Experimental Evaluation of GAN-Based One-Class Anomaly Detection on Office Monitoring*

Ning Dong[1], Yusuke Hatae[1], Muhammad Fikko Fadjrimiratno[2], Tetsu Matsukawa[1][0000−0002−8841−6304], and Einoshin Suzuki[1][0000−0001−7743−6177]

[1] ISEE, Kyushu University, Fukuoka 819-0395, Japan
[2] SLS, Kyushu University, Fukuoka, 819-0395, Japan
dongning.ac@gmail.com, hatae.yusuke.042@s.kyushu-u.ac.jp, muhammadfikko@gmail.com, {matsukawa, suzuki}@inf.kyushu-u.ac.jp

Abstract. In this paper, we test two anomaly detection methods based on Generative Adversarial Networks (GAN) on office monitoring including humans. GAN-based methods, especially those equipped with encoders and decoders, have shown impressive results in detecting new anomalies from images. We have been working on human monitoring in office environments with autonomous mobile robots and are motivated to incorporate the impressive, recent progress of GAN-based methods. Lawson et al.'s work tackled a similar problem of anomalous detection on an indoor, patrol trajectory environment with their patrolbot with a GAN-based method, though crucial differences such as the absence of humans exist for our purpose. We test a variant of their method, which we call FA-GAN here, as well as the cutting-edge method of GANomaly on our own robotic dataset. Motivated to employ such a method for a turnable Video Camera Recorder (VCR) placed at a fixed point, we also test the two methods for another dataset. Our experimental evaluation and subsequent analyses revealed interesting tendencies of the two methods including the effect of a missing normal image for GANomaly and their dependencies on the anomaly threshold.

Keywords: One-Class Anomaly detection · Generative Adversarial Networks · Human monitoring.

1 Introduction

Monitoring an office environment, especially the humans inside, represents an interesting problem for intelligent systems from both scientific and industrial viewpoints. Detecting anomalies is one of the most fundamental and yet important subproblems, though collecting and even knowing such anomalies beforehand are at the same time laborious and difficult. One-class anomaly detection,

* A part of this work is supported by Grant-in-Aid for Scientific Research JP18H03290 from the Japan Society for the Promotion of Science (JSPS).

which takes only normal data in the training stage to detect anomalies in the test stage, solves these shortcomings. Recently Generative Adversarial Networks (GAN) [1], which are deep neural networks capable of learning the probabilistic distribution of the given, originally unlabeled, examples, have shown impressive results in detecting new anomalies from images. For instance, Lawson et al. report that the false positive rate of 4.72% achieved with their previous method [2] dropped to 0.42% with their GAN-based method in an anomaly detection problem by their patrolbot [3].

We have been working on office monitoring including humans inside with autonomous mobile robots, e.g., skeleton clustering [4], facial expression clustering [5], and fatigue detection [6]. Recently our interests are focused on one-class anomaly detection [7, 8], though these works adopt non-GAN-based approaches. Motivated to incorporate the impressive, recent progress of GAN-based methods, we test two most relevant methods, which we explain in Section 2, on our robotic and VCR datasets.

2 Related Work

Recently GAN-based one-class anomaly detection has attracted considerable attention of the machine learning community. Schlegl et al. [9] proposed AnoGAN to detect anomalies on Optical Coherence Tomography (OCT) data. They assumed that the trained latent space represents the true distribution of the training data. However, their method is time-consuming in finding a latent vector that corresponds to an image that is visually most similar to a given query image [10] in the test stage. To cope with the shortcoming that the parameters need to be updated in the test stage of AnoGAN, Zenati et al. proposed an anomaly detection method [11] that is efficient at test time by leveraging BiGAN [12], which simultaneously learns an encoder with a decoder and a discriminator during training. It can avoid the computationally expensive process during the test stage. Sabokrou et al. proposed a framework for one-class novelty detection which consists of a reconstructor and a discriminator [13]. They added noise to the original normal examples to train the reconstructor network to make it more robust and employed the discriminator as a detector to classify whether the input is abnormal. A deep generative model trained on a single class cannot generate examples belonging to other classes. Perera et al. focused on this problem and proposed OCGAN for novelty detection [14]. They restricted the boundary of the latent space and used a latent discriminator and a visual discriminator to ensure the images generated from any latent vector belong to the same class. Different from the traditional GAN-based encoder-decoder approaches, Akcay et al. [10] employed an encoder-decoder-encoder structure to capture the two latent vectors which show significant differences with an abnormal example. The added encoder aids learning the data distribution for the normal examples. We adopt their GANomaly [10] for evaluation in our experiments as we consider it a relevant cutting-edge method.

We have witnessed a number of developments and applications of autonomous mobile robots in anomaly detection. Chakravarty et al. [15] used modified sparse and dense stereo algorithms to detect anomalies that were never shown during the training stage for a patrolbot. However, light intensity has a great impact on their detection accuracy. Lawson et al. [2] proposed a method with clustering normal features from CNNs in a fixed path with a patrolbot. Abnormal features would produce large distances to these clusters. Later, they extended their work in [3] to find anomalies with an autoencoder-decoder GAN, which achieved much better performance as we stated in the previous section. We extend their method by replacing their autoencoder with a more sophisticated encoder [16], and call the extended method FA-GAN in this paper. We also use FA-GAN in our experiments.

The robotic data and the VCR data we use in our experiments have been introduced in our previous work on detecting anomalous image regions with deep captioning [8]. In the work, our anomaly detector represents each salient region with its image, caption, and position features and uses an incremental clustering method [17] to detect anomalies with these features. The point is to exploit another dataset used in training a combination of Convolutional Neural Network (CNN) [18] and Long Short-Term Memory (LSTM) [19, 20] through deep captioning [21]. We will explain the details of our datasets in Section 4.1. Since the method [8] uses deep captioning and conducts evaluation on image region level, we leave its comparison with GANomaly and FA-GAN for our future work.

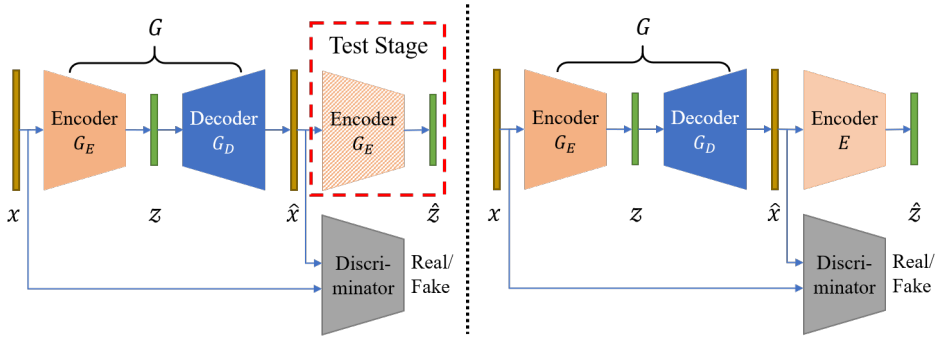
3 Tested Methods

3.1 FA-GAN

Lawson et al. use the DCGAN approach [22], adopting an architecture that is similar to what was proposed in Context Encoders [16]. Unlike DCGAN, they use an autoencoder-style with a bottleneck size of 4096. Considering the more complex nature of our office monitoring problem, we replace their autoencoder with a more sophisticated encoder in [16], as we explained in the previous Section. The resulting FA-GAN has an encoder-decoder architecture, which is shown in Fig. 1 (a). The encoder G_E is composed of convolution layers and batch-normalization layers with LeakyReLU activation function. The decoder G_D adopts the structure of DCGAN [22] with deconvolutional layers to generate images from a latent vector. The discriminator D has a similar structure to G_E and uses Sigmoid function to output whether the input is real or generated.

Given a training set X, which consists of N normal images, $X = \{x_1, \ldots, x_N\}$, the generator G first reads an input image x as the input to its encoder G_E to downscale x by compressing it to a latent vector $z = G_E(x)$. Then the decoder G_D tries to reconstruct z to an image $\hat{x}$. To maximize the capability of reconstructing an image, FA-GAN uses the adversarial loss [1] shown in Eq. 1.

$$Loss\_FA = \mathbb{E}_{x \sim p_x} [\log D(x))] + \mathbb{E}_{x \sim p_x} [\log(1 - D(\hat{x}))] \quad (1)$$



(a) Architecture of FA-GAN (b) Architecture of GANomaly [10]

Fig. 1. Architectures of FA-GAN (the left one) and GANomaly (the right one)

In the test stage, the generator G produces the latent vectors of z and $\hat{z}$ from the original input image and its corresponding generated image through the encoders in the network, respectively. Note that the encoder in the dashed rectangle is not used during the training stage. It is just a copy of the former one after training. Finally, a test example can be determined as normal or abnormal by comparing its abnormal degree A with a user-given threshold ϕ, where $A = \|z - \hat{z}\|_2$ is computed as the distance between the two latent vectors z and $\hat{z}$. If $A > \phi$ then the image is predicted as anomalous, otherwise normal. Since the network is only trained with normal data, the generator cannot reconstruct an anomalous image well, which means there will be a large difference between the two latent vectors.

3.2 GANomaly

GANomaly [10] has a similar architecture to FA-GAN. It also uses an encoder-decoder generator in which the latent vector z of the original input is obtained by $G_E(x) = z$. The difference is that there is one more encoder E to produce $\hat{z}$ after the generated image as shown in Fig. 1 (b). The parameters of the additional encoder E are also optimized during the training stage, in which the distance between z and $\hat{z}$ is considered as a loss. After the second encoder E, the generated image is downscaled to a latent vector $\hat{z} = E(\hat{x})$, which has the same size with z.

The loss function $Loss\_GANomaly$ of GANomaly consists of an adversarial loss L_{adv}, a contextual loss L_{con}, and an encoder loss L_{enc} as in Eq. 2, where w_{adv}, w_{con}, and w_{enc} represent hyper-parameters. In Eq. 3, $f(x)$ is a function that outputs an intermediate layer of the discriminator given x [10]. To predict whether a test example is abnormal, GANomaly uses the same computation flow as FA-GAN, which is to compute the distance between z and $\hat{z}$, except $\hat{z}$ is produced by the additional encoder E, and not G_E.

$$Loss\_GANomaly = w_{adv}L_{adv} + w_{con}L_{con} + w_{enc}L_{enc} \tag{2}$$

Fig. 2. Examples in the robotic dataset (the left two) and the VCR dataset (the right two). From the left, the classes are normal, abnormal, normal, and abnormal.

Table 1. Distributions of the two datasets

	Robotic dataset		VCR dataset	
	training	test	training	test
Normal	4768	343	16800	684
Abnormal	0	15	0	31

$$L_{adv} = \mathbb{E}_{x \sim p_x} \left\| f(x) - \mathbb{E}_{x \sim p_x} f(G(x)) \right\|_2 \tag{3}$$

$$L_{con} = \mathbb{E}_{x \sim p_x} \left\| x - G(x) \right\|_1 \tag{4}$$

$$L_{enc} = \mathbb{E}_{x \sim p_x} \left\| z - \hat{z} \right\|_2 \tag{5}$$

4 Experimental Evaluation

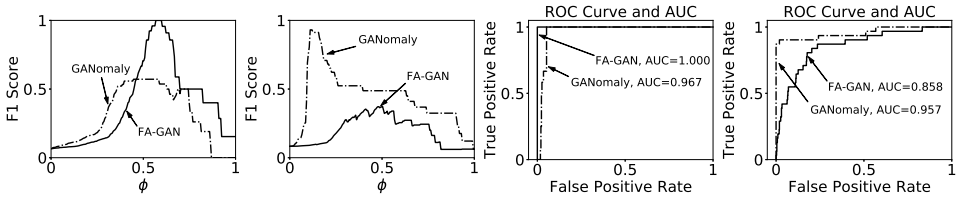
4.1 Experimental Setup

To evaluate the two GAN-based methods on anomaly detection, we conduct experiments with our robotic and VCR datasets, which we introduced in [8]. Fig. 2 shows several examples. The robotic dataset is taken in a room by our TurtleBot2 with Kobuki, which is equipped with a Kinect v2. It contains frequent scene changes as the robot moved in an office. The VCR dataset is taken with a VCR placed on a spandrel wall. It has only a few scene changes as the VCR was put on a fixed point and a human occasionally changed its angle. Table 1 shows the distributions of the datasets.

We install GANomaly [10] and implement FA-GAN [3] in PyTorch (v1.3.1 with Python 3.6.9). The networks are optimized by Adam [23] with an initial learning rate of 0.0001, β_1=0.5, and β_2=0.999. We set the batch size to 32 and each network is trained for 70 epochs. The hyper-parameters are set as $w_{adv} = 1$, $w_{con} = 50$, and $w_{enc} = 1$. The size of latent vector is set to 4096 throughout the experiments. We normalize all the anomaly scores obtained in the test set to the range of [0,1].

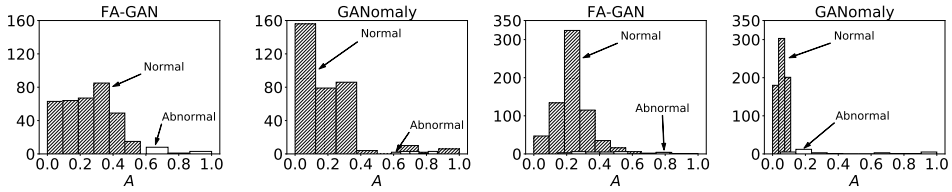
4.2 Results

In this section, we first analyze the dependency of the performance in F1 score on ϕ and then investigate the reasons behind the mistakes by the two methods. The left two plots in Fig. 3 show the results of the dependency. We see that FA-GAN outperforms GANomaly in the robotic dataset but loses to it in the VCR



(a) Robotic dataset (b) VCR dataset (c) Robotic dataset (d) VCR dataset

Fig. 3. F1 scores in terms of threshold ϕ (left two plots) and the ROC curve and AUC (right two plots)



(a) FA-GAN (b) GANomaly (c) FA-GAN (d) GANomaly

Fig. 4. Histogram on the anomaly scores for the test data on the robotic dataset (the left two plots) and the VCR dataset (the right two plots)

dataset. From Fig. 4 we can see the reasons. In (a), there is a clear boundary to distinguish the normal and abnormal examples with FA-GAN on the robotic dataset. GANomaly succeeds in concentrating the anomaly scores of all normal examples in a small interval on the VCR dataset in (d). The right two plots in Fig. 3 show the results in ROC curve and AUC.

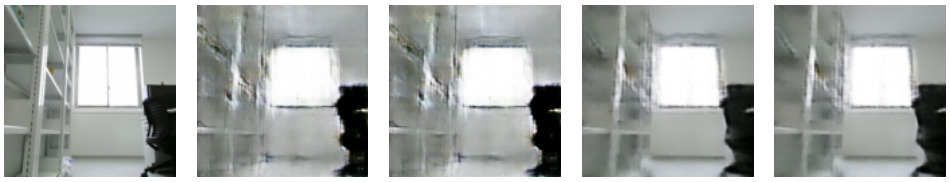
We assume that setting ϕ to its best value is possible as long as the office environment does not change drastically. Based on this assumption we conduct our investigation on the best cases in terms of the value of ϕ. We focus on mistakes committed by the two methods, which are summarized in Table 2. In the Table, FN and FP represent the number of false negatives and the number of false positive, respectively. "Same examples" is the number of images wrongly detected by both methods.

Table 2. Statistics of the wrongly detected examples

	Robotic dataset		VCR dataset	
	FN	FP	FN	FP
FA-GAN	0	0	19	21
GANomaly	0	18	4	0
Same examples	0	0	2	0

(a) (b) (c) (d) (e)

Fig. 5. One of the FP examples with GANomaly in the robotic dataset. (a) Original input. (b) Generated image with z by FA-GAN. (c) Generated image with $\hat{z}$ by FA-GAN. (d) Generated image with z by GANomaly. (e) Generated image with $\hat{z}$ by GANomaly.



(a) (b) (c) (d) (e)

Fig. 6. One of the correctly detected examples by the two methods. See the captions of Fig. 5 for (a)–(e).

On the robotic dataset, we see from Table 2 that FA-GAN made no mistake while GANomaly 18 false positives. Since the 18 examples look all similar, we pick one of them and show it in Fig. 5 (a). The anomaly scores A are 0.968 in GANomamly and 0.506 in FA-GAN. We also show the generated images by z and $\hat{z}$ with both methods in Fig. 5 (b)–(e) and see that the red rectangle region in Fig. 5 (e) accounts for the large A in GANomaly.

We also pick an example which is similar to Fig. 5 (a) but correctly classified by both methods and show it in Fig. 6 (a). The anomaly scores are 0.266 and 0.106 for FA-GAN and GANomaly, respectively. We see from Fig. 6 (b), (c) and (d), (e) that z and $\hat{z}$ are similar in both methods.

Further inspection revealed that no similar image to Fig. 5 (a) exists in the training set while all other images in the test set have similar images in the training set. These analyses show the higher generalization capability of FA-GAN to GANomaly for this dataset. To justify our claim, we added random noise to the 18 FP examples to generate 72 additional examples and added them in the training set. We trained GANomaly on this dataset and obtained a perfect result, i.e., no mistake committed and hence AUC = 1.0.

On the VCR dataset, we see from Table 2 that there are only 4 FN examples with GANomaly but 19 FN examples and 21 FP examples with FA-GAN. This dataset was recorded in one corner by the VCR, so the intuitive complexity of the dataset is relatively reduced compared with the first dataset. Moreover, the

(a) (b) (c) (d) (e)

Fig. 7. One of the FN examples with FA-GAN. See the captions of Fig. 5 for (a)–(e).

(a) (b) (c) (d) (e)

Fig. 8. One of the FN examples with the two methods. See the captions of Fig. 5 for (a)–(e).

training set, which consists of 16800 examples, is large, so the second encoder of GANomaly can well learn the distributions of normal feature in this dataset. Fig. 7 (a) shows an FN example with FA-GAN, which is correctly detected by GANomaly. Hiding under a table[3] is considered as an anomaly because nobody does it in the training set. The subsequent images in (b)–(e) are generated images with z and $\hat{z}$ by the two methods. Unlike in Fig. 5, the anomaly scores A in both methods are small, which is justified by the small differences between (b) and (c) as well as (d) and (e). The different results can be explained by the best values of ϕ. Fig. 4 shows the histograms on the anomaly scores A in the both methods. It can be seen from Fig. 4 (b) that GANomaly concentrates the anomaly scores of all normal examples between 0 and 0.12. Note that FA-GAN and GANomaly adopted a relatively large and small values for ϕ, which results in a false negative and a true positive, respectively.

We also show in Fig. 8 one of the FN examples which were wrongly classified by the two methods. Taking a selfie is considered as an anomaly because nobody does it in the training set. Note that the difference between the images generated by z and $\hat{z}$ is small. Fig. 9 shows an FP example with FA-GAN. We see that there is a large difference between (b) and (c), especially on the middle part. However, the last two generated images by GANomaly, which achieved an anomaly score of 0.051, are similar.

[3] Schools in Japan teach students to take this action under strong shakes during an earthquake.

(a) (b) (c) (d) (e)

Fig. 9. One of the FP examples with FA-GAN. See the captions of Fig. 5 for (a)–(e).

From the experiments above, we see that in overall the two methods show good performance on the two datasets. We conclude that the two GAN-based methods show their ability to solve the problem of anomaly detection on human monitoring. However, the drawback of the methods is also obvious. For some minor anomalies, e.g., the selfie in Fig. 8 (a), the latent vector z after the first encoder does not reflect them, which makes z and $\hat{z}$ be quite similar. It results in a small anomaly score for such examples and thus these abnormal examples will be predicted as normal. We can see these results in Fig. 8.

5 Conclusion

We applied two kinds of GAN-based methods, which are FA-GAN and GANomaly, to the datasets collected by our autonomous robot and with a VCR, so that the anomalies can be detected without any supervision. The results show that FA-GAN performs better on the robotic dataset while GANomaly performs better on the VCR dataset, possibly due to the different frequencies of the scene changes. We analyzed the reason behind the dependency of the performance in F1 score on ϕ through the histograms on the anomaly scores. Also, the methods occasionally make wrong detection with minor anomalies in images due to the scarcity of the scene in the training set or the loss of information in the latent vectors. In general, the two GAN-based methods with encoder-decoder architectures should perform well on our autonomous robot and VCR for human monitoring.

Our future work will take image captions into account to improve the performance of anomaly detection. We think that the captions as weak labels can provide additional useful information on anomaly detection, since we already made some progress with a non-GAN-based approach [8].

References

1. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative Adversarial Nets. In: Proc. NIPS. pp. 2672–2680 (2014)
2. Lawson, W., Hiatt, L., K.Sullivan: Detecting Anomalous Objects on Mobile Platforms. In: Proc. CVPR Workshop (2016)

3. Lawson, W., Hiatt, L., Sullivan, K.: Finding Anomalies with Generative Adversarial Networks for a Patrolbot. In: Proc. CVPR Workshop (2017)
4. Deguchi, Y., Takayama, D., Takano, S., Scuturici, V.M., Petit, J.M., Suzuki, E.: Skeleton Clustering by Multi-Robot Monitoring for Fall Risk Discovery. Journal of Intelligent Information Systems **48**(1), 75–115 (2017)
5. Kondo, R., Deguchi, Y., Suzuki, E.: Developing a Face Monitoring Robot for a Deskworker. In: Ambient Intelligence. LNCS, vol. 8850, pp. 226–241. Springer-Verlag (2014)
6. Deguchi, Y., Suzuki, E.: Hidden Fatigue Detection for a Desk Worker Using Clustering of Successive Tasks. In: Ambient Intelligence. LNCS, vol. 9425, pp. 268–283. Springer-Verlag (2015)
7. Fujita, H., Matsukawa, T., Suzuki, E.: Detecting Outliers with One-Class Selective Transfer Machine. Knowledge and Information Systems (accepted for publication)
8. Hatae, Y., Yang, Q., Fadjrimiratno, M.F., Li, Y., Matsukawa, T., Suzuki, E.: Detecting anomalous regions from an image based on deep captioning. In: Proc. VISIGRAPP, Subvolume for VISAPP, vol. 5. pp. 326–335 (2020)
9. Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. In: Proc. International Conference on Information Processing in Medical Imaging (2017)
10. Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: GANomaly: Semi-Supervised Anomaly Detection via Adversarial Training. In: Proc. ACCV. pp. 622–637 (2018)
11. Zenati, H., Foo, C.S., Lecouat, B., Manek, G., Chandrasekhar, V.R.: Efficient GAN-Based Anomaly Detection. arXiv preprint arXiv:1802.06222 (2018)
12. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial Feature Learning. arXiv preprint arXiv:1605.09782 (2016)
13. Sabokrou, M., Khalooei, M., Fathy, M., Adeli, E.: Adversarially Learned One-Class Classifier for Novelty Detection. In: Proc. CVPR. pp. 3379–3388 (2018)
14. Perera, P., Nallapati, R., Xiang, B.: OCGAN: One-Class Novelty Detection Using GANs with Constrained Latent Representations. In: Proc. CVPR. pp. 2898–2906 (2019)
15. Chakravarty, P., Zhang, A.M., Jarvis, R., Kleeman, L.: Anomaly Detection and Tracking for a Patrolling Robot. In: Proc. Australasian Conference on Robotics and Automation (ACRA) (2007)
16. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context Encoders: Feature Learning by Inpainting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2536–2544 (2016)
17. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: A New Data Clustering Algorithm and its Applications. Data Min. Knowl. Discov. **1**(2), 141–182 (1997)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. In: Proc. NIPS. pp. 1106–1114 (2012)
19. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Compututation **9**(8), 1735–1780 (1997)
20. Zaremba, W., Sutskever, I.: Learning to Execute. CoRR **abs/1410.4615** (2014)
21. Johnson, J., Karpathy, A., Fei-Fei, L.: DenseCap: Fully Convolutional Localization Letworks for Dense Captioning. In: Proc. CVPR. pp. 4565–4574 (2016)
22. Radford, A., Metz, L., Chintala, S.: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv preprint arXiv:1511.06434 (2015)
23. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980 (2014)