Self-attention generative adversarial network*

Tomasz Bury[1,*,†]

[1]Faculty of Applied Mathematics, Silesian University of Technology, Kaszubska 23, 44100 Gliwice, POLAND

Abstract

The possibilities of the generative approach are enormous because they enable the generation of data by using the knowledge used in the learning process. This allows you to create new images based on the given information. In this paper, we propose the architecture of a neural network based on a generative model with a generator and a discriminator, where an attention module is introduced. The attention module allows you to add a weighted matrix assigning importance to appropriate pixels, thus drawing attention to selected features. The proposed architecture was described and tested on a publicly available database using the ADAM algorithm.

Keywords

gan, self-attention, generative adversarial networks, cnn, sagan

1. Introduction

The possibilities of generating new data are important due to their multiple applications. First of all, generating new data can support the operation of machine learning techniques. Many methods, especially arti cial neural networks, are data-dependent. The more data there is, the better the network will likely be able to learn from it. Hence the name data-hungry algorithms is assigned to them. Generating new data is primarily the creation of synthetic data [1], the creation of which is quite often called augmentation. Augmentation can be implemented by classic data processing techniques. An example of such augmentation is the use of interpolation [2]. The proposed idea is based on the use of random resizing of interpolation focusing on the enlargement of the relevant data. Another example is text data augmentation [3], where various techniques were presented and compared and it was shown that generating large amounts of data that have a certain value for further analysis is very time-consuming. Another way is to use arti cial intelligence methods. An example is the hybridization of a network with a heuristic algorithm, where the heuristic supports the training process by analyzing selected features of graphic samples [4]. Such hybridization possibilities are possible by paying attention to the learning process, which is an optimization problem. Consequently, the use of algorithms inspired by nature allows you to obtain results in a shorter time than known, classic solutions. An example is using the grey wolf algorithm for clustering, or stitching images [5, 6]. Another hybridization is made by combining the fuzzy approach [7]. The conducted tests show that such a methodology is interesting and worth investigating. Moreover, fuzzy augmentation is also used for other purposes like expert analysis of obtained results. Such a solution was described in [8].

Moreover, data augmentation may allow for obtaining better values of classifier evaluation matrices. This is possible by balancing the data in each class. When the training database contains data mainly belonging to one class, learning them may result in over-adaptation to that one class at the expense of the others. Generating additional samples allows us to achieve a balance between data in all classes, which is quite often impossible in real conditions [9, 10, 11]. It is also possible to classify or even hierarchize the results using techniques such as multimoora [12]. Apart from the process of generating data, the technique of creating it is also important. By drawing attention to the process of creating or adapting a classi er to data, it allows attention to be drawn to the possibilities of analyzing the features of objects on given samples.

*IVUS2024: Information Society and University Studies 2024, May 17, Kaunas, Lithuania

[1,*] Corresponding author

[†] These author contributed equally.

tb303151@student.polsl.pl (T. Bury)

Feature extraction or their subsequent generation is important to understand the technique itself. Correct feature extraction and their subsequent mapping are crucial in generative models, hence research on new solutions and techniques is also interesting. An example is the possibility of feature extraction using various methods of pooling [13] or multiple convolutions operations [14]. Generative adversarial networks are mostly used to generate and process images, in particular in medical applications [15], so that new data samples can be created, such as X-ray or CT images. There are also versions of GAN that are used to generate text [16] or even .

Generative adversarial networks using convolution have trouble generating images that have a certain number of class-speci c features. Because of the use of a local receptive eld for the network to learn relevant relationships, these features may not be recognized correctly. In addition, traditional GANs perform poorly in generating high-resolution images. These problems are solved by the self-attention module, which focuses on di erent features of the photo instead of using successive xed-size regions. By analyzing the need for new data- generating techniques, in this paper, we propose a new architecture of the generative network model for generating images. The idea is to create two networks, one of which will learn to recognize real and fake samples, and the other will learn to generate samples that can fool the
rst network. The networks used have been extended with an self-attention module, which allows the classi er to draw attention to particular data features during training and solve problem with recognizing speci c class dependencies.

2. Proposed methodology

In this section, we described a used network architecture with detailed information about used layers.

2.1. Generative adversarial networks

The concept of generative adversarial networks was developed in 2014 by Goodfellow et al. [17]. Its operation is based on a zero-sum game between two neural networks, one generates new images from dataset images, and the other is a classi er that decides whether the image it processes is real or fake. Its extension is the DCGAN model created by Radford, Metz, and Chintala [18], and which uses deep convolutional layers to process images, just as it is calculated in convolutional neural networks. The generator and discriminator are trained simultaneously, where the rst model seeks to minimize loss and generate a photo that is indistinguishable from

the real one, while the second model seeks to maximize failure so that it can correctly classify the generated photo and the real photo from the dataset with as much accuracy as possible. The loss function developed in [17] is de ned as follows:

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

where $G(z)$ - generator, $D(z)$ - discriminator. $D(x)$ tries to maximize the function $V(D, G)$, while $G(z)$ minimizes this function, which leads to a minimax relation [19].

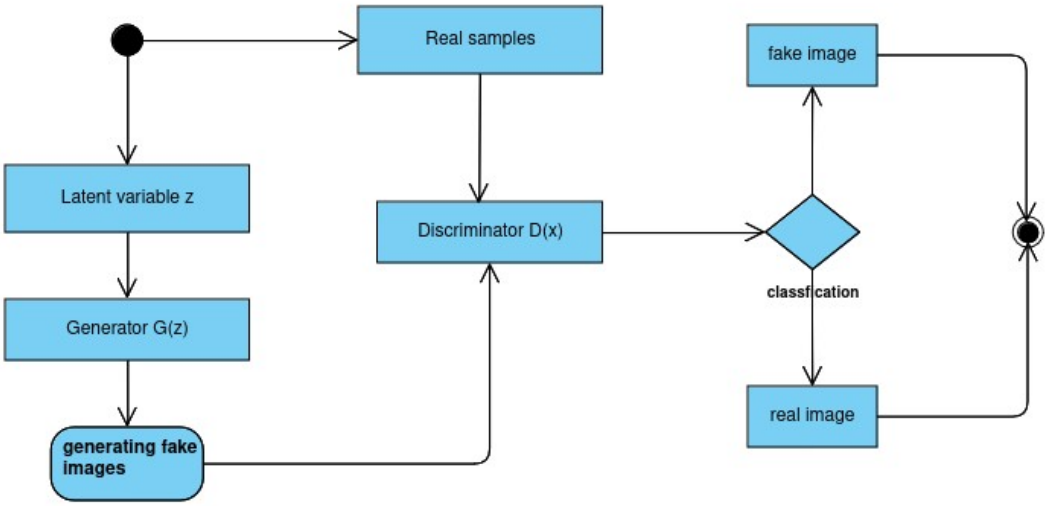

Figure 1: Simple GAN architecture diagram

2.2. Attention module

2.3. GAN architecture

This subsection will describe the self-attention generative adversarial network model used in the project. Before training is started, the images are preprocessed to get the best possible e ciency result and reduce their size to optimize the model's training time and improve its accuracy in generating new images. Each photo is cropped to 160 x 160 pixels to maximize the reduction of unnecessary backgrounds that can disrupt the network results. Then, the cropped photo is scaled to 64 x 64 pixels. The image thus processed is converted into a tensor to be analyzed by the model. The generator and discriminator are stochastically optimized using the ADAM algorithm [20], taking the parameters β_1, β_2, which are the hyperparameters of this model. After this optimization, the training process is started. The loss of both the generator and discriminator is calculated using Binary Cross Entropy loss with logits [21]. Operation of this function is based on combining the value of the sigmoid function with the cross entropy

function. Binary Cross Entropy loss with logits function is expressed by the equation as follows:

$$\mathcal{L} = -\sum_{i=1}^{N}(1 - t_i)z_i + C + log(e^{-C} + e^{-z_i - C}) \tag{2}$$

The detailed algorithm for training the network is described in Alg. 1.

Algorithm 1: Network training algorithm

Initialize generator, discriminator, optimizers for generator and discriminator;
for *epoch* **to** 1, 2, ..., *n* **do**
  Preprocess set of images ;
  Get real images and labels from set;
  Check the real image prediction made by discriminator;
  Prepare latent noise for generating fake photos;
  Generate fake image using generator;
  Check the fake image prediction made by discriminator;
  Calculate discriminator loss as $loss = 1/2(l_r + l_f)$ where l_r - loss at predicting real image, l_f - loss at predicting fake image;
  Calculate gradient penalty and add to discriminator loss;
  Backpropagate;
  Generate fake images with generator;
  Check the generated image prediction made by discriminator;
  Calculate generator loss;
  Backpropagate;
end

The generator and discriminator consist of ve convolution layers. To standardize the result of each layer, the batch normalization method is used, which makes training process much faster and stable [22]. In the case of the generator, the activation function is ReLU, while in the case of the discriminator, the function is LeakyReLU with the parameter *s*, which is responsible for the angle of the straight line in the case of negative values. To normalize the result obtained, a function *tanh* is used at the end of the generator network, which normalizes it to the interval [−1, 1], and in the case of the discriminator - a sigmoid function that normalizes the result to the interval [0, 1]. Each convolution layer of both the generator and discriminator is spectral normalized, which ensures the stability of the network operation and avoids the mode collapse problem [23]. To secure and further improve the stability of the model training process, the gradient penalty technique was used [24], making the convergence process much faster. There are also other heuristic optimization methods for the training process to ensure stability, such as noise injection [25] or minibatch discrimination [26]. The attention module is used between the third and fourth layers.

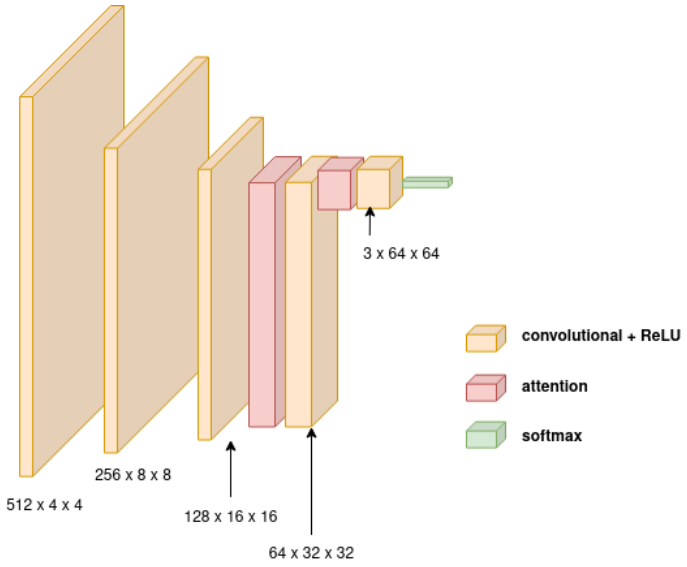

Figure 2: Diagram of generator architecture.

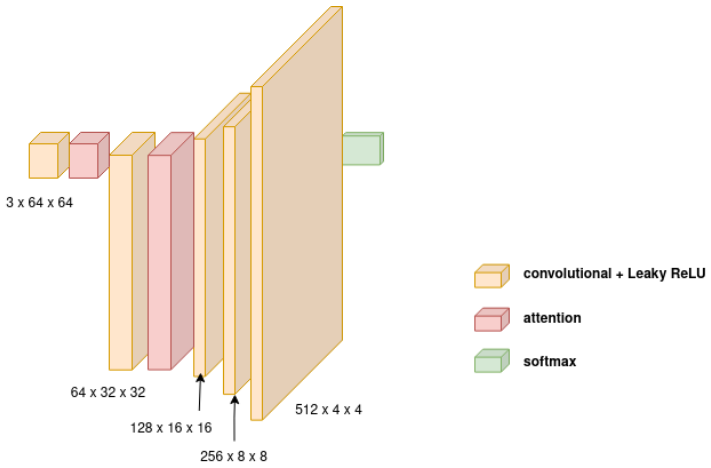

Figure 3: Diagram of discriminator architecture.

3. Experiments

This section will focus on analyzing the results of the generative adversarial network model. To train and test it, the CelebA [27] dataset, consisting of 202,599 celebrity images of 178 x 178 pixels, prepared by researchers at the Chinese University of Hong Kong, was used. The dataset was divided into a validation and training set in a ratio of 80:20. The learning rate for the generator was set to 0.0001, and for the discriminator to 0.0004. The s parameter in the LeakyReLU activation function was set to 0.2. The parameters β_1 and β_2 were set to 0.5 and 0.999, respectively. The results include graphs of how the loss of the generator and discriminator

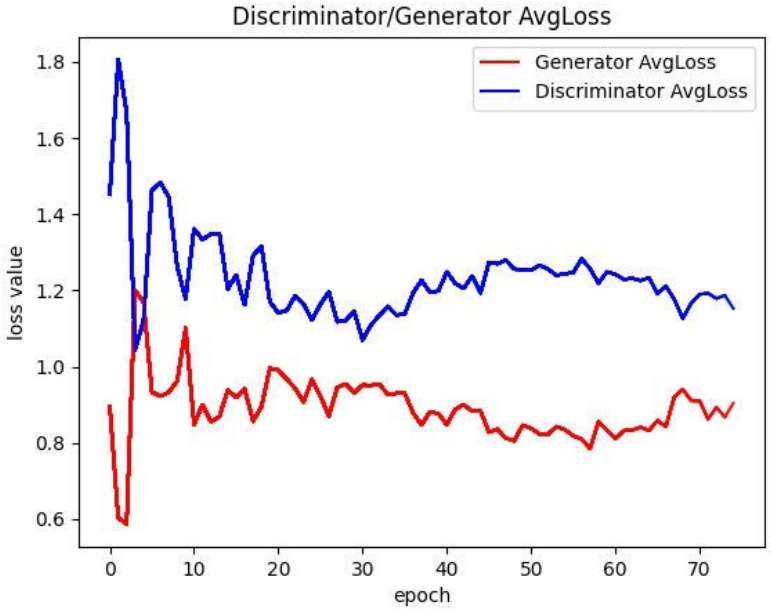

Figure 4: Discriminator and generator loss changing over epochs.

Figure 5: Faces generated a er first epoch.

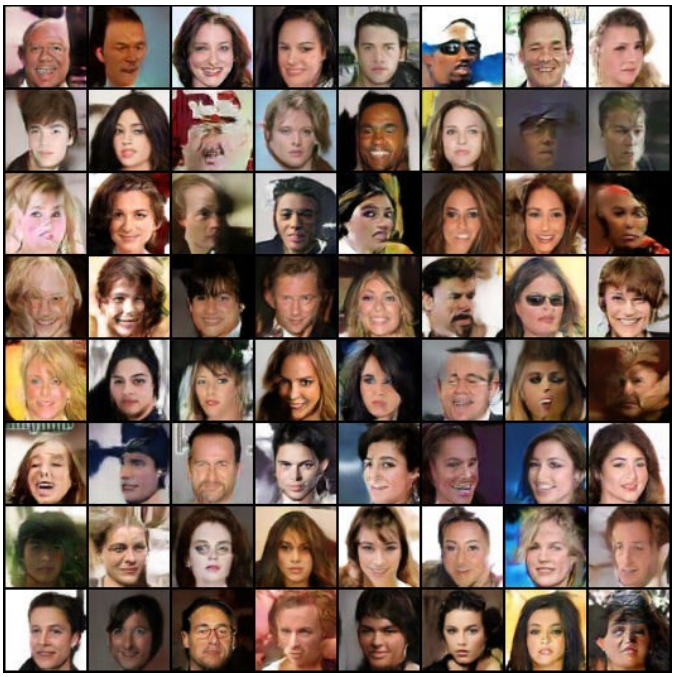

Figure 6: Faces generated a er last epoch.

changed over successive epochs. As can be seen in the chart, the initial average loss values for the generator and discriminator were very jittery, but over the subsequent epochs their value stabilized and converged, respectively, for the generator at a value of about 0.85, and for the discriminator at about 1.2. In Figure 6 we can see a sample of the faces generated by the network. More than half of them have acceptable quality, but there are isolated cases where the network generated only artifacts and the face itself is not recognizable. This is due to the relatively small number of training epochs. It should also be noted that not all samples are of equal quality - some are of great quality, making it di cult for the network to recognize whether a photo is fake or not. In our article, the Fréchet Inception Distance (FID) [28] was selected as a metric for evaluating the performance of various Generative Adversarial Networks (GANs). FID provides a meaningful measure by capturing both the visual delity and diversity of generated samples, making it suitable for comparing di erent GAN architectures. Its ability to consider the feature distributions of real and generated images allows for a comprehensive evaluation, enabling us to make informed comparisons between the models under study. The results of the comparison are presented in Table 1, including FID metric. In the case of this metric, lower values indicate better quality of generated images. Proposed architecture is performing better than BEGAN-CS and PR-BigGan networks. DualGAN achieves the lowest FID score of 13.95, demonstrating best performance in generating high-quality images within the CelebA dataset. This is due to introducing second discriminator and duels both between discriminator and generator and between discriminators, which improves level of diversity within all samples and prevents early convergence.

Table 1
The results of the comparison between generative adversarial networks and proposed architecture.

Method	FID
BEGAN-CS [29]	34.13
PR-BigGAN	22.45
DualGAN [30]	13.95
Proposed architecture	**21.88**

4. Conclusion

Thanks to the attention module, the model is capable of e ectively assigning weights to di erent input elements, which allows better capture of complex relationships in images composed of many elements. However, it is worth noting that there is a need for further research and modi cation of this network. Research should focus on optimizing the hyperparameters and increasing the stability of the learning process, which still is a challenge [31]. The number of potential applications of generative adversarial networks is enormous and allows to solve a number of problems in various scienti c elds. GANs can support the learning process of deep learning models in medical applications by generating images, supporting image reconstruction and repair when data is incomplete or in detecting anomalies, especially in the area of surveillance [32]. Adversarial networks can also be used in biometric attacks, generating for example a photo of the face, iris, ngerprints or voice, as well as to generate deep fake videos, which raises ethical concerns [33]. Securing against using the GAN for biometric attacks is another potential and important direction to follow.

References

[1] F. Garcea, A. Serra, F. Lamberti, L. Morra, Data augmentation for medical imaging: A systematic literature review, Computers in Biology and Medicine 152 (2023) 106391.

[2] D. Wan, R. Lu, T. Xu, S. Shen, X. Lang, Z. Ren, Random interpolation resize: A free image data augmentation method for object detection in industry, Expert Systems with Applications 228 (2023) 120355.

[3] L. F. A. O. Pellicer, T. M. Ferreira, A. H. R. Costa, Data augmentation techniques in natural language processing, Applied Soft Computing 132 (2023) 109803.

[4] D. Połap, A. Jaszcz, Heuristic feedback for generator support in generative adversarial network, Proceedings of the 16th International Conference on Agents and Arti cial Intelligence 3 (2024) 863–870.

[5] K. Prokop, Grey wolf optimizer combined with k-nn algorithm for clustering problem, IVUS 2022: 27th International Conference on Information Technology (2022).

[6] K. Prokop, D. Połap, Heuristic-based image stitching algorithm with automation of parameters for smart solutions, Expert Systems with Applications 241 (2024) 122792.

[7] R. Zhang, W. Lu, J. Gao, Y. Tian, X. Wei, C. Wang, X. Li, M. Yu, R -gan: A reference-guided fuzzy integral network for ultrasound image augmentation, Information Sciences 623 (2023) 709–728.

[8] Y. Gordienko, M. Shulha, Y. Kochura, O. Rokovyi, O. Alienin, S. Stirenko, Fuzzy metadata augmentation for multimodal data classi cation, in: Mobile Computing and Sustainable Informatics: Proceedings of ICMCSI 2023, Springer, 2023, pp. 157–172.
[9] X. Zhang, Y. Wang, N. Zhang, D. Xu, H. Luo, B. Chen, G. Ben, Spectral–spatial fractal residual convolutional neural network with data balance augmentation for hyperspectral classi cation, IEEE Transactions on Geoscience and Remote Sensing 59 (2021) 10473–10487.
[10] T.-C. Pham, A. Doucet, C.-M. Luong, C.-T. Tran, V.-D. Hoang, Improving skin-disease classi cation based on customized loss function combined with balanced mini-batch logic and real-time image augmentation, IEEE Access 8 (2020) 150725–150737.
[11] D. Połap, M. Woźniak, A hybridization of distributed policy and heuristic augmentation for improving federated learning approach, Neural Networks 146 (2022) 130–140.
[12] A. Jaszcz, The impact of entropy weighting technique on mcdm-based rankings on patients using ambiguous medical data, in: International Conference on Information and Software Technologies, Springer, 2023, pp. 329–340.
[13] U. Nandi, A. Ghorai, M. M. Singh, C. Changdar, S. Bhakta, R. Kumar Pal, Indian sign language alphabet recognition system using cnn with di grad optimizer and stochastic pooling, Multimedia Tools and Applications 82 (2023) 9627–9648.
[14] Z. Wang, Z. Wang, C. Zeng, Y. Yu, X. Wan, High-quality image compressed sensing and reconstruction with multi-scale dilated convolutional neural network, Circuits, Systems, and Signal Processing 42 (2023) 1593–1616.
[15] X. Yi, E. Walia, P. Babyn, Generative adversarial network in medical imaging: A review, Medical Image Analysis 58 (2019) 101552. URL: https://www.sciencedirect.com/science/article/pii/S1361841518308430. doi:https://doi.org/10.1016/j.media.2019.101552.
[16] W. Nie, N. Narodytska, A. Patel, RelGAN: Relational generative adversarial networks for text generation, in: International Conference on Learning Representations, 2019. URL: https://openreview.net/forum?id=rJedV3R5tm.
[17] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, 2014. arXiv:1406.2661.
[18] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolu- tional generative adversarial networks, 2016. arXiv:1511.06434.
[19] Y. Hong, U. Hwang, J. Yoo, S. Yoon, How generative adversarial networks and their variants work: An overview, ACM Comput. Surv. 52 (2019). URL: https://doi.org/10.1145/3301282. doi:10.1145/3301282.
[20] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2017. arXiv:1412.6980.
[21] Z. Zhang, M. R. Sabuncu, Generalized cross entropy loss for training deep neural networks
with noisy labels, 2018. arXiv:1805.07836.
[22] S. Io e, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. arXiv:1502.03167.
[23] T. Miyato, T. Kataoka, M. Koyama, Y. Yoshida, Spectral normalization for generative adversarial networks, 2018. arXiv:1802.05957.
[24] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, Improved training of wasserstein gans, 2017. arXiv:1704.00028.
[25] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan, 2017. arXiv:1701.07875.

[26] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved tech- niques for training gans, 2016. arXiv:1606.03498.

[27] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of International Conference on Computer Vision (ICCV), 2015.

[28] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, Gans trained by a two time-scale update rule converge to a local nash equilibrium, in: I. Guyon,
U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_ les/paper/2017/ le/8a1d694707eb0fefe65871369074926d-Paper.pdf.

[29] S.-W. Park, J.-H. Huh, J.-C. Kim, Began v3: Avoiding mode collapse in gans using variational inference, Electronics 9 (2020). URL: https://www.mdpi.com/2079-9292/9/4/688. doi:10. 3390/electronics9040688.

[30] J. Wei, M. Liu, J. Luo, A. Zhu, J. Davis, Y. Liu, Duelgan: A duel between two discriminators stabilizes the gan training, in: S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, T. Hassner (Eds.), Computer Vision – ECCV 2022, Springer Nature Switzerland, Cham, 2022, pp. 290–317.

[31] N. Kodali, J. Abernethy, J. Hays, Z. Kira, On convergence and stability of gans, 2017. arXiv:1705.07215.

[32] M. Sabuhi, M. Zhou, C.-P. Bezemer, P. Musilek, Applications of generative adversarial networks in anomaly detection: A systematic literature review, IEEE Access 9 (2021) 161003–161029. doi:10.1109/ACCESS.2021.3131949.

[33] M. Tschaepe, Pragmatic ethics for generative adversarial networks: Coupling, cyborgs, and machine learning, Contemporary Pragmatism 18 (2021) 95 – 111. URL: https://brill.com/view/journals/copr/18/1/article-p95_95.xml. doi:10.1163/18758185-bja10005.