Generating Synthetic Faces for Data Augmentation with StyleGAN2-ADA

Natália F. de C. Meira[a], Mateus C. Silva[b], Andrea G. C. Bianchi[c] and Ricardo A. R. Oliveira[d]
Department of Computer Science, Federal University of Ouro Preto, Ouro Preto, Brazil

Keywords: Generative Models, GANs, Synthetic Facial Data, Data Anonymization, Face-Swap.

Abstract: Generative deep learning models based on Autoencoders and Generative Adversarial Networks (GANs) have enabled increasingly realistic face-swapping tasks. The generation of representative synthetic datasets is an example of this application. These datasets need to encompass ethnic, racial, gender, and age range diversity so that deep learning models can avoid biases and discrimination against certain groups of individuals, reproducing implicit biases in poorly constructed datasets. In this work, we implement a StyleGAN2-ADA to generate representative synthetic data from the FFHQ dataset. This work consists of step 1 of a face-swap pipeline using synthetic facial data in videos to augment data in artificial intelligence model problems. We were able to generate synthetic facial data but found limitations due to the presence of artifacts in most images.

1 INTRODUCTION

Pre-trained AI models for people detection and facial recognition are becoming increasingly common in industrial and commercial environments. These models are generally trained on large datasets of facial images or images of human traffic in these environments. However, the problems of biased AI models for facial recognition and the need for data augmentation for a well-built solution for these models are known.

In the case of facial images, facial manipulation and facial switching techniques have evolved with a variety of specialized approaches and techniques (Yu et al., 2021). The most common approaches are face swap, synthesized aging, and rejuvenation. Among the solutions most investigated by these AI models are the generation of representative synthetic data to be later used in training deep learning (DL) networks.

The use of synthetic facial data has become necessary because large datasets can contain biases and not represent the test set of the model. Thus, the increasing use of synthetic data is due to the growing concern to avoid biases in the datasets that can lead the model to make errors, such as racism and discrimination against underrepresented groups in the training dataset.

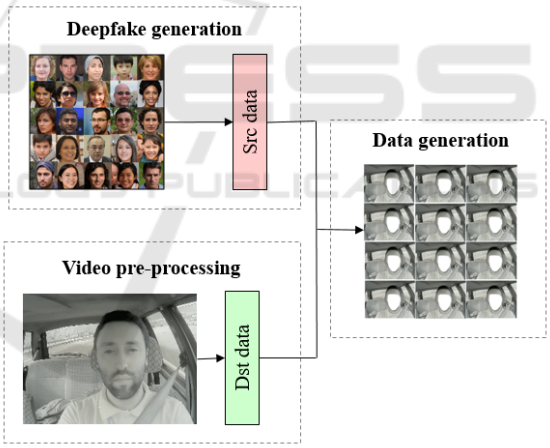

Figure 1: Pipeline for the automatic generation of synthetic facial data: step 1 - Deepfake generation: generation of deepfake images; step 2 - Video pre-processing: pre-processing the videos of the situation applied with the facial change, and; step 3 - Data generation: extraction of the synthetic dataset with face swap - adapted from Meira *et al.* (Meira et al., 2023).

Generative models can potentially learn any data distribution in an unsupervised way (Pavan Kumar and Jayagopal, 2021). The models behind these synthetic facial data generation tasks are based on Autoencoders (AEs) and Generative Adversarial Networks (GANs). Recently, models known as the Diffu-

[a] https://orcid.org/0000-0002-7331-6263
[b] https://orcid.org/0000-0003-3717-1906
[c] https://orcid.org/0000-0001-7949-1188
[d] https://orcid.org/0000-0001-5167-1523

Meira, N., Silva, M., Bianchi, A. and Oliveira, R.
Generating Synthetic Faces for Data Augmentation with StyleGAN2-ADA.
DOI: 10.5220/0011994600003467
In *Proceedings of the 25th International Conference on Enterprise Information Systems (ICEIS 2023) - Volume 1*, pages 649-655
ISBN: 978-989-758-648-4; ISSN: 2184-4992

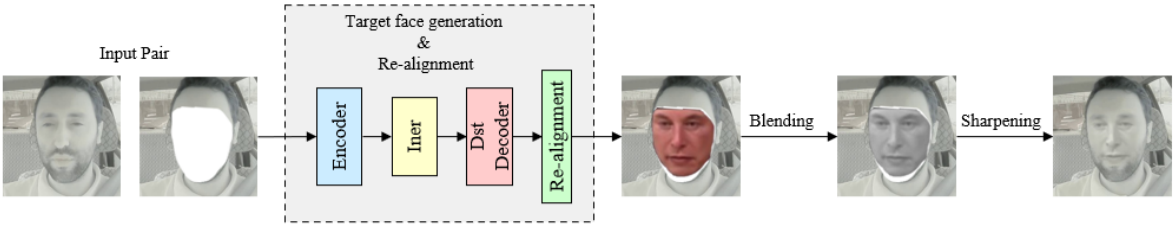

Figure 2: Overview of the conversion phase in the *DeepFaceLab* framework (DFL) - adapted from Perov *et al* (Perov et al., 2020) and Meira *et al* (Meira et al., 2023).

sion Model for generating false images are also being disseminated.

Therefore, this text presents the first steps towards a StyleGAN2-ADA implementation for facial swap in images. We display the theoretical framework, present the first experiments, and discuss the main limitations found in the usage of this technique towards the proposed goals. Therefore, the main objective of this work is:

- Propose and test a pipeline for facial swap using StyleGAN2-ADA.

Based on the discussion presented, the steps to complete the proposed goal are:

1. Investigate the techniques of generative models: Autoencoders, Diffusion Models, and Generative Adversarial Networks in a brief review of the literature and describe them to justify the choice of GANs for the implementation of this work;
2. Propose a pipeline for generating synthetic facial data from the proposed discussion.
3. Introduce the StyleGAN2 architecture for generating synthetic data deepfake;
4. Implement the StyleGAN2-ADA architecture for generating a representative synthetic facial dataset.

Our work presents recent generative model approaches and techniques. In addition, we propose a work pipeline for the generation of facial data to contribute to the community to develop more representative datasets without biases and discriminations, mainly ethnic and racial ones. This work aims to be STEP 1 demonstrated in Figure 1 for constructing a pipeline for generating synthetic facial data that will be replaced by the facial swap technique in videos of situations applied to generate training datasets of other models of AI.

We organized the remainder of this text as follows: We discuss the theoretical background in Section 2, presenting the main techniques of generative models for facial manipulation . Then, we assess the materials and methods in Section 3. Section 4 presents the results of our experiments and the discussion of the limitations and challenges of the current techniques. Finally, we display our conclusions and final remarks in Section 5.

2 THEORETICAL BACKGROUND

Generative models have been around for decades, and with the advancement of DL models, implementing generative modeling techniques has become more popular. Generative models assume that a spatial distribution from a latent space can represent data. Thus, these models make it possible to model a dataset in the form of Markov chains or through an iterative generative process (Harshvardhan et al., 2020).

Generative models generate data through an estimated probability distribution very close to the distribution of the original input data (Harshvardhan et al., 2020). Consider X an independent variable and Y a target variable. The generative models estimate the distribution given by P(X—Y) and P(Y).

Training deep generative models takes more time than discriminative models, as creating a probability distribution similar to the source dataset involves more correlations to learn (Harshvardhan et al., 2020), such as learning features than discriminative models convolutional data could have been ignored to generate a completely new dataset. For Harshvardhan *et al.* (Harshvardhan et al., 2020), generative models are important because:

- Can be used as tools to select indicative features that will improve the classification and accuracy of the model;
- Can be applied to generate realistic data samples.

When it comes to deep generative models, the techniques that allow applications like the one mentioned in this work are: Autoencoders, Diffusion Models, and Adversarial Generative Networks.

2.1 Autoencoders and Variational Autoencoders

The first investigated techniques within the context of Generative Adversarial Networks are Autoencoders. Mainly, they can be classified among traditional implementations and Variational Autoencoders.

2.1.1 Autoencoders

Harshvardhan *et al.* (2020) define Autoencoders as an unsupervised approach to learning feature representations of lower dimensions from unlabeled data. Autoencoders are models consisting of three layers: an input layer, where $\{x^i\}_{i=1}^N \in$ X, an intermediate layer, also called the *coding* or *bottleneck layer* Z and an output layer $\widehat{X}$.

The intermediate layer stores coded and extracted inputs in Z using weights. Then decoding Z generates an output $\widehat{X}$ similar to X. The encoding step can be expressed by a mapping function given in Equation 1 (Harshvardhan et al., 2020):

$$Z = f(WX + b) \tag{1}$$

Where b is the bias and W is the weight vector. The loss is calculated using an L2 loss function at the end of an epoch given by the Equation 2 (Harshvardhan et al., 2020):

$$L = ||x - \widehat{x}|| \tag{2}$$

After calculating the loss, the error is propagated back through the network to adjust the weights. Furthermore, Autoencoders can be used to initialize supervised classification models in which a classifier replaces the decoder. This classifier runs on the extracted feature vector Z to classify only based on the features coded as important (Harshvardhan et al., 2020).

2.1.2 Variational Autoencoders

Similar to the previously described Autoencoders, in the Variational Autoencoders models, we first assume that our data $\{x^i\}_{i=1}^N$ is generated by a prior latent distribution Z, assumed as a Gaussian and given by $p_\theta(z)$, where θ are the parameters learned by the model (Harshvardhan et al., 2020).

To generate the data, we can sample from x from a true conditional $p_\theta(x|Z^i)$ and estimate the proper parameters θ. This conditional can be represented by a neural network (Harshvardhan et al., 2020).

2.2 Generative Adversarial Networks

VAEs are not the most accurate in generating data similar to the original data because, in the case of images, blurring can be noticed in the generated images, (Harshvardhan et al., 2020). The GANs, introduced by Goodfellow *et al.* (Goodfellow et al., 2014), consist of a family of generative models capable of achieving high accuracy in data generation.

GANs are composed of two components: the discriminator and the generator. Both components work and learn features together, rather than one being pretrained.

We denote the distribution of the generator G by p_G over the source real data. We define a prior of input noise, which consists of a latent random variable $p_z(z)$. So G is a differentiable function since it operates on non-discrete data z with parameters θ_G whose data space is represented by $G_{(\theta_G)}$ (Harshvardhan et al., 2020).

We represent the D discriminator data space as $D_{(\theta_D)}(x)$ having parameter θ_D, which is the probability that the data that came from the source data is not false (Harshvardhan et al., 2020).

The objective function of a GAN consists of a function *minmax*, as given by Equation 3 where the generator tries to minimize the objective between a false sample and a real sample while the discriminator tries to maximize to differentiate real and fake samples[1].

$$\begin{aligned} \min_\theta \max_\phi V(G_\theta, D_\phi) = \\ E_{x\sim p_{data}}[\log D_\phi(x)] + \\ E_{x\sim p_z}[\log(1 - D_\phi(G_\theta(z)))] \end{aligned} \tag{3}$$

2.3 Diffusion Models

One of the families of Deep Generative Models is the Diffusion Models (Croitoru et al., 2022). Some of the applications of these models are image synthesis, video generation, speech generation, and natural language processing (Yang et al., 2022; Cao et al., 2022). Despite recent promising results, Diffusion Models have limitations, such as:

- Low time efficiency during inference caused by thousands of evaluation steps (Croitoru et al., 2022);
- Most of the improvements in the existing models for application are based on the original configuration as DDPM (Denoising diffusion probabilistic models). However, researchers need to

[1] https://deepgenerativemodels.github.io/notes/gan/

pay more attention to the widespread configuration of diffusion-based models. Thus, other significant work is exploring other distributions.

Among the Diffusion-based Generative Models that have already been proposed are: Diffusion Probabilistic Models (Sohl-Dickstein et al., 2015), Noise-Conditioned Score Network - NCSN (Song and Ermon, 2019), and Denoising diffusion probabilistic models - DDPM (Ho et al., 2020).

Diffusion Models are based on a forward diffusion stage, and a reverse stage (Croitoru et al., 2022). In the first stage of forward diffusion[2], the input data is perturbed gradually into several Markov chain diffusion steps by adding random Gaussian noise to the (Croitoru et al., 2022) data.

Consider a real data distribution $x_0 \sim q(x)$. Given a sampled point, we define a forward diffusion process by adding a small Gaussian noise to the sample in T steps. The result is a sequence of noise samples $x_1, \ldots, x_T$. The step sizes are controlled by $\{\beta_t \varepsilon(0,1)\}_{t=1}^{T}\}$.

$$q(X_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \quad (4)$$

We can sample x_t at any arbitrary time step by reparameterization. As the t step progresses, the data sample x_0 gradually loses its distinguishable characteristics (Croitoru et al., 2022). The larger the update step, the noisier the sample. In the reverse stage, the model tries to recover the original input data from the gradual reversal of the diffusion process to build desired data samples (Croitoru et al., 2022).

3 METHODOLOGY

This section presents the materials and methods employed in this work. Initially, we discuss the framework used to produce this solution. Then, we present the dataset employed for this case study.

3.1 Framework StyleGan2

The StyleGAN architecture consists of a style-based GAN architecture proposed by Karras *et al.* (Karras et al., 2019). The unsupervised model architecture automatically separates high-level attributes (e.g., pose and identity when trained on human faces) and stochastic variation in the generated images (e.g., freckles, hair). Furthermore, it allows a scale to control the synthesis (Karras et al., 2019). The synthesis

[2]https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

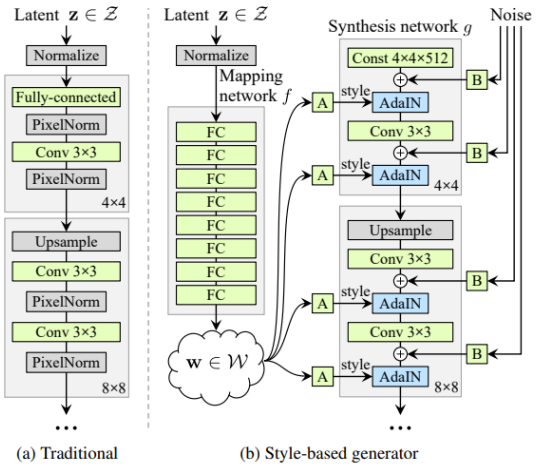

Figure 3: While a traditional generator feeds the latent code the input layer, the new proposal first maps the input to an intermediate latent space W, which then controls the generator through adaptive instance normalization (AdaIN) in each convolution layer. Gaussian noise is added after each convolution before evaluating non-linearity. Here, "A" stands for a learned affine transform, and "B" applies learned per-channel scale factors to the noise input. The mapping network f consists of 8 layers, and the synthesis network g consists of 18 layers, where two of them are for each resolution ($4^2 - 1024^2$). The output of the last layer is converted to RGB using a separate 1×1 convolution. The generator has a total of 26.2 million trainable parameters, compared to 23.1 million in the traditional generator (Karras et al., 2019).

concept is used when a *deepfake* is created without a base target (Mirsky and Lee, 2021).

The authors' motivation was to redesign the generator architecture to control the image synthesis process (Karras et al., 2019). The generator takes a constant learned input and adjusts the "style" of the image in each convolution layer based on the latent code. StyleGAN has latent space as an input that follows the probability density of the training data, and the intermediate latent vector is free of this restriction and, therefore, *disentangled* is possible. Figure 3 shows the architecture designed by the authors.

The work by Karras *et al.* (Karras et al., 2020) proposed changes in the model architecture and training methods to deal with the generated artifacts. The authors improved the state-of-the-art in terms of traditional distribution quality metrics and improved the interpolation properties. They focused on the intermediate latent space W, which is not as impacted by *disentangled* as the input latent space Z due to stochastic variation, making it easier to provide additional random noise maps to the synthesis network.

Regarding the artifacts generated by StyleGAN, Karras *et al.* (Karras et al., 2020) redesigned the

generator normalization to remove the artifacts that were generated to work around an architectural design flaw. The authors also proposed an alternative design to progressive growth related to the generation of artifacts. The training starts by focusing on low-resolution images and then progressively shifts the focus to higher and higher resolutions — without changing the network topology during training (Karras et al., 2020). For implementation, the authors have provided the official StyleGAN2 repository[3].

Then, the authors improved StyleGAN2[4] by proposing an adaptive discriminator augmentation mechanism to avoid discriminator overfitting. This strategy enables training with smaller datasets ($\sim$ 30,000 images).

3.2 *Dataset*

The dataset for this implementation consists of the *Flickr-Faces-HQ Dataset (FFHQ)*[5], a dataset created as a reference for GANs. The dataset consists of high-quality images of human faces, 70,000 PNG images with 1024x1024 resolution, containing variation in terms of ethnicity, background images, age, and accessories (glasses, hat).

Authors (Karras et al., 2020) trained StyleGAN2-ADA with eight high-end NVIDIA GPUs of at least 12GB of GPU memory and provided pre-trained weights from the dataset.

4 RESULTS

This section discusses the results obtained from the experimental apparatus. As presented in the previous section, our starting point was the pre-trained StyleGAN2-ADA model using the FFHQ dataset. Then, we start feeding random seeds to this model to obtain a latent representation as input to generate artificial images.

The seed generates a latent vector containing 512 floating point values. The GAN uses the provided seed to generate these 512 values. Furthermore, a small change in the latent vector results in a slight change in the image. From the observed results, we assess that even tiny changes in the integer seed value will produce radically different images.

We provide three seeds, and the value of steps the model must vary between these images to generate the *deepfakes*. The seed values that provide the most realistic results depend on the type of image being generated. For example, for faces, the best seed values vary between 6000 and 6500. Figure 4 shows some examples of generated realistic images.

In Figure 4, we observe few or almost no artifacts at first sight. However, many images had notable artifacts, particularly in the forehead and teeth/gums region. Figure 5 presents some images with generated artifacts.

Figure 4: These are examples of images generated with realistic aspects with a low degree of artefacs. It is expected to use images with this level of quality for facial swap in videos to generate synthetic facial data for training other DL models, such as fatigue detection, and monitoring of work activity, among others.

Figure 5: These are examples of facial images generated with many artifacts, easily detectable as fake faces, and which cannot be used in the generation of synthetic facial data.

4.1 Limitations and Challenges

We found some limitations in using this architecture to generate faces. These identified challenges can shape the next steps of further research toward this goal:

- Although the generated faces are impressively close to realistic faces, with more careful obser-

[3]https://github.com/NVlabs/stylegan2
[4]https://github.com/NVlabs/stylegan2-ada
[5]https://github.com/NVlabs/ffhq-dataset

vation, one can differentiate the images generated by GAN from a real photo;

- The previous observation limits, or at least generates more work in separating *deepfakes* images for the proposed implementation;
- We found limitations in the implementation to train with the FFHQ dataset since the files and metadata are very extensive, and we could not access them;
- Despite the limitation reported above, the weights made available by training with the FFHQ dataset were sufficient to generate a synthetic image using only the *G* generator;
- We were also unable to add the conditional to the generator, a step that must be overcome in future work.

5 CONCLUSION

This work displays the first steps in employing the StyleGAN2-ADA framework toward the facial swap task in images. We conducted our research starting with a theoretical analysis of the face swap techniques background. Then, we presented the proposed methods. Finally, we identified the main limitations and challenges by observing this process.

In implementing the model for the generation of synthetic faces in phase 1 of our pipeline, we obtained a set of realistic facial images. The limitations found were: hardware for training (the authors of the StyleGAN2-ADA architecture used eight self-performance NVIDIA GPUs for training the FFHQ facial dataset) and a large part of the dataset generated with artifacts in the images. We observed that the generator trained using this dataset was able to generate synthetic images.

Our experiments allowed us to identify several challenges in developing this solution. Among these, the initial observation is the need for more detail in generating synthetic faces. This aspect can jeopardize the production of face swap technologies, as we observed a visible loss in realism. There were also technical challenges, such as adding a conditional into the generator.

The difficulty in accessing the dataset was a relevant challenge in this research. In future works, we intend to carry out our training with our facial dataset, obtained from embedded hardware, 300x300 pixels resolution for comparison purposes. Our dataset has about 45,000 images. We also intend to reassess the strategy for obtaining facial data. In future work, we intend to use a 3D facial generator network to acquire synthetic facial images in all positions of the same identity.

ACKNOWLEDGEMENTS

The authors would like to thank CAPES, CNPq and the Federal University of Ouro Preto for supporting this work. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ) and the Universidade Federal de Ouro Preto (UFOP).

REFERENCES

Cao, H., Tan, C., Gao, Z., Chen, G., Heng, P.-A., and Li, S. Z. (2022). A survey on generative diffusion model. *arXiv preprint arXiv:2209.02646*.

Croitoru, F.-A., Hondru, V., Ionescu, R. T., and Shah, M. (2022). Diffusion models in vision: A survey. *arXiv preprint arXiv:2209.04747*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Harshvardhan, G., Gourisaria, M. K., Pandey, M., and Rautaray, S. S. (2020). A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review*, 38:100285.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.

Karras, T., Laine, S., and Aila, T. (2019). 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.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2020). Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119.

Meira, N., Santos, R., Silva, M., Luz, E., and Oliveira, R. (2023). Towards an automatic system for generating synthetic and representative facial data for anonymization. In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, pages 854–861. INSTICC, SciTePress.

Mirsky, Y. and Lee, W. (2021). The creation and detection of deepfakes: A survey. *ACM Computing Surveys (CSUR)*, 54(1):1–41.

Pavan Kumar, M. and Jayagopal, P. (2021). Generative adversarial networks: a survey on applications and challenges. *International Journal of Multimedia Information Retrieval*, 10(1):1–24.

Perov, I., Gao, D., Chervoniy, N., Liu, K., Marangonda, S., Umé, C., Dpfks, M., Facenheim, C. S., RP, L., Jiang, J., et al. (2020). Deepfacelab: Integrated, flexible and extensible face-swapping framework. *arXiv preprint arXiv:2005.05535*.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR.

Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32.

Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Shao, Y., Zhang, W., Cui, B., and Yang, M.-H. (2022). Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796*.

Yu, P., Xia, Z., Fei, J., and Lu, Y. (2021). A survey on deepfake video detection. *Iet Biometrics*, 10(6):607–624.