailed 3D face models.

Our findings show that our architecture performs significantly better in achieving scale-invariant results, with a value of $0.774 * 10^{-2}$, which is the error between ground truth and our results, and offers real-time inference compared to other optimization-based methods. Additionally, our reconstructed 3D faces exhibit more intricate details, both in the frontal and profile views, compared to other methods.

1.4 Thesis Organization

The remainder of the thesis is organized as follows.

Chapter 2 provides an overview of key concepts such as GAN, GAN inversion, GAN manipulation, and the Diffusion method with Openpose.

Chapter 3 explores topics related to 3D modelling, including single-image 3D reconstruction, unsupervised 3D shape learning, neural mesh renderer, and autoencoders.

Chapter 4 shows our methodology involves utilizing advanced deep-learning techniques to construct a 3D face model from a single-view face image. Our architecture employs a combination of HyperStyle GAN, Autoencoders, Stable Diffusion, and Neural Mesh Renderer, among other innovative techniques.

Chapter 5 delves into the experimental phase of our research. This chapter focuses on conducting various experiments to evaluate the effectiveness and performance of our proposed methodology. Through rigorous experimentation, we aim to validate the capabilities and advantages of our approach in 3D face reconstruction.

Chapter 6 compares our results to those of Unsup3D methods and showcases how our architecture improves the quality of 3D face reconstruction.

Chapter 7 concludes and future works regarding our 3D face construction.

Chapter 2

Backgrounds

2.1 Neural Network and Learning

A Neural Network (NN) [41] is a type of Machine Learning (ML) algorithm inspired by the structure and function of the human brain. It is composed of interconnected nodes or neurons that process and transmit information to one another. Neural networks are capable of learning complex patterns and relationships from large amounts of data, making them a powerful tool for tasks such as image recognition, natural language processing, and prediction. They have become increasingly popular in recent years due to advancements in computing power and data availability.

As shown in Figure 2.1 Each neuron is equipped with both a summation function and an activation function. The summation function typically takes the form of a linear regression function with weights and biases, while the activation function introduces non-linear transformations to the output of the summation function. For a given sample i with n labelled attributes, denoted as x_i, the coefficient for each attribute is represented as x_i. In addition, each neuron is assigned a bias value b. The summation function for each neuron can be expressed as follows:

$$y_i = \sum_i w_i x_i + bias \tag{2.1}$$

A neural network consists of interconnected layers of nodes. Each node can be thought of as a set of algorithms similar to multiple linear regression. The output obtained from

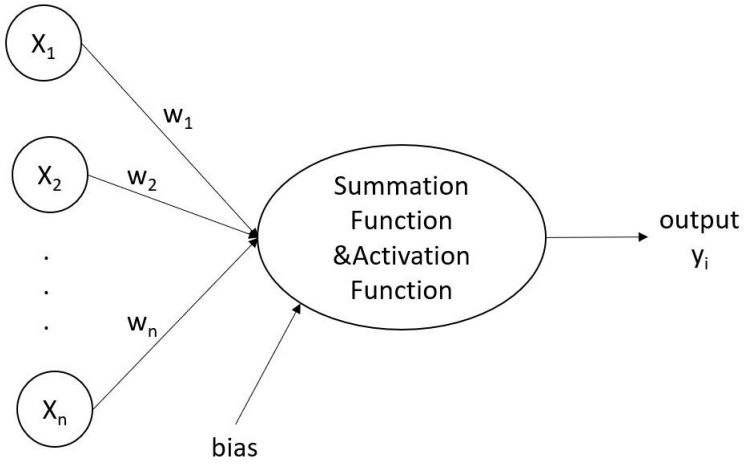


Figure 2.1: Structure of a Basic Neural Network with One Neuron

this regression is then passed through a non-linear activation function using the training algorithm.

Learning in the context of Neural Networks refers to the process by which a network improves its performance at a given task over time. This is typically achieved through a method known as gradient descent, where the network iteratively adjusts its internal parameters (or "weights") in response to the errors it makes on a set of training examples. This process is also called training the network.

In this study, we utilize neural networks as the underlying framework for our GAN networks, autoencoders, and diffusion models. The learning methods employed are designed to comprehend the information from the input image, allowing for the reconstruction of a 3D face.

2.2 Unsupervised Learning

Unsupervised learning [6] is a type of machine learning in which an algorithm is trained on a dataset without any explicit supervision or labelled examples. Unlike supervised learning, where the algorithm is trained on labelled data with known outcomes, unsupervised learning algorithms must identify patterns and relationships within the data themselves.

The goal of unsupervised learning is often to discover hidden patterns, clusters, or relationships within the data, which can then be used for various purposes, such as data com-

pression, anomaly detection, or feature engineering. Some common unsupervised learning techniques include clustering, dimensionality reduction, and generative models.

One key advantage of unsupervised learning is that it can be used in scenarios where labelled data is scarce or unavailable, making it a valuable tool in many real-world applications. However, unsupervised learning can also be more challenging than supervised learning, as there is no ground truth to measure the algorithm's performance, and the results can be more subjective and difficult to interpret. Figure 2.2 illustrates the distinction between unsupervised learning and supervised learning. In our study, we employ unsupervised learning for our autoencoders. This is primarily because the majority of our training dataset lacks labels or ground truth, making our approach align more closely with unsupervised learning methods.

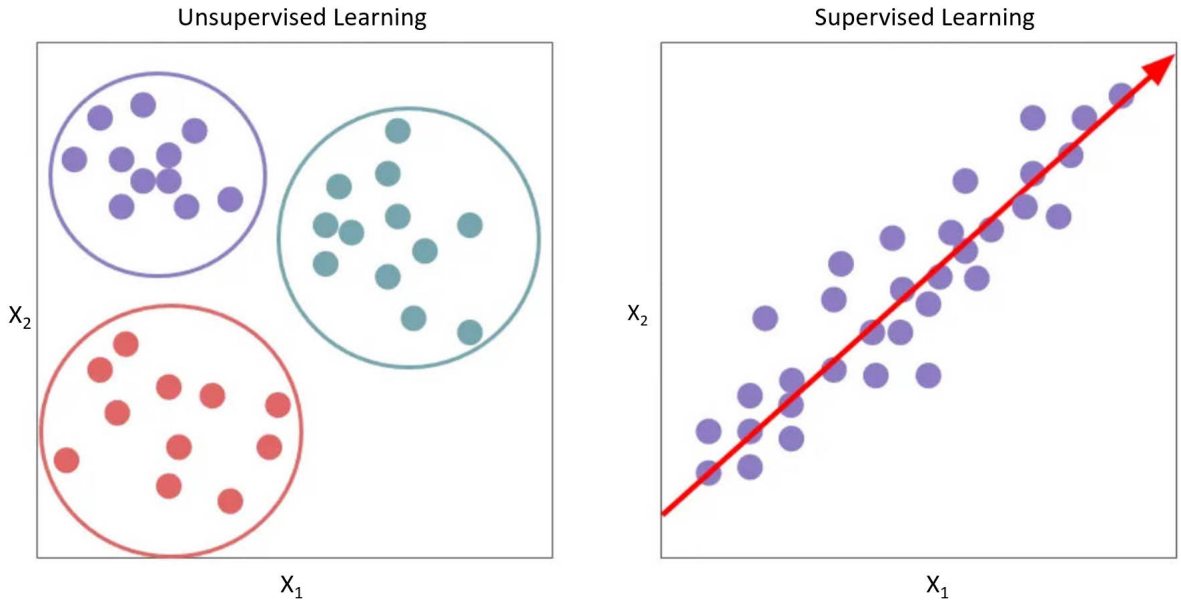


Figure 2.2: Unsupervised Learning and supervised learning explanation in chats. The left image is Unsupervised Learning which uses unlabeled data without any explicit guidance or labels. The right image is Supervised Learning: with guidance as the red line. [2]

Overall, unsupervised learning empowers machines to learn from data in an exploratory manner, uncovering intrinsic structures and patterns and has broad applications in understanding and leveraging the inherent information present in unannotated datasets.

2.2.1 Autoencoders

An Autoencoder (AE) is a type of neural network that is used for unsupervised learning of efficient data representations. It consists of three parts: an encoder, a bottleneck, and a decoder. The encoder takes an input and produces a compressed representation of the data, typically in the form of a lower-dimensional vector. The decoder then takes this compressed representation and attempts to reconstruct the original input. The bottleneck is a module that contains the compressed knowledge representations and is, therefore, the most important part of the network.

The goal of an autoencoder is to learn a compressed representation that preserves the most important features of the input while discarding the less important ones. This can be used for tasks such as data compression, feature extraction, and denoising. Autoencoders can be trained using backpropagation, where the reconstruction error is used as the loss function to update the weights of the network. They have been applied to a wide range of domains, including image and speech recognition, natural language processing, and anomaly detection. Figure 2.3 shows the concept of the autoencoder. And Figure 2.4 shows the example of an architecture of the Autoencoder.

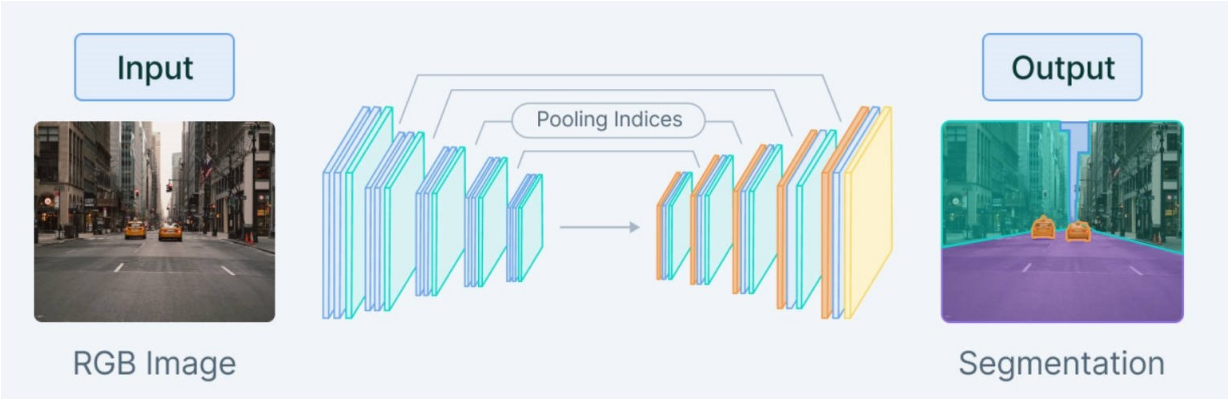


Figure 2.3: Concept of Autoencoder: Convolutions with Encoders and Decoders. The image encodes into a latent space, and decode to recover to original images with segmentation reprinted from [1].

Over the years, various autoencoder architectures have been proposed to enhance their capabilities and address specific tasks. Simple feedforward architectures were initially used, but as deep learning gained prominence, deeper and more complex architectures, such as stacked and deep autoencoders, emerged. Convolutional autoencoders became popular for

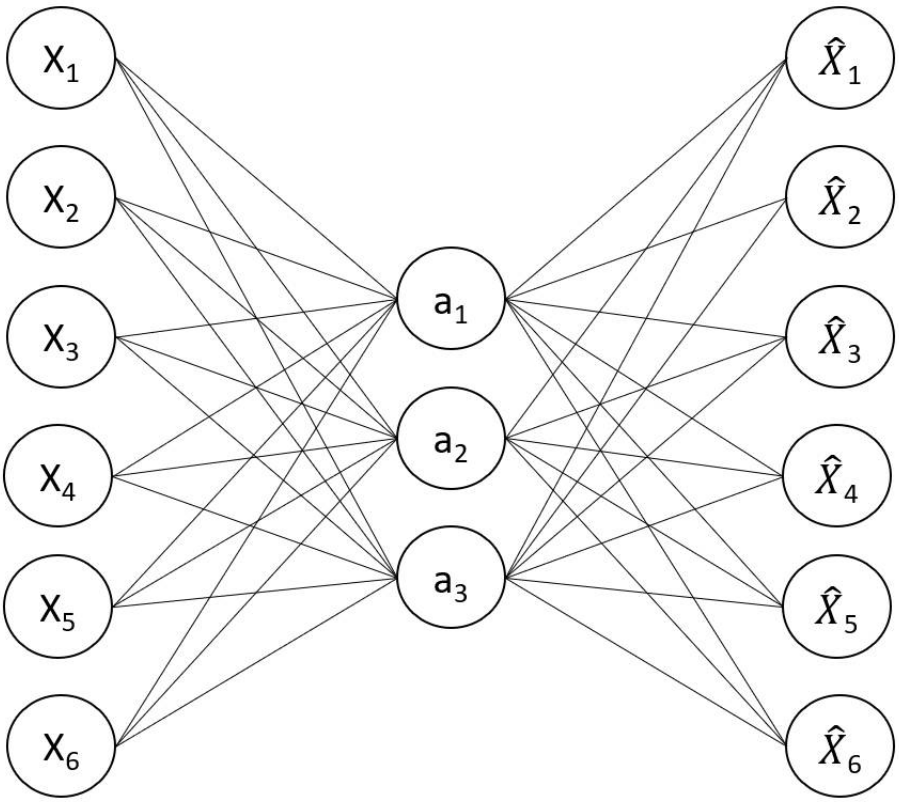


Figure 2.4: Example Architecture of Autoencoder

handling image data, leveraging convolutional layers for spatial hierarchies [38]. Recurrent autoencoders, on the other hand, were designed to capture temporal dependencies in sequential data.

In summary, autoencoders provide a powerful framework for unsupervised learning and feature extraction. They learn to compress and reconstruct data, enabling efficient representation learning and facilitating various downstream tasks in machine learning and data analysis. Autoencoders have emerged as an efficient approach in the field of 3D reconstruction, enabling rapid prediction of 3D features using large datasets. In our research, we utilize autoencoders as our primary tool for predicting the 3D facial structure. The images generated by the GAN network are fed into the autoencoders, which then predict the 3D face.

2.3 Generative Adversarial Network

Generative Adversarial Network (GAN) [15] stands for Generative Adversarial Network, which is a type of deep learning model that is used for generative tasks such as image translation [22, 23, 74], image manipulation [34, 65], and image restoration [61, 68, 71].

GANs consist of two neural networks - a generator and a discriminator - that are trained together in a competitive way. The generator is responsible for producing synthetic data that resembles real data, while the discriminator is trained to distinguish between real and synthetic data. The two networks work in a cycle, with the generator attempting to produce increasingly realistic data that can fool the discriminator, and the discriminator improving its ability to differentiate between real and synthetic data.

During the training of a GAN, the generator initially produces synthetic data that is easily recognized as fake by the discriminator. However, as the training progresses, the generator improves its output to generate synthetic data that becomes increasingly similar to the real data. At the same time, the discriminator becomes better at distinguishing between real and synthetic data.

If the training is successful, the generator will eventually produce synthetic data that is indistinguishable from the real data, causing the discriminator's accuracy to decrease. Initially, the discriminator can easily tell the difference between real and fake data, but as the generator improves, the discriminator begins to classify fake data as real. Figure **??** shows the progress of GAN. GANs have shown remarkable success in various applications, including image and video synthesis, style transfer, and data augmentation.

Despite their success, GAN training can be challenging, with issues such as mode collapse and training instability. Researchers have proposed various architectural modifications and training techniques to address these challenges, resulting in advancements like Deep Convolutional Generative Adversarial Networks (DCGANs) [47], Conditional Generative Adversarial Networks (CGANs) [40], and Progressive Growing GAN (PGGAN) [28].

Comparing GANs with other generative models, GANs excel in generating high-quality samples and capturing complex data distributions. However, they can be computationally demanding and lack explicit inference capabilities. Ongoing research focuses on addressing these limitations and further advancing GAN technology to unleash its full potential in the field of generative modeling.

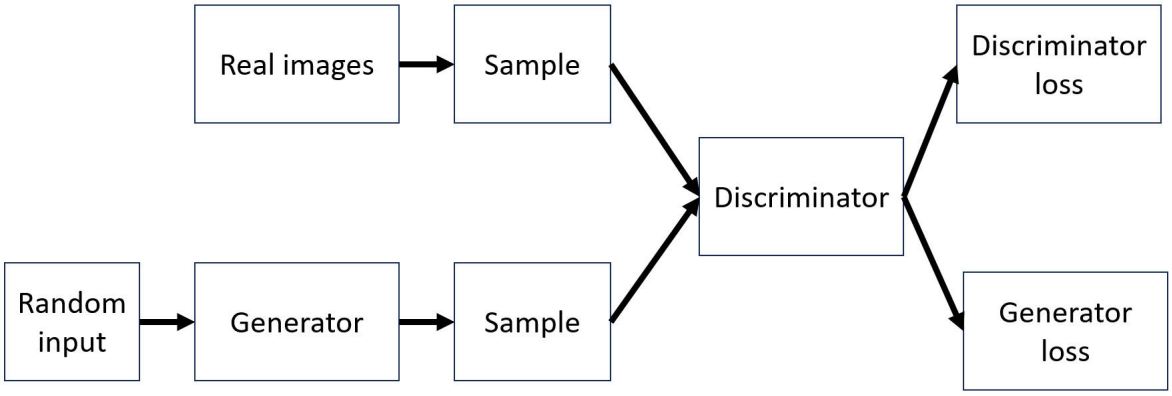


Figure 2.5: System of GAN. The generator takes a random input (latent space) and produces an image, while the discriminator determines whether the generated image is real or fake.

2.3.1 StyleGAN

A Style-Based Generator Architecture for Generative Adversarial Network (StyleGAN) [29–32] is a generative model for creating high-quality synthetic images, particularly of faces, using deep neural networks. It was introduced in a research paper by NVIDIA in 2018 and has since become one of the most popular and widely used GAN models in the research and industry communities.

The key innovation of StyleGAN is "style-based" approach to image generation. Unlike traditional GANs, which generate images from a random noise vector, StyleGAN generates images from a learned distribution of styles, which control various aspects of the image such as the color, texture, and geometric structure. This allows for more fine-grained control over the image generation process, and enables the generation of highly realistic and diverse images.

StyleGAN achieves its remarkable results by utilizing a progressive training approach. It starts with low-resolution images and progressively refines the generated output to higher resolutions, ensuring stability and quality during the training process. This technique enables the generation of highly detailed images with rich textures and realistic appearances.

One of the notable advantages of StyleGAN is its ability to generate diverse and unique images, avoiding mode collapse where the generator produces limited variations. It also provides a smooth latent space interpolation, allowing for smooth transitions between different generated images.

The applications of StyleGAN are extensive, including but not limited to computer graphics, creative arts, and content generation for virtual worlds. It has been used for various tasks such as image synthesis, style transfer, and even generating novel and imaginative visual content.

In summary, StyleGAN is a cutting-edge generative model that pushes the boundaries of image synthesis. With its unique architecture and progressive training approach, StyleGAN enables the generation of high-quality, diverse, and customizable images with a fine level of control over different stylistic aspects. Figure 2.6 shows the architecture of StyleGAN.

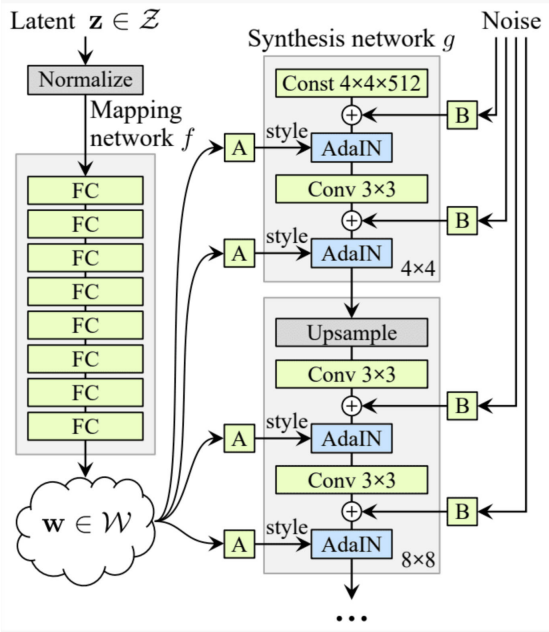


Figure 2.6: Architecture of StyleGAN, reprinted from [31]. Z and W are latent spaces, which Z will transfer to W to feed to the synthesis network. The synthesis network can use the latent code to generate new image.

2.3.2 Latent Space

The latent space of StyleGAN refers to the underlying representation or vector space that captures the variations and attributes of the generated images. It can be thought of as a high-dimensional space where each point corresponds to a unique set of features or characteristics that define the generated image. We can categorize space into two distinct

domains: latent space and image space. While image space pertains to the realm of actual images, latent space refers to a vector representation of these images.

In StyleGAN, the latent space is typically a vector of random numbers or noise variables that serve as input to the generator network. By manipulating the values in the latent space, it is possible to control specific attributes of the generated images, such as facial expressions, hair styles, or even the overall artistic style.

One of the remarkable aspects of the latent space in StyleGAN is its continuity and smoothness. Small changes in the values of the latent vector result in gradual transformations in the generated images. This property allows for smooth interpolations between different images in the latent space, creating visually appealing morphs or blends between distinct visual styles or features. Additionally, the latent space of StyleGAN exhibits disentanglement, meaning that different dimensions of the latent vector control specific attributes of the generated images independently. This disentanglement enables targeted modifications, where individual attributes can be manipulated without affecting others, providing fine-grained control over the generated content. The latent space of StyleGAN has been extensively explored and analyzed, leading to techniques such as latent space exploration, style mixing, and latent space interpolation. These techniques leverage the latent space to generate new images, explore different styles, and discover interesting patterns or combinations [20,56]. The latent space of StyleGAN consists of two distinct spaces: the Z space and the W space. These spaces play a crucial role in controlling and manipulating the generated images. The Z space refers to the initial latent space in StyleGAN, which is typically a vector of random numbers or noise variables. Each point in the Z space represents a unique random seed that serves as input to the generator network. By modifying the values in the Z space, it is possible to introduce randomness and variability in the generated images. However, direct manipulation of the Z space may result in abrupt changes and limited control over specific attributes.

To overcome these limitations, StyleGAN introduces the W space, also known as the style space. The W space is obtained through a mapping network that transforms the Z space into the W space. The mapping network learns a more structured and disentangled representation of the latent space, enabling more precise control over the generated images. In the W space, each dimension corresponds to a specific attribute or feature of the generated image. By modifying the values in the W space, it is possible to manipulate individual attributes independently. This disentanglement allows for targeted modifications, such as adjusting facial expressions, changing hairstyles, or altering artistic styles. Moreover, the W space exhibits a smooth and continuous interpolation property. Small changes in the values of the latent vectors in the W space result in gradual transformations in the generated images. This property enables smooth transitions and seamless blending between

different styles or features, facilitating artistic exploration and creativity. By leveraging the disentangled and continuous properties of the W space, StyleGAN offers fine-grained control over the generated images, allowing for personalized and diverse content generation.

In summary, the latent space of StyleGAN consists of the Z space and the W space. The Z space provides the initial random seed, while the W space offers a structured and disentangled representation. By manipulating the values in the W space, it is possible to control specific attributes and achieve smooth and continuous transformations in the generated images, providing enhanced control and flexibility in the creative process. Figure 2.6 shows how Z space transfers to W space.

2.3.3 GAN Inversion

The primary objective of GAN Inversion is to identify the latent code within the latent space of a pre-trained GAN model based on a provided image. Subsequently, the GAN's generator is employed to accurately reconstruct the given image using this inverted latent code. The prevalent inversion technique involves optimizing the following equation to generate a natural image that closely approximates 'x' (input image). [67].

$$z^* = arg\min_{z \in Z} L(G(z), x) \tag{2.2}$$

where z is the latent space, and L(·,·) denotes the taskspecific objective function, G() is the generator, z is the latent code. Some algorithms under this method, such as Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models (PULSE) [39] optimize the latent code through Low Resolution (LR) image; Multi-code GAN prior (mGAN-prior) [17] optimizes multiple latent codes to increase the expressiveness of the model; The How to Embed Images Into the StyleGAN Latent Space (image2Style GAN) [4] maps a given image into the extended latent space W+ of a pre-trained StyleGAN then optimize the latent code to find better solutions. However, W+ space is known to be less successful for image manipulation compared to W space.

Other than the common inversion method, there are other approaches such as the learning-based inversion method [11, 45, 48, 59]. They tried to learn an encoder to map the relationship between latent code and image. In this way, they can let neural networks generate latent code easier. The equation is as follows:

$$\theta_E^* = arg\min_{\theta_E} \Sigma_n L(G(E(x_n; \theta_E)), x_n) \tag{2.3}$$

where x_n denotes the n-th image in the dataset. E is an encoder; G is a decoder.

A more intricate approach involves the fusion of learning-based algorithms with optimization-based ones. While this method demands more time, it yields superior quality compared to employing only a single approach. This approach is referred to as Hybrid GAN inversion. Such as IDinvert [72], PTI [49], HyperStyle [5] and others [7, 8, 73], They usually use an encoder to learn the mapping relationship to get the latent code, then do some fine-tuning to optimize the latent code to improve the quality.

2.3.3.1 HyperStyleGAN

HyperStyle GAN (HS) [5] is a generative model that combines the power of StyleGAN with the concept of style transfer. It enables the synthesis of images by blending the style of one image with the content of another.

Unlike traditional style transfer techniques that operate on pre-existing images, Hyperstyle GAN generates new images by combining the latent space of StyleGAN with a style image. The model learns to disentangle the style and content information from the input images, allowing for flexible and customizable style transfer.

Hyperstyle GAN consists of two main components: a style encoder and a generator. The style encoder extracts the style information from a given style image and encodes it into a latent style code. The generator then combines this style code with a content code, representing the desired content, to produce a synthesized image that exhibits the style of the input image.

Introduced by Ha et al. [18], hypernetworks are neural networks that serve the purpose of predicting the weights of a primary network. This innovative approach involves training a hypernetwork using a large dataset, which then adjusts the weights of the primary network based on specific inputs. This leads to a more expressive and adaptable model.

Hypernetworks have found applications in various domains, including semantic segmentation [43], 3D modeling [35,58], neural architecture search [70], and continual learning [64]. Their versatility and effectiveness make them a valuable tool for enhancing the performance and capabilities of neural networks across a wide range of tasks and applications.

A hypernetwork has been developed to modulate the weights of StyleGAN in order to accurately express a given image in the editable regions of the latent space. However, the straightforward approach of modulation would require training a hypernetwork with an enormous number of parameters, exceeding three billion.

To overcome this challenge, the network has been meticulously designed to reduce the number of parameters to match those of existing encoders. The resulting HyperStyle

network is capable of producing reconstructions that are comparable to those generated by optimization techniques, while providing near real-time inference capabilities similar to encoders.

In addition to the image inversion task, HyperStyle has also proven effective in other applications, including the editing of out-of-domain images that were not used during training. The architecture of HyperStyle GAN is in Figure 2.7

In our study, we employ HyperStyle GAN as the generator to produce input images with latent code for performing pose editing.

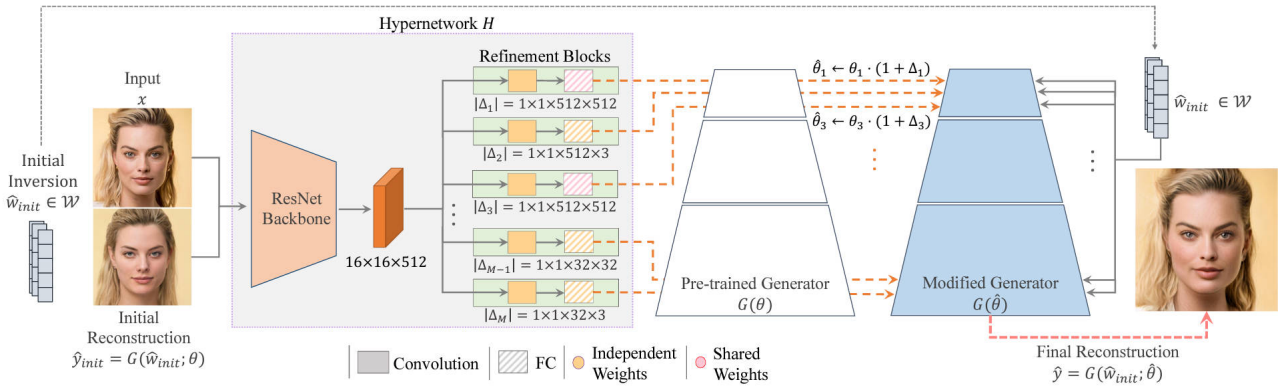


Figure 2.7: HyperStyle scheme. Reprinted from [5]. It uses the real image to tune the Generator to get the best parameters to reconstruct the input image.

2.3.4 GAN Manipulation

GAN manipulation refers to the process of modifying the outputs of a Generative Adversarial Network (GAN). [56] [20] [57]. Unlike traditional image editing techniques that operate on pre-existing images, GAN manipulation allows for the direct modification of the latent space representation of the generated images. By manipulating the latent vectors in the latent space, users can control and change various attributes, such as object appearance, background, lighting conditions, or even semantic features.

GAN manipulation techniques typically involve exploring the latent space to discover patterns and directions that correspond to specific attribute variations. For example, in the case of face images, directions in the latent space might correspond to changes in facial

expressions, hair styles, or age. Once these attribute directions are identified, users can manipulate the latent vectors in those directions to achieve the desired changes in the generated images. This process offers a high degree of control and customization, allowing for creative exploration and modification of the generated content. GAN manipulation techniques have a wide range of applications, including image editing, content generation, virtual reality, and even deepfake generation. They provide a powerful tool for artists, designers, and researchers to generate new and unique variations of the generated content and push the boundaries of creative expression.

InterFaceGAN, developed by Shen et al [56]. in 2020, is a technique that utilizes manual annotations to learn the latent direction of rotating human faces. On the other hand, SeFa [57] and GANSpace [20] are unsupervised methods that can discover significant latent directions, some of which may be associated with the content pose. Figure 2.8 shows the manipulation in the latent space domain.

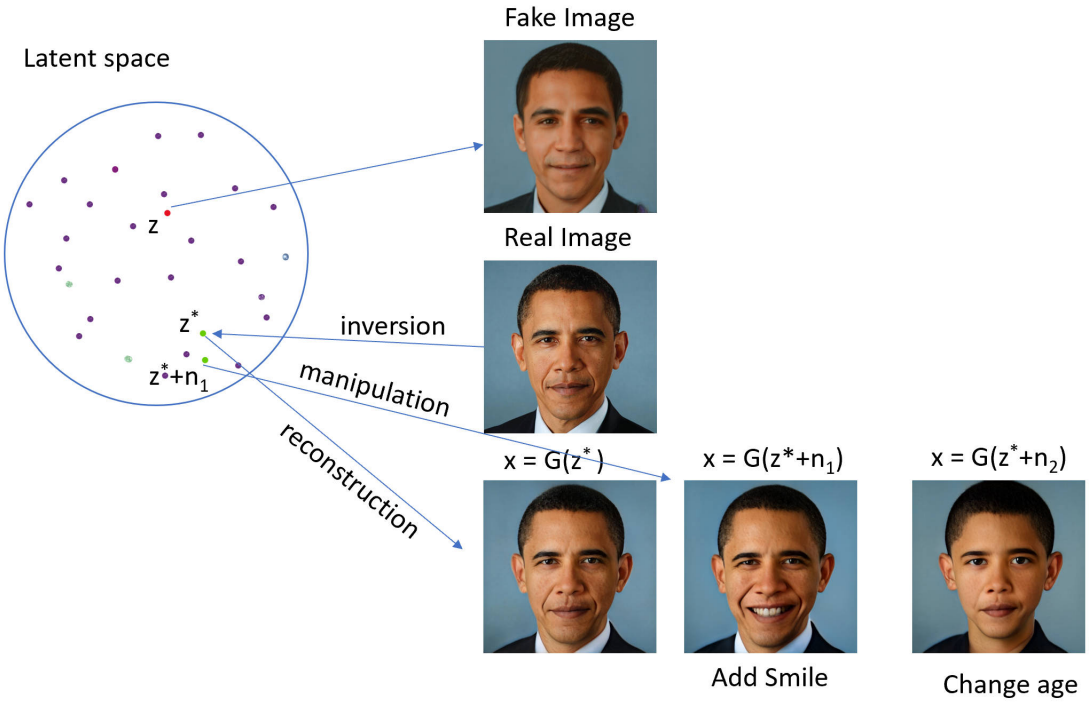


Figure 2.8: Illustration of GAN manipulation and reconstruction. It uses latent code to convert to image space and edit the styles of the image.

2.3.4.1 Interface GAN

Interpreting the Disentangled Face Representation Learned by GANs (InterfaceGAN) [56]is a generative adversarial network (GAN) that is used for image-to-image translation tasks, such as modifying the pose, expression, or attributes of a face image. Unlike traditional GANs, InterfaceGAN is designed to learn an interpretable, disentangled representation of the image content and style, which allows for more precise control over the image generation process.

The key innovation of InterfaceGAN is the use of an "interface" vector, which is a low-dimensional code that controls specific attributes of the image, such as the pose or expression of a face. By manipulating the interface vector, users can modify the image in a controlled and intuitive way, without having to manually specify the desired modifications in pixel space.

InterfaceGAN propose an inversion-based approach to manipulate images by editing multiple attributes simultaneously. The approach involves modifying an image x by computing a new image x' through the equation $x' = G(z^* + \alpha n)$, where n is a unit normal vector representing a hyperplane defined by two latent codes z1 and z2. To manipulate multiple attributes, a set of k attributes $z1, ..., z_k$ is used to form m hyperplanes $n1, ..., n_m$, where m is less than or equal to $k(k-1)/2$. To avoid interference between attributes, the disentangled directions $n1, ..., nm$ must be orthogonal, and the degree of entanglement between the i_{th} and j_{th} attributes can be measured using $n_i \top n_j$. To achieve orthogonality, the method employs projection, which involves finding a projected direction $n1 - (b1 \top n2)n2$ that allows for changes in "attribute one" without affecting "attribute two". For scenarios where multiple attributes are involved, the primal direction is subtracted from the projection onto the plane constructed by all conditioned directions. Figure 2.9 illustrates the approach for two hyperplanes with normal vectors n1 and n2.

2.4 Diffusion Model

Diffusion is a technique used in machine learning and image processing for denoising and enhancing images [50] [21] [42]. It is based on the concept of diffusion, which refers to the process of spreading information or properties across neighbouring pixels in an image.

The goal of stable diffusion is to effectively remove noise from an image while preserving important image details and structures. It achieves this by carefully controlling the diffusion process to ensure stability and avoid excessive blurring or distortion.

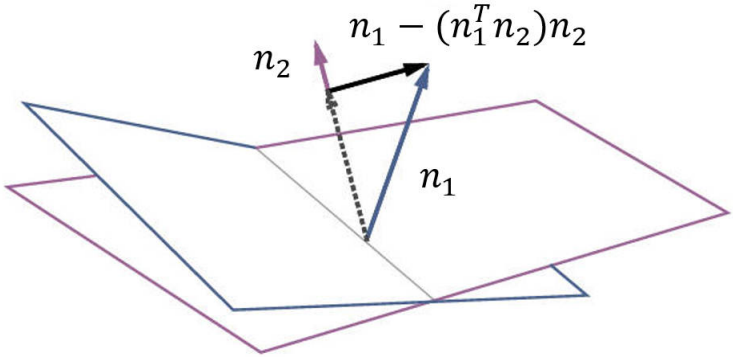


Figure 2.9: Illustration of disentangled directions for multiple attributes. Reprinted from [56]

In diffusion, a diffusion process is modeled as a partial differential equation (PDE) that describes how the image evolves over time. The PDE takes into account the image gradient, which represents the spatial variations in intensity, and incorporates a diffusion coefficient that determines the rate at which information is spread across neighbouring pixels.

One key aspect of diffusion is the use of anisotropic diffusion, which means that the diffusion coefficient is adaptively adjusted based on the image gradient. This allows for selective diffusion in areas with high gradients, such as edges and boundaries, while limiting diffusion in smooth regions. By preserving the sharpness of edges and fine details, stable diffusion produces denoised images that are visually pleasing and retain important image structures. To ensure stability and prevent artifacts, various numerical schemes and techniques are employed in the implementation of stable diffusion. These include explicit or implicit time integration methods, discretization schemes, and regularization techniques. Stable diffusion is an innovative product developed by Stability AI company. It integrates latent diffusion models with various techniques to generate high-quality images. By combining these approaches, Stable diffusion offers enhanced image synthesis capabilities. In our research, we use diffusion models to create realistic profile views. The architecture of the Diffusion model, as depicted in Figure 2.10, illustrates the functioning of this method in generating images.

Stable diffusion has found applications in various areas such as image denoising, image inpainting, and image enhancement. It is particularly effective in scenarios where noise removal is desired without sacrificing important image details, making it a valuable tool in image processing and computer vision tasks. In our research, we utilize Diffusion models to generate profile views for 3D reconstruction.

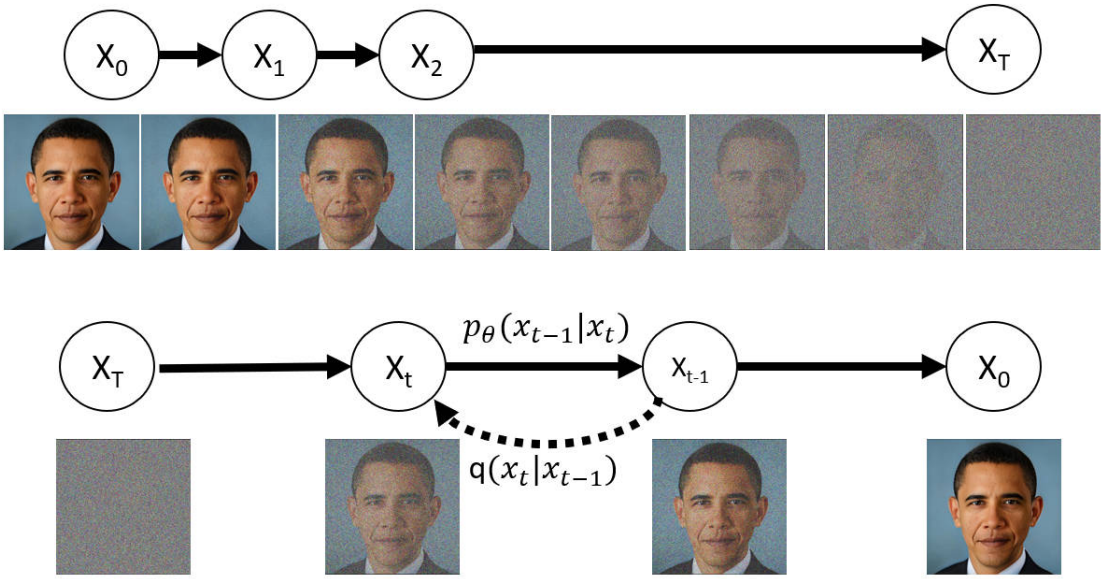


Figure 2.10: Architecture of Diffusion model. Starting from an image, Gaussian noise is added to bring it into the latent space, and then the denoising process is performed to transform the latent code into the desired image.

2.4.1 Dreambooth

Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation (DreamBooth) [51] is an innovative application of Stable Diffusion in the context of image generation by google research. It leverages the power of Stable Diffusion algorithms to create visually captivating and immersive "dream-like" images. DreamBooth takes an input image and applies a series of diffusion steps, gradually transforming and blending the image to produce surreal and artistic renditions.

By incorporating Stable Diffusion, DreamBooth ensures that the image transformation process remains stable and controlled, preventing the generation of undesirable artifacts or excessive distortion. The diffusion parameters and stability constraints are carefully calibrated to achieve the desired artistic effects while maintaining the overall coherence and quality of the generated images.

DreamBooth opens up exciting possibilities for creative expression, allowing users to explore unique visual styles and generate images that evoke a sense of imagination and wonder. Whether used for artistic endeavors, digital design, or simply for personal en-

joyment, DreamBooth harnesses the power of Stable Diffusion to create mesmerizing and dream-like images that push the boundaries of traditional image generation techniques.

The image depicted in Figure 2.11 illustrates the capability of DreamBooth to synthesize images that cater to your imagination. By utilizing the identity of the input images, DreamBooth empowers you to envision and create visuals that transcend the boundaries of reality. With DreamBooth, you can explore and materialize the limitless possibilities of your dreams, taking you on a visual journey to uncharted territories.

Figure 2.11: Examples of how the DreamBooth can synthesize images and extract identity of an object. Reprinted from [51]

The architecture of DreamBooth is illustrated in Figure 2.12. This diagram provides an overview of the underlying structure and components of DreamBooth. It showcases the intricate design and functionality of the system, highlighting how different elements work together to enable the generation of captivating and imaginative images. Figure 2.12 serves as a visual representation of the architecture, offering insights into the inner workings of DreamBooth.

2.5 OpenPose

Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields (OpenPose) [10] is a widely used computer vision framework that focuses on human pose estimation. It is a real-time multi-person keypoint detection system that can accurately identify and track

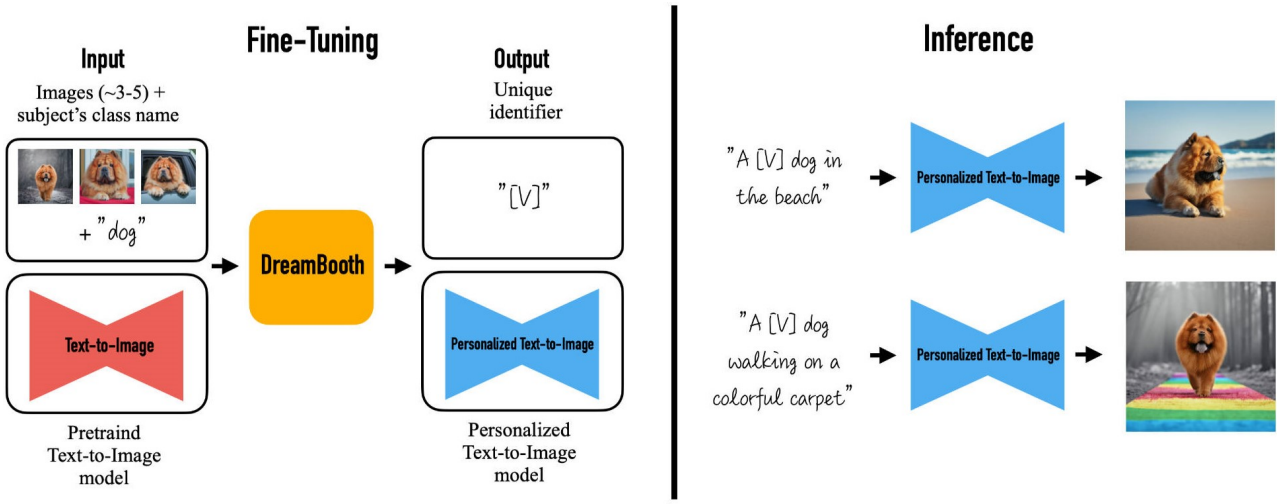


Figure 2.12: Architecture of DreamBooth. **Fine-tuning** is the process of optimizing a pre-trained model on a specific task or dataset to improve its performance and adaptability. It involves adjusting the model's parameters to better fit the new task or dataset, allowing it to learn task-specific features and improve its accuracy. **Inference** is the process of using a trained model to make predictions or generate outputs based on new input data. It applies the learned knowledge of the model to produce useful insights or perform specific tasks on real-world data. Reprinted from [51]

human body keypoints, such as the locations of joints and body parts, from images or video footage.

Due to the inherent limitations of basic diffusion models, they cannot directly generate profile views from prompts. Therefore, in our methodology, we employ OpenPose as a guiding tool for the diffusion models, enabling them to produce accurate profile views.

The OpenPose framework employs deep learning techniques to analyze human poses and detect key body landmarks, including the positions of the head, torso, arms, and legs. It utilizes a two-stage approach, involving a convolutional neural network (CNN) for body part detection and a subsequent refinement step to improve the accuracy of the keypoints. By utilizing OpenPose, developers and researchers can extract valuable information about human poses, movements, and interactions. This information can be applied to a wide range of applications, including action recognition, gesture analysis, augmented reality, sports analysis, and human-computer interaction. OpenPose has gained popularity due to its high accuracy, real-time performance, and availability as an open-source framework. Its versatility and robustness make it a valuable tool for various computer vision tasks

involving human pose estimation.

The example output of OpenPose is illustrated in Figure 2.13, showcasing the detected human body keypoints and their corresponding connections.

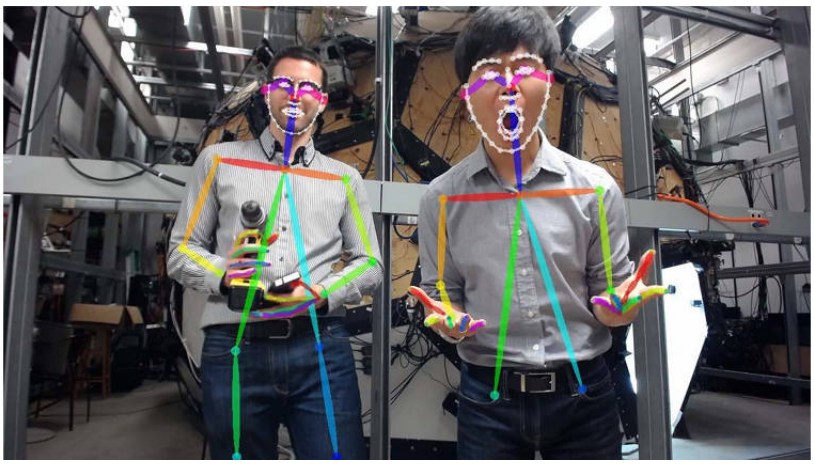


Figure 2.13: Example of OpenPose to get the keypoints of human's face and body. Reprinted from [10]

2.6 Summary

In our work, we cover the fundamental principles of machine learning, specifically unsupervised learning, Autoencoders, GAN networks and Stable diffusion. Our research is centred around the application of these concepts to predict different pose information for 3D models.

We delved into the realm of unsupervised learning, which refers to machine learning techniques that aim to uncover patterns and structures in data without the use of explicit supervision or labels. Unsupervised learning algorithms, such as clustering and dimensionality reduction methods, play a crucial role in discovering underlying patterns and extracting useful information from unlabelled data. we also provided a detailed introduction to autoencoders, which are neural network models used for unsupervised learning tasks. Autoencoders are designed to encode input data into a low-dimensional latent space and then decode it back to reconstruct the original input. This process allows autoencoders to learn meaningful representations of the data without requiring explicit labels. we explored the field of Generative Adversarial Networks (GANs) and examined several relevant works

such as StyleGAN, interfaceGAN, HyperStyleGAN, PTI, and IDomainGAN. These works have served as inspirations for our own architecture, as we integrate their techniques to synthesize human faces in a wide range of poses. By leveraging the advancements made in these GAN-based approaches, we aim to enhance the realism and diversity of the generated facial images in our research. We provided an extensive overview of the diffusion model method and its related techniques, including stable diffusion for high-quality face profile generation. We introduced DreamBooth, utilizing stable diffusion to create images based on input identity, and highlighted OpenPose for body pose extraction. These tools are vital in our approach to construct human face profile views.

Chapter 3

Related Works

3.1 3D Model

A 3D model is a mathematical representation of any three-dimensional object (real or imagined) within a computer. There are two primary types of 3D models: solid and shell (or boundary). Solid models define the volume of the object they represent, like a rock. Shell/boundary models, on the other hand, represent the surface, not the volume, of an object and are therefore more akin to a hollow mask. There are several 3d model techniques available:

Three-Dimensional Morphable Models (3DMM) [9] is a powerful tool for creating realistic 3D models of human faces. It is a statistical model that represents the variations in shape, texture, and lighting of a given population of faces [12].

Deep Learning-Based Methods [24]: Techniques such as convolutional neural networks (CNNs) have been used for 3D face reconstruction. These methods typically learn a mapping from 2D images to 3D face shapes from a large dataset.

Volumetric Methods [46]: These methods often operate directly on the 3D space, using voxels (volume pixels) to represent the 3D face shape. These can be particularly useful when the goal is to create a model that represents the entire volume of the face, not just the surface.

Multi-view Stereo (MVS) Techniques [69]: These methods use multiple images of the subject taken from different viewpoints to reconstruct the 3D face shape. These can be very effective when multiple images are available but less so when only a single image is available.

In our research, we based on deep learning networks to learn the 3D features of the 3D face, such as albedo map, depth map, camera location and lighting to reconstruct the 3D face.

3.2 Unsupervised 3D Shape Learning

In recent years, there has been a growing interest in the unsupervised learning of 3D shapes from monocular view images, which offers the advantage of reducing the heavy burden of manual annotations. Such as other 3D face reconstruction with supervised learning method papers [62] [25], they need a lot of annotations to train their network.

However, a significant challenge arises from the lack of multiple views or lighting images for a single instance. To overcome this challenge, several approaches have been proposed that leverage external 3D shape models or weak supervision, such as 2D key points.

For example, Gecer et al. (2019) [13], Kanazawa et al. (2018a) [26], Sanyal et al. (2019) [54], and Shang et al. (2020) [55] utilize 3D Morphable Models (3DMM) or Skinned Multi-Person Linear model (SMPL) as prior knowledge to reconstruct the 3D shape of human faces or bodies. On the other hand, Tran & Liu (2018) [60] and Kanazawa et al. (2018b) [27] employ 2D key points to learn the 3D shape of faces or objects.

Some recent approaches, such as Tulsiani et al. (2020) [63] and Goel et al. (2020) [14], aim to remove the reliance on previous supervision but still utilize initial category-specific shape templates. In contrast, other methods, such as Sahasrabudhe et al. (2019) [53], Wu et al. (2020) [66], solely rely on natural 2D images and utilize autoencoders to disentangle the viewpoint and shape information of each image.

In our research, we have aimed to implement unsupervised learning techniques to minimize the dependency on image annotations. By providing the neural network with a richer set of data, our goal is to generate more realistic reconstructions of 3D faces.

3.3 Single-image 3D Reconstruction

Single image 3D reconstruction is a task in computer vision that aims to infer the 3D structure and geometry of a scene or object from a single 2D image. The goal is to recover the depth, shape, and spatial layout of the scene or object, even though the information provided by the single image is inherently limited.

Single image 3D reconstruction methods typically employ a combination of geometric reasoning, machine learning techniques, and prior knowledge to estimate the 3D structure. These methods leverage cues such as shading, texture gradients, perspective, and object contours to infer depth and reconstruct the underlying 3D geometry. To achieve accurate 3D reconstruction, single image approaches often rely on learned models or algorithms that generalize from large datasets of 3D shapes and corresponding 2D images. These models capture the statistical relationships between 3D structure and its appearance in 2D images, allowing for inference of the missing depth information from the single image. In the existing literature on unsupervised single-image 3D face reconstruction using autoencoder networks, such as those presented in the works of Wu et al. [66]and Pan et al. [44], there is room for enhancing both the accuracy and the realism of the reconstructed 3D faces. The results from their papers can not make the 3D face looks the same as the input images.

In this paper, we tried to use our proposed 3D face reconstruction architecture with Neural 3D mesh to improve the accuracy and realism of the 3D faces.

3.4 Summary

The related work is centred around the related 3D model reconstruction methods.

We investigate various works related to 3D models, such as unsupervised 3D shape learning, single-image 3D reconstruction. These works provide us with valuable insights and serve as the foundation for developing our 3D faces. By incorporating these techniques into our research, we aim to improve the realistic and expressive 3D models of human faces using a combination of learned shape representations and advanced rendering methods.

Our approach differs from methods that rely on 3D Morphable Models (3DMM) or SMPL as prior knowledge or those that use 2D key points to learn the 3D shape. Instead, our method utilizes autoencoders trained on diverse datasets to capture a wide range of pose variations from a single image, enabling the reconstruction of a comprehensive 3D face model. From our evaluation, we can improve the accuracy and realism of the 3D face.

Chapter 4

Methodology

Over the past few years, 3D face reconstruction, particularly the use of a single image to generate its 3D model, has received considerable attention. However, previous studies have produced 3D faces from a single image with unsatisfactory similarity and quality. While one paper employed 2D GANs to generate manifold faces for 3D face reconstruction, the process was prohibitively slow. Therefore, the challenge of fast and high-quality 3D face reconstruction from a single image remains unresolved.

Building upon the work of Unsup3D by Wu et al. [66], we were inspired to explore ways to enhance the quality and realism of 3D face reconstruction. Recognizing the limitations posed by using a single input image in the Unsup3D method, we hypothesized that incorporating multiple facial poses as inputs could improve the final 3D face model, as this approach would provide more information to the model.

To test this hypothesis, we explored several methods to generate images of varying facial poses, primarily focusing on Generative Adversarial Networks (GANs) and other neural networks. Our research revealed a gap in the current literature, with few studies focusing on the use of GANs and diffusion models for 3D face modelling. Motivated by this finding, we embarked on an exploration of GAN and diffusion methods to generate images of different facial poses, which yielded promising results.

Our proposed architecture combines Hyper Style GAN and Diffusion Model methods to reconstruct 3D models with realistic profile views. By employing an optimization framework, we generate high-quality albedo maps, depth maps, viewpoint estimates, and lighting directions. Additionally, we apply albedo map combination and depth map optimization techniques to enhance the realism of the 3D models. Our approach yields accurate and detailed 3D models that closely resemble the input image with high SIDE, Mean Opinion

Score and MAD while maintaining efficiency in the generation process with fast generating speed.

4.1 System Overview

The overview of our system is shown in Figure 4.1. The roles and functionalities of each component within the pipeline are depicted in the accompanying Figure 4.2. The system has two parts:

- Different pose generation:

  * HyperStyleGAN and InterfaceGAN to generate 14 different poses of input face.

  * Use the 14 different poses of the input face as inputs to feed to stable diffusion model to synthesize profile view faces by dreambooth and Openpose.

- 3D face reconstruction:

  * 3D features prediction network and 3D rendering network to construct the input's 3D face with its different poses images.

Figure 4.3 illustrates the process of albedo map combination in our proposed architecture. After obtaining the HyperStyle results using autoencoders, we integrate the albedo maps to generate the final 3D face model. This step involves combining and blending the albedo maps obtained from different poses to create a comprehensive and realistic representation of the face.

4.2 GAN Inversion and Manipulation

To create 3D facial reconstructions, the first step involves acquiring information on the various poses of the face. However, since the networks are incapable of predicting unknown 3D faces, it becomes necessary to generate believable variations of the input face poses and provide them as input to the network to ensure accurate 3D construction.

With the recent significant progress achieved in GAN inversion and manipulation techniques, employing these methods to create realistic pose variations is a feasible approach. In this study, we employed the HyperStyle GAN technique to perform GAN inversion,

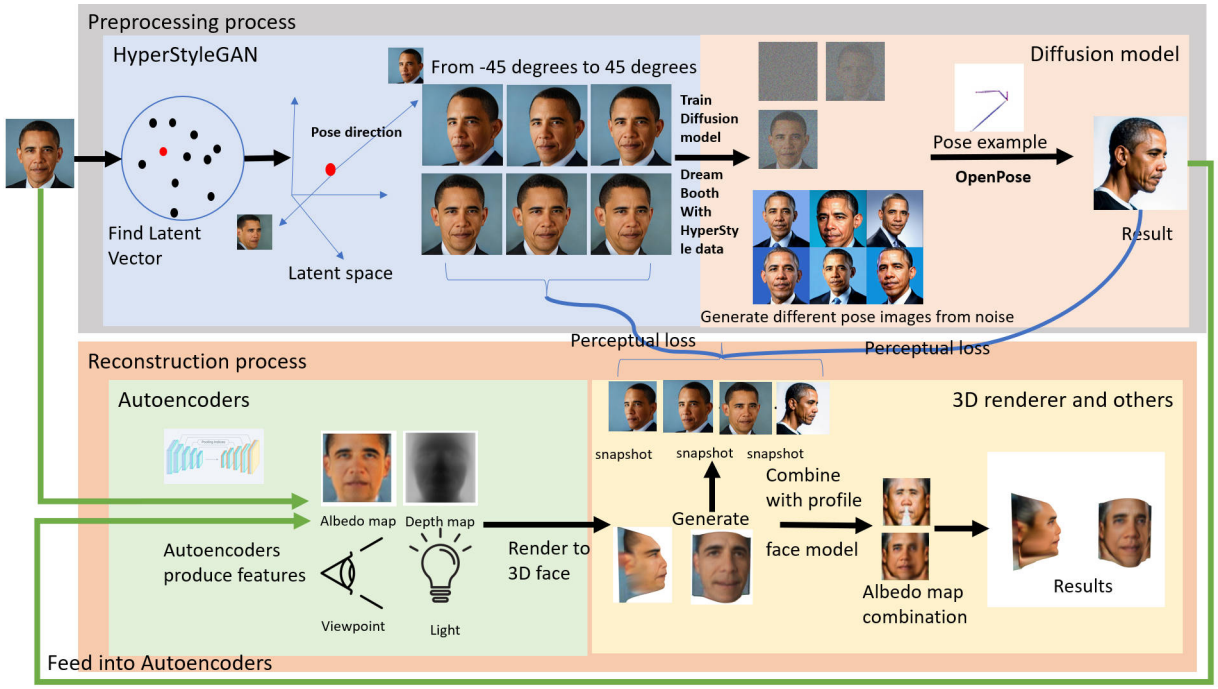


Figure 4.1: Our proposed system operates on a single-view image input and leverages the power of HyperStyle GAN network and Autoencoders. Initially, the HyperStyleGAN network is utilized to reconstruct the input face and extract its latent code representation. Subsequently, the interfaceGAN is employed to manipulate the latent code, enabling the generation of diverse facial poses. To further enhance the realism and accuracy of the generated images, we integrate the stable diffusion model, which specifically focuses on producing accurate profile views of the face. Lastly, we employ a series of carefully designed loss functions to optimize the 3D predicted face in accordance with the various pose images, ensuring a high-quality and realistic output.

allowing us to obtain the latent code for the input image. As the equation shown, we need to minimize the reconstruction distortion with a given input image x:

$$\hat{w} = \arg\min_{w} L(x, G(w; \theta)) \tag{4.1}$$

where G is a pre-trained generator parameterized by weights theta, w is latent code, and Theta is the generator's parameters. Here is StyleGAN. L is the loss function. Figure 4.4 shows how the HyperStyleGAN network does the inversion task.

Once we obtain the latent code using HyperStyle GAN, the next step involves creating various poses of the input face using the interfaceGAN technique. This method allows us

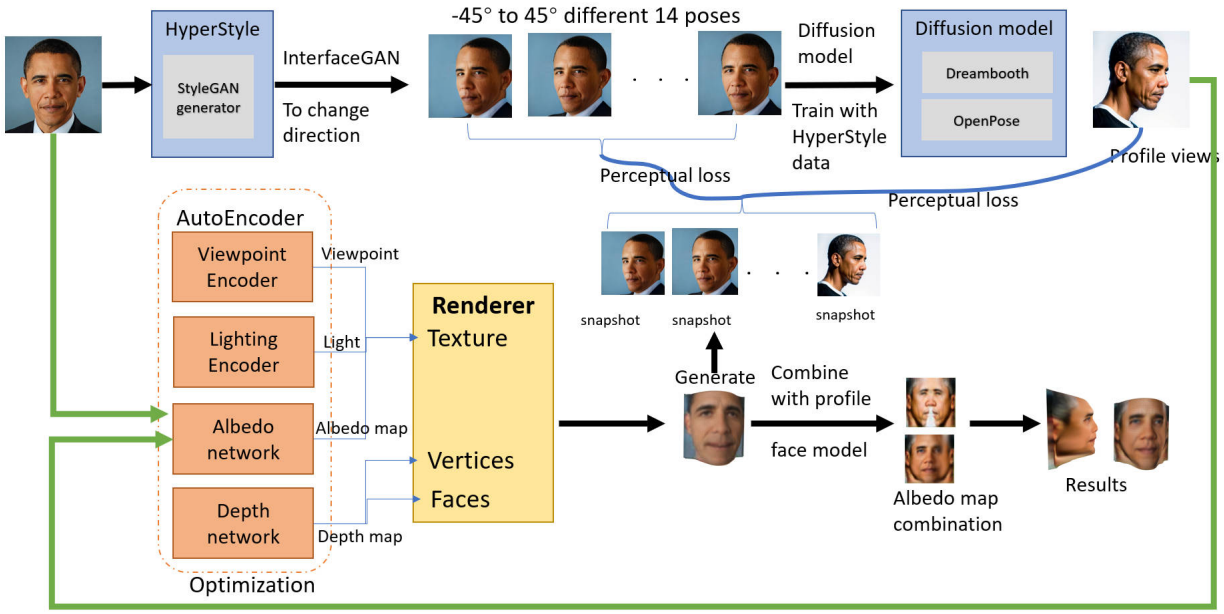


Figure 4.2: Our proposed system operates on a single-view image input and leverages the power of HyperStyle GAN network and Autoencoders. Initially, the HyperStyleGAN network is utilized to reconstruct the input face and extract its latent code representation. Subsequently, the interfaceGAN is employed to manipulate the latent code, enabling the generation of diverse facial poses. To further enhance the realism and accuracy of the generated images, we integrate the stable diffusion model, which specifically focuses on producing accurate profile views of the face. Lastly, we employ a series of carefully designed loss functions to optimize the 3D predicted face in accordance with the various pose images, ensuring a high-quality and realistic output.

to modify each attribute in the latent space by altering the direction by changing the value of α. The equation used for this purpose is:

$$x^{'} = G(w + \alpha n) \tag{4.2}$$

where G represents the StyleGAN generator, w is the latent code obtained from HyperStyleGAN, and n denotes the direction of the pose.

By using this approach, we obtain the necessary data to enhance the performance of our autoencoder networks.

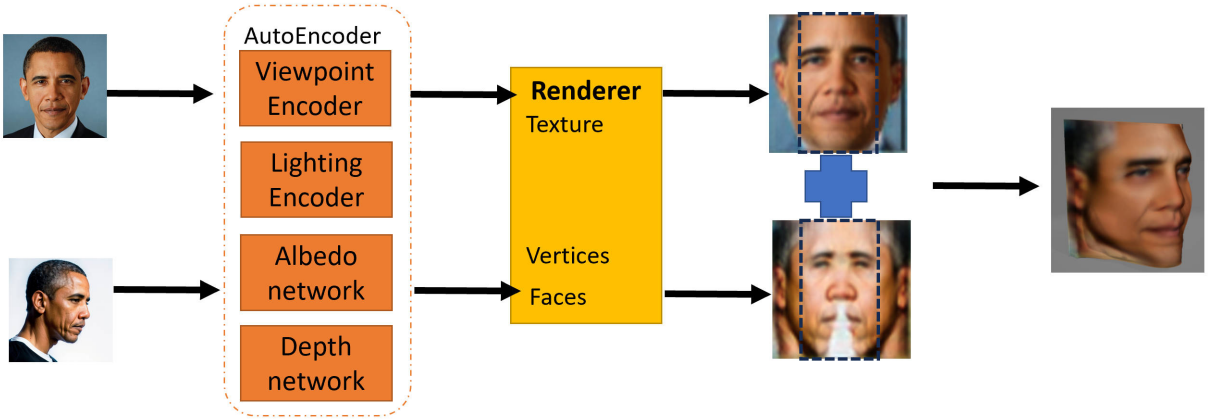


Figure 4.3: Albedo map combination: After optimizing the model, we generate multiple 3D faces based on different pose images. These faces are synthesized by combining their corresponding albedo maps to render a final albedo map which is better. Furthermore, we perform depth map correction to enhance the realism of the 3D faces.

4.3 Diffusion Model to Create Profile Views

Despite the HyperStyle GAN's capability to achieve a balance between reconstruction and editability, it encounters challenges when generating satisfactory profile view poses using its latent code. To address this limitation, we incorporate Stable Diffusion methods into our framework to enhance the quality of profile view poses.

Our approach involves utilizing the results obtained from the HyperStyle GAN to generate 14 different pose images of the same individual through GAN inversion and manipulation techniques. These pose images serve as input for the DreamBooth tool, which trains the model to capture the identity of the input face accurately. By leveraging the DreamBooth tool, we enhance the ability of our system to preserve the facial identity during profile view synthesis.

To further refine the profile view of the input image, we employ the Stable Diffusion model in combination with OpenPose. This allows us to generate high-quality profile view images that align with the natural pose of the individual. The Stable Diffusion model leverages the pose estimation capabilities of OpenPose to ensure accurate alignment and pose representation in the synthesized profile view.

Through this combined approach, we are able to overcome the limitations of the Hy-

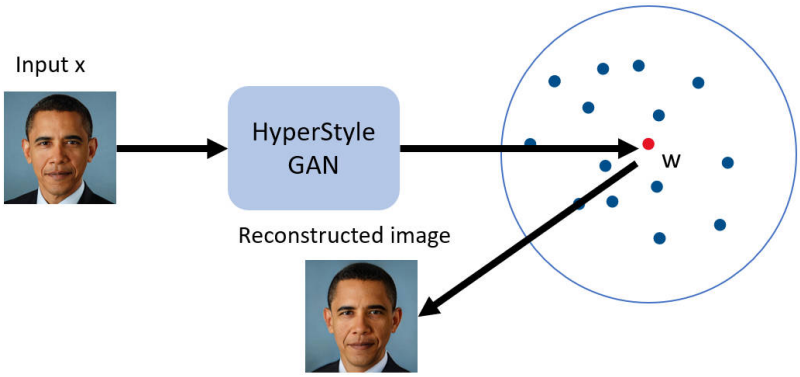


Figure 4.4: Concept of HyperStyle GAN inversion.

perStyle GAN and achieve more realistic and visually pleasing profile view poses for the reconstructed 3D faces.

The architecture depicted in Figure 4.5 illustrates the process of generating profile view images using the combination of HyperStyle, Stable Diffusion model, DreamBooth, and OpenPose.

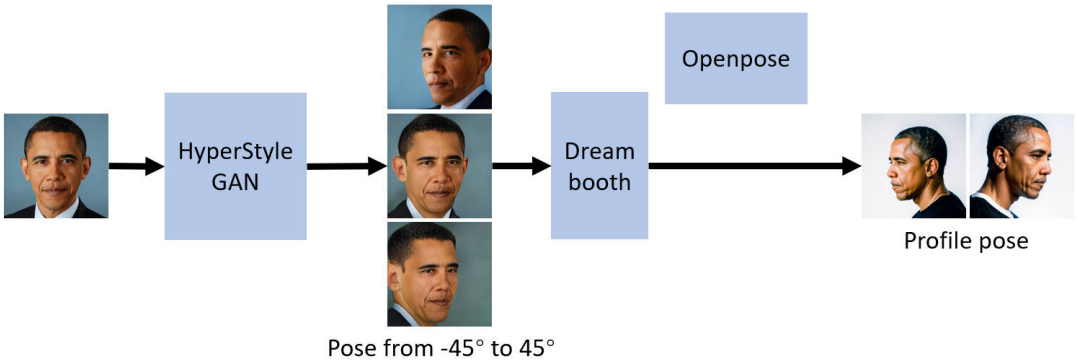


Figure 4.5: Pipeline of the stable diffusion model to generate profile view images. The first step involves generating 14 pose images, which are then used to train the model in DreamBooth for identity extraction. Finally, OpenPose is utilized to obtain the profile view of the face.

4.4 Photo-geometric Autoencoding

Given a grid $\Omega = \{0, ..., W-1\} \times \{0, ..., H-1\}$, an image I can be defined as a function $\Omega \rightarrow R^3$or equivalently, as a tensor in $R^3 \times W \times H$. The object of interest is assumed to be approximately centred in the image. The objective is to train a neural network function ϕ to map the image I to four factors (d, a, w, l) that consist of a depth map $d : \Omega \rightarrow R^+$, an albedo image a: $\Omega \rightarrow R^3$, a global light direction $l \in S^2$, and a viewpoint $w \in R^6$, enabling the reconstruction of the image I from these factors. R^6 refers to a six-dimensional Euclidean space. R^+ means that a depth map is an element of the set of positive real numbers. S^2 represents a two-dimensional sphere centered at the origin (0, 0, 0) in three-dimensional space. R^3 refers to a three-dimensional Euclidean space. The image I is reconstructed using two steps: lighting $\wedge$ and reprojection $\sqcap$ as follows:

$$\hat{I} = \sqcap(\wedge(a, d, l), dw) \tag{4.3}$$

In order to generate an object version, the lighting function $\wedge$ utilizes the albedo a, depth map d, and light direction l, from the perspective of a canonical viewpoint w = 0. The viewpoint w denotes the transformation from the canonical view to the actual input image I's viewpoint. The reprojection function $\sqcap$ then emulates the effect of a viewpoint change, resulting in the image $\hat{I}$ using the canonical depth d and shaded canonical image $\wedge(a, d, l)$. Learning is accomplished through a reconstruction loss, which ensures that the image I is similar to the reconstructed image $\hat{I}$. Figure 4.6 shows the process of autoencoder optimization

4.5 3D Model Rendering

4.5.1 Neural Mesh Renderer

The Neural Mesh Renderer [33] is a neural network-based model developed for the task of image synthesis from 3D models. It was proposed in a research paper by Kato et al. in 2018 and has since become a popular tool for various computer vision and graphics applications.

The key innovation of the Neural Mesh Renderer is its ability to render high-quality images of 3D models with complex geometric structures and material properties. Unlike traditional rendering methods, which require a lot of manual tuning and optimization, the

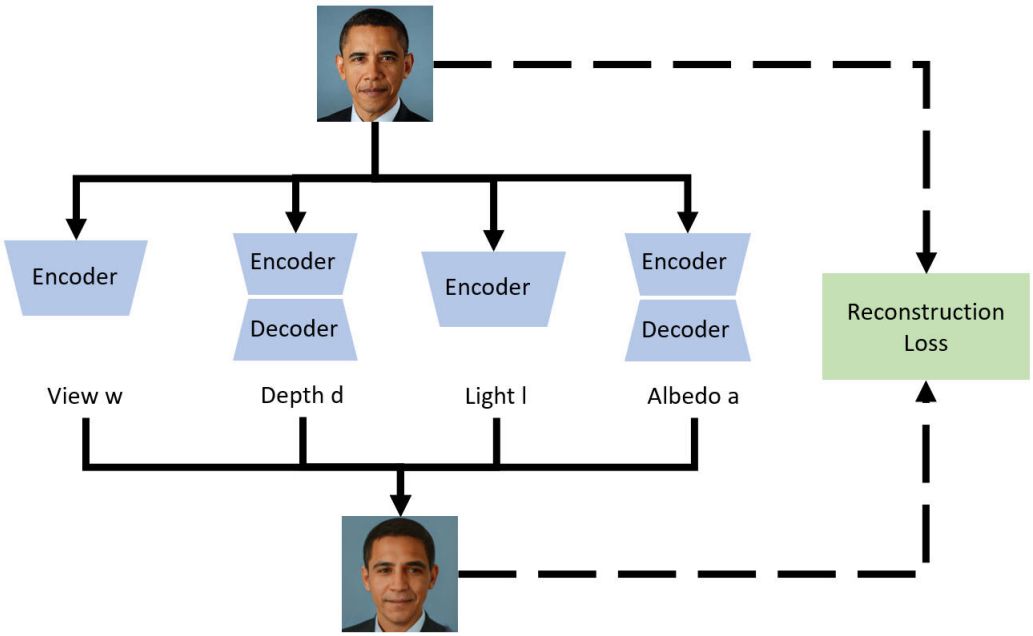


Figure 4.6: Autoencoder optimization.

Neural Mesh Renderer is trained end-to-end using deep neural networks. This allows it to automatically learn the mapping between 3D models and 2D images, and generate highly realistic and visually pleasing renderings.

The Neural Mesh Renderer uses a mesh-based representation of 3D models, where each vertex of the mesh is associated with a texture and a normal vector. The model takes as input the mesh geometry, camera parameters, and lighting conditions, and produces an output image that is rendered from the specified viewpoint. The model is trained on a large dataset of 3D models and corresponding images, and is able to generalize to novel shapes and viewpoints.

The Neural Mesh Renderer has been used for a wide range of applications, including 3D object reconstruction, virtual reality, and augmented reality. It has also been used as a tool for generating synthetic training data for various computer vision tasks, such as object detection and segmentation. Figure 4.7 shows how the neural renderer reconstructs a new mesh with a single image.

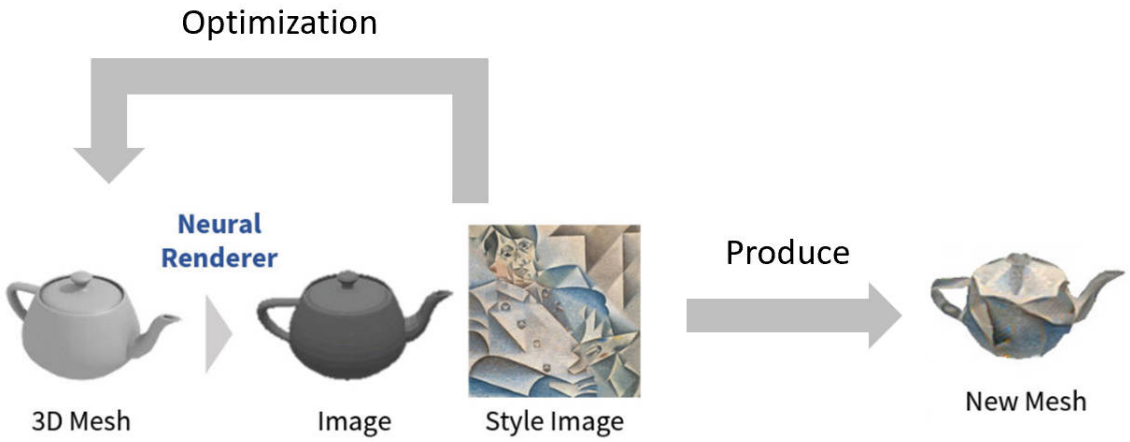


Figure 4.7: Pipeline for single-image 3D mesh reconstruction. Reprinted from [33]

4.5.2 Rendering

In this study, we utilized the Neural mesh renderer to generate our 3D model. The Neural mesh renderer employs an isotropic sphere consisting of 642 vertices, and each vertex v_i is shifted by a local bias vector b_i and a global bias vector c, denoted as $v_i + b_i + c$. The range of vertex movement is restricted within the same quadrant of the original sphere, and the faces f_i remain unchanged. Therefore, the intermediate outputs of the generation function G(x) are represented by $b \in R^{642\times3}$ and $c \in R^{1\times3}$. The mesh is characterized by 642×3 parameters, which is significantly less than the typical voxel representation with a size of 32^3. This low dimensionality is presumed to be advantageous for shape estimation.

The generation function G(x) is trained using silhouette loss L_{sl} and smoothness loss L_{sm}. Silhouette loss measures the difference between the reconstructed silhouettes $\hat{s}_i$ and the correct silhouettes s_i. Smoothness loss acts as a regularizer by ensuring that the surfaces of a mesh are smooth. The objective function is a weighted sum of these two loss functions, $L = \lambda_{sl}L_{sl} + \lambda_{sm}L_{sm}$.

Let si and $\hat{s}_i$ be binary masks, θ_i be the angle between two faces including the i-th edge in G(x), E be the set of all edges in G(x), and $\odot$ be an element-wise product. The loss functions are defined as follows:

$$L_{sl}(x \mid \phi_i, s_i) = -\frac{\mid \hat{s}_i \bigodot s_i \mid_1}{\mid \hat{s}_i + s_i - \hat{s}_i \bigodot s_i \mid_1} \tag{4.4}$$

$$L_{sm}(x) = \sum_{\theta_i \in \varepsilon} (\cos\theta_i + 1)^2 \tag{4.5}$$

L_{sl} corresponds to the negative intersection over union (IoU) between the true and reconstructed silhouettes, while L_{sm} ensures that the intersection angles of all faces are close to 180 degrees. We assume that the object region in an image is segmented beforehand. The mask of the object region is included in the generator as an additional channel of an RGB image.

4.6 Autoencoder Network analysis

The initial step in creating a 3D model resembling the input face involves the recognition of the face in the deep convolution networks. However, recognizing profile view faces poses a significant challenge because the CelebA dataset used for training these networks predominantly features front view faces.

In order to address this issue, it is necessary to train four networks with a considerable amount of profile view data to enhance their proficiency in identifying such views. This is possible due to the networks' capacity to acquire hierarchical features from the images they are trained on, which relies on the backpropagation method for weight updates during training.

4.7 Loss Functions

To optimize the autoencoders, we apply the loss which compares the source image I and the reconstruction $\hat{I}$:

$$L(\hat{I}, I, \sigma) = -\frac{1}{|\Omega|} \sum_{uv \in \Omega} \ln \frac{1}{\sqrt{2}\sigma_{uv}} \exp -\frac{\sqrt{2} l_{1,uv}}{\sigma_{uv}} \tag{4.6}$$

Where $l_{1,uv} =\mid \hat{I}_{uv} - I_{uv}\mid$ is the L1 distance between the intensity of pixels at location uv, and $\sigma < R^{W \times H}$ is a confidence map which expresses the aleatoric uncertainty of the model. Interpreting the loss as the negative log-likelihood of a factorized Laplacian distribution on the reconstruction residuals can lead to optimizing likelihood, which results in model learning.

The $L1$ loss function is prone to producing blurry reconstructions as it is highly sensitive to even minor geometric imperfections. To overcome this issue, we introduce a perceptual loss term. This term involves utilizing an off-the-shelf image encoder "e" that predicts a representation "$e^{(k)}(I)$" for the kth layer, where "$o_{mk} = \{0, \ldots, w_{l-1}\} \times \{0, \ldots, h_{k-1}\}$" represents the corresponding spatial domain. Similar to the equation and assuming a Gaussian distribution, the perceptual loss is calculated as follows:

$$L_p^{(k)}(\hat{I}, I, \sigma^{(k)}) = -\frac{1}{\Omega - k} \sum_{uv \in \Omega_k} \ln \frac{1}{\sqrt{2\pi(\sigma_{uv}^{(k)})^2}} \exp -\frac{(l_{uv}^{(k)})^2}{2(\sigma_{uv}^{(k)})^2} \tag{4.7}$$

Where $l^{(k)}uv = |e_{uv}^{(k)}\hat{I} - e_{uv}^{(k)}I|$ for each pixel index uv in the kth layer. qk is an additional confidence map predicted by our network.

4.8 Albedo Map Combination

Albedo Map Combination involves merging the albedo maps obtained from multiple pose images to create a more complete and visually appealing texture for the reconstructed 3D face. By combining the information from different poses, the resulting albedo map incorporates a wider range of facial details, resulting in a more realistic and high-quality 3D model. The workflow is shown in Figure 4.3.

We utilize both front view and profile view images as input data for the autoencoders. This enables us to create two distinct 3D face models, each accompanied by its own albedo map. To enhance the quality of the albedo map, we merge portions of the profile view albedo map from all four edges with the front view albedo map. Finally, we employ this improved albedo map to render the 3D face models.

In the process of applying the albedo map combination, we begin by taking the output of the albedo map for the frontal pose and designate it as the primary albedo map. Subsequently, we utilize the albedo map output for the profile pose. Since both images share the same dimensions, we employ Python to compute the one-fourth section of both edges. These two one-fourth sections are then integrated into the primary albedo map. To resolve any conflicting areas between the two sections, we apply a Gaussian function for blurring, resulting in the final albedo map.

4.9 Depth Map Correction

To improve the quality of the 3D face reconstruction, we employ neural networks to predict a more accurate depth map. However, we encounter challenges in accurately capturing the details of the forehead, which can affect the overall realism of the reconstructed face.

To address this issue, we introduce a technique to enhance the appearance of the forehead and make it resemble a real forehead more closely. We adjust the gradient descent process specifically for the forehead region, slowing it down to allow for more precise optimization. By applying this method, we ensure that the reconstructed forehead aligns better with the natural shape and characteristics of a real human forehead.

Furthermore, to maintain consistency and coherence in the forehead area, we employ an interpolation technique. We interpolate the predicted forehead depth based on half the length of the forehead itself. This interpolation helps to ensure that the reconstructed forehead seamlessly integrates with the rest of the face, maintaining a natural and authentic appearance.

The algorithmic process, depicted in Figure 4.8, showcases how these techniques are applied to enhance the quality of the reconstructed forehead in our 3D face reconstruction framework. The forehead consists of 17 layers, labeled from 0 to 16, with layer 0 being the uppermost and layer 16 located at the lowest point of the forehead. We opt to employ our algorithm on the middle layer, which corresponds to layer 6.

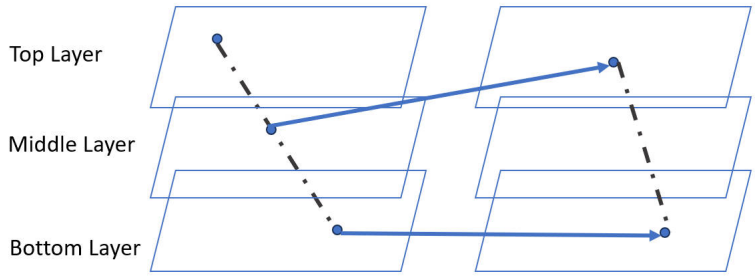


Figure 4.8: Algorithm of Depth Map Correction. We interpolate values between the bottom and the half of the forehead. We utilize the xy panel with varying slice layers, specifically exemplifying layers 0 (Top layer), 6 (middle layer), and 16 (Bottom layer). As an example, we transition a point from layer 6 to layer 0. As a result, we are able to adjust the slope of the forehead to be more vertical.

4.10 Summary

Chapter 4 of our work is dedicated to presenting the architecture we have employed in our research. In this chapter, we provide a comprehensive explanation of how autoencoders are utilized to predict essential features such as the depth map, albedo, viewpoint, and lighting. These predicted features serve as crucial components in the subsequent reconstruction of the 3D model.

Additionally, we emphasize the importance of HyperStyleGAN in capturing latent codes for underlying data characteristics and introduce InterfaceGAN for diverse pose generation in 3D models. We detail the integration of diffusion models to enhance realism in profile view image generation using stable diffusion techniques. We also delve into the utilization of various loss functions, including L1, LSM, and perceptual loss, for autoencoder optimization. These functions refine the reconstructed model. Furthermore, we discuss techniques for expanding autoencoder capabilities to represent input data more comprehensively, improving the 3D model's quality. Lastly, we introduce albedo map combination and depth map correction methods to enhance realism by aligning spatial information with real human faces.

Chapter 4 offers a comprehensive overview of our research's architecture, showcasing the techniques used to achieve realistic 3D face reconstructions.

Chapter 5

Experiment

For conducting our experiments, we utilized a system comprising an i7-9700k Central processing unit (CPU), a GTX 2070 graphics card (Graphics Processing Unit (GPU)), and 64GB of Random Access Memory (RAM). The environment used for this purpose included Windows 11 and Linux, with anaconda installed.

We use Python language with Pytorch framework to develop this pipeline. We combine HyperStyle GAN in our code to provide different pose images to autoencoders and diffusion models. We also develop depth map correction and albedo map combinations in our pipeline. We trained our autoencoders with four datasets (those four datasets has a lot of profile view images).

We use the stable diffusion tool, Neural 3D Mesh tool and other libraries in Python.

5.1 Dataset

Before we can proceed with 3D face generation, it's essential to adequately train our autoencoder networks to recognize and process the input data. Therefore, we trained our autoencoder networks using four substantial datasets, each of which necessarily includes profile views. This training process equips our networks with the ability to generate 3D faces effectively.

5.1.1 Celeb Faces Attributes (CelebA)

The first dataset is Celeb Faces Attributes (CelebA), which includes a diverse range of frontal and profile view faces, allowing us to enhance the network's knowledge through exposure to a wider range of facial variations.

It is a widely used large-scale face attributes dataset that contains over 200,000 celebrity face images, each annotated with 40 binary attributes such as wearing glasses, smiling, or having a beard. This dataset covers large pose variations and background clutter. The images are of varying resolutions and were collected from various online sources.

The dataset is popular in the field of computer vision and machine learning, particularly for tasks related to facial attribute recognition, face detection, face alignment, and face synthesis. It has been used extensively in research for developing and evaluating deep learning models related to these tasks.

Figure 5.1: Samples of CelebA dataset. Reprinted from [37]

5.1.2 Celebrities in Frontal-profile in the Wild

Celebrities in frontal-profile in the wild is a dataset that includes images of various celebrities captured in both frontal and profile views, taken from the internet. The dataset comprises approximately 2000 images, each of which has been annotated with facial landmark points. The landmark annotations facilitate the alignment of the images, enabling frontal and profile views to be easily compared. We use this dataset to add more profile views from real faces.

Figure 5.2: Sample of Celebrities in frontal-profile in the wild dataset. Repainted from [52]

5.1.3 Head Pose Image Database

The Head Pose Image Database is a collection of images that includes 2790 color images of 15 subjects in various head poses. The images were captured using a calibrated camera system, and each image is annotated with the corresponding 3D head pose. The database is useful in developing and evaluating algorithms for head pose estimation, which is a crucial task in many computer vision applications such as human-computer interaction, driver monitoring systems, and virtual reality. The Head Pose Image Database has been widely used in the research community for evaluating and comparing the performance of different head pose estimation methods. The figure 5.3 shows the head pose image dataset

5.1.4 COMSATS Face

Commission on Science and Technology for Sustainable Development in the South (Comsats) Face is a dataset that comprises images of faces captured under varying illumi-

Figure 5.3: Sample of Head pose image dataset. Reprinted from [16]

nation conditions, including indoor and outdoor environments. The dataset includes over 1,000 color images of 52 individuals, with each image annotated with the corresponding gender, age, and ethnicity information. The COMSATS Face dataset is useful in developing and evaluating algorithms for face recognition, face detection, and age and gender estimation tasks, especially in real-world scenarios where lighting conditions can significantly affect the performance of such algorithms. The dataset has been widely used in the research community for developing and evaluating deep learning models related to these tasks.

Figure 5.4: Sample of COMSATS face dataset. Reprinted from [19]

5.1.5 Basel Face Model (BFM) Dataset

To evaluate our results with ground truth, we use the Basel Face Model (BFM) dataset for evaluation. The Basel Face Model (BFM) dataset is a comprehensive 3D face model that includes over 30,000 vertices and provides a detailed representation of facial geometry and texture. The BFM dataset is widely used in computer vision and graphics research,

particularly for developing and evaluating algorithms related to face reconstruction, face recognition, and face animation. The dataset is useful in creating realistic 3D face models and synthesizing new faces from existing ones. The BFM dataset has been extensively used in research for developing and evaluating deep learning models related to these tasks, and it continues to be a popular resource in the field of computer vision and graphics.

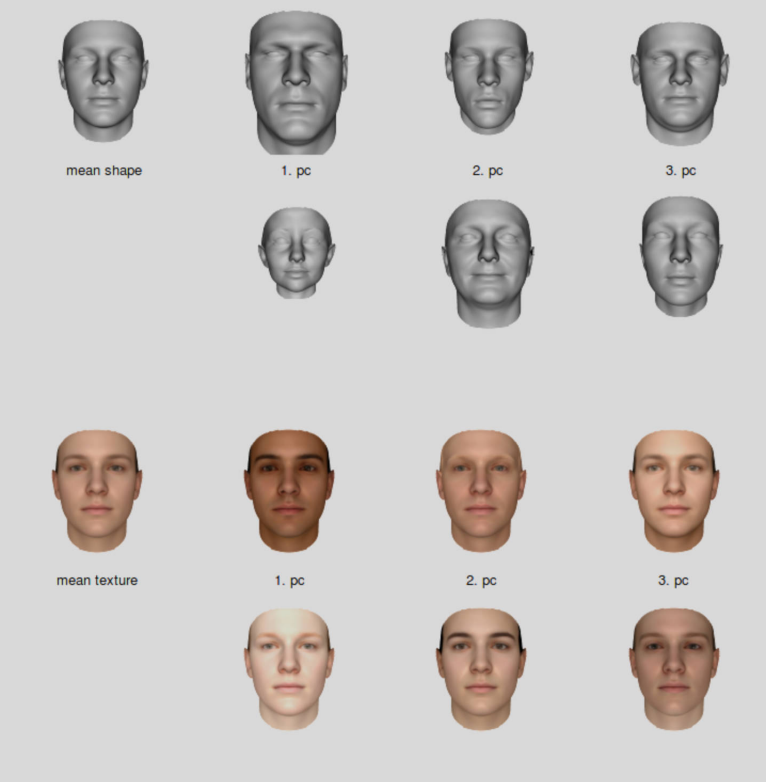


Figure 5.5: Sample of BFM dataset reprinted from [3]

5.2 Implementation details

5.2.1 HyperStyle

As described in our methodology, we utilize HyperStyleGAN to generate the latent code from the input image. We experimented with various state-of-the-art GAN inversion

techniques to obtain the most optimal reconstructed image. In the domain of pose editing, HyperStyleGAN outperforms other GAN inversion methods.

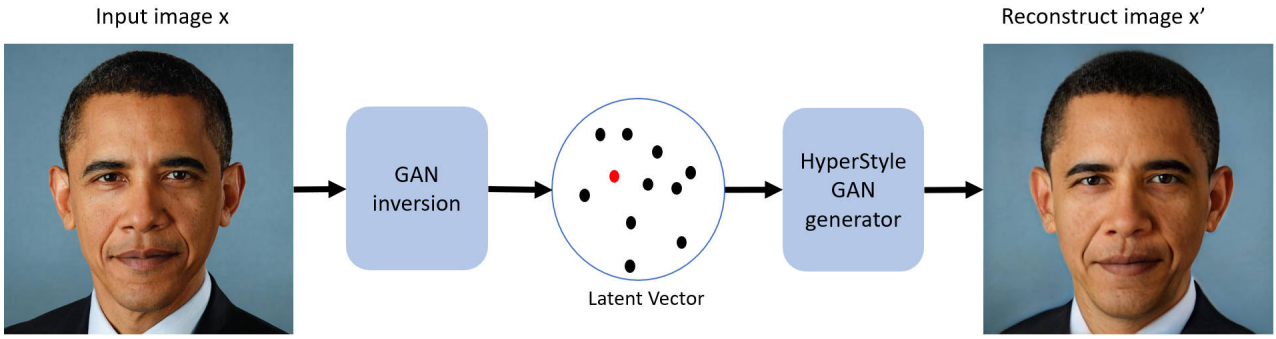


Figure 5.6: Step of getting the latent code of the input face and reconstructing the image from latent space. (Python with HyperStyleGAN code)

5.2.2 Interface GAN

We employed Interface GAN to generate various pose outcomes derived from HyperStyleGAN. As depicted in Figure 5.7, the range spans from -10 to 9, representing -90 degrees to 90 degrees. Notably, we observed that poses from -7 to 6 exhibit satisfactory results, while those from -10 to -8 and 7 to 9 yield unsuitable outcomes for profile views.

Figure 4.1 illustrates the operations of HyperStyleGAN and Interface GAN in the blue area.

5.2.3 Diffusion Model

As discussed in section 5.1.1, the combination of HyperStyleGAN with InterfaceGAN yields satisfactory results for generating diagonal view images. However, when it comes to profile views, specifically poses ranging from -8 to -10 or 8 to 10, the quality of the generated images is not optimal. To address this limitation, we introduce stable diffusion models into our framework.

Initially, we train the model to discern input image identities using a set of pose images derived from HyperStyleGAN, spanning from -7 to 6. We employ the Dreambooth tool [51] for this purpose, integrating identity information during stable diffusion model training, with realism ensured by leveraging the pre-trained realisticVisionV20 model.

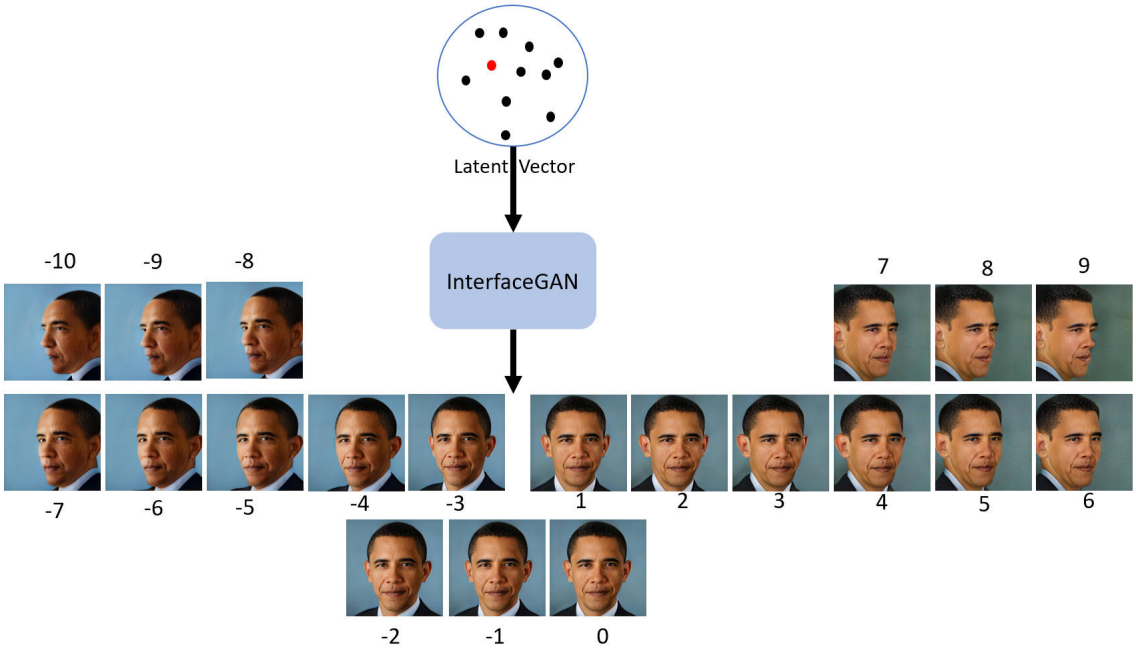


Figure 5.7: Results of interfaceGAN to generate different poses of the face. (Python with pre-trained interfaceGAN data under HyperStyleGAN code.)

Following training with these 14 diverse pose images, the stable diffusion model becomes adept at capturing facial features and producing realistic outputs. Figure 5.8 demonstrates the model's proficiency through synthesized images, validating the effectiveness of our approach.

After incorporating identity into the model, we move on to the next step: using OpenPose to capture the profile view of the input face. This involves a two-step process.

First, we use profile view images from other sources as reference for OpenPose. These images guide OpenPose in extracting pose information, abstracting the necessary details for subsequent synthesis. Second, when synthesizing the target face, we employ this pose information as a guide. This guidance assists the model in generating the intended target pose for the input face, ensuring alignment between the synthesized face and the desired pose. By integrating OpenPose and utilizing pose guidance, our model gains the ability to create precise profile views of the input face, resulting in realistic and aesthetically pleasing outcomes.

Figure 4.1 depicts the functioning of the Diffusion model within our pipeline in the flesh color area..

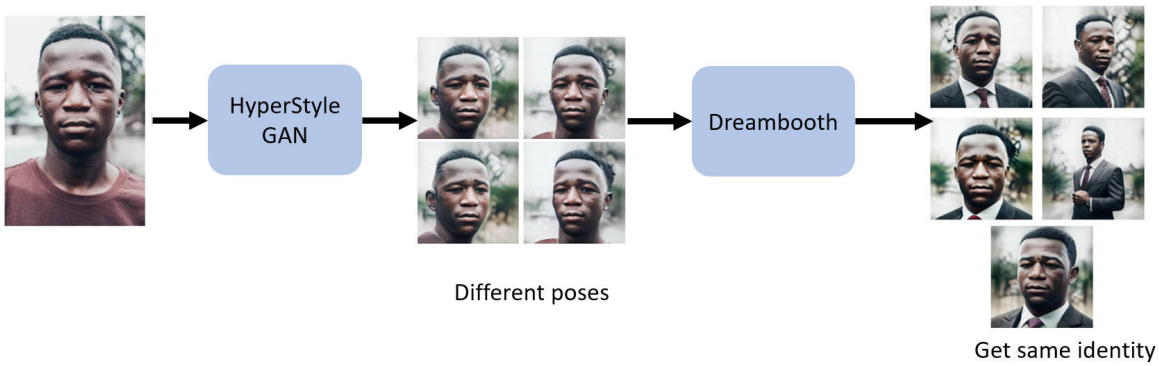


Figure 5.8: Capability of the trained model to generate the same identity. (Python, stable diffusion tool, dreambooth library)

Figure 5.9 showcases the experimental results of our pipeline for generating profile views.

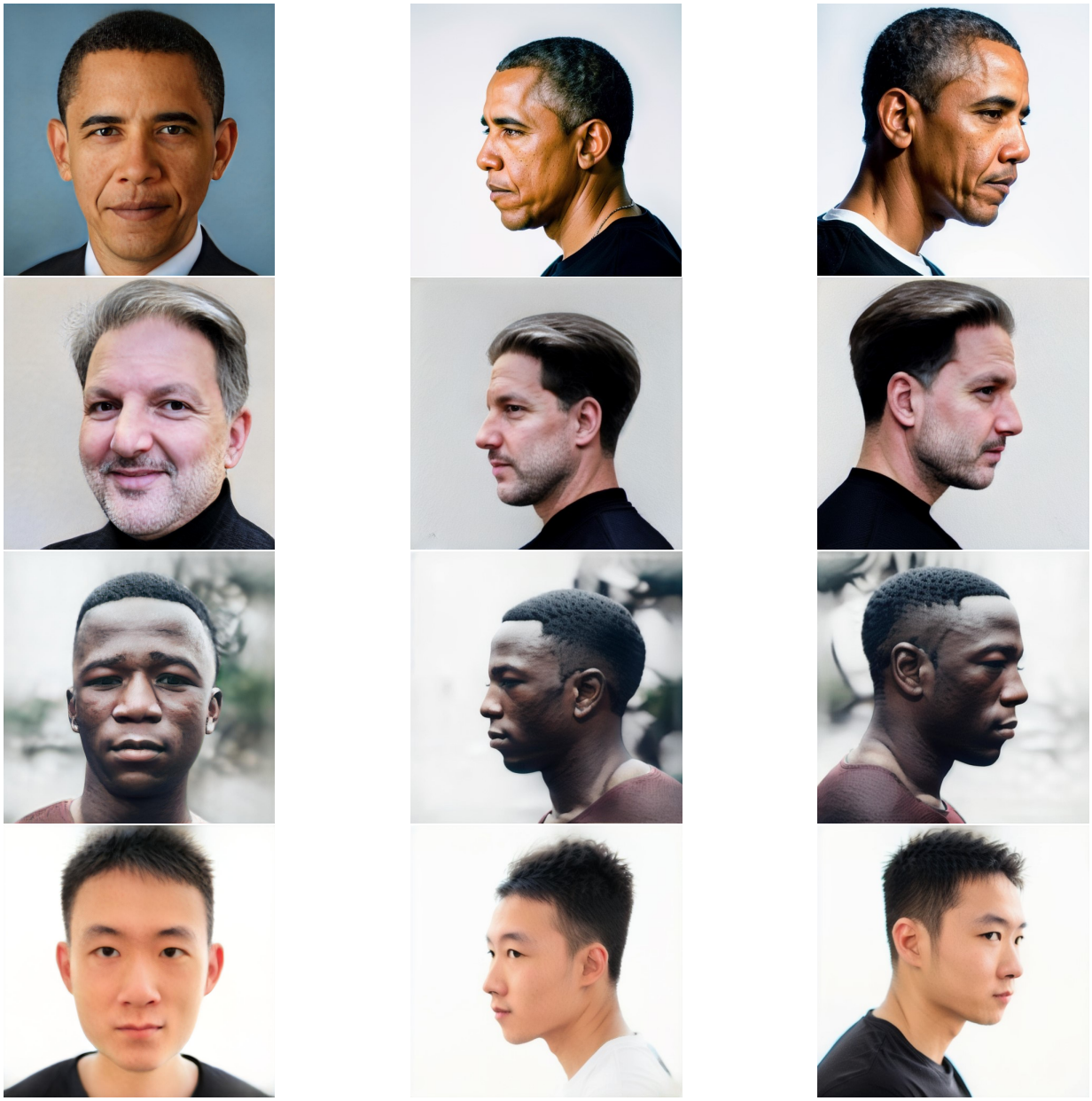
5.2.4 Autoencoders

We enhance the predictive capacity of our 3D face model by training autoencoders using five diverse datasets. Subsequently, we further train these autoencoders using input data that encompasses various poses. These two steps collectively contribute to improved albedo maps, depth maps, accurate viewpoint determination, and lighting considerations. Figure 5.10 visually presents the outcomes achieved through these autoencoder processes.

Figure 4.1 depicts the functioning of the Autoencoders within our pipeline in the green area.

5.2.5 3D Renderer

The neural mesh renderer is applied in this work to render the 3D shape. It takes advantage of the depth map and albedo map obtained through optimization to generate a visually realistic representation of the 3D model. By simulating shading, lighting, and texture mapping, the neural mesh renderer produces high-fidelity renderings that effectively showcase the optimized 3D model. The results are shown in figure 5.11



(a) input image (b) The profile pose 1 (c) The profile pose 2

Figure 5.9: Synthesized profile pose images by Stable diffusion. (Stable diffusion tool with Openpose library. We only use front view to feed to our pipeline to get profile pose of the same person.)

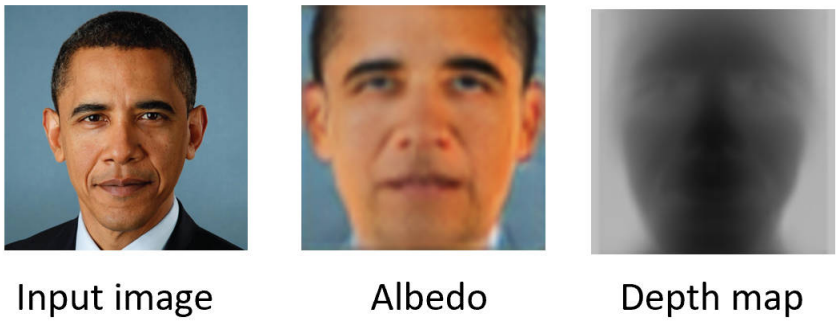


Figure 5.10: Results from autoencoders to predict albedo map and depth map. (Python, autoencoders)

(a) The front pose (b) The profile pose (c) The 20° pose

Figure 5.11: Reconstruct of the 3D face of the input image with neural renderer. (Python, Neural 3D Mesh library)

5.3 Albedo Map Combination

By combining the albedo maps from different poses and profile views, we aim to enhance the overall quality and realism of the synthesized 3D faces. The process involves selectively preserving certain facial features and characteristics from each pose to create a more comprehensive representation of the face.

To achieve this, we analyze the distinctive qualities of the profile view 3D faces and front view 3D faces. The profile view faces excel in capturing the side profile and distinctive contours of the face, while the front view faces provide a clearer view of the frontal features, such as the eyes, nose, and mouth. By combining these complementary aspects, we can

create a more complete and accurate representation of the entire face.

To merge the albedo maps, we apply a blending technique that ensures a smooth transition between the different regions. We use a Gaussian blur function to softly blend the boundary between the profile view and front view areas, resulting in a seamless integration of the albedo maps. This helps to eliminate any noticeable artifacts or abrupt changes in the combined albedo map.

The resulting combined albedo map provides a more comprehensive understanding of the face's appearance, incorporating both profile and front view details. This contributes to the overall realism and accuracy of the synthesized 3D faces, making them visually appealing and closer to the actual human faces.

Figure 5.12 and figure 5.13 serves as a visual example of how the albedo maps are combined, showcasing the blending process and the resulting combined albedo map.

Regarding the albedo map generated by the autoencoders, it's worth noting that we observed the profile view's albedo map to produce a more lifelike 3D model compared to the albedo map from the front view. This advantage stems from the profile view containing richer information about the side of the face, information that the front view lacks. This distinction is illustrated in Figure 5.14.

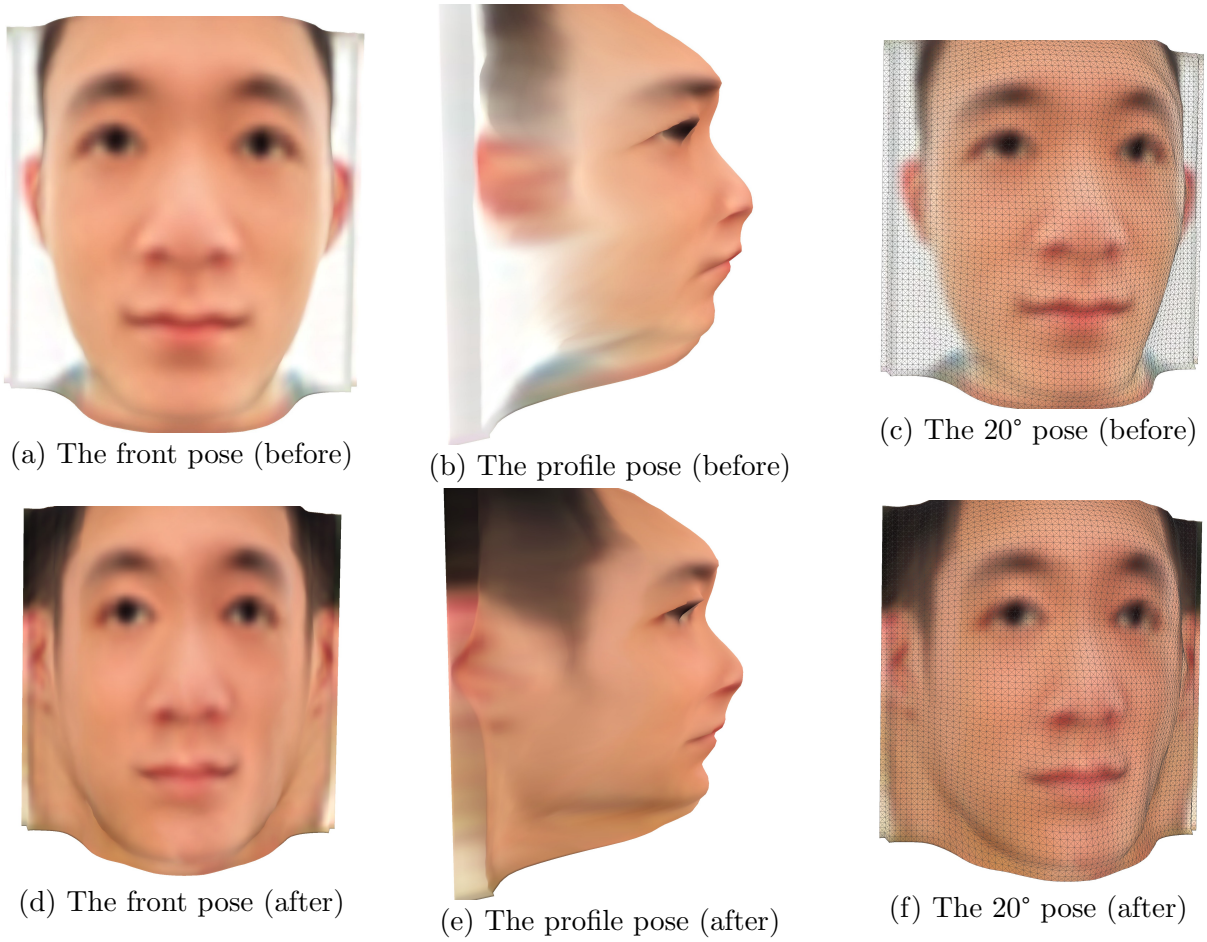
5.4 Depth Map Correction

The subsequent step involves the utilization of the depth map correction method. Upon observing the previous step, it becomes apparent that while the 3D face reconstructions excel in capturing profile views, the forehead region often appears oblique, resulting in an unrealistic appearance.

As detailed in the methodology chapter, our approach addresses this issue by applying the depth map correction method. When generating the 3D face, we specifically target the forehead area and modify the depth map to achieve a more vertical alignment. This is accomplished by interpolating values between the bottom points and the middle points of the forehead region.

The results of this correction method can be observed in Figure 5.15, which serves to illustrate how our approach significantly improves the realism of the forehead region, ultimately enhancing the overall quality of the reconstructed face.

Figure 4.1 depicts the functioning of the 3D renderer, albedo map combination and depth map correction within our pipeline in the yellow area.



(a) The front pose (before)

(b) The profile pose (before)

(c) The 20° pose (before)

(d) The front pose (after)

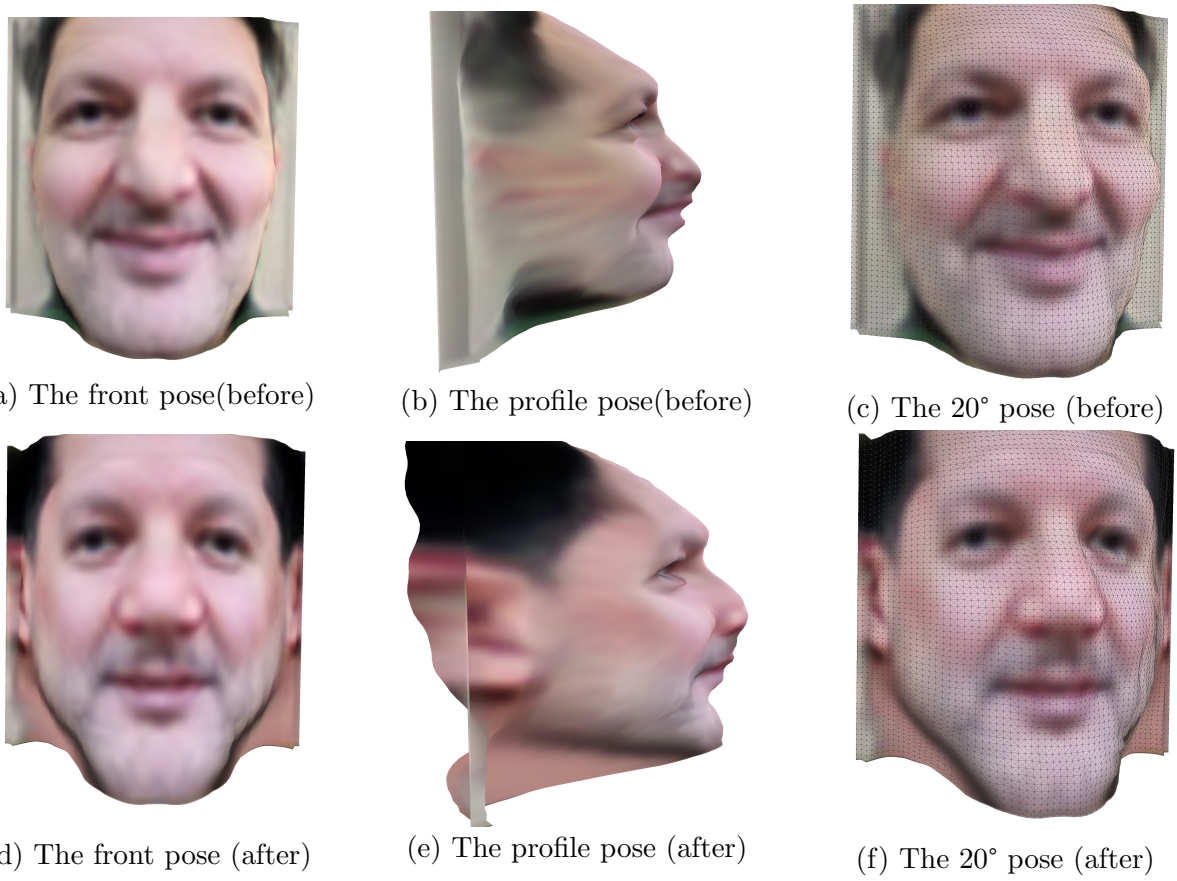
(e) The profile pose (after)

(f) The 20° pose (after)

Figure 5.12: Results before and after albedo map combination. We can see after the albedo map combination, the profile view has more details than before. (Python, new algorithm)

5.5 Summary

In Chapter 5 of our thesis, we provide a comprehensive overview of the experimental methodology employed in our research. This chapter serves as a fundamental basis for our study, outlining the key steps involved in training the prediction networks, extracting the latent code from input images, and ultimately constructing the 3D face model. We begin by detailing the training process of the prediction networks, which are crucial for accurately predicting the essential features of the 3D model. We delve into the architectural design and training procedures employed to ensure reliable and precise predictions. Next,



(a) The front pose(before) (b) The profile pose(before) (c) The 20° pose (before)

(d) The front pose (after) (e) The profile pose (after) (f) The 20° pose (after)

Figure 5.13: Results before and after albedo map combination. We can see after the albedo map combination, the profile view has more details than before. (Python, new algorithm)

we focus on the extraction of the latent code from the input images. The latent code serves as a compact representation capturing the underlying style and content information of the images. We describe the techniques utilized, such as autoencoders or HyperStyle GAN, and emphasize their role in capturing the crucial characteristics of the input images. Addressing the challenge of capturing accurate profile view faces, we introduce the incorporation of the stable diffusion method. By leveraging Dreambooth and Openpose, we utilize stable diffusion to generate profile views, further enhancing the realism and diversity of our 3D face reconstructions. Furthermore, we discuss the optimization process of the prediction networks using multiple pose images. This optimization aims to improve the networks' ability to capture variations in facial expressions, head orientation, and other pose-related attributes. Through this optimization, the prediction networks are fine-tuned to produce

more accurate and diverse predictions. We highlight the significance of obtaining 3D features from the prediction networks, including albedo, depth maps, viewpoints, and lighting information. These features serve as fundamental components for constructing the 3D face model, enabling the capture of intricate details and visual characteristics. Additionally, we introduce the Albedo map combination method, which improves the texture of the 3D face by combining information from both profile views and front views. Finally, we apply the depth map correction method to enhance the realism of the forehead region in the 3D face. By optimizing the depth map specifically for the forehead area, we achieve a more realistic appearance in this region.

Overall, Chapter 5 provides a comprehensive overview of our experimental methodology, and the steps involved in constructing our 3D face model, showcasing the advancements and contributions of our research.

(a) The front pose input albedo

(b) The profile pose input albedo

(c) The combined pose input albedo

(d) The front pose input 20°

(e) The profile pose input 20°

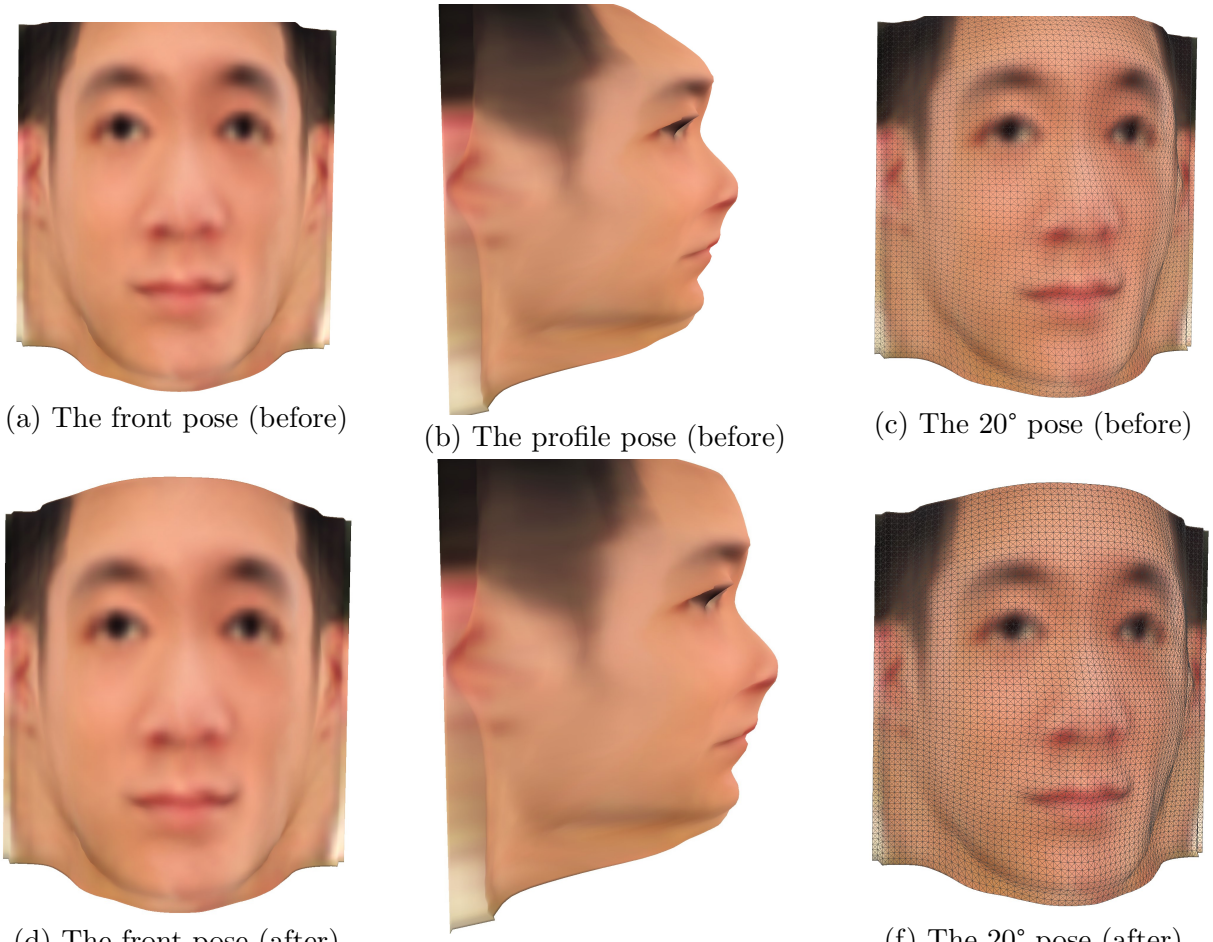
(f) The combined pose input 20°

(g) The front pose input profile

(h) The profile pose input profile

(i) The combined pose input profile

Figure 5.14: Results compared with front pose 3D face, profile pose 3D face and combined 3D face. We can see that profile view 3D face is more realistic. The combined results are the best. (Python, new algorithm)



(a) The front pose (before)

(b) The profile pose (before)

(c) The 20° pose (before)

(d) The front pose (after)

(e) The profile pose (after)

(f) The 20° pose (after)

Figure 5.15: Results before and after depth map correction. The forehead part is more straight than before. (Python, new algorithm)

Chapter 6

Results

In order to assess the effectiveness and performance of our Unsupervised CNN 3D reconstruction model, we conducted a comprehensive evaluation. This evaluation involved comparing the results of our model with those of other existing models, examining its performance from various angles, and analyzing its capabilities on different types of datasets.

Firstly, we compared the results of our Unsupervised CNN 3D reconstruction model with those of other models available in the literature [20, 66]. By evaluating the quality and accuracy of the reconstructed 3D models, we were able to assess the superiority of our approach in capturing the intricate details and overall shape of the objects. This comparison provided valuable insights into the advancements and innovations introduced by our model.

Additionally, we specifically evaluated the performance of our model on untrained profile view datasets. Profile view faces pose a significant challenge in 3D reconstruction due to their unique characteristics and limited availability in training datasets. By assessing the ability of our model to handle profile view inputs and produce accurate 3D reconstructions, we gained valuable insights into its robustness and adaptability.

Furthermore, we investigated the performance of our model using both single image inputs and inputs with 14 different poses with profile view poses. This analysis allowed us to understand the model's capability to handle various input scenarios and determine its effectiveness in capturing the variations in facial expressions, head orientation, and other pose-related attributes. By comparing the results across these different input scenarios, we were able to measure the model's flexibility and accuracy in generating consistent and realistic 3D reconstructions.

Through our extensive evaluation, we obtained valuable empirical evidence of the strengths and limitations of our Unsupervised CNN 3D reconstruction model. The comparisons with other models, the assessment of untrained profile view datasets, and the analysis of different input scenarios collectively contributed to a comprehensive understanding of our model's performance and its potential applications.

6.1 Quantitative Evaluation

Scale-Invariant Depth Error (SIDE) is a widely adopted metric for evaluating the accuracy of depth maps in 3D reconstruction tasks. It's used primarily because of its scale-invariant property, meaning that it is unaffected by the overall scale of the depth map. The equation is typically defined as

$$SIDE = (1/N) * \sum [|log(d_{pred}) - log(d_{gt})|] \tag{6.1}$$

where SIDE: Scale-Invariant Depth Error. N: The total number of depth values being compared (e.g., pixels or points). d_{pred}: The predicted depth value. d_{gt}: The ground truth depth value. In this equation, the logarithm (usually natural logarithm, loge) of the predicted and ground truth depth values is calculated for each point, and the absolute difference is taken. The average of these absolute differences across all points provides a measure of how well the predicted depth values align with the ground truth, while accounting for scale variations. Smaller SIDE values indicate more accurate depth estimation

Mean Angle Deviation (MAD) is another commonly used metric for evaluating the accuracy of depth maps in 3D reconstruction tasks. MAD measures the average angular deviation between the surface normals of the reconstructed model and the ground truth. Surface normals are vectors perpendicular to the surface at each point, providing a measure of the direction the surface is facing. As it provides a meaningful way to measure the quality of the 3D shape and contours. The MAD equation is defined as

$$MAD = arccos(1/N * \sum (cos(\theta_i))) \tag{6.2}$$

where MAD: Mean Angle Deviation. N: The total number of directional data points being compared. θ_i: The angular difference between corresponding data points in the two sets, usually measured in radians.

In the quantitative evaluation, we employ various metrics and measurements to objectively assess the performance of our method. These metrics can include SIDE and MAD.

We compared with Unsup3D [66] and Do 2D GANs [20] method with BFM dataset, and the result is shown in the table 6.1. We create a dataset consisting of 1,000 randomly generated BFM face images along with their corresponding depth maps. To conduct our tests, we divide this dataset into 10 subgroups. For each experiment, we repeat the process ten times to ensure the accuracy and reliability of the results. In our results SIDE is more important for evaluation.

No.	Method	SIDE(x10^2 ↓)	MAD(deg.)	Time
(1)	Unsup3D (Wu et al., 2020)	0.836	16.59	10 seconds
(2)	GAN2SHAPE (Do 2D GAN knows 3d 2021)	0.795	15.01	5 mins
(3)	Ours	0.770	15.17	25 seconds

Table 6.1: Comparison in SIDE, MAD and inference time.

Compared to two other state-of-the-art CNN methods, our approach yields superior SIDE while maintaining similar MAD. Notably, our method significantly reduces inference time in comparison to the GAN2SHAPE [20] optimization method with GAN. Because GAN2Shape can not feed with the input image. So we use the BFM dataset to evaluate it.

Additionally, we conducted comparisons at each stage to assess the impact of the technologies employed on our results. The results of these comparisons are presented in Table 6.2. The table indicates that the Diffusion model has the most significant impact on our algorithm, leading to noticeable improvements in generating better depth maps. Furthermore, it's evident that other technologies also play a role in enhancing the generation of realistic face models through their influence on the depth map.

No.	Method	SIDE(x10^2 ↓)	MAD(deg.)
(1)	Without profile view	0.836	16.59
(2)	Add profile view	0.825	16.48
(3)	Add HyperStyle GAN	0.802	15.93
(4)	Add Diffusion model	0.770	15.17

Table 6.2: Comparison in SIDE, MAD and inference time in each stage.

6.2 Qualitative Evaluation

In the qualitative evaluation, we invited ten people to assess the performance and output of our proposed method visually without knowing which method was ours. We employed the Mean Opinion Score (MOS) as a metric to assess the outcome of our reconstruction. Our method achieved an average score of 8.2, which signifies a marked improvement when compared to the average score of 5.3 yielded by the Unsup3D method. We analyze the generated 3D faces and compare them with the ground truth or target images. We examine the accuracy of the reconstructed facial features, such as the shape, texture, and pose, and assess their resemblance to real human faces. We also evaluate the consistency and realism of the profile views generated by our method. Through this qualitative evaluation, we can gain insights into the visual quality and fidelity of the reconstructed 3D faces.

To assess the Mean Opinion Score (MOS), we enlisted ten individuals from diverse backgrounds to assign scores ranging from 0 to 10 to evaluate the realism of each face model. We conducted this evaluation using four input images, each paired with its corresponding reconstructed 3D face model. We compared the results between the Unsup3D method and our own methods, and the findings are summarized in Table 6.3 Image 1 represents my face, while Image 2 features Barack Obama, Image 3 depicts a European face, and Image 4 showcases an African face.

No.	Method	IMg1	Img2	Img3	Img4	avg. score
(1)	Unsup3D	4.8	5.2	5.6	5.6	5.3
(2)	Ours	8.0	9.1	7.9	7.8	8.2

Table 6.3: Comparison of MOS with Unsup3D method.

As 2D GANs cannot utilize an image without a latent code, we restricted our comparison between Unsep3D and our approach to real faces. The results reveal that our method outperforms Unsup3D [66] in terms of profile view, resulting in a more accurate representation of the original input face. Our method places great emphasis on the importance of incorporating profile view input, which enables the generation of a more realistic facial shape compared with a less profile view information network.The results are shown in Figure tables 6.4 to 6.6

Table 6.4: Examples of face 1 The Unsup3D vs Ours. The first row is Unsup3D method [66], The second row is our method. We can see our methods are more realistic, like real faces, and the profile view has more details.

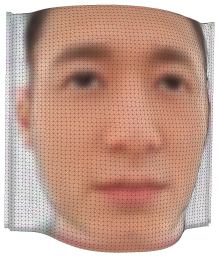
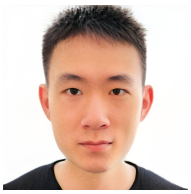
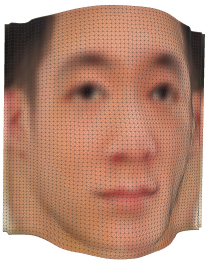
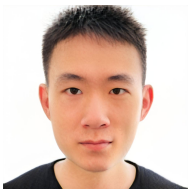
Method	Front pose	45° pose	Profile pose	input
Unsup3D				
Ours				

Table 6.5: Examples of face 2 The Unsup3D vs Ours. The first row is Unsup3D method [66], The second row is our method. We can see our methods are more realistic, like real faces, and the profile view has more details.

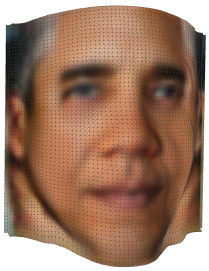
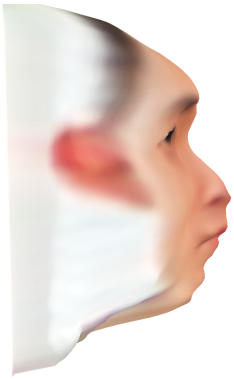
Method	Front pose	45° pose	Profile pose	input
Unsup3D				
Ours				

Table 6.6: Examples of face 3 The Unsup3D vs Ours. The first row is Unsup3D method [66], The second row is our method. We can see our methods are more realistic, like real faces, and the profile view has more details.

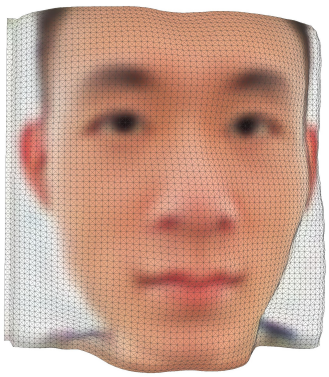
Method	Front pose	45° pose	Profile pose	input
Unsup3D				
Ours				

6.3 Comparison

6.3.1 Single Input Pose vs Multi Input Poses



(a) Single input pose profile view



(b) Single input pose side view

(c) Single input pose front view

(d) Multi poses input pose profile view

(e) Multi poses input pose side view

(f) Multi poses input pose front view

Figure 6.1: In our evaluation, we examined two scenarios: single pose input and Multi poses input. The results clearly showed that single pose input, despite achieving a decrease in loss, led to overfitting of the 3D face. However, using Multi poses input produced better results, generating more accurate and good shape faces.

6.3.2 Pretrained with Profile View Datasets vs Pretained Datasets

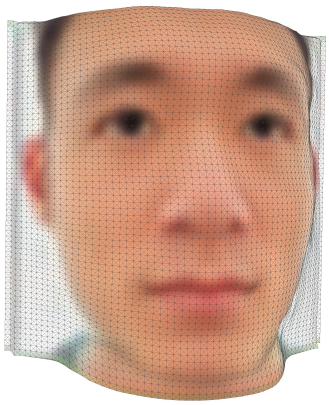
To highlight the impact of training our Autoencoders with a profile view dataset, we sought to demonstrate the differences between the trained and untrained models in capturing profile views. By examining the generated 3D faces, we observed a noticeable distinction between the two sets of results. The results are shown in Figure 6.2

In the trained results, the profile view of the 3D face exhibited a higher level of detail and information compared to the untrained results. This indicates that training the Autoencoders with a profile view dataset enables them to acquire domain knowledge specific to profile views. As a result, the trained models are better equipped to capture the intricacies and nuances of profile facial features, resulting in more accurate and realistic representations.

This comparison emphasizes the significance of incorporating profile view data during the training process. By exposing the Autoencoders to a diverse range of facial orientations, including profile views, we enable them to learn and understand the specific characteristics and variations associated with this particular angle. Consequently, the trained models are able to generate more informative and faithful profile views, enhancing the overall quality and realism of the 3D face reconstructions.

(a) profile view: without pretrained profile view data

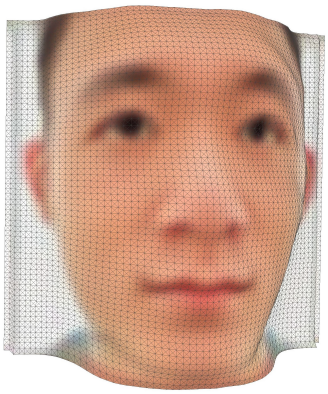
(b) front view: without pretrained profile view data



(c) side view: without pretrained profile view data

(d) profile view: pretrained profile view data

(e) front view: pretrained profile view data



(f) side view: pretrained profile view data

Figure 6.2: The comparison between trained and untrained models using profile data set reveals notable differences. In the case of the trained model, the profile view of the reconstructed 3D face exhibits a higher level of detail and accuracy compared to the untrained model.

6.3.3 Ablation Study

Figure 6.3 showcases a comprehensive comparative analysis, focusing on the effectiveness of incorporating stable diffusion with albedo map combination and depth map optimization techniques in our methodology. This analysis provides valuable insights into the impact of each step in the process of obtaining the final 3D face results, with particular emphasis on the profile views.

By carefully examining the profile views in the figure, we can clearly observe the distinct influence of each step in our methodology. The utilization of stable diffusion, coupled with the albedo map combination, contributes significantly to enhancing the realism and texture of the profile views. This combination of methods ensures that the profile views closely resemble those found in real faces, showcasing the effectiveness of our approach.

Furthermore, the application of the depth map optimization technique further improves the quality of the forehead region in the 3D face. By optimizing the depth map specifically for the forehead area, we achieve a more realistic and natural appearance in this critical region.

The comparative analysis presented in Figure 6.3 and Figure 6.4 not only highlights the impact of each step but also demonstrates the cumulative effect of integrating these techniques in our methodology. This thorough evaluation serves as evidence of the advancements and contributions made in our research, ultimately leading to more realistic and visually appealing 3D face reconstructions, particularly in terms of profile views.

Figure 6.3: First row is the results without pretrained model. The second row is the results after trained with profile data-sets. The third row is the results with stable diffusion. The fourth row is the results with depth map correction.

Figure 6.4: Other examples of each step, the left is the results after training with profile datasets. The middle is the result with Stable diffusion. The right it the results with depth map correction.

6.4 Summary

In Chapter 6 of our work, we present the results obtained from our research, highlighting the superior performance of our method compared to other CNN-based approaches. Our evaluation encompassed various metrics and aspects to provide a comprehensive assessment of our model's capabilities.

One of the key metrics we considered is SIDE, which measures the similarity between the reconstructed 3D face and the ground truth. Our method consistently achieved higher SIDE scores, indicating a closer resemblance to the original face compared to other CNN methods. This demonstrates the effectiveness of our approach in accurately capturing the facial features and geometry during the 3D reconstruction process.

Furthermore, we evaluated the inference time of our model, which measures the speed at which our method can generate the 3D face. Our approach demonstrated competitive performance in terms of inference time, showcasing its efficiency and practicality for real-time applications.

In addition to quantitative metrics, we also assessed the quality of profile views in our results. Profile views present a significant challenge in 3D face reconstruction, requiring accurate modeling of facial features from a side perspective. Our method excelled in this aspect, achieving high-quality profile views with enhanced details and realistic representations.

Moreover, we focused on evaluating the overall realism of the reconstructed faces. Through subjective assessments and visual comparisons, our method consistently produced visually appealing and lifelike 3D faces. This indicates the ability of our approach to capture intricate details, such as skin texture, shading, and facial expressions, resulting in more believable and realistic representations.

By surpassing other CNN methods in terms of SIDE, inference time, profile view quality, and face realism, our results demonstrate the effectiveness and superiority of our proposed approach. These findings validate the advancements and contributions of our research in the field of 3D face reconstruction and highlight its potential for practical applications in computer vision, graphics, and related domains.

Chapter 7

Conclusion and Future Works

7.1 Conclusion

In conclusion, our work has made contributions to the field of 3D face reconstruction using advanced deep learning techniques. By combining HyperStyle GAN, Autoencoders, Stable diffusion, Neural Mesh Renderer, and other innovative methods, we have developed a powerful framework that surpasses existing CNN methods in terms of accuracy, efficiency, and visual quality.

In our methodology, we employed a sequential process to generate high-quality 3D models. We utilized HyperStyle GAN to generate various poses of the face, followed by using Dreambooth to extract the identity information. To capture the profile views, we employed OpenPose with stable diffusion. Next, we fed all the generated images into autoencoders to predict the albedo map, depth map, and other features. To enhance the albedo map, we combined different variations, and to improve the realism of the forehead, we applied a depth map optimization technique. This multi-step approach allowed us to generate realistic and detailed 3D models.

Through extensive experimentation and evaluation, we have demonstrated the effectiveness and superiority of our approach. The inclusion of profile view data during training has greatly enhanced the ability to reconstruct realistic 3D faces, particularly in capturing intricate details and accurate facial profiles. The use of HyperStyle GAN and InterfaceGAN for pose generation has further expanded the versatility of our method, allowing for the synthesis of diverse facial poses and expressions.

Additionally, the integration of the stable diffusion method has improved the overall

realism of the generated 3D faces, making them more consistent with real-world facial characteristics.

Our research has not only advanced the field of 3D face reconstruction but also showcased its practical applications. The ability to generate high-quality and dynamic 3D faces has opened up new opportunities in various domains, including entertainment, virtual avatars, and facial analysis. The potential for creating realistic digital representations of individuals and the ability to manipulate their facial attributes holds promise for numerous applications in computer graphics, animation, virtual reality, and more.

We can summarize our contribution as follows

- With our single front view image, our proposed pipeline enable multi view inputs to create 3D face model by using deep learning network.
- We utilize GAN inversion and GAN manipulation techniques in the domain of single-view 3D modelling.
- We proposed Diffusion methods with our HyperStyle GAN results and Openpose method to synthesize profile view images. This method has not been employed in prior work to generate profile view images. (The Diffusion model cannot generate profile views from a single input image.)
- We proposed to use those synthesized images (profile views and other views) to feed into autoencoders to train to predict 3D model features.
- We proposed to optimize our 3D models with albedo map combination and depth map correction techniques.

7.2 Future Works

While our work has achieved notable success, there are still areas for further exploration and improvement. Future research could focus on enhancing the efficiency and scalability of our method, exploring novel loss functions, or incorporating additional semantic attributes for more diverse facial variations.

Overall, our work has advanced the field of 3D face reconstruction by proposing a novel methodology and showcasing its effectiveness through extensive experimentation. The achievements and contributions made in this research pave the way for future advancements in the field, fostering new possibilities for realistic 3D face modeling and applications.

References

[1] Autoencoders in deep learning: An introduction for beginner — by abhishek mishra — medium. https://medium.com/@abhishekmishra13k/autoencoders-in-deep-learning-an-introduction-for-beginner-f0bf51220d6f. (Accessed on 07/14/2023).

[2] A brief introduction to unsupervised learning — by aidan wilson — towards data science. https://towardsdatascience.com/a-brief-introduction-to-unsupervised-learning-20db46445283. (Accessed on 07/14/2023).

[3] Morphace. https://faces.dmi.unibas.ch/bfm/index.php?nav=1-1-0&id=details. (Accessed on 07/14/2023).

[4] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019.

[5] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *Proceedings of the IEEE/CVF conference on computer Vision and pattern recognition*, pages 18511–18521, 2022.

[6] Horace B Barlow. Unsupervised learning. *Neural computation*, 1(3):295–311, 1989.

[7] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Inverting layers of a large generator. In *ICLR workshop*, volume 2, page 4, 2019.

[8] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4502–4511, 2019.

[9] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999.

[10] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[11] Lucy Chai, Jonas Wulff, and Phillip Isola. Using latent space regression to analyze and leverage compositionality in gans. *arXiv preprint arXiv:2103.10426*, 2021.

[12] Bernhard Egger, William AP Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, et al. 3d morphable face models—past, present, and future. *ACM Transactions on Graphics (TOG)*, 39(5):1–38, 2020.

[13] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1155–1164, 2019.

[14] Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and viewpoint without keypoints. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 88–104. Springer, 2020.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[16] Nicolas Gourier and James Crowley. Estimating face orientation from robust detection of salient facial structures. *FG Net Workshop on Visual Observation of Deictic Gestures*, 01 2004.

[17] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code gan prior. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3012–3021, 2020.

[18] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

[19] Mahmood Ul Haq, Muhammad Sethi, Rehmat Ullah, Aamir Shazhad, Laiq Hasan, and Mohammad Karami. Comsats face: A dataset of face images with pose variations, its design, and aspects. *Mathematical Problems in Engineering*, 2022:1–11, 05 2022.

[20] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *Advances in Neural Information Processing Systems*, 33:9841–9850, 2020.

[21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[22] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 172–189, 2018.

[23] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[24] Aaron S Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In *Proceedings of the IEEE international conference on computer vision*, pages 1031–1039, 2017.

[25] Luo Jiang, Juyong Zhang, Bailin Deng, Hao Li, and Ligang Liu. 3d face reconstruction with geometry details from a single image. *IEEE Transactions on Image Processing*, 27(10):4756–4770, oct 2018.

[26] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7122–7131, 2018.

[27] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018.

[28] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[29] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33:12104–12114, 2020.

[30] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021.

[31] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[32] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[33] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3907–3916, 2018.

[34] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. Manigan: Text-guided image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7880–7889, 2020.

[35] Gidi Littwin and Lior Wolf. Deep meta functionals for shape representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1824–1833, 2019.

[36] Yang Liu, Eunice Jun, Qisheng Li, and Jeffrey Heer. Latent space cartography: Visual analysis of vector space embeddings. In *Computer graphics forum*, volume 38, pages 67–78. Wiley Online Library, 2019.

[37] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[38] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Artificial Neural Networks*

and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21, pages 52–59. Springer, 2011.

[39] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 2437–2445, 2020.

[40] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[41] Berndt Müller, Joachim Reinhardt, and Michael T Strickland. *Neural networks: an introduction*. Springer Science & Business Media, 1995.

[42] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[43] Yuval Nirkin, Lior Wolf, and Tal Hassner. Hyperseg: Patch-wise hypernetwork for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4061–4070, 2021.

[44] Xingang Pan, Bo Dai, Ziwei Liu, Chen Change Loy, and Ping Luo. Do 2d gans know 3d shape? unsupervised 3d shape reconstruction from 2d image gans. *arXiv preprint arXiv:2011.00844*, 2020.

[45] Guim Perarnau, Joost Van De Weijer, Bogdan Raducanu, and Jose M Álvarez. Invertible conditional gans for image editing. *arXiv preprint arXiv:1611.06355*, 2016.

[46] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016.

[47] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[48] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021.

[49] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Transactions on Graphics (TOG)*, 42(1):1–13, 2022.

[50] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[51] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. 2022.

[52] C.D. Castillo V.M. Patel R. Chellappa D.W. Jacobs S. Sengupta, J.C. Cheng. Frontal to profile face verification in the wild. In *IEEE Conference on Applications of Computer Vision*, February 2016.

[53] Mihir Sahasrabudhe, Zhixin Shu, Edward Bartrum, Riza Alp Guler, Dimitris Samaras, and Iasonas Kokkinos. Lifting autoencoders: Unsupervised learning of a fully-disentangled 3d morphable model using deep non-rigid structure from motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[54] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael J Black. Learning to regress 3d face shape and expression from an image without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7763–7772, 2019.

[55] Jiaxiang Shang, Tianwei Shen, Shiwei Li, Lei Zhou, Mingmin Zhen, Tian Fang, and Long Quan. Self-supervised monocular 3d face reconstruction by occlusion-aware multi-view geometry consistency. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV*, pages 53–70. Springer, 2020.

[56] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE transactions on pattern analysis and machine intelligence*, 44(4):2004–2018, 2020.

[57] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1532–1540, 2021.

[58] Przemysław Spurek, Artur Kasymov, Marcin Mazur, Diana Janik, Slawomir K Tadeja, J Tabor, T Trzciński, et al. Hyperpocket: Generative point cloud completion. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6848–6853. IEEE, 2022.

[59] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021.

[60] Luan Tran and Xiaoming Liu. Nonlinear 3d face morphable model. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7346–7355, 2018.

[61] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3789–3797, 2017.

[62] Xiaoguang Tu, Jian Zhao, Mei Xie, Zihang Jiang, Akshaya Balamurugan, Yao Luo, Yang Zhao, Lingxiao He, Zheng Ma, and Jiashi Feng. 3d face reconstruction from a single image assisted by 2d face images in the wild. *IEEE Transactions on Multimedia*, 23:1160–1172, 2020.

[63] Shubham Tulsiani, Nilesh Kulkarni, and Abhinav Gupta. Implicit mesh reconstruction from unannotated image collections. *arXiv preprint arXiv:2007.08504*, 2020.

[64] Johannes Von Oswald, Christian Henning, Benjamin F Grewe, and João Sacramento. Continual learning with hypernetworks. *arXiv preprint arXiv:1906.00695*, 2019.

[65] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.

[66] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2020.

[67] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[68] Xiangyu Xu, Deqing Sun, Jinshan Pan, Yujin Zhang, Hanspeter Pfister, and Ming-Hsuan Yang. Learning to super-resolve blurry face and text images. In *Proceedings of the IEEE international conference on computer vision*, pages 251–260, 2017.

[69] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018.

[70] Chris Zhang, Mengye Ren, and Raquel Urtasun. Graph hypernetworks for neural architecture search. *arXiv preprint arXiv:1810.05749*, 2018.

[71] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017.

[72] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 592–608. Springer, 2020.

[73] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 597–613. Springer, 2016.

[74] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.