Conditional Image Synthesis Using Stacked Auxiliary Classifier Generative Adversarial Networks

Zhongwei Yao[1], Hao Dong[2], Fangde Liu[2], and Yike Guo[2](✉)

[1] Department of Computing, Imperial College London, London, UK
yaozhongwei0131@163.com
[2] Data Science Institute, Imperial College London, London, UK
{hao.dong11,fangde.liu,y.guo}@imperial.ac.uk

Abstract. Synthesizing photo-realistic images has been a long-standing challenge in image processing and could provide crucial approaches for dataset augmentation and balancing. Traditional methods have trouble in dealing with the rich and complicated structural information of objects resulting from the variations in colors, poses, textures and illumination. Recent advancement in Deep Learning techniques presents a new perspective to this task. The aim of our paper is to apply state-of-the-art generative models to synthesize diverse and realistic high-resolution images. Extensive experiments have been conducted on celebA dataset, a large-scale face attributes dataset with more than 200 thousand celebrity images, each with 40 attribute labels. Enlightened by existing structures, we present stacked Auxiliary Classifier Generative Adversarial Networks (Stack-ACGAN) for image synthesis given conditioning labels, which generates low resolution images (e.g. 64 × 64) that sketch basic shapes and colors in Stage-I and high resolution images (e.g. 256 × 256) with plausible details in Stage-II. Inception scores and Multi-Scale Structural Similarity (MS-SSIM) are computed for evaluation of the synthesized images. Both quantitative and qualitative analysis prove the proposed model is capable of generating diverse and realistic images.

Keywords: High-resolution image synthesis · Deep learning
Generative adversarial networks

1 Introduction

Generating photo-realistic images in high resolution is a challenging task, which has enormous applications in scenarios including datasets augmentation and computer-aided design and manufacturing, etc. However, even the most advanced generative models fail to generate plausible images with conditional information, especially in high resolution, due to the fact that objective data space is multi-modal. In other words, there are many possible images that correctly match specific conditional labels.

K. Arai et al. (Eds.): FICC 2018, AISC 887, pp. 423–433, 2019.
https://doi.org/10.1007/978-3-030-03405-4_29

Recently, Generative Adversarial Network (GAN) [1] attracts great attention in the field of image synthesis and a large number of GAN variants ([2]) are proposed to be proficient of generating shaper images. However, due to the unstable training of GAN, most existing GAN networks generate relatively low-resolution images (e.g. 64×64) and the details and object parts added by super-resolution approaches are limited so that large detects in the low-resolution images can hardly be rectified. Therefore, synthesizing high-resolution images with photo-realistic details remain to be a pending challenge.

To solve this problem, we propose Stacked Auxiliary Classifier Generative Adversarial Network (Stack-ACGAN) which divides the synthesis process into two stages. Instead of directly generating high-resolution images, the generator in Stage-I ACGAN produces a 64×64 facial image (Fig. 1(a)) conditioned on given attribute labels and a random noise vector. Conditional labels constrain the image to match corresponding attributes while the random vector encodes other features except for those specified in labels. We observed that the low-resolution images look coarse and suffer from defects such as shape distortion and absence of details. On top of the Stage-I ACGAN, we build Stage-II ACGAN to generate high-resolution images (Fig. 1(b)) given the drafts from Stage-I ACGAN and conditional attribute labels. The aim of Stage-II ACGAN is correcting defects and generating more realistic details. It is much simpler than generating high-resolution images from scratch since Stage-II ACGAN only need to deal with the attributes omitted in Stage-I and correct defects.

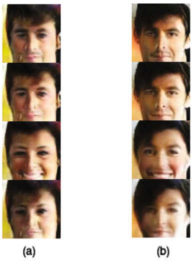

Fig. 1. The comparison between results in Stage-I and Stage-II. The attributes given to images from top to bottom are "man with mustache", "man", "woman smiling" and "woman not smiling", respectively. (a) Given conditional attribute labels, Stage-I ACGAN sketches primitive outlines and colors of human faces, yielding low resolution images. (b) Stage-II ACGAN takes Stage-I results and conditional attribute labels as inputs, and generates high resolution images with photo-realistic details.

The major contribution of our paper is proposing the Stack-ACGAN that is capable of generating realistic high-resolution images conditioned on attribute labels. In comparison with existing generative models, our Stack-ACGAN succeed in synthesizing images with compelling details, proved by an inception score [3] of 1.56 while the mean inception score of ground-truth images in celebA dataset [4] is 1.84. In addition, we evaluate the diversity of samples generated using the Multi-Scale Structural Similarity (MS-SSIM) indices [5]. The MS-SSIM of samples from Stack-ACGAN is 0.65 while the training data has a MS-SSIM of 0.67, indicating that our model generates images as varied as real datasets.

The paper is organized in the following order. Firstly, existing methods for image synthesis are presented and compared in the background section. Secondly, we introduce the basic knowledge about Generative Adversarial Networks. Then we propose our Stack-ACGAN for conditional image synthesis, explaining how the network works and elaborating the training details. We also show the facial images generated in both stages, together with both quantitative and qualitative analysis of experimental results. At last, we summarize the whole paper and add some follow-up works that might be worth trying in the future for better results.

2 Background

Generative models for images synthesis have recently received significant attention, especially in the last decade. The main objective of image synthesis is to synthesize desired images, e.g. photo-realistic, artistic, or high-resolution pictures via given constraints in some semantic domains. These well-studied generative models fall into two categories: parametric and non-parametric.

Non-parametric approaches, whose major idea is searching for matched natural images in existing database, have been widely used in texture synthesis [2], super-resolution [6] and scene completion [7].

Although extensively studied, parametric models haven't achieved much success in generating plausible images until recently. With the emergence of deep learning techniques in the last few years, remarkable progress has been made. Reed et al. [8] proposed to solve visual analogies by learning to map images to neural embeddings and decode the representation after vector arithmetics. Compared to this deterministic approach, Variational Auto-encoder (VAE) [9] was presented by formulating the problem as a probabilistic graphical model whose aim is to fit an approximate inference model to the intractable posterior. Another method called the Deep Recurrent Attentive Writer (DRAW) model was proposed by Gregor et al. [10], which focuses on generating house numbers images by combining recurrent VAE and attention mechanism. As a typical instance of autoregressive models, Pixel Recurrent Neural Networks (PixelRNN) [11] utilizes the strong learning ability of deep neural networks to model the conditional distribution in the pixel space, also yielding appealing results.

Recently, Generative Adversarial Network (GAN) [1] attracts great attention in the field of image synthesis and a large number of GAN variants are proved to be capable of generating shaper images. Build upon these models, image synthesis with conditional information has also been explored. Generally, the conditional information appears in the form of attributes, class labels or text descriptions ([11–15]). Research has also been conducted on generating images conditioned on images, such as photo-editing [14] and super-resolution [16,17]. However, most methods generate relatively low-resolution images(e.g. 64 × 64) and the details and object parts added by super-resolution approaches are limited so that large detects in the low-resolution images can hardly be rectified.

3 Preliminaries of Generative Adversarial Networks

Generative Adversarial Networks (GAN) [1], whose training is game-theoretic, provide an attractive and promising alternative for modeling complex data distribution. The architecture of GAN consists of two parts, which are generator and discriminator, respectively.

As a generative model, GAN interprets data as samples from a high-dimensional probabilistic distribution. By setting up a game between two neural networks, the generator (e.g. deconvolutional neural network) is trained to generate data to fool the discriminator while the discriminator (e.g. convolutional neural network) is trained to tell real data from fake (generated) data. The adversarial learning method is applied so that the generator and discriminator can compete, encouraging each other to learn to perform better on its own target while eventually improve the whole network.

Mathematically, given real samples x and noise z that is randomly sampled from normal Gaussian distribution, to learn the generator's distribution p_g from real training data distribution p_{data}, we first denote the prior of noise z as $p_z(z)$, then refer to the mapping from noise to data space by $G(z;\theta_g)$, where function G is modeled by a multilayer neural network with parameters θ_g. Likewise, we define $D(x = real;\theta_d)$ as the probability of input x to be classified as real data, where function D corresponds to discriminator with parameters θ_d.

During training, the goal for discriminator D is maximizing the probabilities of classifying real samples x as real and generated data $G(z;\theta_g)$ as fake.

$$\max_{\theta_d} E_{x\sim p_{data}}[log(D(x;\theta_d))]+ \tag{1}$$

$$E_{z\sim p_z}[log(1 - D(G(z;\theta_g);\theta_d))] \tag{2}$$

The objective for generator G is minimizing the probabilities of classifying generated data $G(z;\theta_g)$ as fake.

$$\min_{\theta_g} E_{z\sim p_z}[log(1 - D(G(z;\theta_g);\theta_d))] \tag{3}$$

Therefore, the objective function for GAN is to combine them together.

$$\min_{\theta_g}\max_{\theta_d} E_{x\sim p_{data}}[log(D(x;\theta_d))]+ \tag{4}$$

$$E_{z\sim p_z}[log(1 - D(G(z;\theta_g);\theta_d))] \tag{5}$$

4 Stacked Auxiliary Classifier Generative Adversarial Networks

In order to synthesize photo-realistic facial images in high resolution according to conditional attribute labels, we propose stacked Auxiliary Classifier Generative Adversarial Network (Stack-ACGAN) which stacks one ACGAN on the top of the other. The network architecture is shown in Fig. 2.

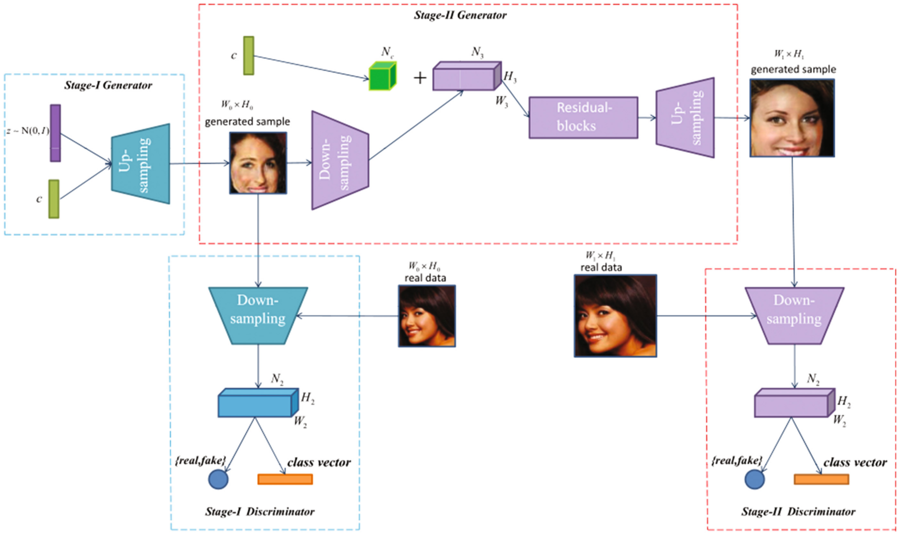

Fig. 2. The architecture of the proposed Stack-ACGAN. Stage-I ACGAN draws a low-resolution image conditioned on attributes labels and a random noise vector. Stage-II ACGAN takes and rectifies the results from Stage-I and synthesize high-resolution images with more realistic details by relearning on the conditional attribute labels.

4.1 Stage-I ACGAN

Instead of directly generating high-resolution images, the generator of Stage-I ACGAN produces a 64 × 64 facial image conditioned on given labels and a random noise vector. Conditional labels constrain the image to match corresponding attributes while the random vector encodes other features except for those specified in labels. For instance, assume we intend to generate images with attributes of "Male" and "Smiling", the result show great diversity. It might be a smiling man in hat or a smiling man with blond wavy hair or even a smiling man in sun glasses. The low-resolution image from stage-I sketches the basic outline and expression of human faces together with some defects and blurry details.

Model Architecture. As shown in Fig. 2, for the generator, a N_z dimensional random noise vector z is sampled from normal Gaussian distribution (e.g. $z \sim N(0, I)$) and then concatenated with conditional labels c before being sent into the Stage-I generator. After a series of upsampling blocks, an image I_0(e.g. $G_0(z, c)$) of size $W_0 \times H_0$ is generated.

For the discriminator, both the real images I_{real} and synthesized images I_0 is processed by a series of downsampling blocks until the spatial size of feature maps becomes $W_2 \times H_2 \times N_2$. Then feature maps are fed to two branches of network. In one branch, a fully-connected layer uses the features to perform binary classification to tell whether the input image is real or not. In the other branch, an auxiliary classifier is trained to predict which class the image belongs to.

Model Training. Stage-I ACGAN trains discriminator D_0 and generator G_0 by alternatively maximizing $\mathcal{L}_{D_0}$ and $\mathcal{L}_{G_0}$. e input image is real or not. In the other branch, an auxiliary classifier is trained to

$$\begin{aligned} L_S = & E_{I_{real}, c \sim p_{data}}[log(D_0(I_{real} = real; \theta_d))] \\ & + E_{z \sim p_z, c \sim p_{data}}[log(D_0(G_0(z, c; \theta_g)) = fake; \theta_d)] \end{aligned} \tag{6}$$

$$\begin{aligned} L_C = & E_{I_{real}, c \sim p_{data}}[log(D_0(I_{real} = c; \theta_d))] \\ & + E_{z \sim p_z, c \sim p_{data}}[log(D_0(G_0(z, c; \theta_g) = c; \theta_d))] \end{aligned} \tag{7}$$

$$\max_{\theta_d} \mathcal{L}_{D_0} = L_C + L_S \tag{8}$$

$$\max_{\theta_g} \mathcal{L}_{G_0} = L_C - L_S \tag{9}$$

4.2 Stage-II ACGAN

Speaking from experience, low-resolution samples synthesized by Stage-I ACGAN fail to show enough compelling details and may be distorted. We suggest that visual information is omitted more or less when we sample random noise vectors as encoded representations of natural images. To regain the features lost in the first stage, we build a Stage-II ACGAN upon the existing framework.

On the basis of Stage-I outputs, Stage-II ACGAN attempts to correct defects and capture omitted information by relearning the conditional labels, yielding $W_1 \times H_1$high-resolution (e.g. 256 × 256) images with more concrete details.

Model Architecture: For the generator, the input is low-resolution images I_0 generated in Stage-I together with conditional labels c. The images are firstly down-sampled to get feature maps with spatial size of $W_3 \times H_3 \times N_3$. In the mean time, the attribute labels adopted in stage-II is transformed to a tensor by spatially replication to $W_3 \times H_3 \times N_c$. The feature maps and label tensor are concatenated along the channel axis before a 1×1 convolutional layer joints the information of image and label together. The joint representation encourages the Stage-II generator to extract previously ignored features for the purpose of providing vivid details and correcting defects. Before processed by a series of upsampling blocks, the joint representation goes through several residual blocks [18] which enriches the level of features by adding more stacked layers. As a consequence, the generator in Stage-II ACGAN is much deeper than that in Stage-I, yielding plausible images in high-resolution.

For the discriminator, the architecture is very similar to that of Stage-I ACGAN except that more convolutional layers is used in downsampling blocks since the input size is larger.

Model Training: When training Stage-II ACGAN, Stage-I ACGAN is fixed. Similarly, Stage-II ACGAN trains discriminator D_0 and generator G_0 by alternatively maximizing $\mathcal{L}_{D_0}$ and $\mathcal{L}_{G_0}$. The objective function is defined as follows. What is different from Stage-I is that L1-norm is applied in $\mathcal{L}_{G_0}$ to force structural coherence between high-resolution output I_1(e.g. $G_1(I_0, c)$) and low-resolution input I_0. We adopt L1-norm rather L2-norm in consideration of preventing blurry results.

$$\begin{aligned} L_S = & E_{I_{real}, c \sim p_{data}}[log(D_1(I_{real} = real; \theta_d))] \\ & + E_{I_0 \sim G_0(z,c), c \sim p_{data}}[log(D_1(G_1(I_0, c; \theta_g)) = fake; \theta_d)] \end{aligned} \tag{10}$$

$$\begin{aligned} L_C = & E_{I_{real}, c \sim p_{data}}[log(D_1(I_{real} = c; \theta_d))] \\ & + E_{I_0 \sim G_0(z,c), c \sim p_{data}}[log(D_1(G_1(G_0(z, c), c; \theta_g) = c; \theta_d))] \end{aligned} \tag{11}$$

$$L_I = -E_{I_{real}, c \sim p_{data}, I_0 \sim G_0(z,c)} ||I_{real} - I_0||_1 \tag{12}$$

$$\max_{\theta_d} \mathcal{L}_{D_0} = L_C + L_S \tag{13}$$

$$\max_{\theta_g} \mathcal{L}_{G_0} = L_C - L_S + L_I \tag{14}$$

4.3 Experimental Details

(1) Stage-I ACGAN: The upsampling blocks are made up of 5 × 5 deconvolutions with stride 2. Batch normalization [19] and ReLU activation [20] are used after every deconvolution except the last one. The down-sampling blocks consist of 5 × 5 convolutions with stride 2 nd Leaky ReLU activation. Batch normalization are applied in every layer except the last one. Two fully-connected layers output the probabilities of being real and the probabilities of matching each attribute, respectively.

In correspondence to the terminology in Fig. 2, $N_c = 100, W_0 = H_0 = 64$, $W_2 = H_2 = 4, N_2 = 512$. For training, we alternatively train D_0 and G_0 for 50 epochs. In each step, the generator updates twice while the discriminator updates once to make sure they both are in the same pace. All networks are trained using ADAM solver with batch size 64 and an initial learning rate of 0.0002. The learning rate is decayed to half of its previous value every 20 epochs.
(2) Stage-II ACGAN: The difference between Stage-I and Stage-II networks lies in three major parts. In the first place, the generator in Stage-II encodes the input images by a series of down-sampling blocks, yielding latent representations of features. In the second place, 16 residual blocks are applied to increase the depth of network with the aim of extracting more detailed features. In the third place, the discriminator in Stage-II perform dimensionality reduction by add 1 × 1 convolutional layers. A bottleneck residual blocks is also used to reduce the parameters.

By default,$W_1 = H_1 = 256, W_3 = H_3 = 16$, $N_3 = 256$. During training, we fix the pretrained Stage-I ACGAN which generates input images for Stage-II ACGAN. The discriminator D_1 and generator G_1 are alternatively optimized for 10 epochs. All networks are trained using ADAM solver with batch size 64 and an initial learning rate of 0.0002, which is later decayed by half every 5 epochs.

4.4 Results and Analysis

We first demonstrate the samples generated from stage-I ACGAN with specific attributes in Fig. 3.

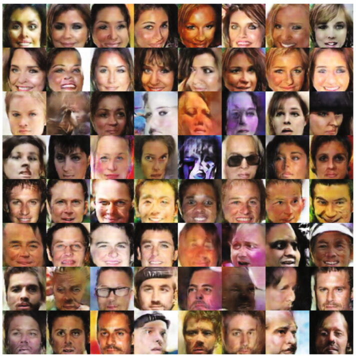

Fig. 3. Random samples generated by stage-I ACGAN. Labels for the first two rows are "Female", "Smiling". Labels for the second two rows are "Female", "Not smiling". Labels for the third two rows are "Male", "Smiling". Labels for the last two rows are "Male", "Not smiling"

We can see that nearly all the images succeed in matching the corresponding attributes suggested by conditional labels. However, similar to the samples generated by DCGAN, the synthesized images in Stage-I suffer severe distortion sometimes (e.g. image in row 3, column 2) and only capture primitive shapes and colors of human faces. In conclusion, Stage-I ACGAN fails to generate high-resolution and high-quality images but manages to generate images with specific attributes.

Next, we present the refined samples generated from Stage-II ACGAN in Fig. 4.

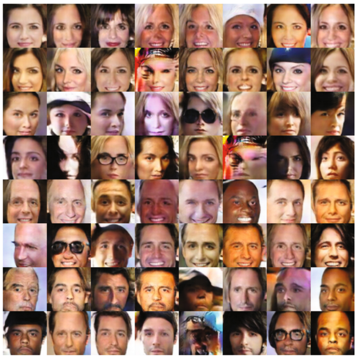

Fig. 4. Refined samples generated by stage-II ACGAN on the basis of results from Stage-I ACGAN. To be noticed, the generated 256 × 256 images are resized to 64 × 64 for evident comparison with images from Stage-I.

We observe that the images from Stage-II outputs match all the conditional attribute labels. Meanwhile, the results are in 4× higher resolution than those in Stage-I, reflecting more convincing details. For instance, the facial image in row 6, column 1 presents a plausible side face, which is generally hard to generate in the case of Stage-I ACGAN. The reason for this is that Stage-II ACGAN focuses on completing details based on the drafts from Stage-I, which is a lot easier than generating side faces from a random vector. Thus, we conclude that photo-realistic image synthesis can be achieved by multiple stages synthesis.

By interpolation on the latent representations in generator, we produce the gradation patterns from one face to another in Fig. 5.

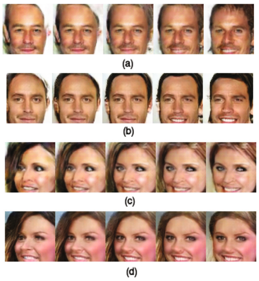

Fig. 5. Comparison between interpolations in Stage-I (a,c) and Stage-II (b,d).

Apart from visually comparing and analyzing the synthesized images, we also adopt quantitative metrics for evaluation of the discriminability and diversity of images generated in both stages.

As the measurement of discriminability of synthesized images, inception score [3] is adopted for evaluation, which correlated well with human visual system. By applying the inception-v3 model [21] to each generated image, we get the conditional label distribution $p(y|x)$. For realistic images that contains meaningful objects, this distribution has a low entropy. The marginal $p(y) = \int p(y|x = G(z))dz$ should have high entropy. Therefore, the metric called inception score is defined as $\mathbb{E}_x KL(p(y|x)||p(y))$. The mean inception scores from three sets, which are 100 images from celebA dataset, 100 images from stage-I ACGAN outputs and 100 images from Stage-II ACGAN outputs respectively, are listed in Table 1. As the figures indicate, both stages are capable of generating relatively convincing and varied images while the results in Stage-II is a little bit better than that in Stage-I.

Another quantitative method we use is Multi-scale Structural Similarity (MS-SSIM), which is created under the assumption that human visual system is considered to be highly adapted for structural feature extraction. The values of MS-SSIM indices of a pairs of images measure the similarity between them. Therefore, the larger the indices are, the more varied images are generated. Results in Table 2 shows that the diversity of generated samples in stage-I and stage-II are in the same level of images in dataset, proving the Stack-ACGAN architecture is proficient in generating varied images.

Table 1. Inception scores of images from celebA dataset, Stage-I ACGAN and Stage-II ACGAN

Image source	Inception score
celebA dataset	$1.84 + 0.31$
Stage-I ACGAN	$1.53 + 0.26$
Stage-II ACGAN	$1.56 + 0.29$

Table 2. MS-SSIM of Images from celebA dataset, Stage-I ACGAN and Stage-II ACGAN

Image source	MS-SSIM index
celebA dataset	0.672
Stage-I ACGAN	0.652
Stage-II ACGAN	0.634

5 Conclusion

In this paper, we propose stacked Auxiliary Classifier Generative Adversarial Networks (Stack-ACGAN) for photo-realistic images synthesis. The proposed method decomposes the synthesis process into two separate stages. Stage-I ACGAN sketches the basic outlines and colors of the object with constraints from conditional attribute labels. Afterwards, Stage-II ACGAN corrects the defects in Stage-I results and adds more photo-realistic details. Extensive quantitative evaluation are conducted to show proficiency of our proposed method. In comparison to existing conditional generative models, our method is capable of generating higher resolution images (e.g., 256×256) with more convincing details.

As for future work, we believe more attention should be devoted to the following two aspects. During experiments we have observed severe mode collapse in the second stage of Stack-ACGAN before preventing it by adopting one-sided label smoothing and dropout tricks. Therefore, it is worthwhile to figure out a general scheme to stabilize the adversarial training in the future. Moreover, we wonder if plausible images in higher resolution (e.g. 1024×1024 or even higher) can be generated simply by stacking more GANs on top of the others. Dozens of experiments need to be conducted to check the convergence of those models and the discriminability and diversity of results.

References

1. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
2. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1033–1038. IEEE (1999)

3. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems, pp. 2234–2242 (2016)
4. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3730–3738 (2015)
5. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers, vol. 2, pp. 1398–1402. IEEE (2003)
6. Freeman, W.T., Jones, T.R., Pasztor, E.C.: Example-based super-resolution. IEEE Comput. Graph. Appl. **22**(2), 56–65 (2002)
7. Hays, J., Efros, A.A.: Scene completion using millions of photographs. ACM Trans. Graph. (TOG) **26**, 4 (2007)
8. Reed, S.E., Zhang, Y., Zhang, Y., Lee, H.: Deep visual analogy-making. In: Advances in Neural Information Processing Systems, pp. 1252–1260 (2015)
9. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes, arXiv preprint arXiv:1312.6114 (2013)
10. Gregor, K., Danihelka, I., Graves, A., Rezende, D.J., Wierstra, D.: Draw: a recurrent neural network for image generation, arXiv preprint arXiv:1502.04623 (2015)
11. Yan, X., Yang, J., Sohn, K., Lee, H.: Attribute2image: conditional image generation from visual attributes. In: European Conference on Computer Vision, pp. 776–791. Springer (2016)
12. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans, arXiv preprint arXiv:1610.09585 (2016)
13. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis, arXiv preprint arXiv:1605.05396 (2016)
14. Zhu, J.-Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: European Conference on Computer Vision, pp. 597–613. Springer (2016)
15. Zhang, H., Xu, T., Li, H., Zhang, S., Huang, X., Wang, X., Metaxas, D.: Stackgan: text to photo-realistic image synthesis with stacked generative adversarial networks, arXiv preprint arXiv:1612.03242 (2016)
16. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z.: Photo-realistic single image super-resolution using a generative adversarial network, arXiv preprint arXiv:1609.04802 (2016)
17. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision, pp. 694–711. Springer (2016)
18. Witten, I.H., Frank, E., Hall, M.A., Pal, C.-J.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Burlington (2016)
19. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456 (2015)
20. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of the ICML, vol. 30 (2013)
21. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)