The Use of Conditional Generative Adversarial Networks in Face Generation Applications

Author: Luc van den Handel (VHNLUC001) **Supervisor:** A/Prof. Deshen Moodley

1. Abstract

Generative Adversarial Networks (GANs) are a kind of neural network which trains by playing an adversarial game between two networks; a generator and a discriminator. Should training succeed the resulting generator will be able to generate output samples with a high degree of accuracy when compared to the real-world samples which were used for training. Conditional GANs have shown their ability to produce images of faces, along with the GANs ability to produce 3D models, they display the possibility for applications in 3D facial model generation. While 3D Morphable Models have displayed their ability to fulfill this function they have shortcomings that may be addressable by conditional GANs.

2. Introduction

Generative Adversarial Networks (GANs) [9] and their variants have been used for many applications involving the synthesis of digital assets, such as the generation of faces[17] and 3D models of every-day objects [8]. In particular the application of 3D face generation has room for additional investigation. The search for a generator that can accurately generate novel 3D models of human faces is driven by its numerous applications in industries such as Computer Games, Film, Cybersecurity, Virtual Reality and Augmented Reality.

There are three broad categories of GANs: progressive [18], controllable [25] and conditional [14] GANs. While progressive and controllable GANs serve their own purposes the domain of conditional GANs for 3D face generation has been largely overlooked. This might be attributed to the success of 3D morphable models (3DMMs) for this exact application. 3DMMs take in parameter values and produce a corresponding facial model, this is very similar to what a conditional GAN might be able to do.

The need for conditions in face generation is pertinent as is speaks to the usability of a program. If an algorithm is to provide users with a theoretically endless supply of novel human faces, the user needs some control over the output. Control such as the ability to specify foundational details such as age, gender and ethnicity, along with situational details such as pose and expression.

To this end, I will explore GANs, their variants and their applications in face generation and model generation.

3. Generative Adversarial Networks

3.1 GAN basics

GANs belong to the neural network class of machine learning algorithms. They are unsupervised learners, which means that they are trained using unlabeled sample data. In general, GANs contain 2 networks, a generator and a discriminator, which work against each other during training [9] – the generator creates output samples that are attempts at fooling the discriminator that they are "real" (where an actual real sample is a member of the training dataset).

Both networks are used during training. The generator will create a series of samples which will be labeled as fake and combined with the input samples which will be labeled as real. The discriminator trains on this dataset as input. After training the fake samples are input and the loss of the discriminator is fed to the generator for it to learn. This process forms a single training epoch. The training can then be run for any specified number of epochs. However, the formal definition of the point when training is complete is when the system reaches equilibrium: the probability that the discriminator output is correct is 50% (i.e. the discriminator can do no more than randomly guess whether or not the input is fake or real, therefore the generator is producing output that is a perfect replication of the real world) [9].

Statistically, the output of the generator follows a certain probability distribution, which in [12] is referred to as $\mathbb{P}_g$. Similarly, the real world data also follows its own probability distribution, which in [12] is referred to as $\mathbb{P}_r$. The goal of training is to have the two distributions align – i.e. shift $\mathbb{P}_g$ such that $\mathbb{P}_g \approx \mathbb{P}_r$. Therefore a key metric is the distance between the two distributions. The loss function is thus derived in terms of this distance, which is commonly defined by the Jensen-Shannon divergence, and the generator aims to minimize the resultant loss value [9, 12].

In the end the training is defined by the following equation:

$$\boxed{\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[1 - \log D(G(z))]}$$

Where $D(x)$ is the output of the discriminator, $G(z)$ is the output of the generator and the generator is training to minimize the values produced by the equation but the discriminator is training to maximize the value.

Once training has completed the generator can then act on its own as an algorithm or program for producing target samples (e.g. a theoretically endless amount of novel images of faces [17]).

The discriminator is typically a classification algorithm with two possible output values: fake or real. Classification algorithms such as this are well known and understood. However, the complexity of the generator is variable, with the most simple, original, generator being made using a Multi-Layer Perceptron with noise (randomly generated seed values) as the only input [9].

Subsequently GANs have been expanded to make use of more complex and dynamic network structures for either the generator, discriminator or both. The generator input has also been expanded to allow for finer control.

3.2 Expanding the GAN

Since the inception of the GAN issues with the architecture have been established and addressed. These improvements address both the structural and theoretical issues with GANs such as poor stability, low output resolution and mode collapse [18, 13] along with usability where some degree of control is needed for the models to be used effectively [14, 15, 17, 22].

The first notable improvement is the incorporation of deep convolutional networks into the GAN architecture. The use of convolutional layers is particularly useful when dealing with higher dimensional outputs such as images. They allow both the generator and the discriminator to traverse the output and input spaces respectively. Thus they can gain a more detailed understanding of the output/input and produce more accurate results [1]. Convolution has been critical in the success of 2D face generation, and thus may have applications in 3D face generation. For example, should voxels be the chosen method for model representation, convolutional layers can be simply expanded with an extra dimension [8].

Furthermore, with the goal of improving stability and eliminating with issues such as mode collapse, the Wasserstein GAN (WGAN) was created [13]. WGAN does this by using a different loss function compared to the original GAN. This new loss function was formulated using the Wasserstein distance to measure the distance between the probability distributions of the generator and the real input, where in the past the Jensen-Shannon divergence had be used. Another key difference is that the discriminator output is no longer bounded by 0 and 1 via the use of a Sigmoid activation function, but rather discriminator uses weight clipping where during each training iteration every weight in the network is bounded between some value and its negative (for example with a clipping value of c=0.01, all of the weights will lie in the range: -0.01<=w<=0.01).

The WGAN was then further improved by using a gradient penalty instead of weight clipping for the discriminator, this helps guarantee convergence and improves the quality of the output [10].

Lastly, GANs have been expanded to allow for the inclusion of additional input. For example; a class of GANs which achieves this are Conditional GANs [14]. Conditional GANs achieve this by fusing an additional conditional information vector to the noise input of the generator and to one the layers of the discriminator. In [7] the vector was added to the first layer of the discriminator, but it has been added at other points [1, 8].

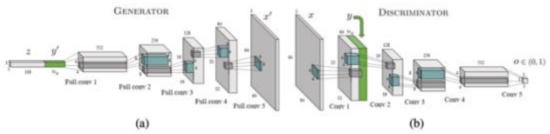

Figure 1: showing the generator and discriminator from a conditional GAN [7]. Here the conditional space y is joined to the input noise z and the first convolutional layer of the discriminator.

The inclusion of the conditional information vector (y) can also be seen in the conditional GAN learning equation [14]:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_{z}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))].$$

4. GANS and Face Generation

4.1 2D Face Generation

The generation of images using GANs is an area that has been avidly explored as it was one of the first major applications of GANs [9]. Subsequently the technique has been applied to images displaying faces. A particularly applicable subject matter considering that while each face is unique, they all follow a similar structure and are therefore good candidates for GAN applications. Algorithms have been developed to produce generators that can produce images of novel human faces with a high degree of accuracy [17].

This accuracy has been achieved through a number of improvements on the GAN architecture. As previously mentioned, the use of convolutional layers has been particularly critical. This is because convolution is not only extremely good at dealing with the 2-dimensional data of images but can also be expanded to include channels (essentially extra dimensions) which can be used to provide additional image data such as color [1].

GANs such as this use convolutional layers in two different ways. The generator can make use of transpose convolutional or deconvolutional layers which are the reverse of the typical convolutional layer. Instead of traversing the input space and extracting critical information, the layer traverses the output space populating it based on data provided by the previous network layers. The discriminator can then use a more typical application of convolution where the input image is traversed as the first layer of the network that passes critical information to the rest of the network[1].

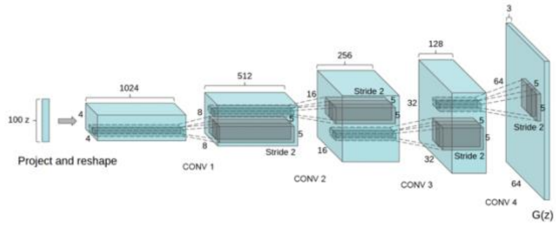

Figure 2: showing the use of transpose convolutional layers for the generator of a GAN using channels – see output layer which has 3 channels for RGB color [1].

Another key improvement that has been shown in the area of 2D face generation is the incorporation of high-level attributes (general details such as age or ethnicity) and stochastic variation (more specific details such as hair color) in generated images [17], essentially using an extended variant of conditional GANs. The GAN in [17] achieved this by mapping the input parameters to their own latent space which controls the generator. Values from this latent space are then input into many layers of the generators' "synthesis network" along with noise to produce outputs which were both novel and yet still conformed to the desired parameters.

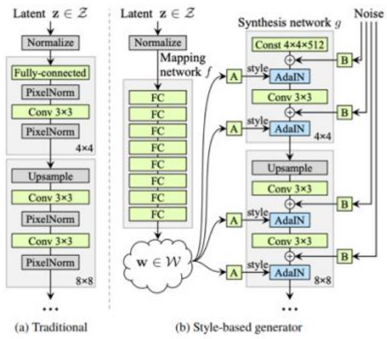

Figure 3: an excerpt from [17] showing a comparison between a traditional generator and the proposed, style-based generator architecture. Here the parameters are mapped an intermediate latent space which provides style values which are input at each layer of the synthesis network.

However, while the progress made and results produced are impressive, the number of useful applications of 2D face generation – or industries where 2D face generation can be applied – is limited. Primarily it is useful for applications where an image of a person is needed but one wants to avoid any privacy breaches.

4.2 3D Models

Before the topic of 3D faces can be addressed we should consider 3D models, the different kinds of models, their advantages and disadvantages and how they are stored. This knowledge will be critical in forming GANs that both read in and output 3D models. Then an investigation into how GANs have delt with this issue in the past can be performed.

There are a number of ways in which a 3D model can be stored and represented. Understanding some of the more popular methods will be critical when the time comes to decide on which dataset(s) will be used to train and test the model.

There are 3 common methods for constructing 3D objects in computer graphics: polygon meshes, voxels and point clouds [11].

Polygon meshes define models that exist in a 3 dimensional space and are formed by a collection of polygons (typically triangles) which are defined in that space. For example a triangle will consist of three points – each with their own x, y and z coordinates – and additional orientation information such as a normal vector encoding which side of the triangle is the "front." The triangle mesh is typically constructed such that they have shared edges and points. The final result from rendering all tringles will be the model.

Voxels are the closest approximation pixels in the 3D space. They are volumetric pixels, where each voxel is a cube in an evenly divided 3D space, in the same way a pixel is a square on a 2D space, which can be stored as a 3D Boolean array.

Lastly, point clouds are simply a collection of points in 3D space. Both point clouds and polygon meshes are ways of defining surfaces, while point clouds can me much simpler they can encounter issues where rendering is incorrect for complex surfaces.

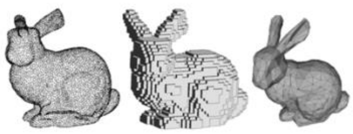

Figure 4: showing the 3 kinds of models all representing the same object, point cloud (left), voxel (middle) and polygon mesh (right) [11].

While voxel models display the potential for being more easily generated by GANs [8] the resolution required for enough detail in a face will require an extremely large network and questions training feasibility. Furthermore a large portion of voxel data is effectively meaningless as the internal and external volumes are contained in the data where only the surface of the model is what matters.

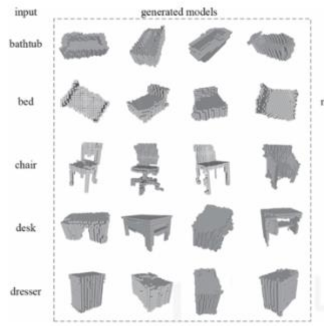

Figure 5: Showing 3D models generated by a conditional GAN [8]. Here we can see that while the objects do resemble their real world counterparts, their low voxel resolution has severely impacted the quality of the model.

4.3 3D Face Generation

Recently, GANs have been used for the generation of 3D face models. A variety of GAN architectures have been used to achieve this output.

Significant success has been found in making use of a 3D morphable model (3DMM) architecture [6, 21]. A morphable model is a 3D model that is formed to be a standard middle-ground between all possible 3D faces. The model is embedded with parameters which can be adjusted to morph the model into the desired face [21]. They essentially map low-dimensional data (the parameters) to a high-dimensional outputs (the 3D faces). 3DMMs have been combined with GANs for the generation of novel facial models [16] but are limited by the 3DMM parameters. Since these parameters are so low-dimensional, the models output display limited novelty and runs the risk that important information is lost. However, 3DMMs have displayed great potential for applications involving expression synthesis and pose variation as in these cases the synthesis of the model itself is not the issue but rather performing conditional adjustments to an input.

More traditional GAN architectures have been used for this application, such as the GAN in

[20] which makes use of an MLP generator to create 3D faces. This GAN is still somewhat complex in that the discriminator is a convolutional neural network that takes in a UV map of the 3D model to determine if it is fake or real. This network also makes use of simple conditional techniques in that along with noise it also takes in 2 conditional values; identity and expression. This was done with the intention of decoupling these classifications from the data. Therefore while they extended the usefulness of the resultant generator, these conditional factors were not included with a particular application of the generator in mind.

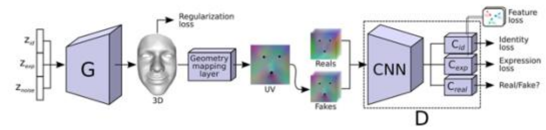

Figure 6: diagram showing the GAN used in [20]. Here z_{id} represents the identity condition and z_{exp} represents the expression condition.

5. Conditional GANs for 3D Face Generation

As it currently stands, 3D face generation is still very much at the cutting edge of what is possible for the output of a GAN. Conditional GANs have been used to produce 3D models that conform to a specific set of conditions. The majority of these GANs make use of the 3DMM, this is due to the focus on the conditional nature of the problem and less of the problem of generating 3D objects.

5.1 3DMMs for Face Frontalization

In [22] Yin et al. proposed a GAN that makes use of a 3DMM to produce frontalized images of a person based on an image displaying side-on pose of said person. This FF-GAN makes use of a generator to produce the 3DMM parameters which is conditioned by the input image. In this case the input image is the conditional information vector. It encodes all the necessary facial information such as age, gender and expression. To further ensure that this identity information has been correctly conveyed to the 3DMM, FF-GAN makes use of a recognition engine to preserve identity. This way FF-GAN mitigates effects the information loss due to the low-dimensionality of the 3DMM. Affine transformations are then applied to the 3DMM to rotate the model and produce the frontal image. Since the output of FF-GAN is an image the discriminator can remain fairly simple, as a 2D CNN to classify if an image is generated or real.

The 3DMM was used to reduce training complexity and to increase performance with limited data. Thus key to the success of FF-GAN was the technique for converting an image into 3DMM parameters. This function was performed by their Recognition Module *R* such that: **p** = *R*(x) where x is an input image and **p** is the vector containing the 3DMM parameters. Essentially x can be seen as highly configured noise with respect to GAN training [9], where in the past the noise and conditional vector were separate, now they are highly coupled. *R* was constructed using a CNN because, as previously mentioned, CNNs deal extremely well with image data. *R* is therefore trained using the following loss equation:

$$\boxed{\min_{\mathbf{p}} L_R = (\mathbf{p} - \mathbf{p}^g)^\top \mathbf{W}(\mathbf{p} - \mathbf{p}^g),}$$

Where p^g are the generated parameters, p are the example parameters and **W** is the importance matrix such that the diagonals of **W** are the weight of each parameter.

FF-GAN is able to produce somewhat accurate results from a qualitative perspective. The expressly facial features – eyes, nose, mouth, etc. – were accurately transformed to the frontal view with the surrounding features lacking significant adjustment such as hair, ear(s) and face shape which remained in their side-on pose. This suggests that FF-GAN was able to successfully configure the 3DMM as this is what enables the kind of success seen in the results.

5.2 3D-GAN

Similarly, Marriott et al. proposed their 3D-GAN [15] which was aimed at producing a series of facial images of a subject from multiple angles for use in facial recognition software. While 3D-GAN outputs 2D images, it also uses an internal 3DMM to produce those images. However, unlike FF-GAN, 3D-GAN produces images of random, novel, faces. It does this by using a two generators to create textures for the face and the background, the face texture is then mapped onto a 3DMM which is configured by shape, pose and expression information. The pose and expression information is conditioned by conditional information vectors which are passed along with noise to the 3DMM. To ensure the texture maps perfectly the texture generator is provided with the same shape information as the 3DMM. While the faces are completely random, these parameters are still open to receive controlled values to condition face.

This design has demonstrated higher quality results compared to FF-GAN. The images are relatively free from artifacts and have a high level of detail. This displays the possibilities enabled by the 3DMM for novel face generation. In this case the FLAME (Faces Learned with an Articulated Model and Expressions) 3DMM was used [19].

5.3 The FLAME 3DMM

The FLAME 3DMM is a highly detailed morphable model that was created using 33000 facial scans [19]. While it is not a GAN it still can be used to achieve the same goals as a conditional GAN that produces 3D facial models. In this case FLAME takes in 3 parameters to configure the model; shape, pose and expression. In order to map these parameters to the output models the algorithm was trained using Principle Component Analysis (PCA). The model can be expressed by the following equation:

$$\boxed{M(\vec{\beta},\vec{\theta},\vec{\psi}) = W(T_P(\vec{\beta},\vec{\theta},\vec{\psi}),\mathbf{J}(\vec{\beta}),\vec{\theta},\mathcal{W}),}$$

Where $\vec{\beta}$, $\vec{\theta}$ and $\overrightarrow{\Psi}$ are, effectively, conditional information vectors which define the modifications to based face model. The other functions which define this equation are internal functions such as **J** which manipulates the joint points of the model. The effects of these parameters are seen in Figure below.

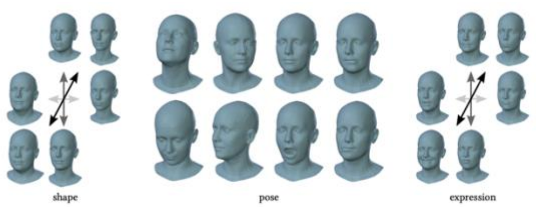

Figure 7: showing the different face models that are output by the FLAME 3DMM when one of its three parameters (shape, pose and expression) are manipulated [19].

The FLAME 3DMM is available as an open source tool that can be added as an extension to Blender – a popular 3D modeling application. It performs well and enables users to create novel models on the fly as needed. It is also well suited to be integrated with gaming and graphics pipelines. However it is entirely statistical and does not make use of any kind of GAN structure. This demonstrates that 3DMMs can be used to achieve the goal of novel 3D face generation, but the question of whether or not they are the best or only solution still remains.

5.4 Generic 3D Model Generation

Moving away from 3DMMs, generating 3D faces can be seen as an extension to the generation of 3D models. As such it will be worth exploring the 3D Improved Wasserstein Generative Adversarial Network (3D-IWGAN), proposed by Smith et al [5].

The output of the generator is a voxel grid which is produced using a series of 3D deconvolutional layers and subsequently the discriminator makes use of 3D convolutional layers to produce the single value output typical of the GAN discriminator [9]. The input to the generator is a 200 dimensional latent vector. As can be seen by its name, 3D-IWGAN makes use of the improved WGAN design as seen in [10]. This architecture was able to produce adequate results considering the use of voxels and the low output resolution chosen as the output models do closely resemble their real world counterparts, with however a low degree of detail.

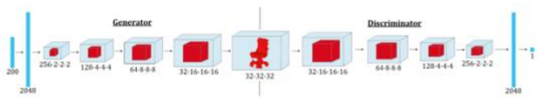

Figure 8: showing the network architecture of 3D-IWGAN [5].

3D-IWGAN has been expanded upon by Öngün et al. to produce a conditional GAN to produce 3D models based on a trained label [4]. It can contain knowledge of a number of kinds of objects (chairs, beds, sofas, etc.) as long as the training data is correctly labeled. It stands to reason that it can be trained using highly detailed labels to describe different kinds of faces – detailing features such as age, gender, ethnicity, etc.

In [4] the use of conditional GANs for model generation was driven by the design decision to have two generators. These two generators would take in the same noise, but alternate conditions, in this case viewing angle, and the results would be merged into the final output to ensure that the output maintained correctness under both conditions. While this does improve output quality and stability, the improvements degrade for symmetric objects – which the human face typically is. Furthermore, the output is susceptible to the same shortcomings as 3D-IWGAN, where the use of low resolution voxel objects hinders the detail of the output. However, the output does correctly conform to the training labels, demonstrating the applicability of conditional GANs in this problem domain.

A further improvement to this approach was made by Li et al. and their conditional GAN for 3D model generation [8] which added a second network to the discriminator end of the GAN called the classifier. This network worked as a secondary discriminator by generating a classification label for the generated model with the goal of this label matching generator input. Li et al. was also able to scale up the resolution of the output voxels to 64x64x64 which displayed and increase in the quality and detail of the output

Figure 9: showing the 64^3 output of the GAN in [8]

6. Datasets

In order to develop a GAN to produce models of 3D faces, one would require a dataset containing example models. Furthermore said dataset would need to include labels to provide input for a conditional information vector, such age, gender, and most importantly, expression. Datasets that could fulfill this purpose include BU-3DFE [24] or the OUY 3D face database from the University of York.

7. GAN Issues

When it comes to implementing and training GANs there are a number of issues which can be encountered. Issues can be mitigated with the network design such as the way that WGANs mitigate mode collapse as previously mentioned.

7.1 Vanishing Gradients

In [12] it was identified that a difference in performance between the discriminator and the generator, specifically when the discriminator is too good, might cause generator training to fail. This is because the disparity has caused vanishing gradients.

Vanishing gradients occur when the gradients that form the weight values between the neurons become extremely small. This causes the updates from each learning iteration to have little to no influence on the weight value. However, this influence is critical to learning as it is what improves the network. Therefore it causes the learning cycle to fail.

7.2 Mode Collapse

As was seen in [13], mode collapse is an issue which can cause undesirable performance. When it occurs the generator appears to function normally, producing acceptable output, however it is only able to produce one correct output. For example, a generator trained to produce handwritten digits may be able to produce acceptable results, but the digit is always an 8.

7.3 Failure to Converge

As mentioned in [9] there is a theoretical point where discriminator and generator reach an equilibrium. However, in practice this point is only reached in approximation, this is why GANs don't bother checking for equilibrium and they rather train for a set number of epochs.

An issue is encountered when the equilibrium point is never reached, not even in approximation. In this situation, training is effectively pointless, and no improvement is seen over additional training iterations. This is also an issue which WGAN aims to address [12]

8. Evaluating GANs

In order to determine if a GAN is functioning well, correctly or better than the alternative designs, we need a way to evaluate GANs, this can either be done qualitatively by people, or quantitively using numerical techniques [2]

8.1 Manual Evaluation

As the easiest form of evaluation, manual evaluation involves using the generator to produce a batch of output samples which are then evaluated by a single evaluator. The evaluator will assess quality and diversity of the output with respect to the target domain. However, this method is very unreliable as human vision is extremely biased for a single person.

8.2 Qualitative Evaluation

Qualitative evaluation is another human based method. Evaluations are either subjective or comparative. The most popular technique is called "Rating and Preference Judgement," which involves providing a number of human judges with a batch of real and generated output samples and asking them to rank the samples.

8.3 Quantitative Evaluation

In order to completely counteract biases, quantitative evaluation is used. In this case we use numerical metrics to assess the performance of a GAN. In [9] the Average Log-likelihood method was used. This particular technique involves estimating how well the generator probability distribution matched that of the real data.

In addition of Average Log-likelihood, there are two other popular approaches; Inception Score which uses a pretrained network to classify the quality of generated samples and Fréchet Inception Distance (FID), which embeds the generated samples in a feature space which is given by a layer of the network used for inception scores.

The FID score is described by the following equation:

$$\boxed{FID(r,g) = ||\mu_r - \mu_g||_2^2 + Tr\left(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}}\right)}$$

Where r are the real samples and g are the generated samples.

9. Conclusion

GANs have displayed an ability to produce novel output samples with a high degree of quality . These output samples include: images of target objects and faces[9,17], 3D models of objects [8] and even the application of a transformation to an input image [22]. Furthermore, conditional GANs add the ability for users to administer control over the output allowing for increased usability of the resulting generator [14]. Furthermore, in addition to the ability of GANs to produce 3D models, there is a high level of feasibility for a controllable GAN to produce 3D facial models. Such a task is already performed well by 3DMMs [19], however they are entirely statistical and may lack flexibility in their ability to produce extremely unique faces. Such shortcomings may be overcome by a GAN and is therefore worth investigating.

10. References

[1] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv. DOI:https://doi.org/10.48550/ARXIV.1511.06434

[2] Ali Borji. 2018. Pros and Cons of GAN Evaluation Measures. arXiv. DOI:https://doi.org/10.48550/ARXIV.1802.03446

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large Scale GAN Training for High Fidelity Natural Image Synthesis. arXiv. DOI:https://doi.org/10.48550/ARXIV.1809.11096

[4] Cihan Öngün and Alptekin Temizel. 2019. Paired 3D Model Generation with Conditional Generative Adversarial Networks. In Computer Vision – ECCV 2018 Workshops, Springer International Publishing, Cham, 473–487.

[5] Edward Smith and David Meger. 2017. Improved Adversarial Systems for 3D Object Generation and Reconstruction. arXiv. DOI:https://doi.org/10.48550/ARXIV.1707.09557

[6] Gil Shamai, Ron Slossberg, and Ron Kimmel. 2019. Synthesizing facial photometries and corresponding geometries using generative adversarial networks. arXiv. DOI:https://doi.org/10.48550/ARXIV.1901.06551

[7] Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, and Jose M. Álvarez. 2016. Invertible Conditional GANs for image editing. arXiv. DOI:https://doi.org/10.48550/ARXIV.1611.06355

[8] Haisheng Li, Yanping Zheng, Xiaoqun Wu, and Qiang Cai. 2019. 3D Model Generation and Reconstruction Using Conditional Generative Adversarial Network. International Journal of Computational Intelligence Systems 12, (January 2019), 697. DOI:https://doi.org/10.2991/ijcis.d.190617.001

[9] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. arXiv. DOI:https://doi.org/10.48550/ARXIV.1406.2661

[10] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. Improved Training of Wasserstein GANs. arXiv. DOI:https://doi.org/10.48550/ARXIV.1704.00028

[11] Long Hoang, Suk-Hwan Lee, Oh-Heum Kwon, and Ki-Ryong Kwon. 2019. A Deep Learning Method for 3D Object Classification Using the Wave Kernel Signature and A Center Point of the 3D-Triangle Mesh. Electronics 8, 10 (2019). DOI:https://doi.org/10.3390/electronics8101196

[12] Martin Arjovsky and Léon Bottou. 2017. Towards Principled Methods for Training Generative Adversarial

Networks. arXiv.
DOI:https://doi.org/10.48550/ARXIV.1701.04862

[13] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein GAN. arXiv.
DOI:https://doi.org/10.48550/ARXIV.1701.07875

[14] Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. arXiv.
DOI:https://doi.org/10.48550/ARXIV.1411.1784

[15] Richard T. Marriott, Sami Romdhani, and Liming Chen. 2021. A 3D GAN for Improved Large-Pose Facial Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 13445–13455.

[16] Ron Slossberg, Gil Shamai, and Ron Kimmel. 2018. High Quality Facial Surface and Texture Synthesis via Generative Adversarial Networks. arXiv.
DOI:https://doi.org/10.48550/ARXIV.1808.08281

[17] Tero Karras, Samuli Laine, and Timo Aila. 2018. A Style-Based Generator Architecture for Generative Adversarial Networks. arXiv.
DOI:https://doi.org/10.48550/ARXIV.1812.04948

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive Growing of GANs for Improved Quality, Stability, and Variation. arXiv.
DOI:https://doi.org/10.48550/ARXIV.1710.10196

[19] Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. 2017. Learning a Model of Facial Shape and Expression from 4D Scans. ACM Trans. Graph. 36, 6 (November 2017).
DOI:https://doi.org/10.1145/3130800.3130813

[20] Victoria Fernandez Abrevaya, Adnane Boukhayma, Stefanie Wuhrer, and Edmond Boyer. 2019. A Decoupled 3D Facial Shape Model by Adversarial Training. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).

[21] Volker Blanz and Thomas Vetter. 1999. A Morphable Model for the Synthesis of 3D Faces. In Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99), ACM Press/Addison-Wesley Publishing Co., USA, 187–194.
DOI:https://doi.org/10.1145/311535.311556

[22] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. 2017. Towards Large-Pose Face Frontalization in the Wild. arXiv.
DOI:https://doi.org/10.48550/ARXIV.1704.06244

[23] Yanfei Liu and Junhua Chen. 2021. Unsupervised face Frontalization for pose-invariant face recognition. Image and Vision Computing 106, (2021), 104093.

[24] "A 3D Facial Expression Database For Facial Behavior Research" by Lijun Yin; Xiaozhou Wei; Yi Sun; Jun Wang; Matthew J. Rosato, 7th International Conference on Automatic Face and Gesture Recognition, 10-12 April 2006 P:211 – 216

[25] Minhyeok Lee and Junhee Seok. 2019. Controllable Generative Adversarial Network. IEEE Access 7, (2019), 28158–28169.
DOI:https://doi.org/10.1109/ACCESS.2019.2899108