



Real-Time Deepfake Image Generation Based on Stylegan2-ADA

Doaa A. Talib^{*}, Ali A. Abed^{}

Department of Computer Engineering, College of Engineering, University of Basrah, Basrah 61004, Iraq

Corresponding Author Email: duaaalaaa@gmail.com

<https://doi.org/10.18280/ria.370216>

Received: 10 February 2023

Accepted: 28 March 2023

Keywords:

Deepfake, Generative Adversarial Networks (GAN), stylegan2-ADA, stylegan, FFHQ

ABSTRACT

Training Generative Adversarial Networks (GAN) usually leads to hyper-specialization due to few data and this causes training to diverge. This paper proposes a method that significantly stabilizes training without making changes. The method which will be used is stylegan2-ADA method to get fake images, the images will be entered in several steps, where the first step is using 76,400 Flickr-Faces-HQ (FFHQ) images and training them to get fake images. The program will be dividing images inside seven test folders, as the performance rate of 1000 images is 83.3%, which is a very good percentage when compared with the stylegan2 method because our proposed method contains augmentation that generates many images through the use of few images. The second step is represented by using personal images, we used two personal images and made a projection between them. The result of the performance of generating 200 images is 99.9%. Additionally, will be took a direct photo using the computer camera in real-time mode, and i generated 300 images. The generation performance is 99.9%, and our approach outperformed earlier ones in terms of accuracy, the ability to produce images without noise, and ability to create fake images of people who are not actually there.

1. INTRODUCTION

In recent years, the research community has watched and studied the so-called. Deepfake problem with growing concerns and interests. There is broad consensus regarding the nature and disruptive potential of this technology: they synthesis incredibly realistic multimedia content based on a specific application of Generative Adversarial Networks (GAN) [1]. The interesting issue of Generative Adversarial Networks (GAN) applies to both semi-supervised and unsupervised learning. They accomplish this because high-dimensional data distributions are intrinsically modeled. A pair of neural networks striving to outperform one another is how the framework for GANs, which was first suggested in 2014, characterizes the technology. Consider one network as a plagiarist and the other as an authority on the subject when analyzing visual data. The imitator, known as generator G in the literature, creates copies in an effort to create genuine images [2]. The expert, discriminator D, gathers both original and fake samples and attempts to determine which ones are genuine (as presented in Figure 1). Both are simultaneously trained for and engaged in competition [2]. The discriminator is exposed to both samples taken from the read world dataset and synthetic images. Using the ground truth image dataset, which includes both fake and real images, the discriminator's error is calculated. After each epoch, higher-quality images are produced as a result of the generated mistake being back-propagated throughout the network.

If the generator network is thought of as a functional mapping from a seemingly random, high-dimensional space known as a latent space to the space of actual picture data, then mathematically it is denoted by G: $G(z)$.

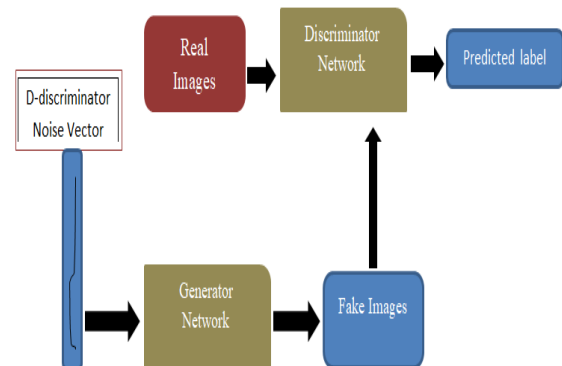


Figure 1. General architecture of Generative Adversarial Networks

Generative Adversarial Networks (GANs) have become a very popular image generating paradigm. As an illustration, StyleGAN is presently the preferred technique for producing images that are nearly image realistic for numerous classes (e.g., human faces, cars, and landscapes). However, producing extremely high-quality results becomes more difficult for classes that show intricate variations. Due to the significant degree of variation in human attitude, shape, and appearance, full-body human generation, for instance, is still an unsolved problem [3]. The limitless number of images accessible online has helped Generative Adversarial Networks (GAN) produce better impressive results. It is still difficult to gather a sizable enough collection of images for a particular application that has requirements for topic matter, image quality, location, time, privacy, copyright status, etc. [4]. The majority of research on the fairness and bias of GANs seeks to either discover and

explain the biases or to remove the detrimental impact of using unbalanced data on generation outputs. The three primary areas of bias and fairness research are: discovering and explaining biases; debiasing pre-trained GANs; and improving the training and generation performance of GANs using biased datasets [5].

This paper aims to generate a large number of images of fake individual that will be used as dataset in researches and train images using two methods, stylegan2 and stylegan2-ADA, and compare between them. Generative Adversarial Networks (GANs) have considerably improved face image synthesis in recent years. Modern image creation techniques now produce images with astounding realism thanks to their high levels of visual quality and accuracy. Most significantly, StyleGAN [6, 7] achieves state-of-the-art visual quality on high-resolution images and presents a revolutionary style-based generator architecture. Furthermore, it has been shown to possess a disentangled latent space that allows for control and editing.

Images, music, and text are just a few of the many sorts of data that GANs may produce. Popular GAN examples from the real world include the following:

Creating faces of people. GANs are capable of creating precise images of human faces. For instance, Nvidia's StyleGAN2-ADA can create great, lifelike photos of fictional persons. Many people mistake these images for real persons because they are so convincing [8].

Creating innovative clothing designs. GANs may be used to produce new clothing designs that mimic older styles. For instance, the clothing chain H&M employed GANs to develop fresh clothing designs for its products [9].

Creating animal images that are realistic. GANs are also capable of producing lifelike animal pictures. For instance, BigGAN, a GAN model created by Google researchers, can generate excellent pictures of animals like dogs and birds [10].

Creating characters for video games. New characters for video games may be made using GANs. For the popular video game Final Fantasy XV, Nvidia used GANs to design new characters [11].

Previous research lacks the use of a high-specification Computer, and this makes it difficult to run the code and train the model. In our research, we will use a high-specification Computer to get the best accuracy, as a computer with a high screen card will be used. We will generate more images and high accurate, and the images will also contain labels, gender and age. When Generative Adversarial Networks (GAN) are trained with insufficient data, discriminator overfitting frequently results, causing training to diverge. Therefore, we will use an adaptive discriminator augmentation mechanism that significantly stabilizes training in limited data regimes.

2. RELATED WORK

This section discusses Previous studies for Deepfake that make use of Generative Adversarial Networks (GAN). Research by Tariq et al. [12] presented neural network-based techniques. This strategy uses pr-processing to analyse the

statistical aspects of the image and improves the recognition of human-made artificial faces for the purpose of identifying fraudulent images produced by GANs. Research by Do et al. [13] also offer another method based on a deep convolution neural network. In order to extract face characteristics from face recognition networks, the model first uses a deep learning network. The facial traits are then fine-tuned to make them appropriate for real/fake image recognition. Research by Xuan et al. [14] present a forensics convolution neural network (CNN) that employs Gaussian Blur and Gaussian Noise as two picture pr-processing phases to identify fraudulent human images. The purpose of this model is to enhance high frequency pixel noise in low level pixel statistics while ignoring low level high frequency hints artifact in GAN pictures. This makes it possible for the forensic classifier to pick up more telling differences between genuine and phony photos, improving its ability to discern between their faces. Research by McCloskey et al. [15] presented the Intensity Noise Histograms network to categorize the histograms created using the R and G chromaticity coordinates as two variables. They used the features discovered by measuring the quantity of saturated and underexposed pixels to train a Support Vector Machine (SVM). Research by Thies et al. [16] presented the facial reenactment method maintains the target individual's identity while transferring the expressions of one person in a source video to another in a target video. With this technique, faces are condensed into a small expression space that makes it simple to translate expressions from the source to the destination. Research by Rössler et al. [17] created the FaceForensics++ dataset of modified movies. Conducted tests using the FaceForensics++ dataset, which contains films with solely face forgery, even suggested attacks are generally applicable to CNN [18] detectors for any sort of forgery in images and videos. Research by Hsu et al. [19] introduced a deep forgery discriminator that concatenates two classifiers with a contrastive loss For the purpose of identifying GANs-based false pictures. Research by Li and Lyu [20] proposed A deep learning approach that leverages affine face wrapping artifacts for Deepfake detection. The suggested method replicated these pictures using image processing procedures to produce artifacts that are present in Deepfake material rather than utilizing Deepfake images as negative examples. Research by Karras et al. [21] demonstrated how GANs create new faces by gradually expanding the generator and discriminator. It is challenging for humans to tell whether the facial images produced by this approach are of actual individuals since they are so lifelike. Although this technology has many useful uses, GANs may be maliciously used to hurt individuals by creating false human faces. Additionally, the development of these convincing fake faces can confuse facial recognition systems, and attackers can produce numerous examples of these fake images to trick individuals and perhaps even cause societal issues. For instance, they can use the image of an imaginary person to fabricate fraudulent social identities. Table 1 summarizes the findings of these experiments in terms of Deepfake.

Table 1. Summary of the previous works for Deepfake

Ref	Dataset	Method	Best Performance
[22]	Own (StarGAN, Glow, ProGAN, StyleGAN)	Deep Learning Features (CNN,AE)	99.8%
[23]	100K-Faces (StyleGAN),	Deep Learning Features (CNN)	EER=0.3%

	IFakeFaceDB		EER=0.4.5%
[24]	Handcrafted Facial Manipulation (HFM)	FakeFace Net (SFFN)	72.52%
[25]	FaceForescs++	(DNNs)	N/P

3. METHODOLOGY

This section outlines the procedures used in this paper, including the details of generating fake images by using real images with saved these images with labels, age, and gender.

3.1 GAN

Numerous factors, including the distribution of training datasets, network architecture, loss design, optimization method, and hyper parameter settings, define each GAN model. The values of model weights are sensitive to their random initializations and do not converge to the same values throughout each training due to the non-convexity of the objective function and the instability of adversarial equilibrium between the generator and discriminator in GANs [26]. The work starts by training the generator and discriminator from scratch. Where the work of the discrimination is by discriminating the image if it is real from the dataset or fake from the generator. The generator's job is to generate very distorted images, while the discriminator's work is to give random commands that don't know if the image is real or fake. Over time, when you provide it with a lot of data, about, say, a million images, it will start to discriminate between fake and real, and the generator will start making images that look like the real ones. In general, the work of the discriminated is to train the generator to generate an image that is closer to the truth. The following law represents the mechanism of action.

$$\frac{\partial}{\partial \theta} \frac{1}{g} \frac{1}{m} [\ln \ln [1 - D(G(z))]] \quad (1)$$

The above Generator is modeled by a neural network $G(z, 1)$. Its function is to translate the required data space, x , to the input noise variables, z . On the other hand, a second neural network $D(x, 2)$ simulates the discriminator and generates the likelihood that the data originated from the genuine dataset, in the range $(0,1)$.

The Discriminator is then trained to accurately identify whether the incoming data is authentic or fraudulent. This implies that its weights are adjusted to increase the likelihood that any actual data input x will be identified as being part of the real dataset and to reduce the likelihood that any false image will be identified as being part of the real dataset. In more precise words, the loss/error function maximizes $D(x)$ and reduces $D(G(z))$ functions.

3.2 Dataset

Previous studies contain many data that have been worked out extensively. In our research, we downloaded the FFHQ dataset of 76,400 images from kaggle, then trained the images and generated fake images after resizing the real images to $32*32$ through the data tool. Also, we will use a dataset-tool for the main image folder to change the format of the real images in it to png to fit our proposed integration in order to obtain consistent and standard images for use in generating fake images shown in Figure 2.

Also, we used our real data by inserting personal images that

are being projected to generate fake images that are close to the real images, as presented in Figure 3. Also, we used our real data by inserting personal images that are being projected to generate fake images that are close to the real images. In addition, we trained the model in real-time using the laptop's camera, taking live images of two individual, and generating fake images by moving from the first image to the second, see Figure 4.



Figure 2. Real images dataset



Figure 3. Personal images

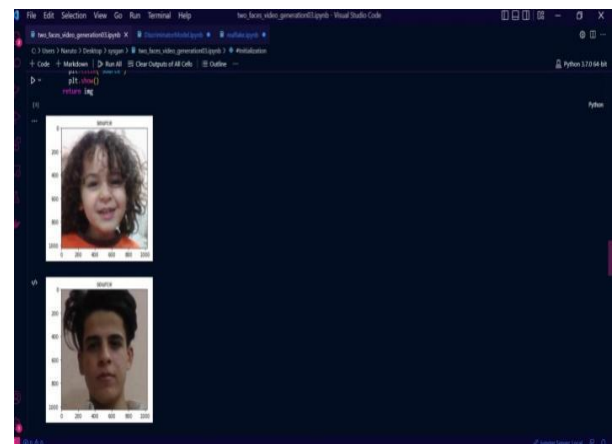


Figure 4. Live images from Laptop camera

3.3 Random projection

The cutout augmentation may be seen of as projecting a random subset of the dimensions to zero, for instance, the pixel and patch blocking in AmbientGAN [27]. Let P_1, P_2, \dots, P_N represent a collection of deterministic projection augmentation operators, where $P_j^2 = P_j$ serves as their distinguishing characteristic. Each of these operators, for instance, has the ability to set a separate fixed rectangular region to zero. It is obvious that each projection has a null space (unless it is the identity projection) and that it cannot be inverted independently.

Take into account a stochastic augmentation that chooses either the identity or one of these projections at random. The discrete probabilities of selecting the identity operator I for p_0 and P_K for the subsequent p_k are denoted by p_0, p_1, \dots, p_N . Define the projections' combination as:

$$T = p_0 I + \sum_{j=1}^N x p_j P_j \quad (2)$$

T is a collection of operators once more, but unlike the preceding examples, some (but not all) of them are not invertible. T is invertible on the probability distribution p , but under what circumstances? Assume that T is not invertible, meaning that a probability distribution with the shape $x \neq 0$ exists that makes $Tx=0$.

$$0 = T = p_0 I + \sum_{j=1}^N x p_j P_j \quad (3)$$

$$\sum_{j=1}^N x p_j P_j x = -p_0 x \quad (4)$$

We can consider the inner product of both sides of this equation with x if we make some technical assumptions (such as the discreteness of the pixel intensity values, which is justified in Theorem 5.4 of Bora et al. [27]):

$$\sum_{j=1}^N x p_j \langle x, P_j x \rangle = -p_0 \langle x, x \rangle \quad (5)$$

If the probability p_0 of identity is larger than zero, the right side of this equation is absolutely negative since $x \neq 0$. Since the inner product of a vector with its projection is non-negative, the left side is a non-negative sum of non-negative components. Therefore, unless $p_0=0$, the assumption contradicts itself; on the other hand, if there is a non-zero chance that it would yield the identity, random projection augmentation does not leak.

3.4 The proposed method

The development of Deepfake's neural networks, which excel at producing convincing human-sounding images and videos, has significantly enhanced the technology [28]. We presented a system that could identify forgeries image files in order to lessen the negative impacts brought on by these types of images. Recently, several new methods have emerged in this field. Many Deepfake models exist, including CycleGAN,

GANimation, StarGAN, and others. GANimation can only change one whereas StyleGAN can edit many characteristics. Since it is robust enough, we create a model to counteract StyleGAN manipulation. The StyleGAN is supplied with real images [29]. The altered photos are produced by StyleGAN. In this study, we will examine and contrast the StyleGAN 2 and StyleGAN-ADA. Understanding (and managing) the image synthesis process in the convolutional GAN generator is the goal of SyleGAN, See Figure 5.

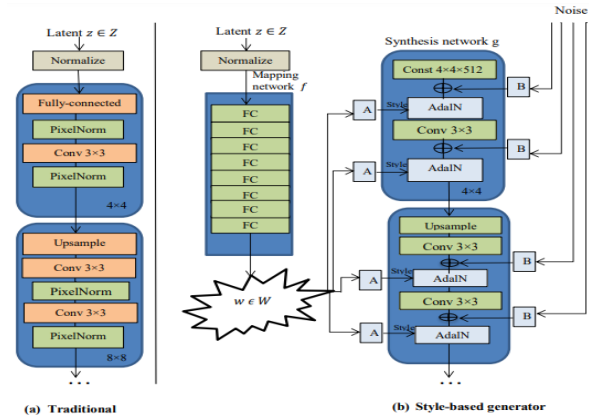


Figure 5. Traditionally, a feed forward network's input layer, or the first layer, is where the generator receives the latent code (a). By completely removing the input layer and beginning from a learnt constant instead, we stray from this architecture (b)

The architecture of the generator would be altered, according to the authors. A feed-forward network projects and detangles the input into a middle latent space rather than giving the generator the latent code (Z) directly (W). To directly adjust the adaptive instance normalization (AdaIN) after each convolution, an affine transformation may be generated from W . As a result, W will be encouraged to specialize in various styles by the affine transform settings. Keep in mind that a learning constant tensor serves as the generator's input. Each feature map is then given an additional dose of Gaussian noise to facilitate the creation of stochastic details.

The AdaIN operation is defined as:

$$AdaIN(x_i, y) = y_{s,i} \frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i} \quad (6)$$

Since the learnt mapping induces its sampling density, the intermediate latent space W does not need to support sampling in accordance with any fixed distribution (whereas Z must). Since disentangled factor representation makes it simpler to produce realistic images, the generator has an incentive to linearize the factor of variation in Z while they are learning. The authors found no advantage to giving the latent code as input to the generator when adding the mapping network and AdaIN. They streamlined the design by requiring an input of a constant (learned) tensor. These modifications have the significant benefit of enabling both high-level and low-level control over the styles, as each style only has a local impact (specific to a convolution operation).

Where the matching scalar components from style y are used to scale and bias each feature map x_i after it has been individually normalized. Since there are twice as many feature

maps on that layer, y has two dimensions. Instead of using an example picture in our method as in style transfer, we calculate the spatially invariant style y from vector w . As comparable network topologies are previously utilized for feedforward style transfer [30], unsupervised image-to-image translation [31], and domain mixes [32], we decided to reuse the term "style" for y . AdaIN is especially well suited for our aims because to its effectiveness and concise representation when compared to other generic feature transformations [33, 34].

3.4.1 StyleGan2

StyleGAN-2 outperforms StyleGAN-1 in two different ways. In order to address the "blob" issue, it first applies the style latent vector to alter the weights of the convolution layer. The created image has a "blob" problem since important information is lost when the resulting image is normalized using the style latent vector. As a result, the generator learns to produce a big blob that acts as a "distraction," absorbing the majority of the normalization's effects (somewhat similar to using flares to distract a heat-seeking missile). Two, it makes advantage of residual connections to assist it to avoid the problem where some features become stuck at pixel intervals [7]. There are two elements to the StyleGAN2 generator. An original, normally distributed latent is first converted into an intermediate latent code, $w \sim W$, via a mapping network. Finally, a synthesis network G uses a learnt $4 \times 4 \times 512$ constant Z_0 to create an output image $Z_N = G$ by applying a series of N layers made up of convolutions, nonlinearities, upsampling, and per-pixel noise ($Z_0; w$). The modulation of the convolution kernels in G is controlled by the intermediate latent code w . Two layers are run at each resolution according to a strict $\times 2$ upsampling schedule, and the number of feature maps is trimmed in half after each upsampling. Skip connections, mixing regularization, and path length regularization are further techniques used by StyleGAN2.

Stylegan2 was updated by the StyleGAN-2-ADA (where "ADA" stands for "adaptive") [4] which employs invertible data augmentation. The term "adaptive" refers to how it adjusts the amount of data augmentation used by beginning at zero and progressively increasing it until a "overfitting heuristic" reaches a goal level.

3.4.2 StyleGan2-ADA

Designing a technique to train GAN with less data is the goal of StyleGAN2-ADA, where ADA stands for Adaptive Discriminator Augmentation, see Figure 6.

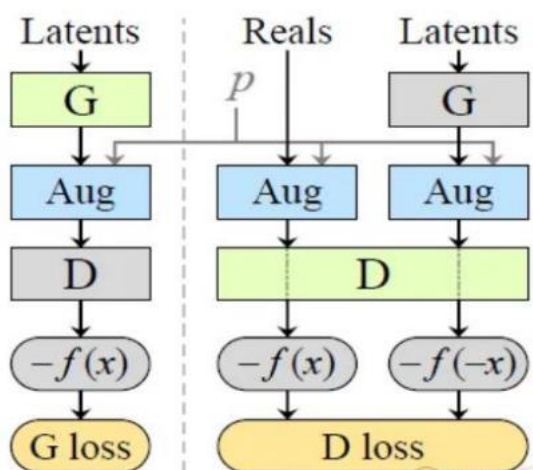


Figure 6. Stochastic Discriminator Augmentation [35]

The letters G and D stand for the discriminator and generator, respectively. The augmentation probability, p , which determines the intensity of the augmentations, lies between 0 and 1. We often add augmentations to the pipeline, so the discriminator D seldom gets a clean picture. The value of p will be set at about 0.8.

Before the discriminator D during training evaluates the generated images, they will be enhanced. The generator G is instructed to only create clean images because the augmentation process is performed after the generation.

4. RESULTS AND DISCUSSION

In this section the obtained results using the stylegan2-ada method will be presented and compared with the old method stylegan2. Proposed system was implemented in python using OpenCV [36] library for age processing and face detection, in addition, it was implemented using PyTorch due to its high speed compared to TensorFlow. In addition, the used data was trained using the NVIDIA RTX 20786 processor. We first generated images on a previously trained model to test the programs if they were working correctly. then, we downloaded data from kaggle and created a dataset-tool for it through which the images were initialized and moved from normal images to images ready for training.

4.1 Dataset

To evaluate the performance of our used method, we downloaded the FFHQ data set consisting of 76,400 images from Kaggle that were trained after resizing them to suit our computer used to create fake images by dividing the number into two groups where the first group contains 6400 images and through the dataset tool the program puts Every 1000 images in a folder, then the program performs training of the master folder that contains the prepared images, and the result is as seen in Figure 7, which shows the fake images that were generated from the real images.

As presented in Table 2, the total dataset is 6400 images, but 1000 images were trained, and the amount of wrong was 167. Then we trained 101 images, and the amount of wrong was 20. As for the performance, it was 83.3 and 80.2, respectively. Figure 8 presents the graph of image setup and performance rating of how effective fake image detection.

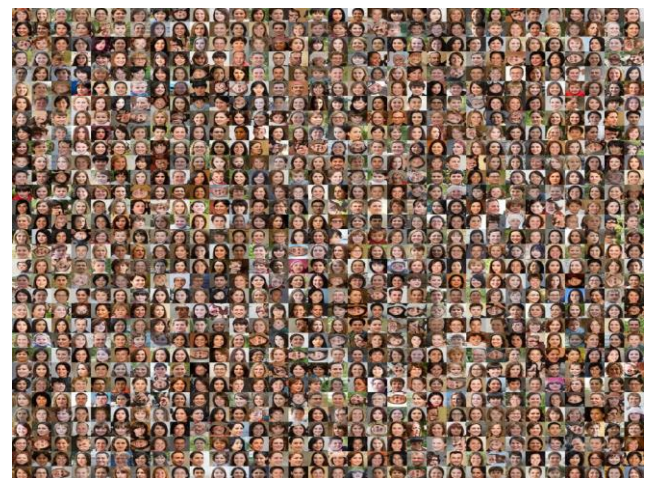


Figure 7. Fake images dataset

Table 2. Results of FFHQ dataset

Dataset	Generate	Wrong generate	Performance
6400	1000	167	83.3
	101	20	80.2
	201	40	80.2

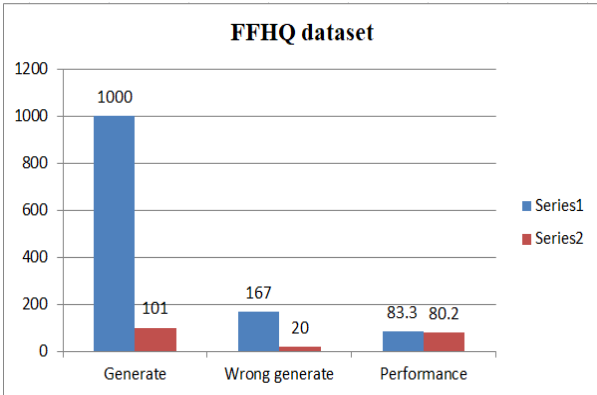
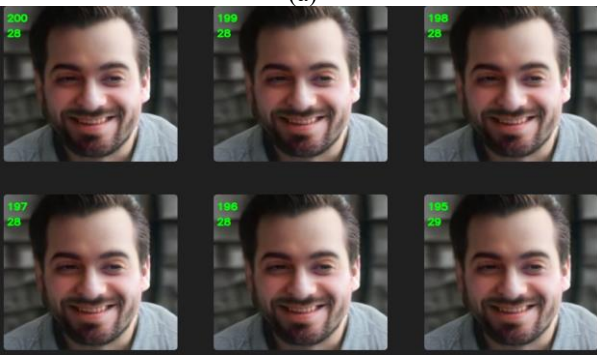


Figure 8. Performance of FFHQ dataset



(a)



(b)

Figure 9. (a) Personal images projection. (b) generating images with number and age



Figure 10. Result of live images from laptop camera

As for the personaed images, where we inserted two images to create a projection between them. Selecting the number of images is not optional, for instance, random number will be selected such as 200 the program generates 200 fake images by moving between the two images that were entered while saving the image with a label, each image contains the generated number and age as presented in Figure 9. Since there were not any false images generated, the result is 100% of generating fake images by entering real personal images.

In addition, we trained the model in real-time using the laptop's camera, taking live images of two individuals, and generating fake images by moving from the first image to the second as presented in Figure 10.

4.2 StyleGan2-ADA

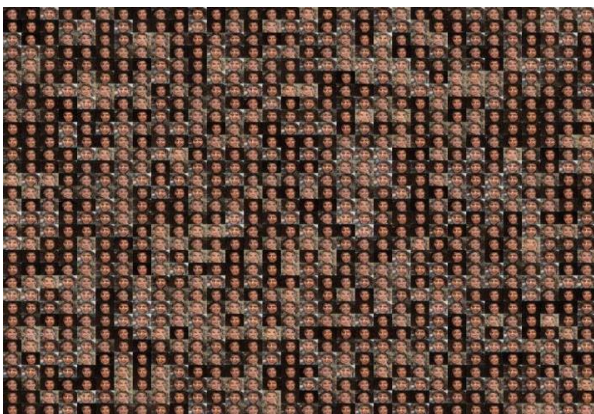
An architecture known as StyleGAN2 [37, 38] is regarded as the initial attempt at the Deepfake model challenge. Given its ease of fine-tuning training and ability to produce almost artifact-free. Deepfake face images with minimal training samples, StyleGAN2-ADA [39, 40] was the implementation which was used for the same reason in producing almost artifact-free fake face images. The Stylegan2-ada implementation also makes it simple to freeze the weights and parameters of a user-specified number of layers in the "Discriminator" during training operations. The initial StyleGAN2-ADA implementation was modified in a way that freezing capability was also accessible for the layers of the "Generator". This process was crucial for the work of training many models with slightly varied parameters (also known as weights), from which images that were almost "identical" to real images were generated.

Stylegan2-ada has the advantage that it works with less data because it has an augmentation that turns a few images into many images which is not found in stylegan2. As presented in Figure 11, throughout the training on the same amount of data for both methods, which displays the accuracy and clarity of images in stylegan2-ada. StyleGAN2-ada obviously produces the most realistic-appearing expressions while accurately preserving the input's individuality and facial traits. Despite the fact that stylegan2 mostly maintains the identity of the input, many of their outcomes are displayed fuzzy and do not keep the degree of clarity as seen in the input. By producing hazy images, GAN even fails to maintain the individual's identity in the images. The implicit data augmentation effect from a multi-task learning scenario, in our opinion, is the reason styleGAN2-ada outperforms other image quality algorithms. Images have a limited number of samples, such as 500 photos per domain. Stylegan2 can only utilize 1,000 training images at a time when trained on two domains, whereas styleGAN2-ada may use 4,000 images in total from all the accessible domains. This enables styleGAN2-ada to correctly learn how to preserve the standard and clarity of the output that is produced.

In Figure 12(a), the level of the curve using stylegan2-ada rose to a high level, as if the training was done correctly, which means that the generator could deceive the discriminator in the training. As for the second curve, which was done using stylegan2, when the images were given for training, the result was high, but with time it began to decline because the generator could not train well. For this, the discriminator was able to take enough time to distinguish between the real and the fake images.

In Figure 12(b), the level of the curve decreased using stylegan2-ada at the same time that the curve rose in Figure 11(a), and this indicates that the training was done well, but in Stylegan2, the generator increased because the training was not done well, and in this case, the discriminator is the best. And he was able to detect the fake images from the real ones.

In Figure 13(a), the two methods stylegan2 and stylegan2-ada were used and could not generate fake images, and after training better, the stylegan2-ada method maintained a high level of training, while stylegan2 only decreased its training level and its inability to generate good fake images. As for Figure 13(b), the detection score was low in the stylegan2-ada method, meaning that the discriminator could not distinguish between the real and the fake, while in stylegan2 the discriminator was able to overcome the generator in distinguishing between the real and the fake.

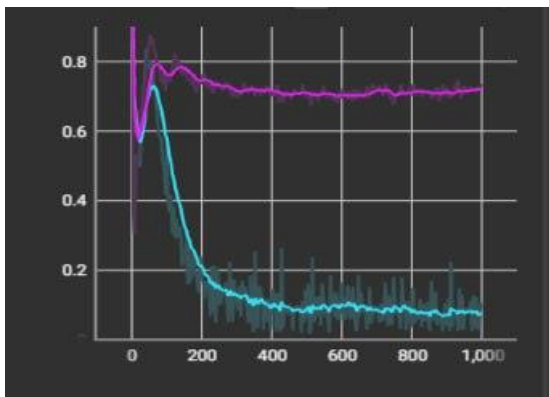


(a)

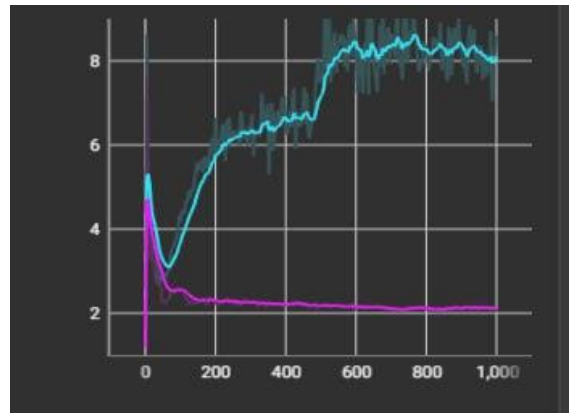


(b)

Figure 11. Difference between (a) stylegan2 and (b) stylegan2-ada

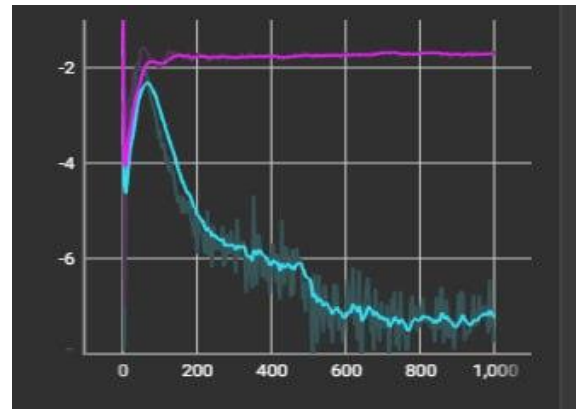


(a)

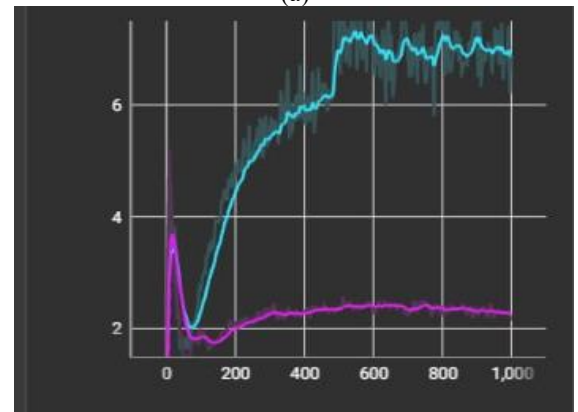


(b)

Figure 12. Difference between stylegan2 and stylegan2-ada (a) loss discriminator (b) loss generator



(a)



(b)

Figure 13. Difference between stylegan2 and stylegan2-ada (a) score fake (b) score real

5. CONCLUSION

This paper aims to create a large number of images of fake individuals by using images of real individuals. 76400 FFHQ images were used from Kaggle, these images were entered into python program using the stylegan2-ada method then compared it with stylegan2 where the images generated by the stylegan2-ada method were accurate and high quality, unlike stylegan2 which needs a large amount of data in order to read the images correctly. Previous research lacks the use of a high-specification computer which makes it difficult to train the model. This problem was addressed by using a high-

specification computer with a high screen card. In addition, real data was used by inserting personal images that are being projected to generate fake images that are close to the real images to create any number of images that were entered, for example, number 200 was entered and the program created a folder containing images as a result of the projection between two images, and the program also calculated the age for each image with the work of the label for each of them. Then a camera was used directly from the computer and recorded live image. We also entered the required number for the purpose of generating it. Our proposed approach was accurate and it was proven that this method possesses a latent, non-overlapping space that allows for control and liberation. To further improve the performance of the proposed method, a computer a high ram card will be used for the experiment, and we will also study a new method in which the error rate is low.

REFERENCES

- [1] Guarnera, L., Giudice, O., Nießner, M., Battiato, S. (2022). On the exploitation of Deepfake model recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 61-70. <https://doi.org/10.1109/CVPRW56347.2022.00016>
- [2] Mohana, Shariff, D.M., Abhishek, H., Akash, D. (2021). Artificial (or) fake human face generator using generative adversarial network (GAN) machine learning model. In 2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT), IEEE, 1-5. <https://doi.org/10.1109/ICECCT52121.2021.9616779>
- [3] Frühstück, A., Singh, K.K., Shechtman, E., Mitra, N.J., Wonka, P., Lu, J.W. (2022). InsetGAN for full-body image generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 7723-7732. <https://doi.org/10.1109/CVPR52688.2022.00757>
- [4] Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T. (2020). Training generative adversarial networks with limited data. In Proceedings of the 34th International Conference on Neural Information Processing Systems, 33: 12104-12114.
- [5] Karakas, C.E., Dirik, A., Yalçınkaya, E., Yanardag, P. (2022). FairStyle: Debiasing styleGAN2 with style channel manipulations. In Computer Vision-ECCV 2022: 17th European Conference, Proceedings, Part XIII, 570-586. https://doi.org/10.1007/978-3-031-19778-9_33
- [6] Karras, T., Laine, S., Aila, T. (2021). A style-based generator architecture for generative adversarial networks. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(12): 4217-4228. <https://doi.org/10.1109/TPAMI.2020.2970919>
- [7] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T. (2020). Analyzing and improving the image quality of stylegan. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 8110-8119. <https://doi.org/10.1109/CVPR42600.2020.00813>
- [8] Zou, H., Ak, K.E., Kassim, A.A. (2020). Edge-gan: Edge conditioned multi-view face image generation. In 2020 IEEE International Conference on Image Processing (ICIP), IEEE, 2401-2405. <https://doi.org/10.1109/ICIP40778.2020.9190723>
- [9] Liu, L.L., Zhang, H.J., Xu, X.F., Zhang, Z., Yan, S.C. (2020). Collocating clothes with generative adversarial networks cosupervised by categories and attributes: a multidiscriminator framework. In IEEE Transactions on Neural Networks and Learning Systems, 31(9): 3540-3554. <https://doi.org/10.1109/TNNLS.2019.2944979>
- [10] Khatri, K., Asha, C.S., D'Souza, J.M. (2022). Detection of animals in thermal imagery for surveillance using GAN and object detection framework. In 2022 International Conference for Advancement in Technology (ICONAT), 1-6. <https://doi.org/10.1109/ICONAT53423.2022.9725883>
- [11] Torrado, R.R., Khalifa, A., Green, M.C., Justesen, N., Risi, S., Togelius, J. (2020). Bootstrapping conditional GANS for video game level generation. In 2020 IEEE Conference on Games (CoG), IEEE, 41-48. <https://doi.org/10.1109/CoG47356.2020.9231576>
- [12] Tariq, S., Lee, S., Kim, H., Shin, Y., Woo, S.S. (2018). Detecting both machine and human created fake face images in the wild. In Proceedings of the 2nd International Workshop on Multimedia Privacy and Security, 81-87. <https://doi.org/10.1145/3267357.3267367>
- [13] Do, N.T., Na, I.S., Kim, S.H. (2018). Forensics face detection from GANS using convolutional neural network. ISITC, 376-379.
- [14] Xuan, X.S., Peng, B., Wang, W., Dong, J. (2019). On the generalization of GAN image forensics. In Biometric Recognition: 14th Chinese Conference, Proceedings, 134-141. https://doi.org/10.1007/978-3-030-31456-9_15
- [15] McCloskey, S., Albright, M. (2018). Detecting GAN-generated imagery using color cues. arXiv preprint arXiv: 1812.08247. <https://doi.org/10.48550/arXiv.1812.08247>
- [16] Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., Nießner, M. (2016). Face2face: Real-time face capture and reenactment of RGB videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2387-2395. <https://doi.org/10.1109/CVPR.2016.262>
- [17] Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Nießner, M. (2019). Faceforensics++: Learning to detect manipulated facial images. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, 1-11. <https://doi.org/10.1109/ICCV.2019.00009>
- [18] Hakim, H., Fadhil, A. (2021). Survey: Convolution Neural networks in Object Detection. Journal of Physics: Conference Series, 1804: 12095. <https://doi.org/10.1088/1742-6596/1804/1/012095>
- [19] Hsu, C.C., Lee, C.Y., Zhuang, Y.X. (2018). Learning to detect fake face images in the wild. In 2018 International Symposium on Computer, Consumer and Control (IS3C), IEEE, 388-391. <https://doi.org/10.1109/IS3C.2018.00104>
- [20] Li, Y.Z., Lyu, S. (2018). Exposing Deepfake videos by detecting face warping artifacts. arXiv preprint arXiv: 1811.00656. <https://doi.org/10.48550/arXiv.1811.00656>
- [21] Karras, T., Aila, T., Laine, S., Lehtinen, J. (2017). Progressive growing of GANS for improved quality, stability, and variation. arXiv preprint arXiv: 1710.10196. <https://doi.org/10.48550/arXiv.1710.10196>
- [22] Hulzebosch, N., Ibrahim, S., Worring, M. (2020). Detecting CNN-generated facial images in real-world

- scenarios. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2729-2738. <https://doi.org/10.1109/CVPRW50498.2020.00329>
- [23] Neves, J.C., Tolosana, R., Vera-Rodriguez, R., Lopes, V., Proença, H., Fierrez, J. (2020). Ganprintr: Improved fakes and evaluation of the state of the art in face manipulation detection. In IEEE Journal of Selected Topics in Signal Processing, 14(5): 1038-1048. <https://doi.org/10.1109/JSTSP.2020.3007250>
- [24] Lee, S., Tariq, S., Shin, Y., Woo, S.S. (2021). Detecting handcrafted facial image manipulations and GAN-generated facial images using Shallow-FakeFaceNet. Applied Soft Computing, 105: 107256. <https://doi.org/10.1016/j.asoc.2021.107256>
- [25] Hussain, S., Neekhara, P., Jere, M., Koushanfar, F., McAuley, J. (2021). Adversarial Deepfakes: Evaluating vulnerability of Deepfake detectors to adversarial examples. In 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 3347-3357. <https://doi.org/10.1109/WACV48630.2021.00339>
- [26] Tang, C.Y., Lu, H.H., Qiao, Z.P., Jiang, Y., Du, F.Z., He, J.Q., Jiang, Y.L., Wang, Q., Yu, H.Y. (2022). Ohmic contact with a contact resistivity of $12 \Omega \cdot \text{mm}$ on p-gan/algan/gan. In IEEE Electron Device Letters, 43(9): 1412-1415. <https://doi.org/10.1109/LED.2022.3193004>
- [27] Bora, A., Price, E., Dimakis, A.G. (2018). Ambientgan: generative models from lossy measurements. In International conference on learning representations.
- [28] Nasar, B.F., Sajini, T., Lason, E.R. (2020). Deepfake detection in media files-audios, images and videos. In 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS), IEEE, 74-79. <https://doi.org/10.1109/RAICS51191.2020.9332516>
- [29] Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J. (2018). Stargan: unified generative adversarial networks for multi-domain image-to-image translation. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8789-8797. <https://doi.org/10.1109/CVPR.2018.00916>
- [30] Huang, X., Belongie, S. (2017). Arbitrary style transfer in real-time with adaptive instance normalization. In 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, 1510-1519. <https://doi.org/10.1109/ICCV.2017.167>
- [31] Zhang, C., Xi, W., Liu, X.H., Bai, G., Sun, J.D., Yu, F. (2022). Unsupervised multimodal image-to-image translation: generate what you want. In 2022 International Joint Conference on Neural Networks (IJCNN), IEEE, 1-8. <https://doi.org/10.1109/IJCNN55064.2022.9892018>
- [32] Hao, G.Y., Yu, H.X., Zheng, W.S. (2018). Mixgan: learning concepts from different domains for mixture generation. arXiv preprint arXiv: 1807.01659. <https://doi.org/10.48550/arXiv.1807.01659>
- [33] Li, Y.J., Fang, C., Yang, J.M., Wang, Z.W., Lu, X., Yang, M.H. (2017). Universal style transfer via feature transforms. Advances In Neural Information Processing Systems, 30.
- [34] Siarohin, A., Sangineto, E., Sebe, N. (2018). Whitening and coloring batch transform for gans. arXiv preprint arXiv: 1806.00420. <https://doi.org/10.48550/arXiv.1806.00420>
- [35] StyleGAN vs StyleGAN2 vs StyleGAN2-ADA vs StyleGAN3. <https://medium.com/@steinsfu/stylegan-vs-stylegan2-vs-stylegan2-ada-vs-stylegan3-c5e201329c8a>, accessed on Mar. 14, 2023.
- [36] Abed, A., Rahman, S. (2017). Python-based raspberry pi for hand gesture recognition. International Journal of Computer Applications, 173: 975-8887. <https://doi.org/10.5120/ijca2017915285>
- [37] Ayanthi, D.M.A., Munasinghe, S. (2022). Text-to-face generation with stylegan2. arXiv preprint arXiv: 2205.12512. <https://doi.org/10.48550/arXiv.2205.12512>
- [38] Viazovetskyi, Y., Ivashkin, V., Kashin, E. (2020). Stylegan2 distillation for feed-forward image manipulation. In Computer Vision—ECCV 2020: 16th European Conference, Proceedings, Springer International Publishing, Part XXII 16: 170-186. https://doi.org/10.1007/978-3-030-58542-6_11
- [39] Fox, G., Tewari, A., Elgharib, M., Theobalt, C. (2021). Stylevideogan: A temporal generative model using a pretrained stylegan. arXiv preprint arXiv: 2107.07224. <https://doi.org/10.48550/arXiv.2107.07224>
- [40] de Curtò, J., de Zarzà, I., Yan, H., Calafate, C.T. (2022). Signature and log-signature for the study of empirical distributions generated with GANs. ArXiv preprint arXiv: 2203.03226. <https://doi.org/10.48550/arXiv.2203.03226>