Computers & Graphics 81 (2019) 32–40

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on CAD & Graphics 2019

Image generation from bounding box-represented semantic labels

Congying Liu[a], Zexi Yang[b], Feng Xu[a,*], Jun-Hai Yong[a,*]

[a] BNRist and School of Software, Tsinghua University, Beijing 100084, PR China
[b] Yi Tunnel Technology Co., Ltd., Beijing 100084, PR China

ARTICLE INFO

Article history:
Received 9 March 2019
Accepted 23 March 2019
Available online 13 April 2019

Keywords:
Computer vision
Generative adversarial networks
Image generation

ABSTRACT

Image generation with pixel-wise semantic information is suitable for the development of adversarial learning techniques. In this study, we propose a method for synthesizing objects with class-specific textures and fine-scale details based on bounding box-represented semantic labels. To achieve this goal, we note that the traditional generative adversarial network (GAN) uses noise as an input to generate realistic images with sufficient textures and details, but it cannot be guided by specific targets and requirements. By contrast, conditional GAN (cGAN) can involve various types of guiding information but it often ignores specific textures and details, thereby leading to less realistic results and low resolution. Thus, we propose a new translator-enhancer framework by combining cGAN and GAN to achieve high quality image generation. cGAN is used as a translator to match the semantic constraints whereas GAN is employed as an enhancer to provide details and textures. We also propose a new form of semantic label map as an input, which is represented by instance-level bounding boxes rather than segmentation masks. The semantic label map represented by bounding boxes makes it easier for users to provide the inputs and it also gives greater flexibility when generating object boundaries. The results obtained from qualitative and quantitative experiments showed that our method can generate realistic images of objects with semantic labels represented by bounding boxes. Our method can be used to generate images of novel scenes to support learning tasks during training with various scenes, which are difficult to capture in the real world.

1. Introduction

Image generation and editing are key research areas in computer graphics and computer vision. Virtual reality applications such as Parallax360 [1], three-dimensional (3D) reconstruction applications including Ddrnet [2], and image editing methods such as style transfer [3] all involve image generation and editing. Deep learning can simplify the real world modeling problem to a model learning problem, and thus automatically generating realistic images has attracted much interest among deep learning researchers. Due to the rapid development of the generative adversarial network (GAN) method [4], random noise can be used to generate realistic images such as chairs [5], faces, and room interiors [6]. Conditional GAN (cGAN) can also be employed to generate specific images that match certain conditions, such as generating an image of a dog. Previous studies employed semantic layouts [7], semantic label maps [8], sketches [8], and other techniques as constraint conditions to guide the image generation process.

The GAN method has various applications but its low resolution and quality prevent this approach from being used widely in various applications. Thus, attempts have been made to improve the image resolution and quality with GAN in previous studies. For example, Karras et al. [9] managed to generate high-quality human faces with sufficiently realistic details and textures. Isola et al. [8] proposed a more general image-to-image translation solution based on cGAN with visually appealing results. Wang et al. [10] further improved pix2pix [8] to generate 2048×1024 results. In the present study, inspired by pix2pix and pix2pixHD [10], we propose the use of GAN to generate images with class-specific details and textures based on semantic label maps in a two-step framework, as shown in Fig. 1. This method has a wide range of applications. For example, we can generate images as additional training data for various vision tasks, such as image classification and object detection. Furthermore, if we only have images captured from tables and certain layouts of objects, we can generate images of different scenes with random layouts of the objects.

In our approach, we employ the pix2pix method to generate images from semantic labels. The pix2pix method utilizes a cGAN framework to translate images from one style into another, e.g., day to night, edge map to photo, or grey to color. To further improve pix2pix, pix2pixHD employs a new robust adversarial

* Corresponding author.
E-mail addresses: feng-xu@tsinghua.edu.cn (F. Xu), yongjh@tsinghua.edu.cn (J.-H. Yong).

https://doi.org/10.1016/j.cag.2019.03.009
0097-8493/

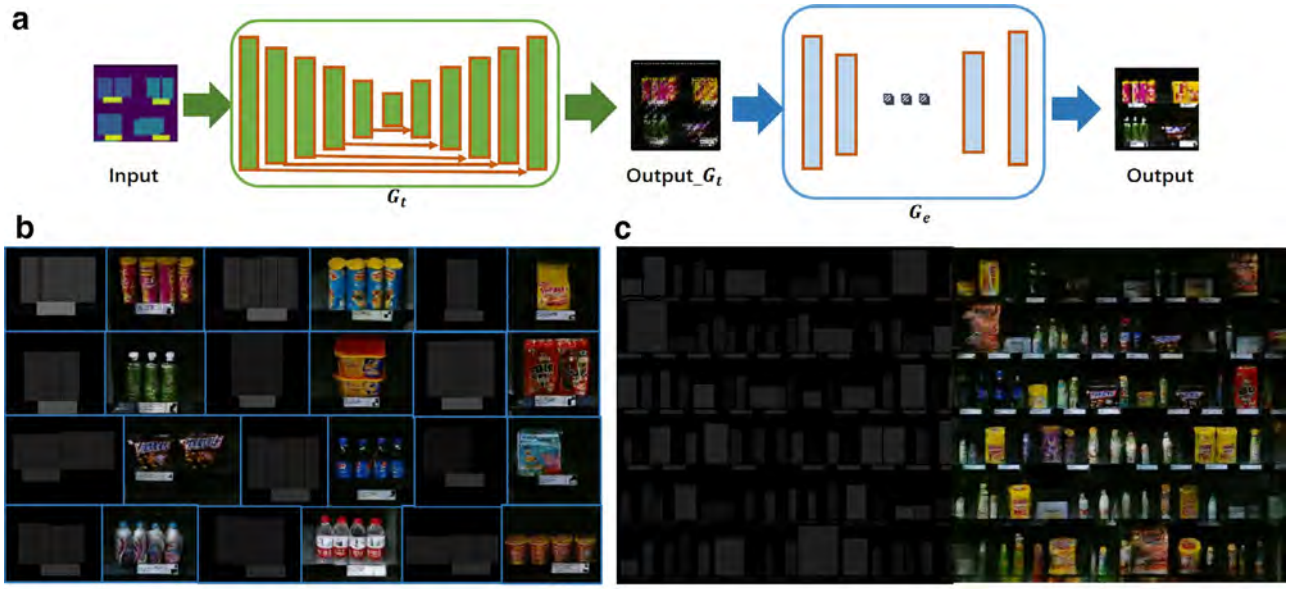

Fig. 1. (a) Generation pipeline: given a semantic label map represented by bounding boxes, G_t uses the U-Net architecture to generate a middle result output G_t. G_e accept the output G_t as the input to generate a more realistic image. Examples are shown of semantic label maps represented by bounding boxes as inputs and the generated outputs. (b) Object areas cropped from images generated with a test semantic label map. (c) Example obtain from our randomly generated data.

learning objective together with novel multi-scale architectures as the generator and discriminator to generate photo-realistic images at a resolution of 2048×1024. The multi-scale generator can be regarded as a generator with enhancers to increase the resolution. Based on the idea of an enhancer, we propose the use of an additional GAN as an enhancer to improve the quality of the generated images by including more details and textures. The adversarial learning objective proposed for pix2pixHD uses the perceptual loss to guarantee robust training and excellent results. However, the perceptual loss [11] depends on the pre-trained networks (e.g., VGGNet [12]) for its calculation, which limits the applicability because many scenarios lack sufficient data to pre-train a network for a specific task and general networks that are pre-trained with public data sets will not work well with all scenarios. Instead, we use the simple L1 loss to help the enhancer to integrate low-level information.

The specific scenario mainly considered in the present study is object generation, which is quite different from other common scenarios, such as street scene generation. In a street scene scenario, semantic classes are defined at coarse scales such as cars, people, streets, buildings, and trees. By contrast, in object generation, semantic classes are defined at fine scales such as 330 ml Coca-Cola and 600 ml Pepsi-Cola. We can translate a semantic area labeled as a car into a realistic car without considering whether the car is a BMW X5 or a Mercedes Benz GLE, but when we translate a semantic labeled image with instance-level bounding boxes, we must ensure that the result resembles a certain object as well as including its fine scale details, e.g., its logo and color.

Furthermore, we propose the use of instance-level bounding boxes to represent the input semantic information, which is quite different from previous methods where semantic segmentation masks were used as the input label maps. This difference has the following advantages. First, bounding-box labeling costs less than labeling the exact mask for each object, and it may further enable further applications to image generation. Second, the use of bounding boxes gives the network more flexibility to decide the exact boundaries of all the instances compared with directly providing the masks for the instances used by pix2pixHD and better results may be obtained. Finally, regular bounding boxes can help to lay out different objects in an image by simply placing one box next to or on top of another. In addition, the use of bounding boxes makes it easier for users to modify the layout or change the number of objects simply by changing the number of boxes.

To evaluate our framework, we mainly compared our method with the baseline pix2pix method and the experimental results demonstrated that our method performed better in terms of both the structural similarity (SSIM) and peak signal-to-noise ratio (PSNR) in quantitative evaluations, as well as in a qualitative user study. In order to help readers to assess our framework, we also conducted comparisons with cycleGAN and pix2pixHD, which deal with quite similar tasks. We performed experiments to demonstrate the requirements for the enhancer GAN, residual blocks, multi-scale discriminator, and instance-level bounding box information. In summary, our main contribution is the development of a novel GAN framework that combines cGAN and GAN to generate images from semantic label maps with fine scale details and class-specific textures. Our second main contribution is the use of instance-level bounding boxes to represent semantic information, which makes it easier for users to lay out a target image in a more flexible manner and generate better results based on object boundaries.

2. Related work

GAN was first proposed by Goodfellow et al. [4] in 2014 and this method has attracted much attention from computer vision researchers who have developed various GAN-based methods to solve many problems. The original GAN based on the idea of a min-max two-player game can estimate the distributions in real images and generate novel images. The generator aims to generate more realistic images from random noise, whereas the discriminator distinguishes the generated images from the real images. The original GAN method can generate digital images from MNIST, but some generated images are noisy and unexpected. Denton et al. [13] extended GAN to generate images in a coarse-to-fine scheme by using a Laplacian pyramid. To overcome the problem of unstable training, Radfor et al. [6] proposed DCGAN by further improving GAN with a deeper convolutional network architecture. Zhang et al. [14] used two generators to progressively render more realistic images. Salimans et al. proposed InfoGAN [15] and

determined several tricks that can help when training a GAN. InfoGAN is an information-theoretic extension of GAN for learning disentangled representations. Arjovsky et al. [16] introduced the Wasserstein GAN, which is less concerned with the balance between the discriminator and generator during training.

Image generation has been a focus of computer vision researchers for many years. Several state-of-the-art patch-based synthesis methods were summarized by Barnes and Zhang [17] and numerous models have been developed for generating highly realistic images. The architectures of these models include variational autoencoders [18], auto-regressive models [19], and GAN [4,13]. Due to its great success and wide range of applications, GAN and its variants, especially cGAN, have become the most popular methods for dealing with tasks related to image generation. Additional methods such as GAN Lab [20] and GANViz [21] have been proposed to help people better understand GAN and its adversarial mechanism.

cGAN aims to generate images that are realistic while also meeting the constraints imposed by specific conditions. Previous proposed methods employ discrete labels [13,22], attributes [23,24], or text [25] as conditions to guide the image generation process. For example, DCGAN [6] applies GAN with class names as conditions. Reed et al. [25] used GAN as a bridge to generate images from detailed natural language descriptions. Reed et al. [26] also proposed GAWWN to learn where the specific content should be drawn when giving the bounding boxes and object key points. Liu and Tuzel [27] proposed coupled GAN (CoGAN) for learning a joint distribution of multi-domain images, e.g., generating a smiling human face with eye glasses attributes. Other studies synthesized or created images by using GAN conditioned on different types of images and they achieved excellent results. Zhu et al. [28] used mouse strokes to generate user constraints on a low dimensional latent vector representation and then used GAN to generate a natural image from the modified latent vector to perform image editing. Pathak et al. [29] added adversarial loss with a standard pixel-wise reconstruction loss to train a context encoder with the capability of semantic inpainting tasks. Ledig et al. [30] presented SRGAN as a GAN method for image super-resolution (SR) to infer photo-realistic images when super-resolving at $4\times$ upscaling factors. Wang and Gupta [31] combined Style-GAN and Structure-GAN for image generation. Structure-GAN accepts noise to generate a surface normal map, which is used as the input for Style-GAN to generate natural indoor scenes images. Wu et al. [32] used GAN with global and local views to complete and extrapolate portrait images. Yoo et al. [33] used GAN to generate images of clothing to dress a person in the input image. Li and Wand [34] focused on the synthesis of efficient textures and proposed Markovian GANs (MGANs) to generate novel images with the expected style by capturing certain feature statistics. GAN has also been extended to other image-related area. For example, Wu et al. [35] extended GAN to the 3D domain and proposed 3D-GAN for generating 3D object silhouettes. Vondrick et al. [36] extended GAN to videos with a spatial-temporal convolutional architecture, which helps the GAN method to use the current frame as a constrain when generating future frames.

The image-conditional GAN method has achieved great success. Isola et al. [8] described an image-to-image translation problem as an image-conditional generation problem and proposed a simple but more general framework called pix2pix. This novel framework based on cGAN aims to learn the common pattern in the input image and output image instead of the output image itself. However, pix2pix is limited by its low resolution and the lack of realistic details and textures. Wang et al. [10] extended pix2pix to pix2pixHD, which is capable of generating images at a resolution of 2018×1024. The resolution is high and the details appear to be sufficient, but the details and textures of a certain object may be a mixture of all objects in the same class, and thus the visually appealing objects obtained might not exist in the real scene. Therefore, it is not possible to guarantee that the generated image matches the original appearance of the object and it cannot be used to train a fine scale classifier. Our method is inspired by pix2pix and it is capable of generating images of objects while retaining their class-specific textures and fine scale details.

3. Proposed method

GAN is combined with a generator G and a discriminator D. The generator G is designed to generate images that can deceive the discriminator by learning a mapping from the input noise z to output image y, $G : z \rightarrow y$ [4]. The discriminator D accepts real images and generates images to learn how to accurately distinguish one from another. Image-conditional GAN is GAN conditioned on an image, so the generator G accepts both the random noise z and observed image x as inputs to generate an output image y, $G : f(x; z) \rightarrow y$ [4].

3.1. pix2pix baseline

The pix2pix method is a cGAN framework designed to learn the translation between an input image and output image. An image from one style, such as a sketch, semantic label map, or grey image, is used as an input to guide the generation of the output with another style, e.g., a photo-realistic image or colored image. The discriminator designed in this framework can distinguish a fake image pair $(x, G(x))$ from a real pair (x, y). In order to further improve the performance and training stability, pix2pix runs in a supervised manner by mixing the adversarial loss with the L1 loss, which forces the generator to create a novel image that deceives the discriminator but it also resembles the real image in the L1 distance.

The pix2pix method uses a set of pairs of corresponding images (x, y) as training data. The generator uses x as the input to generate a fake image $G(x)$. The discriminator takes (x, y) as a real image pair and $(x, G(x))$ as a fake image pair. The pix2pix method then uses the following objective to train the cGAN:

$$G^* = arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_1(G, y), \tag{1}$$

which can be formulated in detail as:

$$arg\min_G \mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,z}[log(1 - D(x, G(x, z)))] \tag{2}$$

$$arg\max_D \mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[logD(x, y)] + \mathbb{E}_{x,z}[log(1 - D(x, G(x, z)))] \tag{3}$$

$$\mathcal{L}_1(G) = \mathbb{E}_{x,y,z}[\|y - G(x, z)\|_1], \tag{4}$$

where x is the observed image, y is the target image with the expected style, such as a colored image, and z is random noise to guarantee the variety of the output.

The pix2pix method is a general solution to the image-to-image translation problem, but the resolution of the image generated is only up to 256×256. We also directly apply the pix2pix method in our object generation scenario to generate images of several objects at a resolution of 512×512, but the training process is unstable, especially when decreasing the weight of the L1 loss. The results may appear to be the correct translation but the class-specific textures and fine-scale details are not sufficient and realistic. Therefore, we improve the pix2pix framework by applying several modifications.

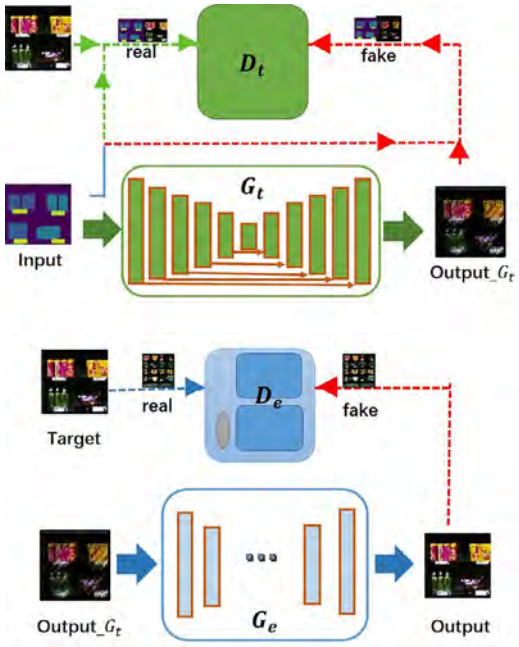

Fig. 2. (a) Translator: the generator G_t uses a semantic label map to generate the output and the discriminator D_t is trained with (Input,Target) as the real image and (Input,OutputG_t) is the fake image. (b) Enhancer: the generator G_e accepts the result from G_t to generate the output and the discriminator D_e trained with Target as the real image and Output as the fake image.

Table 1
G_t architecture details. Conv1-6 is the encoder and Dconv1-6 is the decoder. Except for the last layer, the Conv and Dconv layers are formed as convolution-batchnorm-relu.

Layer	Stride	Kernel	Dilate	Chl(in/out)
Conv1	2	4 × 4	1	3/64
Conv2	2	4 × 4	2	64/128
Conv3	2	4 × 4	2	128/256
Conv4	2	4 × 4	2	256/512
Conv5	2	4 × 4	2	512/512
Conv6	2	4 × 4	2	512/512
Dconv6	1	4 × 4	1	512/512
Dconv5	1	4 × 4	1	512/512
Dconv4	1	4 × 4	1	512/256
Dconv3	1	4 × 4	1	256/128
Dconv2	1	4 × 4	1	128/64
Dconv1	1	4 × 4	1	64/3

3.2. Network architectures

We apply pix2pix as a translator in our framework for image-to-image translation. In order to generate images of objects with class-specific textures and fine-scale details, we add another GAN as an enhancer to augment the outputs from the translator. The translator and enhancer operate as shown in Fig. 2.

3.2.1. Generator

We use two generators in our framework comprising G_t and G_e. We regard G_t as a translator and G_e as an enhancer. In our scenario, G_t takes a semantic label map represented by bounding boxes as the input to generate a photo-realistic image. G_e takes the image generated by G_t to further improve the image details. Further details of the architectures of the generators are given in Tables 1 and 2.

For G_t, we expect that G_t can achieve translation at a high level, e.g., the image structure level. Thus, we adapt the U-Net architecture by combining it with an encoder-decoder network and skip connections, as used by Kumar et al. [37] and Zhang et al. [38] in the image processing field. The encoder-decoder network uses a series of layers to progressively downsample the input and encode it into high-level features, before another series of layers are used to decode the high-level features into a novel output with the same size as the input. In order to help the encoder understand the image well, we also use dilated convolution to obtain larger reception fields. Furthermore, the output image loses some of its low-level features because the output image is decoded from a high level but it contains less features. Thus, the use of skip connections allows the decoder to share the low-level features encoded from the input. In particular, we use six layers for separate encoding and decoding.

Table 2
G_e architecture details. Residual blocks 1–6 each comprise two Conv layers. Except for the last layer, Conv layer, Downsample layer, and Upsample layer are also formed as convolution-batchnorm-relu.

Layer	Stride	Kernel	Chl(in/out)
Downsample layer1	1	7 × 7	3/64
Downsample layer2	2	3 × 3	64/128
Downsample layer3	2	3 × 3	128/256
Residual Block1-6	1	3 × 3	256/256
Upsample layer3	2	3 × 3	256/128
Upsample layer2	2	3 × 3	128/64
Upsample layer1	1	7 × 7	64/3

For G_e, inspired by the idea of a coarse-to-fine framework applied in pix2pixHD, we design G_e to learn low-level information, especially when the classes are defined on a fine scale. The features may be similar among some classes in our object generation scenario, so we use residual blocks and adapt the architecture proposed by Johson et al. [39]. The architecture comprises components: a convolutional head, set of residual blocks, and transposed convolutional end. We can use this architecture to take full advantage of the low-level features because the residual blocks perform better than encoder-decoder style convolutional layers when extracting image features. By using a large kernel in the head and end components separately, G_e can obtain a larger reception field. In addition, by using residual blocks, G_e can also obtain sufficient information to improve the final output with more class-specific textures and fine scale details. In particular, considering the resolution of our input semantic image, we only use six residual blocks and three convolutional layers separately for downsampling or upsampling.

3.2.2. Discriminator

In order to function better in certain tasks comprising translation and enhancement, the two discriminators do not share their parameters. The discriminator D_t operates with G_t in an image translation task is a patch-discriminator for generating fake/real details for each image patch. The other discriminator D_e function with G_e enhances the detail is a multi-scale discriminator comprising two identity patch-discriminators that accept images at different resolutions, where one accepts images at a resolution of 512×512 and the other accepts images at a resolution of 1024×1024 by upsampling the input image by a factor of $2\times$. Some previous studies used downsampling and a multi-scale discriminator to obtain better results from a global perspective. The enhancer is added to improve the details so using an upsampler can provide the discriminator with a more local view in order to focus on that details that may be omitted from a global view. Thus, D_e can help the enhancer to improve the quality of globally and locally generated images. More details of the architecture are given in Table 3.

Table 3
Patch-discriminator architecture details. Each patch-discriminator has six layers. Except for the last layer, each layer is formed as convolution-batchnorm-relu.

Layer	Stride	Kernel	D_t:chl(in/out)	D_e:chl(in/out)
Conv1	2	4 × 4	6/64	3/64
Conv2	2	4 × 4	64/128	64/128
Conv3	2	4 × 4	128/256	128/256
Conv4	2	4 × 4	256/256	256/256
Conv5	1	4 × 4	256/256	256/256
Conv6	1	4 × 4	256/1	256/1

3.3. Objective

We follow the objective of pix2pix by using $\mathcal{L}_1$ loss to provide a low-level constraint and adversarial loss to provide a high-level constraint. We also consider the popular perceptual loss but the VGG feature distance does not work well with our data set.

For the image translation part, components objective can be expressed as:

$$G_t = arg\min_{G_t}\max_{D_t} \mathcal{L}_{GAN_t}(G_t(x,z),D_t) + \lambda\mathcal{L}_1(G_t(x,z),y), \tag{5}$$

where x is an input semantic label map, z is random noise, y corresponds to the ground truth image, and $G(x, z)$ is the image generated based on the input semantic label map x and random noise z.

For the image enhancer component, the objective can be expressed as:

$$G_e = arg\min_{G_e}\max_{D_e} \mathcal{L}_{GAN_e}(G_e(G_t(x,z)),D_e) + \lambda\mathcal{L}_1(G_e(G_t(x,z)),y), \tag{6}$$

where $G_t(x, z)$ is the output from G_t. Thus, the final objective is as follows.

$$\begin{aligned} G^* = & arg\min_{G_t}\max_{D_t} \mathcal{L}_{GAN_t}(G_t(x,z),D_t) \\ & + arg\min_{G_e}\max_{D_e} \mathcal{L}_{GAN_e}(G_e(G_t(x,z)),D_e) \\ & + \lambda_t\mathcal{L}_1(G_t(x,z),y) + \lambda_e\mathcal{L}_1(G_e(G_t(x,z)),y) \end{aligned} \tag{7}$$

3.4. Training details

Our training process is divided into three stages. In stage 1, the translator GAN is trained. We want the translator GAN to learn the image translation at a high level, so we weaken the $\mathcal{L}_1$ constraint by decreasing the loss weight every 20 epochs. In stage 2, we fix the translator GAN and train the enhancer GAN. In stage 3, we fine tune both the translator GAN and the enhancer GAN. We actually remove $\mathcal{L}_1(G_t(x,z),y)$ from the final objective during the third training stage because it provides the same guidance as the $\mathcal{L}_1(G_e(G_t(x,z)),y)$ component and we do not want G_t to be disrupted frequently and violently. We set a certain layer in the generator with a dropout rate of 0.5 to simulate the random noise z mentioned above. We use the Adam solver with momentum parameters $\beta_1 = 0.5,\ \beta_2 = 0.999$. The learning rate is 0.0002 and the batch size is 1.

3.5. Evaluation criteria

Recent studies [8,10,31,40,41] evaluated the generation models with a re-segmentation method. This type of evaluation method uses a pre-trained segmented network, such as FCN [42] or PSPNet [43], to segment the generated images and calculate the mean IoU and pixel accuracy. However, the existing segmentation methods do not perform well with small objects. pix2pixHD only uses the re-segmentation evaluation method with the cityscapes data set, which has large semantic segmentation areas. By contrast, the objects are usually small in our object generation scenario. In addition, we expect class-specific textures and fine scale details, which means that high quality generated image should be similar to the target. Therefore, we employ image reconstruction evaluation criteria comprising SSIM and PSNR to quantify the quality of the generated images compared with the ground truth.

We use SSIM to measure the similarity between the generated image and ground truth image. The SSIM between two images x and y is defined as:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \tag{8}$$

where x is the generated image and y is the ground truth image. μ and δ are the average and variance of the image, respectively, and c_1 and c_2 are two variables that stabilize the division and they are determined by the dynamic range of the images.

The PSNR is also used widely to measure the quality of reconstructed images. The PSNR between two images x and y can be defined as:

$$PSNR(x,y) = 10log\left(\frac{\mathrm{Max}^2}{MSE(x,y)}\right), \tag{9}$$

where Max is the maximum possible pixel value of the image, and MSE is the sum of each pixel's mean squared error between x and y divided by the image size and by the channel number, e.g., 3 for an RGB image.

We tested our framework based on some randomly generated semantic label images and generated maps without corresponding ground truth data. We conducted a user study for these generated images, where we displayed each pair of images for an unlimited time and the users were asked to select the most realistic. Two images generated using the pix2pix method and our framework with the same inputs were paired randomly in a left to right order. Each user was randomly presented with 10 from 100 pairs in the study.

4. Experiments

Our framework is designed to generate realistic images of objects based on semantic label maps. In our object generation scenario, we expect image translation with more fine-scale details and class-specific textures, which were not considered in previous studies. Next, we explain the specific data forms used in our study. Our object data set used for training and testing comprised RGB images, where each image contained a number of objects at a resolution of 512 × 512. Each type of object belonged to a different semantic class. The outputs from our framework comprised simple RGB images that satisfied the semantic constraints, and they contained many class-specific textures and fine-scale details. The two types of semantic label maps used in our experiments are shown in Fig. 3.

We used the corresponding test data set to evaluate the trained model. In order to verify that our framework could generate images for novel scenarios, we used another test data set to evaluate its performance. The semantic label maps in this test data set contained instance-level bounding boxes generated by the algorithm with random layouts of the objects. In the experiments, we determined the mean SSIM and mean PSNR for the test data with the ground truth, and we conducted a user study with the test data without ground truth data.

4.1. Evaluation

Our framework comprised a translator GAN and enhancer GAN. The enhancer GAN was used to augment the generated images

Table 4
Experimental results obtained based on evaluations with different architecture designs for our networks. We trained our method based on different data sets and various versions of the enhancer GAN, and also without the enhancer GAN. OBD denotes the semantic label map (512×512) based on the object data set with information for the instance-level bounding boxes. OD denotes the semantic label map ((512×512) based on the object data set without information for the instance-level bounding boxes. The G_e column shows the different architecture designs. w or w/o indicate whether an enhancer GAN was applied or not. R shows the use of residual blocks whereas U indicates the use of U-Net in G_e. M denotes the use of the multi-scale discriminator in the enhancer GAN, whereas S shows that a single-scale discriminator was employed in the enhancer GAN. Higher values of *SSIM/PSNR* are better.

Method	G_e	Data	SSIM(mean)	PSNR(mean)
Ours	w;R;M	OBD	0.3856	17.7215
Ours-OD	w;R;M	OD	0.3705	17.1173
Ours-noG_e	w/o	OBD	0.3720	17.6126
Ours-sD_e	w;R;S	OBD	0.3766	17.9120
Ours-2U	w;U;M	OBD	0.3721	17.7826

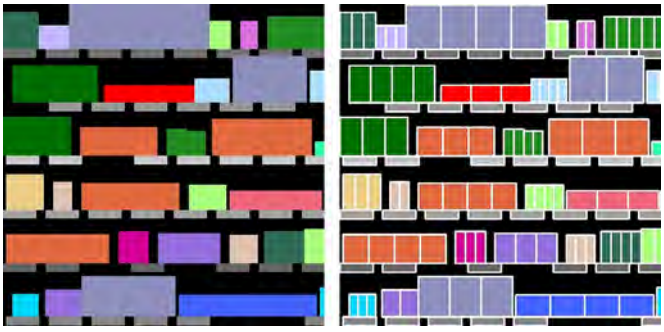

Fig. 3. Left: semantic label map. Right: instance-level semantic label map represented by bounding boxes. One semantic area could correspond to more than one object and the information for the instance-level bounding boxes is shown by white lines in the image on the right.

Fig. 4. Comparison of the results obtained using the models trained with: (1) ours: our method trained using object data with information for the instance-level bounding boxes, and with the enhancer GAN using residual blocks and a multi-scale discriminator; (2) ours-OD: the same design as (1) but trained using object data without information for the instance-level bounding boxes; (3) ours-noG_e: the same design as (1) but without the enhancer GAN; (4) ours-sD_e: the multi-scale discriminator was replaced with a single-scale discriminator in the enhancer; and (5) ours-2U: the GAN enhancer was used in the U-Net mode.

with higher quality and more details. Thus, we visualized the outputs from the translator GAN and enhancer GAN by evaluating the results generated using two versions of our framework. One version comprised our framework with the translator GAN trained based on our objects data set at a resolution of 512×512 with an instance-level semantic label map represented by bounding boxes. The other version comprised our whole framework trained with the same data. The experimental results are shown in Table 4 and Fig. 4.

The experimental results shown in Fig. 4 and Table 4 demonstrate that the division and cooperation between the translator and enhancer were as expected. The difference in the results obtained with or without the enhancer verified the ability of G_e to guide the method to produce results with more details and textures.

Table 5
Experimental results: We used the pix2pix method and our method trained separately based on CMP Facades at a resolution of 256×256 and the object data set (OBD) at a resolution of 512×512. Higher values of *SSIM/PSNR* are better.

Method	Data	SSIM(mean)	PSNR(mean)
Pix2pix	CMP Facades	0.1501	9.3567
Ours	CMP Facades	0.1559	9.4546
Pix2pix	OBD	0.3742	17.4331
Ours	OBD	0.3856	17.7215

In the network architecture design evaluation, we compared our framework with different enhancer GAN designs. We employed residual blocks for the enhancer GAN. In order to prove that the residual blocks performed better than U-Net in the enhancer GAN, we compared our framework with two U-Nets stacked architectures. We also compared the use of a single-scale discriminator and a multi-scale discriminator in the enhancer GAN. The experimental results with different architecture designs are compared in Table 4 and Fig. 4.

The results in Table 4 and Fig. 4 show that better performance was obtained with residual blocks. The translator GAN and enhancer GAN both accept images as inputs, but the two GANs have different roles at different levels of the image generation process. The translator GAN aims to translate semantic label maps represented by bounding boxes into photo-realistic images, whereas the enhancer GAN is designed to augment the input image with more textures and details. The semantic label maps represented by bounding boxes provide coarse level details and the corresponding information for different region. The enhancer GAN should preserve the original textures and details in the inputs, but also improve the fine scale information. Residual blocks performed better than U-Net at a fine scale and this method lost less low-level information. The single-scale discriminator obtained a higher PSNR but the generated images contained more blur with less strong texture and details compared with the images generated by the multi-scale discriminator. We consider that upsampling with the multi-scale discriminator could focus on more of the fine scale information because the discrimination process corresponded to more small areas in the images. In particular, the single-scale discriminator received inputs at a resolution of 64×64 and yielded discrimination outputs at a resolution of 16×16, whereas the discriminator with upsampling received inputs at a resolution of 128×128 and the discrimination output had a resolution of 32×32. The discrimination output actually corresponded to the same resolution of 64×64, which can be regarded as the output generated by the discriminator with upsampling focused at a finer scale. The results obtained using different discriminator designs are compared in Table 4 and columns two and six in Fig. 4.

We also conducted experiments to test the improvements achieved by using the information for the instance-level bounding boxes. We trained the whole framework based on our object data set with or without information for the instance-level bounding boxes. The results shown in Fig. 4 and Table 4 demonstrate that the model trained with information for the instance-level bounding boxes improved the quality of the generated images. The information for the bounding boxes helped the GAN to simplify the class-and-number generation problem to a class-only generation problem, so the GAN did not need to consider the number of the objects that a semantic area should contain.

4.2. Comparison

Our framework is based on pix2pix and it focuses on a highly specific scenario, so we mainly conducted comparisons with pix2pix as the baseline to demonstrate that our modifications are

Fig. 5. Comparison of the results obtained using models trained with: (1) our method based on CMP Facades; and (2) the pix2pix method based on CMP Facades.

Fig. 6. Comparison of the results obtained using models trained with: (1) our method based on our object data set; and (2) the pix2pix method based on our object data set.

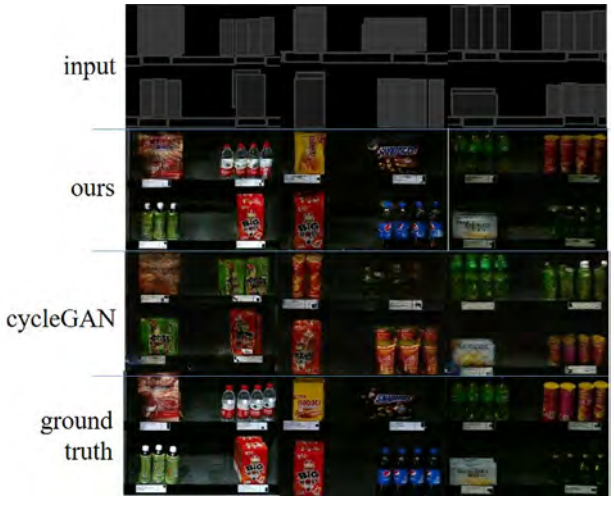

Fig. 7. Results generated using our method and cycleGAN. cycleGAN lack constraints on the object semantic class so there were many incorrect matches among the semantic label maps and generated images.

effective. We conducted comparisons with pix2pix based on CMP Facades data at a resolution of 256×256 and our object data set at a resolution of 512×512.

According to Table 5, our framework performed better than pix2pix using both CMP Facades and our object data set, as also shown by the generated images in Figs. 5 and 6. We also found that the images generated using our method contained more fine-scale details and class-specific textures.

Table 6
Experimental results: We compared our method with cycleGAN and pix2pixHD based on the object data set (OBD). Higher values of *SSIM*/*PSNR* are better.

Method	Data	SSIM(mean)	PSNR(mean)
Ours	OBD	0.3856	17.7215
cycleGAN	OBD	0.3704	15.3467
pix2pixHD	OBD	0.3065	15.8088

We also conducted comparisons with cycleGAN and pix2pixHD, and the results are shown in Fig. 7 and Table 6. If we consider that style transfer is a type of image translation, then we can conduct comparisons with cycleGAN, which performs particularly well in the style transfer task. Our method obtained higher SSIM and PSNR scores than cycleGAN, but we cannot claim that it is better. First, there were many incorrect matches between the semantic label map and the images generated using cycleGAN, which led to low SSIM and PSNR scores. Second, the cycleGAN method is mainly focused on image style transfer, so the sample network design was not suitable for translating the semantic label map represented by bounding boxes containing only coarse level information into images with sufficient textures and details. Finally, the cycleGAN method was developed for cycle learning with unpaired images, whereas our framework can be regarded as

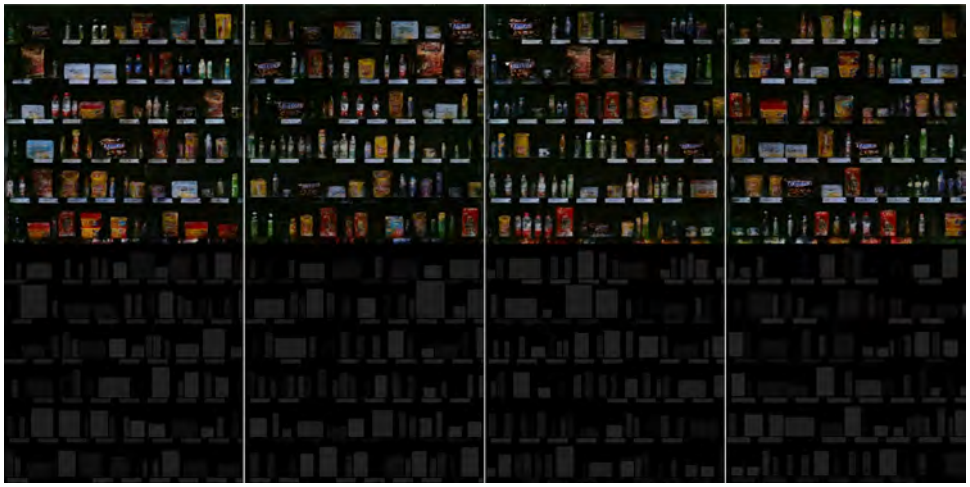

Fig. 8. Results generated by our framework with a randomly generated semantic label map as the input. The first row shows the generated images and the second row presents the corresponding semantic label map represented by randomly generated bounding boxes.

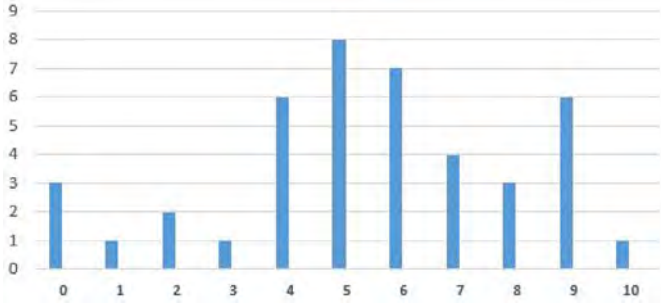

Fig. 9. User study results. x-axis: number of times the images generated by our method were selected as real by a user; *y*-axis: number of users who gave the same number of votes.

task-separated learning. These two types of learning can also be combined, such as by using our framework to replace the basic GAN in cycleGAN. We lacked instance boundary information, so we conducted comparisons with pix2pixHD by removing the instance-related information, including the instance boundary, instance feature, and VGG loss. Our method and pix2pixHD can both generate images from the semantic label map, but they focus on different improvements. pix2pixHD aims to generate images at a high resolution using instance information, whereas our method generates images with more details by using the new semantic label represented by bounding boxes. pix2pixHD obtained fairly low SSIM and PSNR scores because many instances and perceptual loss-related components were are unavailable and it was only guided by the GAN-related loss. Fig. 7 shows that the images generated using our method contained more details and textures.

We invited 42 people to participate in a user study with randomly generated data. Each user was presented with 10 pairs of images generated by our method and the pix2pix method, where they were placed randomly in a left to right order. For all of the pairs, each user selected the more realistic generated image according to their opinion without any time limitation. The images generated using our method received more than half of the votes. The images generated using our method that received more votes were more realistic in terms of their shape, color, and details. Some of the images generated using our method contained regions with sufficient details but also regions with less details, which led to a strong contrast and it may have reduce the realistic effect to some degree. The results of the user study demonstrated that our method performed better compared with pix2pix, but the global consistency and the uniform distribution of the details should be improved further. The votes given by the participants are shown in Fig. 9. Some of the images generated using our method and tested in the user studies are showed in Fig. 8.

5. Conclusions

In this study, we proposed a framework for generating images based on semantic information labeled with instance-level bounding boxes. Our framework extends the previous state-of-the-art method by adding a novel translator-enhancer structure and it improved the image quality in terms of both the SSIM and PSNR scores, and according to the results of a user study. Representing the semantic information based on regular boxes makes it easier and more flexible for users to prepare the input and change the layout of the target image, and the representations with bounding boxes can help to generate better results based on object boundaries. Our method only focuses on generating some specific objects and the aim is to provide images with more class-specific details and textures. The translator-enhancer pipeline could also be useful in various other applications, which requires further exploration. In addition, we only use the L1 loss to support adversarial learning and other loss function designs should be tested in future research.

Acknowledgments

This study was supported by the NSFC (Nos. 61822111, 61727808, 61671268, 61672307, and 61562063) and Beijing Natural Science Foundation (L182052).

References

[1] Luo B, Xu F, Richardt C, Yong J-H. Parallax360: Stereoscopic 360 scene representation for head-motion parallax. IEEE Trans. Vis. Comput. Graph. 2018;24(4):1545–53.
[2] Yan S, Wu C, Wang L, Xu F, An L, Guo K, et al. Ddrnet: Depth map denoising and refinement for consumer depth cameras using cascaded cnns. In: Proceedings of the European conference on computer vision (ECCV); 2018. p. 151–67.
[3] Lu M, Zhao H, Yao A, Xu F, Chen Y, Zhang L. Decoder network over lightweight reconstructed feature for fast semantic style transfer. In: Proceedings of the IEEE international conference on computer vision; 2017. p. 2469–77.
[4] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. In: Proceedings of the advances in neural information processing systems; 2014. p. 2672–80.
[5] Dosovitskiy A, Tobias Springenberg J, Brox T. Learning to generate chairs with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 1538–46.
[6] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:151106434 2015.

[7] Karacan L, Akata Z, Erdem A, Erdem E. Learning to generate images of outdoor scenes from attributes and semantic layouts. arXiv preprint arXiv:161200215 2016.
[8] Isola P, Zhu J-Y, Zhou T, Efros AA. Image-to-image translation with conditional adversarial networks. In: Proceedings of the 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE; 2017. p. 5967–76.
[9] Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks. arXiv preprint arXiv:181204948 2018.
[10] Wang T-C, Liu M-Y, Zhu J-Y, Tao A, Kautz J, Catanzaro B. High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018a. p. 8798–807.
[11] Dosovitskiy A, Brox T. Generating images with perceptual similarity metrics based on deep networks. In: Proceedings of the advances in neural information processing systems; 2016. p. 658–66.
[12] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556 2014.
[13] Denton EL, Chintala S, Fergus R, et al. Deep generative image models using a? Laplacian pyramid of adversarial networks. In: Proceedings of the advances in neural information processing systems; 2015. p. 1486–94.
[14] Zhang H, Xu T, Li H, Zhang S, Wang X, Huang X, et al. Stackgan: text to photo-realistic image synthesis with stacked generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision; 2017. p. 5907–15.
[15] Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel P. Infogan: interpretable representation learning by information maximizing generative adversarial nets. In: Proceedings of the advances in neural information processing systems; 2016. p. 2172–80.
[16] Arjovsky M, Chintala S, Bottou L. Wasserstein gan. arXiv preprint arXiv:170107875 2017.
[17] Barnes C, Zhang F-L. A survey of the state-of-the-art in patch-based synthesis. Comput. Vis. Med. 2017;3(1):3–20.
[18] Kingma DP, Welling M. Auto-encoding variational bayes. arXiv preprint arXiv:13126114 2013.
[19] van den Oord A, Kalchbrenner N, Espeholt L, Vinyals O, Graves A, et al. Conditional image generation with pixelcnn decoders. In: Proceedings of the advances in neural information processing Systems; 2016. p. 4790–8.
[20] Kahng M, Thorat N, Chau DHP, Viégas FB, Wattenberg M. Gan lab: understanding complex deep generative models using interactive visual experimentation. IEEE Trans. Vis. Comput. Graph. 2019;25(1):310–20.
[21] Wang J, Gou L, Yang H, Shen H-W. Ganviz: a visual analytics approach to understand the adversarial game. IEEE transactions on visualization and computer graphics 2018b;24(6):1905–17.
[22] Gauthier J. Conditional generative adversarial nets for convolutional face generation. Class project for stanford CS231N: convolutional neural networks for visual recognition, winter semester, 2014; 2014. p. 2.
[23] Donahue J, Krähenbühl P, Darrell T. Adversarial feature learning. arXiv preprint arXiv:160509782 2016.
[24] Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Proceedings of the advances in neural information processing systems; 2017. p. 6626–37.
[25] Reed S, Akata Z, Yan X, Logeswaran L, Schiele B, Lee H. Generative adversarial text to image synthesis. arXiv preprint arXiv:160505396 2016a.
[26] Reed SE, Akata Z, Mohan S, Tenka S, Schiele B, Lee H. Learning what and where to draw. In: Proceedings of the advances in neural information processing systems; 2016b. p. 217–25.
[27] Liu M-Y, Tuzel O. Coupled generative adversarial networks. In: Proceedings of the advances in neural information processing systems; 2016. p. 469–77.
[28] Zhu J-Y, Krähenbühl P, Shechtman E, Efros AA. Generative visual manipulation on the natural image manifold. In: Proceedings of the European conference on computer vision. Springer; 2016. p. 597–613.
[29] Pathak D, Krahenbuhl P, Donahue J, Darrell T, Efros AA. Context encoders: feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 2536–44.
[30] Ledig C, Theis L, Huszár F, Caballero J, Cunningham A, Acosta A, et al. Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE; 2017. p. 105–14.
[31] Wang X, Gupta A. Generative image modeling using style and structure adversarial networks. In: Proceedings of the European conference on computer vision. Springer; 2016. p. 318–35.
[32] Wu X, Li RL, Zhang FL, Liu JC, Wang J, Shamir A, et al. Deep portrait image completion and extrapolation. arXiv preprint arXiv:180807757 2018.
[33] Yoo D, Kim N, Park S, Paek AS, Kweon IS. Pixel-level domain transfer. In: Proceedings of the European conference on computer vision. Springer; 2016. p. 517–32.
[34] Li C, Wand M. Precomputed real-time texture synthesis with Markovian generative adversarial networks. In: Proceedings of the European conference on computer vision. Springer; 2016. p. 702–16.
[35] Wu J, Zhang C, Xue T, Freeman B, Tenenbaum J. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In: Proceedings of the advances in neural information processing systems; 2016. p. 82–90.
[36] Vondrick C, Pirsiavash H, Torralba A. Generating videos with scene dynamics. In: Proceedings of the advances in neural information processing systems; 2016. p. 613–21.
[37] Kumar P, Nagar P, Arora C, Gupta A. U-segnet: fully convolutional neural network based automated brain tissue segmentation tool. In: Proceedings of the 2018 25th IEEE international conference on image processing (ICIP). IEEE; 2018. p. 3503–7.
[38] Zhang F-L, Wu X, Li R-L, Wang J, Zheng Z-H, Hu S-M. Detecting and removing visual distractors for video aesthetic enhancement. IEEE Trans Multimed 2018;20(8):1987–99.
[39] Johnson J, Alahi A, Fei-Fei L. Perceptual losses for real-time style transfer and super-resolution. In: Proceedings of the European conference on computer vision. Springer; 2016. p. 694–711.
[40] Zhang R, Isola P, Efros AA. Colorful image colorization. In: Proceedings of the European conference on computer vision. Springer; 2016. p. 649–66.
[41] Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X. Improved techniques for training gans. In: Proceedings of the advances in neural information processing systems; 2016. p. 2234–42.
[42] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 3431–40.
[43] Zhao H, Shi J, Qi X, Wang X, Jia J. Pyramid scene parsing network. In: Proceedings of the ieee conference on computer vision and pattern recognition (CVPR); 2017. p. 2881–90.