AUTOSTAR-GAN: MULTI-DOMAIN IMAGE-TO-IMAGE TRANSLATION WITH UNLABELED DATA

[1]*Po-Wui Wu (吳柏威),* [2]*Chiou-Shann Fuh (傅楸善)*

[1]Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan

ABSTRACT

Recent studies have shown remarkable success in image-to-image translation for multiple domains. However, those models are hard to use in real case, since many image are unlabeled images and we can not know in advance which domain those unlabeled images are. In this paper, we indicate two main problems: fix image attributes problem and middle attributes compulsory change problem. To address this limitation, we propose AutoStar-GAN, a novel and scalable approach that can automatically recognize the attributes of unlabeled image and perform image-to-image translations for multiple domains.

1. INTRODUCTION

The task of image-to-image translation is always a classical problem in computer vision field. After invention of Generative adversarial network, the result have been significant improved[1]. From pair-domain style transfer model[2][3] to multi-domain image-to-image translation[4], currently the state of the art model of multiple domain translate can use only one single model to training with large number of domain data[4].

1.1. Problems with current network

Along with the growth of the number of domain, we can find there are some problems when we want to use it with some unlabeled image. In usual case, we have some expectation and assumption to do the task of image-to-image translation.

- We want to change small part of attributes in testing image.
- We can preserve the other part of attributes in testing image.
- We assume that the attribute in many image is not fully compliant, this attribute may be in the middle of two attributes.

To achieve those expectation and assumption, this is hard to do it only with existing image-to-image translation model. We indicate two main problem we will encounter as follows: recognize original image domain problem and middle attributes compulsory change problem.

1.2. Recognize Original Image Domain Problem

To achieve the first two expectation we mentioned, we have to find all the attributes it have before we change few attributes. We know that every image must have some positive attributes except the changing attribute, so We can not directly assign other attributes is zero, it means other attributes in this image are negative, but it's not the truth from original image. We can do it by human labeling or using another recognition model to detect attributes, but whatever method we choose it consumes lots of time and computing power. Figure 1 shows the more clear example for this problem.

1.3. Middle Attributes Compulsory Change Problem

Even we have known the all attributes the image have. it is still hard to preserve original attributes. When we use model to transform the image and we want to preserve a attribute, but the attribute may be the middle of two attributes. At this situation, we will choose a more obvious attribute to represent this image instead of another attribute. However, we can find that after image translation the small part of another attribute is eliminated because the model will let this image fit this positive attribute completely. This result violated our second expectation. Figure 2 shows the clearer example for this problem.

1.4. Solution

In this paper, we present a novel concept called domain translation vector. Instead of telling which goal the output should be, we tell the model which direct the input should go. For those unknown or preserved attributes, we just give the zero value in domain translation vector to tell the model it don't need to change on this attribute. In addition, we

Fig. 1. Multiple Domain Image to image translation can apply on face attribute transfer (Input, Zero Vector, Blond hair, Brown hair, Gender, Mustache(Gender, Goatee, Mustache), Pale, Smile, Bangs, Glasses, Aged(Gray hair, Aged)). Our goal is let the model easy to input with Testing image without labels.

Fig. 2. If we only want to change hair color in this face and we don't want to change other attributes, we have to recognize all the attributes this face have (Gender, Aged, Smiled ...etc) and then change the hair color attribute.

present a novel model base on the text2image model[5], starGAN model[4] and segmentation GAN model[6]. Our propose model consists of generator and discriminator. Generator inputs original image and domain translation vector to generate fake image, and discriminator inputs original image ,target image and domain translation vector to tell the degree of the reality for the translation. We refer to as proposed networks as AutoStar-GAN, and our main contributions are summarized as follows:

- We build an model changing the needed input to let us just focus on the attributes we want to change. For those unknown or unchangeable attributes we just assign a zero value in domain translation vector.
- Our proposed model can automatically detect the attributes the input image have and judgment which attributes should be change and which attributes should be preserve.

Fig. 3. If we only want to change mustache in this face, and we can find that the hair color is between blond and black. Because the blond color is more obvious, we will choose blond hair attribute becoming positive instead of black hair. After translation, the result shows that the hair color becomes purely blond and totally eliminates the black color. The hair color has been changed.

- We show that our proposed architecture favors multi-domain image-to-image translation, supported by quantitative and qualitative evaluation results.

2. RELATED WORK

2.1. Generative Adversarial Networks

Generative adversarial networks (GANs) have shown remarkable results in various computer vision tasks such as image generation, image translation, super-resolution imaging, and face image synthesis. A typical GAN model consists of two modules: a discriminator and a generator. The discriminator learns to distinguish between real and fake samples, while the generator learns to generate fake samples that are

indistinguishable from real samples. Our approach also leverages the adversarial loss to make the generated images as realistic as possible[4].

2.2. Conditional GANs

GAN-based conditional image generation has also been actively studied. Prior studies have provided both the discriminator and generator with class information in order to generate samples conditioned on the class. Other recent approaches focused on generating particular images highly relevant to a given text description. The idea of conditional image generation has also been successfully applied to domain transfer, super-resolution imaging, and photo editing. In this paper, we propose a scalable GAN framework that can flexibly steer the image translation to various target domains, by providing conditional domain information[4].

2.3. Image-to-Image Translation

Recent work have achieved impressive results in image-to-image translation. For instance, pix2pix learns this task in a supervised manner using cGANs. It combines an adversarial loss with a L1 loss, thus requires paired data samples. To alleviate the problem of obtaining data pairs, unpaired image-to-image translation frameworks have been proposed. UNIT combines variational autoencoders (VAEs) with CoGAN, a GAN framework where two generators share weights to learn the joint distribution of images in cross domains. CycleGAN and DiscoGAN preserve key attributes between the in- put and the translated image by utilizing a cycle consistency loss. However, all these frameworks are only capable of learning the relations between two different domains at a time. Their approaches have limited scalability in handling multiple domains since different models should be trained for each pair of domains. Unlike the aforementioned approaches, our framework can learn the relations among multiple domains using only a single model[4].

3. AUTOSTAR GENERATIVE ADVERSARIAL NETWORKS

To solve these problems, we propose two novel method: domain translation vector and zero vector consistency. we use domain translation vector to change the input format and use zero vector consistency to preserve those attributes we do not want to change. In this section, we introduce both method and necessary objective function of generative adversarial network.

3.1. Domain Translation Vector

We use Domain Translation Vector instead of giving target domain directly. We want to let our generator learn how to detect attributes from original image and we give domain translation vector to tell the generator where you should go and which domain is your target domain. We simply get the raw vector by subtracting original image domain value from target image domain value. However,the raw vector can not be used for training because the negative value will be ignored after processing through the relu activation in generator.

To solve this problem, we use positive vector and negative vector to represent the raw vector. The positive vector removes the negative number in raw vector. We can say that the positive vector save the attribute value those target image have but original image don't. To generate negative vector, we plus minus on the raw vector and removes the negative number in negative vector. We save the attribute value those the original image have but the target image don't. Finally, we concatenate the positive vector and negative vector as our domain translation vector. We define algorithm 1 as below to generate vector from attribute lists.

Algorithm 1 Domain Translation Vector algorithm

c_a: Attributes list of original image
c_b: Attributes list of target image
v_{ab}: Domain translation vector from original to target image
v_{ba}: Domain translation vector from target to original image

procedure $Vector(c_a, c_b)$
  $v_{positive} \leftarrow c_b - c_a$
  $v_{negative} \leftarrow c_a - c_b$
  $v_{positive} \leftarrow maximum(v_{positive}, 0)$
  $v_{negative} \leftarrow maximum(v_{negative}, 0)$
  $v_{ab} \leftarrow concatenate(v_{positive}, v_{negative})$
  $v_{ba} \leftarrow concatenate(v_{negative}, v_{positive})$
  return v_{ab}, v_{ba}

3.2. Objective function

3.2.1. Adversarial loss

We use adversarial loss to make the fake data indistinguishable from real data.

$$L_{adv} = E_x[logD(x)] + E_{x,v}[log(1 - D(G(x, v)))]$$

where generator G generates an image $G(x, c)$ conditioned on both the input image x and the target domain label c, while discriminator D tries to distinguish between real and fake images. We refer the discriminator D outputs the probability distribution over the given data. The generator G tries to minimize it and discriminator D tries to maximize it.

3.2.2. Cycle Consistency

By minimizing the adversarial and classification losses, G is trained to generate images that are realistic and classified

to its correct target domain. However, minimizing the losses does not guarantee that translated images preserve the content of its input images while changing only the domain-related part of the inputs. To alleviate this problem, we apply a cycle consistency loss[2] to the generator, defined as:

$$v = v_{positive}||v_{negative}$$
$$v' = v_{negative}||v_{positive}$$
$$L_{cycle} = \mathrm{E}_{x,v,v'}[||x - G(G(x,v),v')||]$$

where G takes in the translated image $G(x, v)$ and the original domain label v' as input and tries to reconstruct the original image x. We adopt the L_1 norm as our reconstruction loss. Note that we use a single generator twice, first to translate an original image into an image in the target domain using the domain translation vector and then to reconstruct the original image from the translated image using the negative vector. As we know from Algorithm 1, the negative vector is a swap between positive vector and negative vector.

3.2.3. Zero Vector Consistency

To make the other features we do not want to change more identical to original image, we apply a zero vector consistency to the generator, define as:

$$\overrightarrow{0} = (0, 0, ..., 0)$$
$$L_{zero} = \mathrm{E}_{x,\overrightarrow{0}}[||x - G(x, \overrightarrow{0})||]$$

where G takes in the translated image $G(x, \overrightarrow{0})$and we expect that the target image should be identical with original image because $\overrightarrow{0}$ means all the value in domain translation vector is zero and any attribute will not change after translation. We adopt the L_1 norm as our reconstruction loss too.

3.2.4. Total Loss

Finally, the objective functions to optimize G and D are written, respectively, as

$$L_D = L_d$$
$$L_G = L_g + \lambda_{cycle}L_{cycle} + \lambda_{zero}L_{zero}$$

where λ_{cycle} and λ_{zero} are hyper-parameters that control the relative importance of domain classification and reconstruction losses, respectively, compared to the adversarial loss. We use $\lambda_{cycle} = 5$ and $\lambda_{zero} = 5$ in all of our experiments.

4. IMPLEMENTATION

4.1. Network Architecture

4.1.1. Generator

Instead of changing the architecture of generator, we use advance discriminator to teach a more smart generator. After training, we can use generator with same architecture and same number of weights to do multi-domain image-to-image translation and automatically detect attributes of input image. In generator, the vector concatenates to each channel of input image. The processed data is encoded by three convolution layers and nine residual blocks and decoded by three convolution transpose layer. Finally, the generator outputs target image. The more detailed architecture is shown in Figure 3.

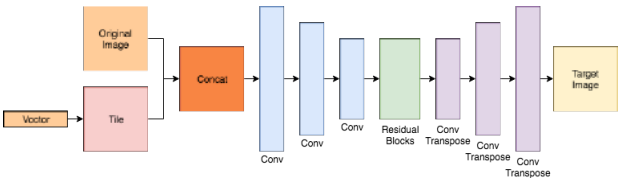

Fig. 4. Generator model: Input with original domain image and domain translation vector. Generator model architecture is a classical ResNet network with nine residual blocks.

4.1.2. Discriminator

To let our discriminator to learn the difference between two images and distinguish if the difference is meet the domain translation vector, which is the subtraction of target image from original image. First, we use numbers of convolution layers to encoding two images. Then,we concatenate both encodes and the vector and use local connection layer to output final conclusion, one or zero. one means the difference between two images is meet the vector and vice versa. The more detail architecture is shown on below figure 4.

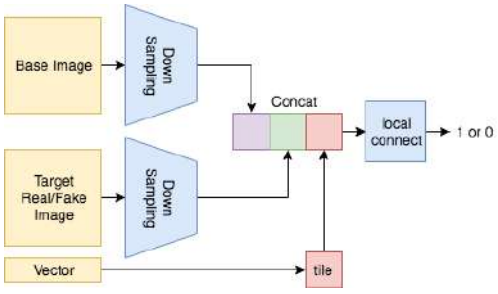

Fig. 5. Discriminator model: Input with real or fake target domain image, original domain image and domain translation vector.

4.2. Matching-aware Discriminator

To let our discriminator to learn the relation between images and the vector, we need to input the wrong data instead of only training with right image and fake image. The set of original image A, target image B and vector AB is correct. The set of original image A, fake image F and vector AB is wrong. The set of wrong image C, target image B and vector AB is wrong. The set of wrong image C, original image A and vector AB is wrong. The set of original image A, target image B and wrong vector AC is wrong. The set of original image A, target image B and wrong vector CB is wrong. We visualize the relation of these sets and its correctness on figure 5.

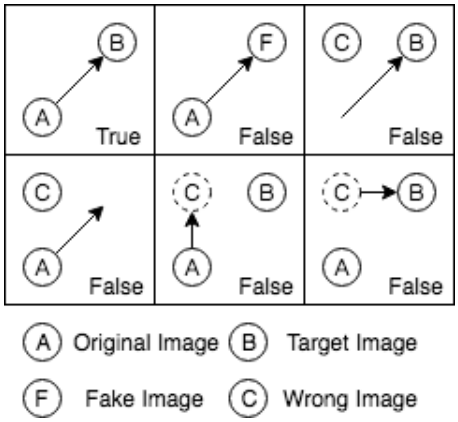

Fig. 6. To let the discriminator learn if the domain translation vector is valid, we conclude six situations and teach the discriminator the first situation is right, and the other are wrong.

4.3. Least square loss

Instead of using negative log likelihood, we use least square loss to stabilize out training and improve our performance. In particular, we train the D loss to minimize $\mathrm{E}_{x,y,v}[(1-D(x,y,v))^2]+\mathrm{E}_{x,y,v}[(0-D(G(x,v),y,v))^2]+\mathrm{E}_{x,v}[(0-D(x',y',v'))^2]$ and train the G loss to minimize $\mathrm{E}_{x,y,v}[(1-D(G(x,v),y,v))^2]$. The x means the real target data, y means the real origin data and v means the domain vector from y to x. We indicate that the x',y',v' are the wrong data. To let the discriminator learn the feature from vector and origin data, we give a more detailed algorithm in algorithm 2.

4.4. Training Dataset

We use Celeba-HQ and Celeba dataset to get high resolution images and attributes lists of these images.

Algorithm 2 AutoStar-GAN training algorithm

A: Real original domain image
B: Real target domain image
C: Wrong target domain image
$\overrightarrow{0}$: Zero vector
V_{AB}: domain translation vector
V_{BA}: domain translation reverse vector
V_{AC}: Wrong target vector
V_{CB}: Wrong origin vector
S: Number of training batch step

for n=1 to S **do**
  $s_f \leftarrow D(A, G(A, V_{AB}), V_{AB})$ ▷ real, fake, real
  $s_{wo} \leftarrow D(C, B, V_{AB})$ ▷ wrong, real, real
  $s_{wt} \leftarrow D(A, C, V_{AB})$ ▷ real, wrong, real
  $s_{wv1} \leftarrow D(A, B, V_{AC})$ ▷ real, real, wrong
  $s_{wv2} \leftarrow D(A, B, V_{CB})$ ▷ real, real, wrong
  $g_{cycle} \leftarrow |A - G(G(A, V_{AB}), V_{BA})|$
  $g_{zero} \leftarrow |A - G(A, \overrightarrow{0})|$
  $L_D \leftarrow (1-s_r)^2 + s_f^2 + s_{wt}^2 + s_{wo}^2 + s_{wv1}^2 + s_{wv2}^2$
  $L_G \leftarrow (1-s_f)^2 + \lambda_{cycle} g_{cycle} + \lambda_{zero} g_{zero}$
  $D \leftarrow D - \alpha \frac{\delta L_D}{\delta D}$ ▷ Update Discriminator
  $G \leftarrow G - \alpha \frac{\delta L_G}{\delta G}$ ▷ Update Generator

4.4.1. Celeba

CelebA is a dataset provided by Ziwei Liu, Ping Luo, Xiaogang Wang, Xiaoou Tang[7]. It can provide a huge number of image of celebrity faces. It also provide a list records the attributes of CelebA images. Each image map to a attributes list, which records if this face is man face or woman face or if this face is old face or young face and so on.

4.4.2. Celeba-HQ

Celeba-HQ is a dataset provided by NVIDIA Research. It can provide super high resolution celebA images, which are 1024 pixel-wide. It is used to training Progressive Growing of GANs[8]. We use this dataset and we compress the images to 256 pixel-wide. Each Celeba-HQ image map to one of the low resolution image in celebA and Celeba-HQ provides a list to retrive it.

Now we have a list high resolution image map to original low resolution image and a list original low resolution image map to its attributes. We can make a list high resolution image map to its attributes and train our model with it and high resolution images. We take 17/40 of the attributes to train the model.

Fig. 7. Single and multiple attribute transfer on CelebA (Input, Zero Vector, Blond hair, Brown hair, Gender, Mustache(Gender, Goatee, Mustache), Pale, Smile, Bangs, Glasses, Aged(Gray hair, Aged)).

5. EXPERIMENTS

5.1. Baseline Models

As our baseline models, we adopt StarGAN, which performs image-to-image translation between two different domains. For comparison, we trained the model using the same dataset and same attrbute list.

5.1.1. StarGAN

StarGAN uses an adversarial loss to learn the mapping between multiple domains $A, B, C, D, E....$ This method regularizes the mapping via cycle consistency losses,$||x - (G_{XY}(G_{YX}(x)))||$ the domain variable X and Y can apply to many domains A,B,C,D,E.... This method requires input an image and a target domain: an attribute list.

5.2. Setting Parameters

Lambda cycle: 10
Lambda zero: 10
Training steps: 400000
Batch size: 2
Image size: 256
Optimizer: $Adam(\alpha = 0.00004, \beta 1 = 0.5, \beta 2 = 0.999)$

We use these parameter to train our model. We perform one generator update after one discriminator updates.For data augmentation we flip the images horizontally with a probability of 0.5 and scale its size from 0.8 to 1.2 randomly. Training takes about three day on a single NVIDIA GTX 1080ti GPU.

5.3. Result

As seen in Fig. 6, AutostarGAN clearly generates the most natural-looking expressions while properly maintaining the personal identity and facial features of the input. Additionally, as seen in Fig. 7 AutostarGAN can automatically detect the attributes of input data, performing translation on unlabeled data and also generates the most natural-looking expressions too.

6. CONCLUSION

In this paper, we proposed Autostar-GAN, a novel, scalabel model to perform image-to-image translation among multiple domains using a single model. With the domain translation vector, we can easily use unlabeled data to preform translation task. More over, because we don't need to label all the unlabeled data first, the batch work of it becomes possible. Besides the advantages in scalability and usability, AutostarGAN generated images of same high visual quality compared to existing methods with bigger images and less data. In principle, our proposed model can be applied to translation between any other types of domains, e.g., style transfer, which will be one of our future work.

[9] [8] [10] [11] [12] [13] [14] [15] [16] [5] [17] [18] [19] [20] [21] [22] [23] [6] [24] [25] [7] [26] [27] [28]

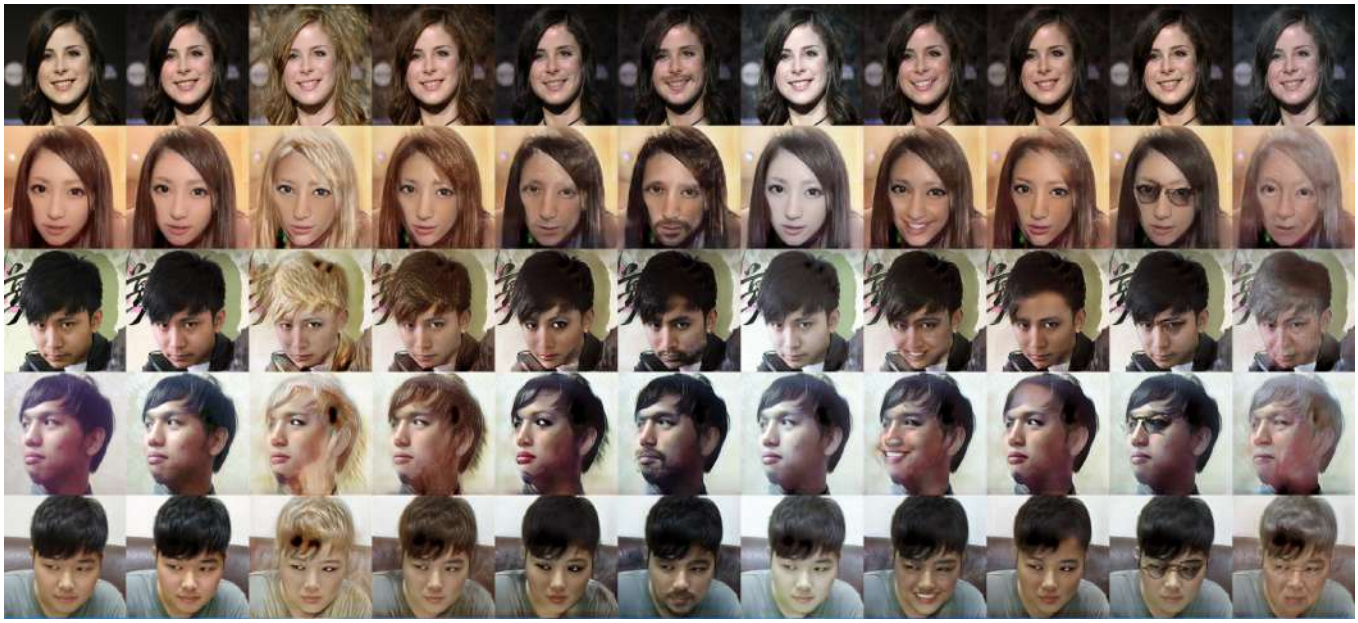

Fig. 8. Single and multiple attribute transfer on Testing image without labels (Input, Zero Vector, Blond hair, Brown hair, Gender, Mustache(Gender, Goatee, Mustache), Pale, Smile, Bangs, Glasses, Aged(Gray hair, Aged)).

REFERENCES

[1] Hyunsoo Kim Jung Kwon Lee Jiwon Kim. Taeksoo Kim, Moonsu Cha, "Learning to discover cross-domain relations with generative adversarial networks," 2017.

[2] Phillip Isola Alexei A. Efros. Jun-Yan Zhu, Taesung Park, "Unpaired image-to-image translation using cycle-consistent adversarial networks," 2017.

[3] Ping Tan Minglun Gong. Zili Yi, Hao Zhang, "Dualgan: Unsupervised dual learning for image-to-image translation," 2017.

[4] Munyoung Kim Jung-Woo Ha Sunghun Kim Yunjey Choi, Minje Choi and Jaegul Choo., "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," 2017.

[5] Xinchen Yan Lajanugen Logeswaran Bernt Schiele Honglak Lee. Scott Reed, Zeynep Akata, "Generative adversarial text to image synthesis," 2016.

[6] Soumith Chintala Jakob Verbeek. Pauline Luc, Camille Couprie, "Semantic segmentation using adversarial networks," 2016.

[7] Chen Change Loy Xiaoou Tang. Shuo Yang, Ping Luo, "From facial parts responses to face detection: A deep learning approach," 2015.

[8] Samuli Laine Jaakko Lehtinen. Tero Karras, Timo Aila, "Progressive growing of gans for improved quality, stability, and variation," 2017.

[9] Xueting Li Ming-Hsuan Yang Jan Kautz. Yijun Li, Ming-Yu Liu, "A closed-form solution to photorealistic image stylization," 2018.

[10] Sebastian Nowozin Thomas Hofmann. Kevin Roth, Aurelien Lucchi, "Stabilizing training of generative adversarial networks through regularization," 2017.

[11] Martin Arjovsky Vincent Dumoulin-Aaron Courville. Ishaan Gulrajani, Faruk Ahmed, "Improved training of wasserstein gans," 2017.

[12] Léon Bottou. Martin Arjovsky, Soumith Chintala, "Wasserstein gan," 2017.

[13] Hongsheng Li Shaoting Zhang-Xiaogang Wang Xiaolei Huang Dimitris Metaxas. Han Zhang, Tao Xu, "Stackgan++: Realistic image synthesis with stacked generative adversarial networks," 2017.

[14] Hyunsoo Kim Jung Kwon Lee Jiwon Kim. Taeksoo Kim, Moonsu Cha, "Learning to discover cross-domain relations with generative adversarial networks," 2017.

[15] Stephan Gouws Fred Bertsch Inbar Mosseri Forrester Cole Kevin Murphy. Amélie Royer, Konstantinos Bousmalis, "Xgan: Unsupervised image-to-image translation for many-to-many mappings," 2017.

[16] Zhichao Liu Guangyu Sun. Bingzhe Wu, Haodong Duan, "Srpgan: Perceptual generative adversarial network for single image super resolution," 2017.

[17] Ferenc Huszar Jose Caballero Andrew Cunningham Alejandro Acosta Andrew Aitken Alykhan Tejani Johannes Totz Zehan Wang Wenzhe Shi. Christian Ledig, Lucas Theis, "Photo-realistic single image super-resolution using a generative adversarial network," 2016.

[18] Jonathon Shlens. Augustus Odena, Christopher Olah, "Conditional image synthesis with auxiliary classifier gans," 2016.

[19] Haoran Xie Raymond Y.K. Lau Zhen Wang Stephen Paul Smolley. Xudong Mao, Qing Li, "Least squares generative adversarial networks," 2016.

[20] Lior Wolf. Yaniv Taigman, Adam Polyak, "Unsupervised cross-domain image generation," 2016.

[21] Hongsheng Li Shaoting Zhang Xiaogang Wang Xiaolei Huang Dimitris Metaxas. Han Zhang, Tao Xu, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," 2016.

[22] Tinghui Zhou Alexei A. Efros. Phillip Isola, Jun-Yan Zhu, "Image-to-image translation with conditional adversarial networks," 2016.

[23] Wojciech Zaremba Vicki Cheung Alec Radford Xi Chen. Tim Salimans, Ian Goodfellow, "Improved techniques for training gans," 2016.

[24] Shaoqing Ren Jian Sun. Kaiming He, Xiangyu Zhang, "Deep residual learning for image recognition," 2015.

[25] Soumith Chintala. Alec Radford, Luke Metz, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015.

[26] Simon Osindero. Mehdi Mirza, "Conditional generative adversarial nets," 2014.

[27] Mehdi Mirza Bing Xu David Warde-Farley Sherjil Ozair Aaron Courville Yoshua Bengio. Ian J. Goodfellow, Jean Pouget-Abadie, "Generative adversarial networks," 2014.

[28] Ziwei Liu Ping Luo Xiaogang Wang Xiaoou Tang, "Large-scale celebfaces attributes (celeba) dataset," 2016.