TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 0X/XX ppXXX-XXX
Volume XX, Number 3, June 20XX

A Survey of Image Synthesis and Editing with Generative Adversarial Networks

Xian Wu, Kun Xu*, Peter Hall

Abstract: This paper presents a survey of image synthesis and editing with generative adversarial networks (GANs). GANs consist of two deep networks, a generator and a discriminator, which are trained in a competitive way. Due to the power of deep networks and the competitive training manner, GANs are capable of producing reasonable and realistic images, and have shown great capability in many image synthesis and editing applications. This paper surveys recent GAN papers regarding topics including, but not limited, to texture synthesis, image inpainting, image-to-image translation, and image editing.

Key words: image synthesis; image editing; constrained image synthesis; generative adversarial networks; image-to-image translation

1 Introduction

With the rapid development of Internet and digital capturing devices, huge volumes of images have become readily available. There are now widespread demands for tasks requiring synthesizing and editing images, such as: removing unwanted objects in wedding photographs; adjusting the colors of landscape images; and turning photographs into artwork – or vice-versa. These and other problems have attracted significant attention within both the computer graphics and computer vision communities. A variety of methods have been proposed for image/video editing and synthesis, including texture synthesis[1–3], image inpainting[4–6], image stylization[7,8], image deformation[9,10], and so on. Although many methods have been proposed, intelligent image synthesis and editing remains a challenging problem. This is because these traditional methods are mostly based on pixels[1,4,11], patches[8,10,12,13] and low-level image features[3,14], lacking high-level semantic information.

In recent years, deep learning techniques have made a breakthrough in computer vision. Trained using large-scale data, deep neural networks substantially outperform previous techniques with regard to the semantic understanding of images. They claim state of the art in various tasks, including image classification[15–17], object detection[18,19], image segmentation[20,21], *etc.*

Deep learning has also shown great ability in content generation. In 2014 Goodfellow *et al.* proposed a generative model, called generative adversarial networks (GANs)[22], GANs contain two networks, a generator and a discriminator. The discriminator tries to distinguish fake images from real ones; the generator produces fake images but it tries to fool the discriminator. Both networks are jointly trained in a competitive way. The resulting generator is able to synthesise plausible images. GAN variants have now achieved impressive results in a variety of image synthesis and editing applications.

In this survey, we cover recent papers that leverage generative adversarial networks(GANs) for image synthesis and editing applications. This survey discusses the ideas, contributions and drawbacks of these networks. This survey is structured as follows. Section 2 provides a brief introduction to GANs and related variants. Section 3 discusses applications in image synthesis, including texture synthesis, image impainting, face and human image synthesis. Section 4 discusses applications in constrained image synthesis,

• Xian Wu, Kun Xu are with TNList, the Department of Computer Science and Technology, Tsinghua University, Beijing, China; Peter Hall is with the Department of Computer Science, University of Bath, Bath, UK.

∗ Corresponding author, xukun@tsinghua.edu.cn

Manuscript received: year-month-day; revised: year-month-day; accepted: year-month-day

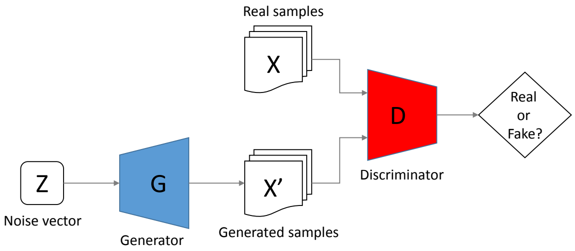


Fig. 1 Structure of GANs.

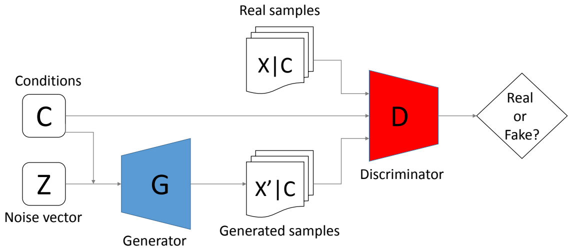


Fig. 2 Structure of cGANs.

including general image-to-image translation, text-to-image, and sketch-to-image. Section 5 discusses applications in image editing and video generation. Finally, Section 6 provides a summary discussion and current challenges and limitations of GAN based methods.

2 Generative Adversarial Networks

Generative adversarial networks (GANs) were proposed by Goodfellow *et al.*[22] in 2014. They contain two networks, a *generator* G and a *discriminator* D. The generator tries to create fake but plausible images, while the discriminator tries to distinguish fake images (produced by the generator) from real images. Formally, the generator G maps a noise vector $\mathbf{z}$ in the latent space to an image: $G(\mathbf{z}) \rightarrow \mathbf{x}$, and the discriminator is defined as $D(\mathbf{x}) \rightarrow [0,1]$, which classifies an image as a real image (*i.e.*, close to 1) or as a fake image (*i.e.*, close to 1).

To train the networks, the loss function is formulated as:

$$\min_G \max_D \mathbb{E}_{\mathbf{x}\in\mathcal{X}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\in\mathcal{Z}}[\log(1-D(G(\mathbf{z})))], \tag{1}$$

where $\mathcal{X}$ denotes the set of real images, $\mathcal{Z}$ denotes the latent space. The above loss function (Equation 1) is referred to as the *adversarial loss*. The two networks are trained in a competitive fashion with back propagation. The structure of GANs is illustrated as Fig. 1.

Compared with other generative models such as variational autoencoders (VAEs)[23], images generated by GANs are usually less blurred and more realistic. It is also theoretically proven that optimal GANs exist, that is the generator perfectly produces images which match the distributions of real images well, and the discriminator always produces 1/2[22]. However, in practice, training GANs is difficult because of several reasons: firstly, networks converge is difficult to achieve [24]; secondly, GANs often get into 'mode collapse', in which the generator produces the same or similar images for different noise vectors $\mathbf{z}$. Various extensions of GANs have been proposed to improve training stability [24–28].

cGANs. Mirza *et al.*[29] introduced conditional generative adversarial networks (cGANs), which extends GANs into a conditional model. In cGANs, the generator G and the discriminator D are conditioned on some extra information $\mathbf{c}$. This is done by putting $\mathbf{c}$ as additional inputs to both G and D. The extra information could be class labels, text, or sketches. cGANs provide additional controls on which kind of data are being generated, while the original GANs do not have such controls. It makes cGANs popular for image synthesis and image editing applications. The structure of cGANs is illustrated as Fig. 2.

DCGANs. Radford and Metz presented deep convolutional generative adversarial networks (DCGANs)[24]. They propose a class of architecturally constrained convolution networks for both generator and discriminator. The architectural constraints include: 1) replacing all pooling layers with strided convolutions and fractional-strided convolutions; 2) using batchnorm layers; 3) removing fully connected hidden layers; 4) in the generator, using $\tanh$ as the activation function and using rectified linear units (ReLU) activation for other layers; 5) in the discriminator, using LeakyReLU activation in discriminator for all layers. DCGANs have shown to be more stable in training and are able to produce higher quality images, hence they have been widely used in many applications.

LAPGAN. Laplacian generative adversarial networks (LAPGAN)[30] are composed of a cascade of convolutional GANs with the framework of a Laplacian pyramid with K levels. At the coarsest level, K, a GAN is trained which maps a noise vector to an image with the coarsest resolution. At each level of the pyramid except the coarsest one (*i.e.*, level k, $0 \le k < K$), a separate cGAN is trained, which takes the output image in the coarser level (i.e., level $k+1$) as a conditional variable to generate the residual image at this level. Due to such a coarse-to-fine manner, LAPGANs are able to produce images with higher resolutions.

Other extensions. Zhao *et al.* proposed an energy-based generative adversarial network (EBGAN)[25] , which views the discriminator as an energy function instead of a probability function. They show that EBGANs are more stable in training. To overcome the vanishing gradient problem, Mao *et al.* proposed least squares generative adversarial networks(LSGAN)[26], which replace the log function by least square function in the adversarial loss. Arjovsky et al. proposed Wasserstein generative adversarial networks (WGAN)[27]. They first theoretically show that the Earth-Mover (EM) distance produces better gradient behaviors in distribution learning compared to other distance metrics. According, they made several changes to regular GANs: 1) removing the sigmoid layer and adding weight clipping in the discriminator; 2) removing the log function in the adversarial loss. They demonstrate that WGANs generate images with comparable quality compared to well designed DCGANs. Berthelot *et al.* proposed boundary equilibrium generative adversarial networks (BEGAN)[28], trying to maintain an equilibrium which can be adjusted for the trade-off between diversity and quality.

Creswell *et al.*[31] provides an overview of GANs. They mainly focus on GANs themselves, including architectures, and training strategies of GANs. Our survey differs because it focuses on image synthesis and editing applications with GANs.

3 Image Synthesis

This section discusses applications including texture synthesis, image super-resolution, image inpainting, face image synthesis and human image synthesis.

3.1 Texture synthesis

Texture synthesis is a classic problem in both computer graphics and computer vision. Given a sample texture, the goal is to generate a new texture with identical second order statistics.

Gatys *et al.*[32] introduced the first CNN-based method for texture synthesis. To characterize a texture, they define a Gram-matrix representation. By feeding the texture into a pre-trained VGG19[16], the Gram matrices are computed by the correlations of feature responses in some layers. The target texture is obtained by minimizing the distance between the Gram-matrix representation of the target texture and that of the input texture. The target texture starts from random noise, and is iteratively optimized through back propagation, hence, its computational cost is expensive.

MGANs. Li *et al.*[33] proposed a real-time texture synthesis method. They first introduced Markovian deconvolutional adversarial networks (MDANs). Given a content image $\mathbf{x}_c$ (*e.g.*, a face image) and a texture image $\mathbf{x}_t$ (*i.e.*, a texture image of leaves), MDANs synthesize a target image $\mathbf{x}_s$ (*e.g.*, a face image textured by leaves). *Feature maps* of an image are defined as feature maps extracted from a pre-trained VGG19 by feeding the image into it[34], and *neural patches* of an image are defined as patch samples on the feature maps[16]. A discriminator is trained to distinguish neural patches from real and fake images. The objective function includes a *texture loss* and a *feature loss*. The texture loss is computed from the classification scores of neural patches of $\mathbf{x}_s$ from the discriminator. The feature loss considers the distance between the feature maps of $\mathbf{x}_s$ and $\mathbf{x}_c$. The target image is initialized with random noise, and is iteratively updated through back propagation by minimizing the objective function. They further introduced Markovian generative adversarial networks (MGANs), which take feature maps of a content image $\mathbf{x}_c$ as input to generate a texture image. MGANs are trained using content and target image pairs synthesized by MDANs. The objective function of MGANs is defined similar to MDANs. MGANs are able to achieve real-time performance for neural texture synthesis, which is about 500 times faster than previous methods.

SGAN and PSGAN. Regular GANs map a random vector to an image. Instead, Jetchev and Bergmann proposed spatial GANs (SGAN)[35], which extend to map a spatial tensor to an image. The network architecture follows DCGANs [24]. The architectural properties of SGAN make it suitable for the task of texture synthesis. Bergmann *et al.* further extend SGAN to periodic spatial GAN (PSGAN)[36]. In PSGAN, the input spatial tensor contains three parts: a local independent part, a spatially global part, and a periodic part. PSGAN is able to synthesize diverse, periodic and high-resolution textures.

3.2 Image super-resolution

Given a low-resolution image, the goal of super-resolution is to upsample it to a high-resolution one. Essentially, this problem is ill-posed because high frequency information is lacking, especially for large upscaling factors. Recently, some deep learning based methods[37–39] were proposed to tackle this problem, results are good for low upsampling factors, but less satisfactory for larger scales. Below, we discuss GAN-based super-resolution methods.

SRGAN. Ledig *et al.*[40] proposed super-resolution generative adversarial network (SRGAN), which takes a low-resolution image as input, and generates an upsampled image with $4\times$ resolution. The network architecture follows the guidelines of DCGAN[24], and the generator uses very deep convolutional network with residual blocks. The objective function includes an adversarial loss and a feature loss. The feature loss is computed as the distance between the feature maps of the generated upsampled image and the ground truth image, where the feature maps are extracted from a pre-trained VGG19 network. Experiments show that SRGAN outperforms the state-of-art approaches on public datasets.

FCGAN. Based on boundary equilibrium generative adversarial networks (BEGAN)[28], Huang *et al.*[41] proposed face conditional generative adversarial network (FCGAN), which specializes on facial image super resolution. Within the network architecture, both the generator and discriminator use an encoder-decoder along with skip connections. For training, the objective function includes a content loss, which is computed by the $L1$ pixelwise difference between the generated upsampled image and the ground truth. FCGAN generates satisfactory results with $4\times$ scaling factor.

3.3 Image inpainting

The goal of image inpaiting is to fill holes in images. It has been always a hot topic in computer graphics and computer vision. Traditional approaches replicate pixels or patches from the original image[4, 5, 10] or from image library[6, 8] to fill the holes. GANs offer a new way for image inpainting.

Context encoder. Pathak *et al.*[42] presented a network called context encoder, which is the first image inpainting method based on GANs. The network is based on an encoder-decoder architecture. The input is an 128×128 image with holes. The output is a 64×64 image content in the hole (when the hole is central) or the full 128×128 inpainted image (when the hole is arbitrary). The objective function includes an adversarial loss and a content loss measuring $L2$ pixel-wise difference between the generated inpainted image and the ground truth image. Experiments used the Paris StreetView dataset[43] and the ImageNet dataset[44]. It achieves satisfactory inpainting results for central holes but less satisfactory results for arbitrary holes.

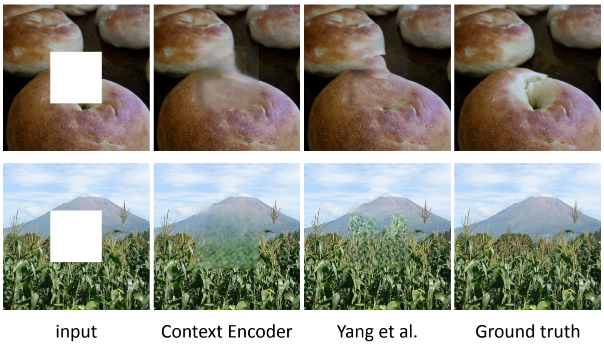


Fig. 3 Comparison between Context Encoder[42] and Yang *et al.*[45].

Multiscale method. Yang *et al.*[45] present a multiscale synthesis method for high-resolution image inpainting. They first train a context network which is similar to context encoder[42] with minor changes on some layers. To complete a 512×512 image with a 256×256 hole, they first downsample the image by a factor of 4, then obtain an initial reconstructed hole image $\mathbf{x}_0$ at resolution 64×64 through the trained context network. The final reconstructed hole image (at resolution 256×256) is then obtained in a coarse-to-fine manner. At each scale, the reconstructed hole image is iteratively updated by optimizing a joint objective function, including a content loss, a texture loss and a total variation (TV) loss. The content loss measures the $L2$ difference between the currently optimized image and the resulting image from the coarser level. The texture loss is computed by comparing the neural patches inside the hole and the neural patches outside the hole. The texture loss enforces the image content inside and outside the hole have similar texture details. It shows nice image inpainting results with 512×512 resolution, but is slow due to the iterative approach. Some results of this method and Context Encoder are shown in Figure 3.

Consistent completion. Iizuka *et al.*[46] proposed a GAN-based approach for global and local consistent image inpainting. The input is an image with an additional binary mask to indicate the missing hole. The output is an inpainted image with the same resolution. The generator follows the encoder-decoder architecture and uses dilated convolution layers[47] instead of standard convolution layers for larger spatial support. There are two discriminators, a global discriminator that takes the entire image as input and a local discriminator that takes a small region covering the hole as input. The two discriminators ensure that the resulting image is consistent at both global and local scale. This work produces natural image inpainting results for high-resolution images with arbitrary holes.

Other methods. Yeh *et al.*[48] proposed a GAN-based iterative method for semantic image inpainting. It first pre-trained a GAN, whose generator G maps a latent vector $\mathbf{z}$ to an image. Given an image with missing contents $\mathbf{x}_0$, they recover the latent vector $\mathbf{z}*$ by minimizing an objective function including an adversarial loss and a content loss. The content loss is computed by a weighted $L1$ pixel-wise distance between the generated image $G(\mathbf{z}*)$ and $\mathbf{x}_0$ on uncorrupted regions, where pixels near the hole are given higher weights. The objective function is iteratively optimized through back propagation. Li *et al.*[49] proposed a GAN-based specialized approach for face image impainting. Following the network architecture of Iizuka *et al.*[46], it incorporates a global discriminator and a local discriminator to enforce image consistency in both global and local scale. It additionally includes a pre-trained parsing network to enforce the harmony of the inpainted face image.

3.4 Face image synthesis

Face image synthesis is a specialized but important topic. Because human vision is sensitive to facial irregularities and deformations, it is not an easy task to generate realistic synthesized face images. GANs have shown a good ability in creating face images of high perceptual quality and with detailed textures.

Face aging. Face aging methods transform a facial image to another age, while still keeping identity. Zhang *et al.*[50] present a conditional adversarial autoencoder (CAAE) for this problem, which consists of an encoder E, a generator G, and two discriminators D_z and D_{img}. The encoder E maps a face image $\mathbf{x}$ to a vector $\mathbf{z}$ indicating personal features. The output vector $\mathbf{z}$, together with a conditional vector $\mathbf{c}$ indicating a new age, are fed into the generator G to generate a new face image. D_z takes the vector $\mathbf{z}$ as input, and enforces $\mathbf{z}$ to be uniformly distributed. D_{img} forces the face image generated by G to be realistic and to conform with the given age. Besides the two adversarial losses of the two discriminators, the objective function also includes an $L2$ content loss and a TV (total variation) loss. The content loss enforces the input face image and the generated face image to be similar: $\mathbf{x} \approx G(E(\mathbf{x}), \mathbf{c})$. The TV loss is introduced to remove ghosting effects. All the networks are jointly trained. CAAE is able to generate map input face images to plausibly appear as a different age.

Antipov *et al.* [51] proposed an age conditional generative adversarial network (Age-cGAN) for face aging. Age-cGAN consists of an encoder and a cGAN. Like CAAE, the encoder E maps a face image $\mathbf{x}$ to a latent vector $\mathbf{z}$, and the conditional generator G maps a latent vector $\mathbf{z}$ with an age condition $\mathbf{c}$ to a new face image. The cGAN is first trained, and the encoder is trained using pairs of latent vectors and generated face images of the cGAN. After training, given an input face image $\mathbf{x}_0$ with age $\mathbf{c}_0$, face aging is achieved by: 1) feeding $\mathbf{x}_0$ into the encoder to obtain an initial latent vector $\mathbf{z}_0$; 2) iteratively updating $\mathbf{z}_0$ to a new latent vector $\mathbf{z}*$ through an identity preserving optimization, which enforces the reconstructed face image with the same age to be close to the input image: $G(\mathbf{z}*, \mathbf{c}_0) \approx \mathbf{x}_0$; 3) feeding the optimized latent vector $\mathbf{z}*$ and the target age into the generator to obtain the new face image.

Face frontalization. Face frontalization aims to transform a face image from rotated or perspective views to frontal views. Tran *et al.*[52] proposed Disentangled Representation learning-GAN (DR-GAN) for face synthesis with new poses. The generator G uses an encoder-decoder architecture. It learns a disentangled representation for face images which are the output of the encoder and also the input of the decoder. Specifically, the encoder maps a face image $\mathbf{x}$ to an identity feature $\mathbf{f}$, and the decoder synthesize a new face image, given an identity feature $\mathbf{f}$, a target pose, and a noise vector. The discriminator D has two parts, one for identity classification (*i.e.*, also contains an additional identity class for fake images), and the other for pose classification. The goal of D is to predict both identity and pose correctly for real images and also to predict identity as fake for fake images. The goal of G is to fool D to classify fake images to its input identity and pose. The objective function for training only contains the newly introduced adversarial loss. Experiments show that DR-GAN is superior to existing methods on pose invariant face recognition.

Yin *et al.*[53] proposed a face frontalization generative adversarial network (FF-GAN), which incorporates 3D Morphable Model (3DMM)[54] into the GAN structure. Since 3DMM provides geometry and appearance priors for face images and the representation of 3DMM is also compact, FF-GAN has the advantage of fast convergence, and produces high-quality frontal face images.

Huang *et al.*[55] proposed two-pathway generative adversarial network (TP-GAN) for frontal face image synthesis. The network has a two pathway architecture, a global generator for generating global structures and a local generator for generating details around facial landmarks. The objective function for training consists of an $L1$ pixel-wise content loss which measures the difference between the generated face image $\widehat{\mathbf{x}}$ and the ground truth, a symmetry loss which enforces $\widehat{\mathbf{x}}$ to be horizontally symmetric, an adversarial loss, an identity preserving loss, and a TV loss.

3.5 Human image synthesis

Human image processing is important in computer vision. Most existing works focused on problems such as pose estimation, detection and re-identification, while generating novel human images attracted few attention until GANs were presented. For the purpose of improving person re-identification precision, Zheng *et al.*[56] utilize GANs to generate human images as extra training data. Other GANs are designed for human image synthesis *per se*.

VariGAN. Variational GAN (VariGAN)[57] aims to generate multi-view human images from a single-view. It follows a coarse-to-fine manner and consists of three networks: a coarse image generator, a fine image generator, and a conditional discriminator. The coarse image generator G_c uses a conditional VAE architecture[58]. Given an input image $\mathbf{x}_0$ and a target view t, it is separately trained to generate a low-resolution image with the target view $\mathbf{x}^t_{LR}$. The fine image generator G_f uses a dual-path U-Net[21] architecture. It maps $\mathbf{x}^t_{LR}$ to a high-resolution image $\mathbf{x}^t_{HR}$ conditioned on $\mathbf{x}_0$. The discriminator D examines the high-resolution image $\mathbf{x}^t_{HR}$ conditioned on the input image $\mathbf{x}_0$. G_f and D are jointly trained with an objective function consisting of an adversarial loss and a content loss measuring $L1$ difference between $\mathbf{x}^t_{HR}$ and ground truth.

Pose guided generation PG^2. Ma *et al.*[59] proposed a pose guided human image generation method (PG^2). Given an input human image and a target pose, it generates a new image with the target pose. They also use a coarse-to-fine two-stage approach. In the first stage, a generator G_1 produces a coarse image $\mathbf{x}_{LR}$ from the input image $\mathbf{x}_0$ and the target pose, capturing the global structure. In the second stage, a generator G_2 generates a high-resolution difference image $\Delta\mathbf{x}_{HR}$ from the coarse image $\mathbf{x}_{LR}$ and the input image $\mathbf{x}_0$. The final image is obtained by summing up $\mathbf{x}_{LR}$ and $\Delta\mathbf{x}_{HR}$. A conditioned discriminator is also involved. Both G_1 and G_2 uses the U-Net[21] architecture. It is able to produce new 256×256 images.

4 Constrained Image Synthesis

This section discusses constrained image synthesis, which is synthesizing a new image with respect to some specified constraints from users, such as another image, text description, or sketches. We will discuss applications on image-to-image translation, text-to-image, and sketch-to-image.

4.1 Image-to-image translation

Image-to-image translation refers to a constrained synthesis process which maps an input image to an output image.

pix2pix. Isola *et al.*[60] proposed a general image-to-image translation framework *pix2pix* using cGANs. Their network architecture follows the guidelines of DCGANs[24] with some additional changes: 1) applying modules of the form convolution-BatchNorm-ReLU; 2) adding skip connections between the deep layers and the shallow layers for the generator. The discriminator uses PatchGAN[33], which runs faster and penalizes unreal structure at the patch scale. Since the goal is not only to produce realistic images (which fool the discriminator), but also require the generated image to be close to ground truth; hence, besides the adversarial loss, they additionally include a content loss in the objective function. The content loss measures the $L1$ distance between the output image and the ground truth image. Pix2pix was demonstrated to be effective for a variety of image-to-image translation tasks, including labels to cityscape, labels to façade, edges to photo, day to night, *etc.* It produces convincing results at the 256×256 resolution, as shown in Figure 4.

cycleGAN. Pix2pix[60] requires paired images (an image before translation and the corresponding image

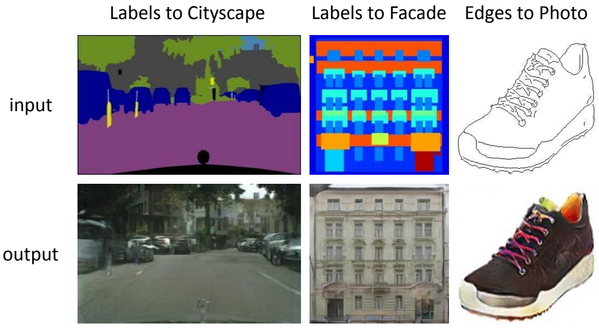


Fig. 4 Results produced by pix2pix[60].

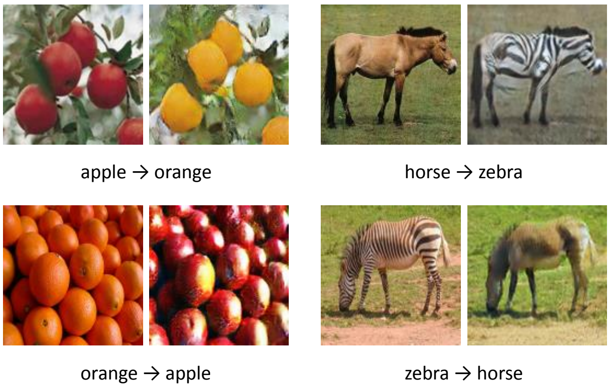


Fig. 5 Results produced by cycleGAN[61].

after translation) as training data, however, in many cases, such image pairs do not exist. To address this issue, Zhu *et al.*[61] proposed an unpaired image-to-image translation framework, named cycle-consistent adversarial networks (cycleGAN). CycleGAN consists of two separate GANs, one translates an image from one domain to another (*e.g.* horse to zebra): $\mathbf{x}_{trans} = G(\mathbf{x})$, the other does the inverse translation (*e.g.* zebra to horse): $\mathbf{x} = G_{inv}(\mathbf{x}_{trans})$. Their network architecture follows Johnson *et al.* [62] which has shown to be effective in style transfer. Similar to pix2pix, the discriminators use PatchGAN[33]. The two GANs are jointly trained. Following LSGAN[26], the adversarial loss uses least square function instead of a log function for more stable training. Beside the two adversarial losses of the two GANs, the objective function additionally includes an $L1$ cycle consistency loss, which enforces that an image translates to itself after the translation cycle: $\mathbf{x} \approx G_{inv}(G(\mathbf{x})), \mathbf{x}_{trans} \approx G(G_{inv}(\mathbf{x}_{trans}))$. Their method has been successfully applied to several translation tasks, including collection style transfer, season transfer, *etc*., as shown in Figure 5.

AIGN. The adversarial inverse graphics network (AIGN)[63] also utilizes unpaired training. AIGN consists of a generator G, a discriminator D, and a task specific renderer P. Consider image-to-image translation as example: the generator G maps an input image $\mathbf{x}$ to an output image $G(\mathbf{x})$, and the renderer maps the output of the generator back to its input. The objective function for training includes an adversarial loss and a reconstruction loss enforcing $\mathbf{x} \approx P(G(\mathbf{x}))$. Beside image-to-image translation, AIGN could be also used for 3D human pose estimation, face super-resolution, image inpainting, *etc*.

4.2 Text-to-image

Text-to-image refers to the process of generating an image which corresponds to a given text description. For example, we could imagine an image describing ”a red bird with a black tail” or ”a white flower with a yellow anther”. This is a difficult problem, but recently, it is able to generate images depicting simple scenes with the help of GANs.

GAN-INT-CLS. Reed *et al.*[64] proposed a text-to-image synthesis method using GANs. The input text is encoded into a text embedding vector $\Phi(t)$ using a recurrent network. Conditioned on the text embedding vector $\Phi(t)$, the generator maps a noise vector $\mathbf{z}$ to a synthesized image. The discriminator is also conditioned on $\Phi(t)$, and is designed to judge whether the input image is real or fake, and that it matches the texture description. The network architecture follows the guidelines of DCGAN[24]. The objective function only includes an adversarial loss. Note that the noise vector $\mathbf{z}$ could be used to control styles of generated images.

GAWWN. Reed *et al.*[65] introduced a generative adversarial what-where network(GAWWN), which considers location constraints in addition to text descriptions. A location constraint could be given by a bounding box, or by keypoints. Specifically, for bounding box constraints, a bounding-box-conditional GAN is proposed. The networks (both the generator and the discriminator) are conditioned on the bounding box and the text embedding vector which represents text description. The networks have two pathways, a global pathway that operates on the full image, and a local pathway that operates on the region inside the bounding box. For keypoint constraints, a keypoint-conditional GAN is also proposed. The keypoint constraints are represented using binary mask maps.

Other methods. Stacked generative adversarial networks (StackGAN)[66] are able to generate high-resolution images conditioned by given text descriptions. This method has two stages of GANs. Stage-1 GAN generates a low-resolution (64×64) image from a noise vector conditioned to some text description. The output 64×64 image from Stage-1 and the text descriptions are both fed into another Stage-2 GAN to generate a high-resolution (256×256) image. It is the first work to generate images with 256×256 resolution from texts. A text conditioned auxiliary classifier GAN (TAC-GAN)[67] is another text-to-image synthesis method. It is built upon auxiliary classifier GAN (AC-GAN)[68], but replaces the class label condition by a text description condition.

Current text-to-image synthesis approaches are capable of generating plausible images of single object, such as a bird or a flower. But they are still not well adapted to complex scenes with multiple objects, which is an important direction for future works.

4.3 Sketch-to-image

Sketches are a convenient way for users to draw what they want, but they lack detail, color *etc*. Therefore, automatically mapping the input sketches to the user desired images is an attractive problem for researchers. Sketch2Photo[69] provides a fantastic way to synthesize images with sketch and text labels by the composition of Internet images, but text labels are necessary in their work. Recently, GAN-based methods are able to generate images from sketches without text labels, showing better flexibility.

Scribbler. Sangkloy *et al.*[70] proposed GAN-based synthesis method named Scribbler, which converts sketch images with color strokes to realistic images. The generator employs an encoder-decoder architecture with residual blocks[17], and generates a new image with the same resolution as the input sketch image. The objective function consists of a content loss, a feature loss, an adversarial loss, and a TV loss. The content loss measures $L2$ pixel-wise difference between the generated image $\widehat{\mathbf{x}}$ and ground truth $\mathbf{x}_G$. Similar to MDANs[33], the feature loss is defined as the feature distance between $\widehat{\mathbf{x}}$ and $\mathbf{x}_G$, where the features are extracted from a pre-trained VGG19 network. The TV loss is included to improve the smoothness of the generated images[62]. It is able to generate realistic, diverse, and controllable images.

TextureGAN. Xian *et al.*[71] proposed TextureGAN, which converts sketch images to realistic images with the additional control of object textures. The generator takes a sketch image, a color image, and a texture image $\mathbf{x}_t$ as input to generate a new image $\widehat{\mathbf{x}}$. The network structure follows Scribbler[70]. The objective function consists of a content loss, a feature loss, an adversarial loss, and a texture loss. The content loss, feature loss and adversarial loss are defined similar to Scribbler[70]. Following the CNN based texture synthesis method[32], the texture loss is computed as the distance between the Gram-matrix representation of patches in $\widehat{\mathbf{x}}$ and $\mathbf{x}_t$, enforcing texture appearance of $\widehat{\mathbf{x}}$ close to $\mathbf{x}_s$. Segmentation masks are also introduced to enforce computing texture loss and content loss only in the foreground region.

Other methods. Magic Pencil[72] is another GAN-based sketch-to-image synthesis method. Beside a generator and a discriminator, it additionally includes a classifier to enforce the generated image and the input sketch are in the same category. With the help of the newly included classifier, it is able to achieve multi-category image synthesis. Auto-painter[73] converts sketches to cartoon images based on GANs. It adopts the network architecture of pix2pix[60] and additionally includes texture loss and TV loss into the objective function.

Sketch dataset are rare, so researchers often utilize edge detection algorithm to extract sketches for training. However, extracted sketches often contain lots of details and their low level statistics are very different from hand drawings. That is not the only difference, there can be changes in geometry and connectedness too. Additionally, while these methods work on single sketched object well, GAN-based generation of complex scenes from sketches is still be a challenging problem.

5 Image editing and Videos

5.1 Image editing

Image editing is an important topic in computer graphics, in which users manipulate an image through color and (or) geometry interactions. A lot of work has investigated tasks such as image warping[74–76], colorization[77–79], blending[80–82]. These works mainly work on pixels or patches, and do not necessarily keep semantic consistency in editing.

iGAN. Zhu *et al.*[83] proposed iGAN, which uses GAN as a manifold approximation and constrains the edited images on the manifold. They pre-train a GAN

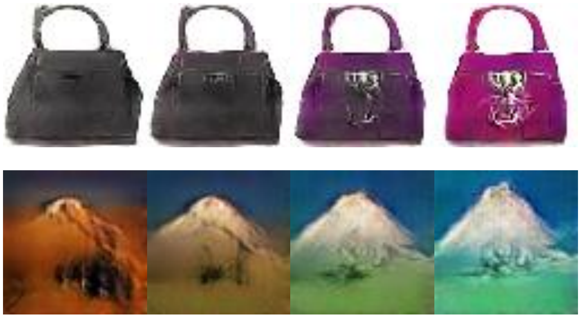


Fig. 6 Results produced by iGAN[83].

from a large image collection, which maps a latent vector $\mathbf{z}$ to a natural image. Given an original image, the method works as follows: 1) They project the original image $\mathbf{x}_0$ to the latent space and obtain a latent representation vector $\mathbf{z}_0$. The projection is done through a hybrid method combining optimization and a pre-trained encoder network. 2) After specifying shape and color edits, they optimize for a new vector $\mathbf{z}*$ minimizing an objective function containing a data loss, a manifold smoothness loss, and an adversarial loss. The data loss measures differences with user edit constraints so as to enforce satisfying user edits. The manifold smoothness loss is defined as the $L2$ difference between $\mathbf{z}*$ and $\mathbf{z}_0$, so that the image is not changed too much. 3) By interpolating between $\mathbf{z}_0$ and $\mathbf{z}*$, a sequence of continuous edited images are generated. 4) Finally, the same amount of edits are transferred to the original image $\mathbf{x}_0$ using optical flow to obtain the final results. iGAN achieves realistic image editing, on various edit operators such as coloring, sketching and geometric warping. Figure 6 shows the interpolation between generated images of a bag and an outdoor scene.

IAN. Brock *et al.*[84] proposed an introspective adversarial network (IAN) for image editing. IAN consists of a generator G, a discriminator D, and an encoder E. The network architecture follows DCGAN[24]. The generator G maps a noise vector $\mathbf{z}$ to a generated image. The encoder E uses the discriminator D as a feature vector, and is built on top of the final convolutional layer of D. The discriminator D inputs an image and determines whether it is real, fake, or reconstructed. The networks are jointly trained with respect to an objective function consisting of a content loss, a feature loss, the ternary adversarial loss, and the KL divergence of VAE[23]. The content loss measures $L1$ pixel-wise difference between reconstructed and original images. The feature loss measures $L2$ feature difference between reconstructed and original images. Such designs enforce high-quality reconstruction.

Other methods. Cao *et al.*[85] applies GAN for image colorization. Its high-level network architecture follows pix2pix[60]. It generates diverse colorization results by feeding different input noise into the generator. Gaussian-Poisson GAN(GP-GAN)[86] applies GAN for high-resolution image blending. It combines GAN with traditional gradient based blending techniques. GAN is used to generate an initial low-resolution blended image, and the final result is obtained by optimizing the Gaussian-Poisson equation.

5.2 Video Generation

Inspired by the success of GANs in image synthesis applications, researchers have also applied GANs to video generation. Compared to image synthesis, video generation is more difficult since video has an extra temporal dimension requiring much larger computation and memory cost. It is also not trivial to keep temporal coherence. We will discuss some important works for such attempts.

VGAN. Vondrick *et al.*[87] proposed a generative adversarial network for video (VGAN). They assume the whole video is combined by a static background image and a moving foreground video. Hence, the generator has two-streams. The input to both streams is a noise vector. The background stream generates the background image with 2D convolutional layers, and the foreground stream generates the 3D foreground video cube and the corresponding 3D foreground mask, with spatial-temporal 3D convolutional layers. The discriminator takes the whole generated video as input, and tries to distinguish it from real videos. Since VGAN treats videos as 3D cubes, it requires large memory space; it can generate tiny videos of about one second duration.

TGAN. Saito *et al.*[88] proposed temporal generative adversarial nets (TGAN) for video generation. TGAN consists of a temporal generator, an image generator, and a discriminator. The temporal generator produces a sequence of latent frame vectors $[\mathbf{z}_1^1, ..., \mathbf{z}_1^K]$ from a random variable $\mathbf{z}_0$, where K is the number of video frames. The image generator takes $\mathbf{z}_0$ and a frame vector $\mathbf{z}_1^t$ $(1 \leq t \leq K)$ as input, and produces the $t-$th video frame. The discriminator takes the whole video as input and tries to distinguish it from real ones. For stable training, they follow WGAN[27], but further apply singular value clipping instead of the weight clipping to the discriminator.

MocoGAN. Tulyakov *et al.* proposed motion and content decomposed GAN (MoCoGAN)[89] for video generation. The basic idea is to use a motion and content decomposed representation. Given a sequence of random variables $[\epsilon^1, ..., \epsilon^K]$, a recurrent network maps them to a sequence of motion vectors $\mathbf{z}_M^1, ..., \mathbf{z}_M^K$, where K is the number of video frames. A content vector $\mathbf{z}_C$, together with a motion vector $\mathbf{z}_M^t$ $(1 \leq t \leq K)$ are fed into the generator to produce the $t-$th video frame. There are two discriminators, one for distinguishing real from fake single frames, while the other for distinguishing real from fake videos.

Video prediction. Video prediction refers to the process of predicting one (or a few) future frames conditioned by a few existing video frames. Various GAN-based approaches[90–93] have been proposed for this goal. With a multiscale architecture, Mathieu *et al.*[90] generate future frames by minimizing a MSE loss, an adversarial loss, and a gradient difference loss. Zhou *et al.*[91] learn temporal transformations of specific phenomenon from videos, such as flower blooming, ice melting, *etc*. Vondrick and Torralba[92] learn pixel transformation and generate future frames by transforming pixels from existing frames. Liang *et al.*[93] proposed dual motion GAN, which enforces predicted future frames to be consistent with predicted optical flows.

6 Discussion and Conclusions

There have been great advances in image synthesis and editing applications using GANs in recent years. By exploring large amounts of images, GANs are able to generate more reasonable, more semantically consistent results than classical methods. Besides that, GANs can produce texture details and realistic content, which is beneficial to many applications, such as texture synthesis, super-resolution, image inpainting, *etc*.

However, GANs are still facing many challenges. First, it is difficult to generate high-resolution images. At present, most GAN-based applications are limited to handle images with resolution not larger than 256 × 256. When applied to high-resolution images, blurry artifacts usually occur. Although some approaches use coarse-to-fine iterative approaches to generate high-resolution images, but they are not end-to-end and are usually slow. Recently, Chen *et al.*[94] introduced cascaded refinement networks for photographic image synthesis at 2-megapixel resolution, which gives us a novel perspective for high-resolution image generation.

Secondly, the resolutions of input and output images are usually required to be fixed. In comparison, traditional image synthesis approaches are more flexible and could be adapted to arbitrary resolution. Recently proposed PixelRNN[95] draws images pixel to pixel and allows arbitrary resolution, which gives a good insight.

Thirdly, as a common issue in deep learning, ground truth data (for training) are crucial but hard to get. This is more important in GAN-based image synthesis and editing applications, because usually it is not easy to find ground truth of synthesized or edited images (or they simply do not exist). CycleGAN[61] and AIGN[63] proposed to use unpaired data for training, which might be a feasible solution for similar problems but this needs more attention and exploration.

Finally, although GANs have been applied to video generation and synthesis of 3D models[96–99], the results are far from perfect. It is still hard to extract temporal information from videos or decrease memory costs.

Acknowledgements

This work was supported by the National Key Technology R&D Program(Project Number 2016YFB1001402), the Natural Science Foundation of China (Project Number 61521002), the Joint NSFC-ISF Research Program (project number 61561146393), Research Grant of Beijing Higher Institution Engineering Research Center and Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology. This work was also supported by the EPSRC CDE (Project Number EP/L016540/1).

References

[1] A. A. Efros and T. K. Leung, “Texture synthesis by non-parametric sampling,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, pp. 1033–1038 vol.2.

[2] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, “Graphcut textures: Image and video synthesis using graph cuts,” in *ACM SIGGRAPH*, 2003, pp. 277–286.

[3] Q. Wu and Y. Yu, “Feature matching and deformation for texture synthesis,” *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 364–367, 2004.

[4] A. Criminisi, P. Perez, and K. Toyama, “Object removal by exemplar-based inpainting,” in *Proc.*

Applications	Input	Output	Characteristics	Loss Function	Resolution	Code
Texture Synthesis						
MGAN[33]	image	texture	real-time	$L_{adv} + L_f$	arbitrary	T
PSGAN[36]	noise tensor	texture	periodical texture	L_{adv}	arbitrary	Th
Image Super-Resolution						
SRGAN[40]	image	image	high upscaling factor	$L_{adv} + L_f$	arbitrary	TF
FCGAN[41]	face	face		$L_{adv} + L_1$	128×128	
Image Inpainting						
Context Encoder[42]	image+holes	image		$L_{adv} + L_2$	128×128	T+TF
Yang *et al.*[45]	image+holes	image	high-quality but slow	$L_2 + L_t + L_{tv}$(update)	512×512	T
Iizuka *et al.*[46]	image+holes	image	arbitrary holes	$L_{adv} + L_2$	256×256	
Yeh *et al.*[48]	image+holes	image	high missing rate	$L_{adv} + L_1$(update)	64×64	TF
Li *et al.*[49]	face+holes	face	semantic regularization	$L_{adv} + L_2 + L_{seg}$	128×128	C
Face Aging						
CAAE[50]	face+age	face	smooth interpolation	$L_{adv} + L_2 + L_{tv}$	128×128	TF
Age-cGAN[51]	face+age	face	identity preserved	L_{ip}(update)		
Face Frontalization						
DR-GAN[52]	face+pose	face	arbitrary rotation	L_{adv}	96×96	
FF-GAN[53]	face	face	3DMM coefficients	$L_{adv} + L_1 + L_{tv} + L_{ip} + L_{sym}$	100×100	
TP-GAN[55]	face	face	two pathway	$L_{adv} + L_1 + L_{tv} + L_{ip} + L_{sym}$	128×128	
Human Image Synthesis						
VariGAN[57]	human+view	human	coarse-to-fine	$L_{adv} + L_1$	128×128	
PG^2[59]	human+pose	human	coarse-to-fine	$L_{adv} + L_1$	256×256	
Image-to-Image Translation						
pix2pix[60]	image	image	general framework	$L_{adv} + L_1$	256×256	T+PT
cycleGAN[61]	image	image	unpaired data	$L_{adv} + L_{cyc}$	256×256	T+PT+TF
AIGN[63]	image	image	unpaired data	$L_{adv} + L_{cyc}$	128×128	
Text-to-Image						
GAN-INT-CLS[64]	text	image		L_{adv}	64×64	T
GAWWM[65]	text+location	image	location-controllable	L_{adv}	128×128	T
StackGAN[66]	text	image	high-quality	L_{adv}	256×256	TF+PT
TAC-GAN[67]	text	image	diversity	L_{adv}	128×128	TF
Sketch-to-Image						
Scribbler[70]	sketch(+color)	image	guided colorization	$L_{adv} + L_2 + L_f + L_{tv}$	128×128	
TextureGAN[71]	sketch+texture+color	image	texture-controllable	$L_{adv} + L_2 + L_f + L_t$	128×128	
Magic Pencil[72]	sketch	image	multi-class	$L_{adv} + L_{cls}$	64×64	
Auto-painter[73]	sketch	cartoon		$L_{adv} + L_1 + L_f + L_{tv}$	512×512	
Image Editing						
iGAN[83]	image+manipulation	image	interpolation sequence	$L_{data} + L_{smooth} + L_{adv}$(update)	64×64	Th
IAN[84]	image+manipulation	image	fine reconstruction	$L_{adv} + L_1 + L_f + D_{KL}$	64×64	Th
Cao *et al.*[85]	grayscale image	image	diversity	$L_{adv} + L_1$	64×64	TF
GP-GAN[86]	composited image	image	coarse-to-fine	$L_{adv} + L_2$		Ch
Video Generation						
VGAN[87]	noise vector	video	two streams	L_{adv}	64×64	T
TGAN[88]	noise vector	video	temporal generator	L_{adv}	64×64	Ch
MoCoGAN[89]	noise vector	video	unfixed-length	L_{adv}	64×64	PT

Table 1 Comparisons between different GAN-based methods. Each row gives the information of a specific method. From left to right, we give the method name, its input format, its output format, its characteristic, the composition of its loss function, its maximal allowed image/video resolution, and the framework of its provided code. In the column of loss function, L_{adv}, L_1, L_2, L_f, L_t, L_{tv}, L_{seg}, L_{ip}, L_{sym}, L_{cyc}, L_{cls}, D_{kl} denote adversarial loss, L_1 distance, L_2 distance, feature loss, texture loss, tv loss, segmentation loss, identity preserving loss, symmetry loss, cycle consistency loss, classification loss, and KL divergence, respectively. In the column of code, T, Th, TF, C, PT, Ch denote torch, theano, tensorflow, caffe, pytorch, chainer, respectively.

IEEE Computer Vision and Pattern Recognition (CVPR), June 2003.

[5] N. Komodakis and G. Tziritas, “Image completion using efficient belief propagation via priority scheduling and dynamic pruning,” *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2649–2661, Nov 2007.

[6] J. Hays and A. A. Efros, “Scene completion using millions of photographs,” *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 26, no. 3, 2007.

[7] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, “Image analogies,” in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ser. ACM SIGGRAPH, 2001, pp. 327–340.

[8] C. Barnes, F.-L. Zhang, L. Lou, X. Wu, and S.-M. Hu, “Patchtable: Efficient patch queries for large datasets and applications,” in *ACM Transactions on Graphics (Proc. SIGGRAPH)*, Aug. 2015.

[9] H. Fang and J. C. Hart, “Detail preserving shape deformation in image editing,” in *ACM SIGGRAPH*, 2007.

[10] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, “PatchMatch: A randomized correspondence algorithm for structural image editing,” *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 28, no. 3, Aug. 2009.

[11] T. Welsh, M. Ashikhmin, and K. Mueller, “Transferring color to greyscale images,” in *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3, 2002, pp. 277–280.

[12] Z. Zhu, R. R. Martin, and S.-M. Hu, “Panorama completion for street views,” *Computational Visual Media*, vol. 1, no. 1, pp. 49–57, Mar 2015.

[13] C. Barnes and F.-L. Zhang, “A survey of the state-of-the-art in patch-based synthesis,” *Computational Visual Media*, vol. 3, no. 1, pp. 3–20, Mar 2017.

[14] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays, “Transient attributes for high-level understanding and editing of outdoor scenes,” *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 149, 2014.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105.

[16] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *ICLR*, 2015.

[17] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[18] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” in *Advances in Neural Information Processing Systems 28*, 2015, pp. 91–99.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[20] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[21] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: convolutional networks for biomedical image segmentation,” in *MICCAI: 18th International Conference, Munich, Germany, October 5-9*, 2015.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27*, 2014, pp. 2672–2680.

[23] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *ICLR*, 2014.

[24] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” in *ICLR*, 2016.

[25] J. J. Zhao, M. Mathieu, and Y. LeCun, “Energy-based generative adversarial network,” in *ICLR*, 2017.

[26] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, “Least squares generative adversarial networks,” in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[27] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[28] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: boundary equilibrium generative adversarial networks," *arXiv preprint arXiv:1703.10717*, 2017.

[29] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[30] E. L. Denton, S. Chintala, a. szlam, and R. Fergus, "Deep generative image models using a laplacian pyramid of adversarial networks," in *Advances in Neural Information Processing Systems 28*, 2015, pp. 1486–1494.

[31] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *arXiv preprint arXiv:1710.07035*, 2017.

[32] L. A. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," in *Advances in Neural Information Processing Systems 28*, 2015, pp. 262–270.

[33] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *European Conference on Computer Vision (ECCV)*, 2016.

[34] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[35] N. Jetchev, U. Bergmann, and R. Vollgraf, "Texture synthesis with spatial generative adversarial networks," *arXiv preprint arXiv:1611.08207*, 2016.

[36] U. Bergmann, N. Jetchev, and R. Vollgraf, "Learning texture manifolds with the periodic spatial GAN," in *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[37] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision (ECCV)*, 2016.

[38] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[39] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[40] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[41] H. Bin, W. Chen, X. Wu, and L. Chun-Liang, "High-quality face image SR using conditional generative adversarial networks," *arXiv preprint arXiv:1707.00737*, 2017.

[42] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros, "Context encoders: Feature learning by inpainting," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[43] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros, "What makes paris look like paris?" *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 31, no. 4, pp. 101:1–101:9, 2012.

[44] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec 2015.

[45] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[46] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and Locally Consistent Image Completion," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 36, no. 4, pp. 107:1–107:14, 2017.

[47] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *ICLR*, 2016.

[48] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[49] Y. Li, S. Liu, J. Yang, and M.-H. Yang, “Generative face completion,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[50] Z. Zhang, Y. Song, and H. Qi, “Age progression/regression by conditional adversarial autoencoder,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[51] G. Antipov, M. Baccouche, and J.-L. Dugelay, “Face aging with conditional generative adversarial networks,” in *IEEE International Conference on Image Processing, 17-20 September*, 2017.

[52] L. Tran, X. Yin, and X. Liu, “Disentangled representation learning GAN for pose-invariant face recognition,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[53] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, “Towards large-pose face frontalization in the wild,” in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[54] V. Blanz and T. Vetter, “A morphable model for the synthesis of 3d faces,” in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’99, 1999, pp. 187–194.

[55] R. Huang, S. Zhang, T. Li, and R. He, “Beyond face rotation: Global and local perception GAN for photorealistic and identity preserving frontal view synthesis,” in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[56] Z. Zheng, L. Zheng, and Y. Yang, “Unlabeled samples generated by GAN improve the person re-identification baseline in vitro,” in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[57] B. Zhao, X. Wu, Z.-Q. Cheng, H. Liu, and J. Feng, “Multi-view image generation from a single-view,” *arXiv preprint arXiv:1704.04886*, 2017.

[58] K. Sohn, H. Lee, and X. Yan, “Learning structured output representation using deep conditional generative models,” in *Advances in Neural Information Processing Systems 28*, 2015, pp. 3483–3491.

[59] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, “Pose guided person image generation,” *arXiv preprint arXiv:1705.09368*, 2017.

[60] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[61] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[62] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in *European Conference on Computer Vision (ECCV)*, 2016.

[63] H.-Y. Fish Tung, A. W. Harley, W. Seto, and K. Fragkiadaki, “Adversarial inverse graphics networks: Learning 2d-to-3d lifting and image-to-image translation from unpaired supervision,” in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[64] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, “Generative adversarial text-to-image synthesis,” in *Proceedings of The 33rd International Conference on Machine Learning*, 2016.

[65] S. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, “Learning what and where to draw,” in *Advances in Neural Information Processing Systems 29*, 2016, pp. 217–225.

[66] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, “StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks,” in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[67] A. Dash, J. C. B. Gamboa, S. Ahmed, M. Z. Afzal, and M. Liwicki, “TAC-GAN-Text conditioned auxiliary classifier generative adversarial network,” *arXiv preprint arXiv:1703.06412*, 2017.

[68] A. Odena, C. Olah, and J. Shlens, “Conditional image synthesis with auxiliary classifier GANs,” in *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[69] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu, “Sketch2photo: Internet image montage,” *ACM Transactions on Graphics*, vol. 28, no. 5, pp. 124:1–124:10, Dec. 2009.

[70] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays, "Scribbler: Controlling deep image synthesis with sketch and color," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[71] W. Xian, P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays, "TextureGAN: controlling deep image synthesis with texture patches," *arXiv preprint arXiv:1706.02823*, 2017.

[72] H. Zhang and X. Cao, "Magic pencil: Generalized sketch inversion via generative adversarial nets," in *SIGGRAPH ASIA Posters*, 2016, pp. 42:1–42:2.

[73] Y. Liu, Z. Qin, Z. Luo, and H. Wang, "Auto-painter: Cartoon image generation from sketch by using conditional generative adversarial networks," *arXiv preprint arXiv:1705.01908*, 2017.

[74] M. Alexa, D. Cohen-Or, and D. Levin, "As-rigid-as-possible shape interpolation," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '00, 2000, pp. 157–164.

[75] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," *ACM Transactions on Graphics*, vol. 26, no. 3, July 2007.

[76] Y. Liu, L. Sun, and S. Yang, "A retargeting method for stereoscopic 3d video," *Computational Visual Media*, vol. 1, no. 2, pp. 119–127, Jun 2015.

[77] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *ACM SIGGRAPH*, 2004, pp. 689–694.

[78] X. Li, H. Zhao, G. Nie, and H. Huang, "Image recoloring using geodesic distance based color harmonization," *Computational Visual Media*, vol. 1, no. 2, pp. 143–155, Jun 2015.

[79] S.-P. Lu, G. Dauphin, G. Lafruit, and A. Munteanu, "Color retargeting: Interactive time-varying color image composition from time-lapse sequences," *Computational Visual Media*, vol. 1, no. 4, pp. 321–330, Dec 2015.

[80] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," in *ACM SIGGRAPH*, 2003, pp. 313–318.

[81] Z. Farbman, G. Hoffer, Y. Lipman, D. Cohen-Or, and D. Lischinski, "Coordinates for instant image cloning," in *ACM SIGGRAPH*, 2009, pp. 67:1–67:9.

[82] M. Afifi and K. F. Hussain, "MPB: a modified poisson blending technique," *Computational Visual Media*, vol. 1, no. 4, pp. 331–341, Dec 2015.

[83] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *European Conference on Computer Vision (ECCV)*, 2016.

[84] A. Brock, T. Lim, J. Ritchie, and N. Weston, "Neural photo editing with introspective adversarial networks," in *ICLR*, 2017.

[85] Y. Cao, Z. Zhou, W. Zhang, and Y. Yu, "Unsupervised diverse colorization via generative adversarial networks," *arXiv preprint arXiv:1702.06674*, 2017.

[86] H. Wu, S. Zheng, J. Zhang, and K. Huang, "GP-GAN: towards realistic high-resolution image blending," *arXiv preprint arXiv:1703.07195*, 2017.

[87] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Advances in Neural Information Processing Systems 29*, 2016, pp. 613–621.

[88] M. Saito, E. Matsumoto, and S. Saito, "Temporal generative adversarial nets with singular value clipping," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[89] S. Tulyakov, M. Liu, X. Yang, and J. Kautz, "MoCoGAN: decomposing motion and content for video generation," *arXiv preprint arXiv:1707.04993*, 2017.

[90] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," in *ICLR*, 2016.

[91] Y. Zhou and T. L. Berg, "Learning temporal transformations from time-lapse videos," in *European Conference on Computer Vision (ECCV)*, 2016.

[92] C. Vondrick and A. Torralba, "Generating the future with adversarial transformers," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[93] X. Liang, L. Lee, W. Dai, and E. P. Xing, "Dual motion GAN for future-flow embedded video prediction," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[94] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[95] A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proceedings of The 33rd International Conference on Machine Learning*, 2016.

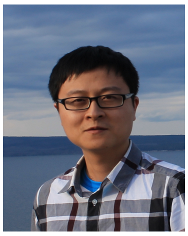
[96] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," in *Advances in Neural Information Processing Systems 29*, 2016, pp. 82–90.

[97] E. J. Smith and D. Meger, "Improved adversarial systems for 3d object generation and reconstruction," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 87–96.

[98] W. Wang, Q. Huang, S. You, C. Yang, and U. Neumann, "Shape inpainting using 3d generative adversarial network and recurrent convolutional networks," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[99] B. Yang, H. Wen, S. Wang, R. Clark, A. Markham, and N. Trigoni, "3d object reconstruction from a single depth view with adversarial learning," in *International Conference on Computer Vision Workshops (ICCVW)*, 2017.

Xian Wu is a PhD student in the Department of Computer Science and Technology, Tsinghua University. Before that, he received his bachelor's degree in the same university in 2015. His research interests include image/video editing and computer vision.



Kun Xu is an associate professor in the Department of Computer Science and Technology, Tsinghua University. Before that, he received his bachelor and doctor's degrees from the same university in 2005 and 2009, respectively. His research interests include realistic rendering and image/video editing.

Peter Hall is a professor in the Department of Computer Science at the University of Bath. He is also the director of the Media Technology Research Centre, Bath. He founded to vision, video, and graphics network of excellence in the United Kingdom, and has served on the executive committee of the British Machine Vision Conference since 2003. He has published extensively in computer vision, especially where it interfaces with computer graphics. More recently he is developing an interest in robotics.