High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs

Ting-Chun Wang[1] Ming-Yu Liu[1] Jun-Yan Zhu[2] Andrew Tao[1] Jan Kautz[1] Bryan Catanzaro[1]
[1] NVIDIA Corporation [2] UC Berkeley

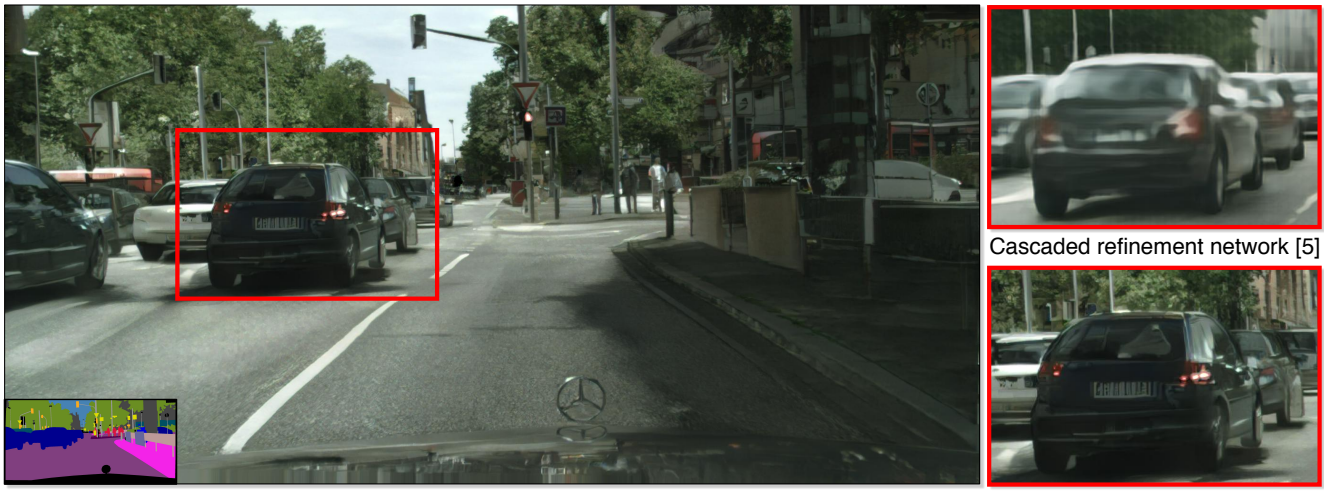


Figure 1: We propose a generative adversarial framework for synthesizing 2048×1024 images from semantic label maps (lower left corner in (a)). Compared to previous work [5], our results express more natural textures and details. (b) We can change labels in the original label map to create new scenes, like replacing trees with buildings. (c) Our framework also allows a user to edit the appearance of individual objects in the scene, e.g. changing the color of a car or the texture of a road. Please visit our website for more side-by-side comparisons as well as interactive editing demos.

Abstract

We present a new method for synthesizing high-resolution photo-realistic images from semantic label maps using conditional generative adversarial networks (conditional GANs). Conditional GANs have enabled a variety of applications, but the results are often limited to low-resolution and still far from realistic. In this work, we generate 2048×1024 *visually appealing results with a novel adversarial loss, as well as new multi-scale generator and discriminator architectures. Furthermore, we extend our framework to interactive visual manipulation with two additional features. First, we incorporate object instance segmentation information, which enables object manipulations such as removing/adding objects and changing the object category. Second, we propose a method to generate diverse results given the same input, allowing users to edit the object appearance interactively. Human opinion studies demonstrate that our method significantly outperforms existing methods, advancing both the quality and the resolution of deep image synthesis and editing.*

1. Introduction

Rendering photo-realistic images using standard graphics techniques is involved, since geometry, materials, and light transport must be simulated explicitly. Although existing graphics algorithms excel at the task, building and editing virtual environments is expensive and time-consuming. That is because we have to model every aspect of the world explicitly. If we were able to render photo-realistic images using a model learned from data, we could turn the process of graphics rendering into a model learning and inference problem. Then, we could simplify the process of creating new virtual worlds by training models on new datasets. We could even make it easier to customize environments by allowing users to simply specify semantic information rather than modeling geometry, materials, or lighting.

In this paper, we discuss a new approach that produces high-resolution images from semantic label maps. This method has a wide range of applications. For example, we can use it to create synthetic training data for training visual recognition algorithms, since it is much easier to create semantic labels for desired scenarios than to generate training images. Using semantic segmentation methods, we can transform images into a semantic label domain, edit the objects in the label domain, and then transform them back to the image domain. This method also gives us new tools for higher-level image editing, e.g., adding objects to images or changing the appearance of existing objects.

To synthesize images from semantic labels, one can use the pix2pix method, an image-to-image translation framework [21] which leverages generative adversarial networks (GANs) [16] in a conditional setting. Recently, Chen and Koltun [5] suggest that adversarial training might be unstable and prone to failure for high-resolution image generation tasks. Instead, they adopt a modified perceptual loss [11, 13, 22] to synthesize images, which are high-resolution but often lack fine details and realistic textures.

Here we address two main issues of the above state-of-the-art methods: (1) the difficulty of generating high-resolution images with GANs [21] and (2) the lack of details and realistic textures in the previous high-resolution results [5]. We show that through a new, robust adversarial learning objective together with new multi-scale generator and discriminator architectures, we can synthesize photo-realistic images at 2048×1024 resolution, which are more visually appealing than those computed by previous methods [5,21]. We first obtain our results with adversarial training only, without relying on any hand-crafted losses [43] or pre-trained networks (e.g. VGGNet [47]) for perceptual losses [11,22] (Figs. 7c, 9b). Then we show that adding perceptual losses from pre-trained networks [47] can slightly improve the results in some circumstances (Figs. 7d, 9c), if a pre-trained network is available. Both results outperform previous works substantially in terms of image quality.

Furthermore, to support interactive semantic manipulation, we extend our method in two directions. First, we use instance-level object segmentation information, which can separate different object instances within the same category. This enables flexible object manipulations, such as adding/removing objects and changing object types. Second, we propose a method to generate diverse results given the same input label map, allowing the user to edit the appearance of the same object interactively.

We compare against state-of-the-art visual synthesis systems [5, 21], and show that our method outperforms these approaches regarding both quantitative evaluations and human perception studies. We also perform an ablation study regarding the training objectives and the importance of instance-level segmentation information. Our code and data are available at our website.

2. Related Work

Generative adversarial networks Generative adversarial networks (GANs) [16] aim to model the natural image manifold by forcing the generated samples to be indistinguishable from natural images. GANs enable a wide variety of applications such as image generation [1, 41, 60], representation learning [44], image manipulation [62], object detection [32], and video applications [37, 52]. Various coarse-to-fine schemes [4] have been proposed [9, 19, 55] to synthesize larger images (e.g. 256×256) in an unconditional setting. Inspired by their successes, we propose a new coarse-to-fine generator and multi-scale discriminator architectures suitable for conditional image generation at a much higher resolution.

Image-to-image translation Many researchers have leveraged adversarial learning for image-to-image translation [21], whose goal is to translate an input image from one domain to another domain given input-output image pairs as training data. Compared to L_1 loss, which often leads to blurry images [21, 22], the adversarial loss [16] has become a popular choice for many image-to-image tasks [10, 24, 25, 31, 40, 45, 53, 58, 64]. The reason is that the discriminator can learn a trainable loss function and automatically adapt to the differences between the generated and real images in the target domain. For example, the recent pix2pix framework [21] used image-conditional GANs [38] for different applications, such as transforming Google maps to satellite views and generating cats from user sketches. Various methods have also been proposed to learn an image-to-image translation in the absence of training pairs [2, 33, 34, 46, 49, 50, 54, 63].

Recently, Chen and Koltun [5] suggest that it might be hard for conditional GANs to generate high-resolution images due to the training instability and optimization issues. To avoid this difficulty, they use a direct regression objective

based on a perceptual loss [11, 13, 22] and produce the first model that can synthesize 2048×1024 images. The generated results are high-resolution but often lack fine details and realistic textures. Our method is motivated by their success. We show that using our new objective function as well as novel multi-scale generators and discriminators, we not only largely stabilize the training of conditional GANs on high-resolution images, but also achieve significantly better results compared to Chen and Koltun [5]. Side-by-side comparisons clearly show our advantage (Figs. 1, 7, 8, 9).

Deep visual manipulation Recently, deep neural networks have obtained promising results in various image processing tasks, such as style transfer [13], inpainting [40], colorization [56], and restoration [14]. However, most of these works lack an interface for users to adjust the current result or explore the output space. To address this issue, Zhu *et al*. [62] developed an optimization method for editing the object appearance based on the priors learned by GANs. Recent works [21, 45, 57] also provide user interfaces for creating novel imagery from low-level cues such as color and sketch. All of the prior works report results on low-resolution images. Our system shares the same spirit as this past work, but we focus on object-level semantic editing, allowing users to interact with the entire scene and manipulate individual objects in the image. As a result, users can quickly create a novel scene with minimal effort. Our interface is inspired by prior data-driven graphics systems [6, 23, 28]. But our system allows more flexible manipulations and produces high-res results in real-time.

3. Instance-Level Image Synthesis

We propose a conditional adversarial framework for generating high-resolution photo-realistic images from semantic label maps. We first review our baseline model pix2pix (Sec. 3.1). We then describe how we increase the photorealism and resolution of the results with our improved objective function and network design (Sec. 3.2). Next, we use additional instance-level object semantic information to further improve the image quality (Sec. 3.3). Finally, we introduce an instance-level feature embedding scheme to better handle the multi-modal nature of image synthesis, which enables interactive object editing (Sec. 3.4).

3.1. The pix2pix Baseline

The pix2pix method [21] is a conditional GAN framework for image-to-image translation. It consists of a generator G and a discriminator D. For our task, the objective of the generator G is to translate semantic label maps to realistic-looking images, while the discriminator D aims to distinguish real images from the translated ones. The framework operates in a supervised setting. In other words, the training dataset is given as a set of pairs of corresponding images $\{(\mathbf{s_i}, \mathbf{x_i})\}$, where $\mathbf{s_i}$ is a semantic label map and $\mathbf{x_i}$ is a corresponding natural photo. Conditional GANs aim to model the conditional distribution of real images given the input semantic label maps via the following minimax game:

$$\min_G \max_D \mathcal{L}_{\text{GAN}}(G, D) \tag{1}$$

where the objective function $\mathcal{L}_{GAN}(G, D)$ [1] is given by

$$\mathbb{E}_{(\mathbf{s},\mathbf{x})}[\log D(\mathbf{s}, \mathbf{x})] + \mathbb{E}_{\mathbf{s}}[\log(1 - D(\mathbf{s}, G(\mathbf{s}))]. \tag{2}$$

The pix2pix method adopts U-Net [42] as the generator and a patch-based fully convolutional network [35] as the discriminator. The input to the discriminator is a channel-wise concatenation of the semantic label map and the corresponding image. However, the resolution of the generated images on Cityscapes [7] is up to 256×256. We tested directly applying the pix2pix framework to generate high-resolution images, but found the training unstable and the quality of generated images unsatisfactory. We therefore describe how we improve the pix2pix framework in the next subsection.

3.2. Improving Photorealism and Resolution

We improve the pix2pix framework by using a coarse-to-fine generator, a multi-scale discriminator architecture, and a robust adversarial learning objective function.

Coarse-to-fine generator We decompose the generator into two sub-networks: G_1 and G_2. We term G_1 as the global generator network and G_2 as the local enhancer network. The generator is then given by the tuple $G = \{G_1, G_2\}$ as visualized in Fig. 2. The global generator network operates at a resolution of 1024×512, and the local enhancer network outputs an image with a resolution that is $4\times$ the output size of the previous one ($2\times$ along each image dimension). For synthesizing images at an even higher resolution, additional local enhancer networks could be utilized. For example, the output image resolution of the generator $G = \{G_1, G_2\}$ is 2048×1024, and the output image resolution of $G = \{G_1, G_2, G_3\}$ is 4096×2048.

Our global generator is built on the architecture proposed by Johnson *et al*. [22], which has been proven successful for neural style transfer on images up to 512×512. It consists of 3 components: a convolutional front-end $G_1^{(F)}$, a set of residual blocks $G_1^{(R)}$ [18], and a transposed convolutional back-end $G_1^{(B)}$. A semantic label map of resolution 1024×512 is passed through the 3 components sequentially to output an image of resolution 1024×512.

The local enhancer network also consists of 3 components: a convolutional front-end $G_2^{(F)}$, a set of residual blocks $G_2^{(R)}$, and a transposed convolutional back-end

[1] we denote $\mathbb{E}_{\mathbf{s}} \triangleq \mathbb{E}_{\mathbf{s}\sim p_{\text{data}}(s)}$ and $\mathbb{E}_{(\mathbf{s},\mathbf{x})} \triangleq \mathbb{E}_{(\mathbf{s},\mathbf{x})\sim p_{\text{data}}(\mathbf{s},\mathbf{x})}$ for simplicity.

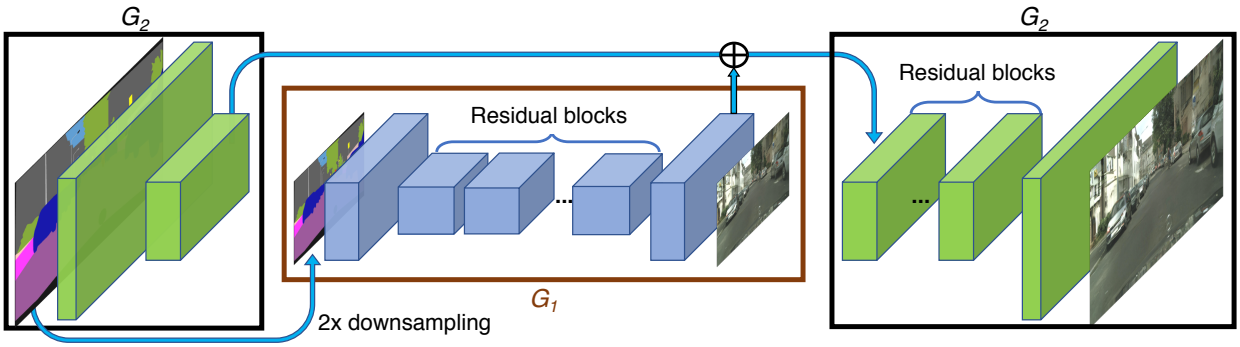


Figure 2: Network architecture of our generator. We first train a residual network G_1 on lower resolution images. Then this network is used to initialize our final network G_2 trained on high resolution images. Specifically, the input to the residual blocks in G_2 is the element-wise sum of the feature map from G_2 and the last feature map from G_1.

$G_2^{(B)}$. The resolution of the input label map to G_2 is 2048×1024. Different from the global generator network, the input to the residual block $G_2^{(R)}$ is the element-wise sum of two feature maps: the output feature map of $G_2^{(F)}$, and the last feature map of the back-end of the global generator network $G_1^{(B)}$. This helps integrating the global information from G_1 to G_2.

During training, we first train the global generator and then train the local enhancer in the order of their resolutions. We then jointly fine-tune all the networks together. We use this generator design to effectively aggregate global and local information for the image synthesis task. We note that such a multi-resolution pipeline is a well-established practice in computer vision [4] and two-scale is often enough [3]. Similar ideas but different architectures could be found in recent unconditional GANs [9, 19] and conditional image generation [5, 55].

Multi-scale discriminators High-resolution image synthesis poses a great challenge to the GAN discriminator design. To differentiate high-resolution real and synthesized images, the discriminator needs to have a large receptive field. This would require either a deeper network or larger convolutional kernels. As both choices lead to an increased network capacity, overfitting would become more of a concern. Also, both choices require a larger memory footprint for training, which is already a scarce resource for high-resolution image generation.

To address the issue, we propose using multi-scale discriminators. We use 3 discriminators that have an identical network structure but operate at different image scales. We will refer to the discriminators as D_1, D_2 and D_3. Specifically, we downsample the real and synthesized high-resolution images by a factor of 2 and 4 to create an image pyramid of 3 scales. The discriminators D_1, D_2 and D_3 are then trained to differentiate real and synthesized images at the 3 different scales, respectively. Although the discriminators have an identical architecture, the one that operates at the coarsest scale has the largest receptive field. It has a more global view of the image and can guide the generator to generate globally consistent images. On the other hand, the discriminator operating at the finest scale is specialized in guiding the generator to produce finer details. This also makes training the coarse-to-fine generator easier, since extending a low-resolution model to a higher resolution only requires adding an additional discriminator at the finest level, rather than retraining from scratch. Without the multi-scale discriminators, we observe that many repeated patterns often appear in the generated images.

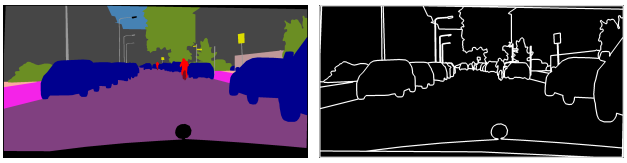
With the discriminators, the learning problem in Eq. (1) then becomes a multi-task learning problem of

$$\min_G \max_{D_1,D_2,D_3} \sum_{k=1,2,3} \mathcal{L}_{\text{GAN}}(G, D_k). \tag{3}$$

Using multiple GAN discriminators at the same image scale has been proposed in unconditional GANs [12]. Iizuka et al. [20] add a global image classifier to conditional GANs to synthesize globally coherent content for inpainting. Here we extend the design to multiple discriminators at different image scales for modeling high-resolution images.

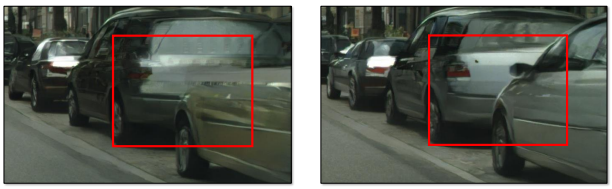
Improved adversarial loss We improve the GAN loss in Eq. (2) by incorporating a feature matching loss based on the discriminator. This loss stabilizes the training as the generator has to produce natural statistics at multiple scales. Specifically, we extract features from multiple layers of the discriminator, and learn to match these intermediate representations from the real and the synthesized image. For ease of presentation, we denote the ith-layer feature extractor of discriminator D_k as $D_k^{(i)}$ (from input to the ith layer of D_k). The feature matching loss $\mathcal{L}_{\text{FM}}(G, D_k)$ is then calculated as:

$$\mathcal{L}_{\text{FM}}(G, D_k) = \mathbb{E}_{(\mathbf{s},\mathbf{x})} \sum_{i=1}^{T} \frac{1}{N_i} [||D_k^{(i)}(\mathbf{s}, \mathbf{x}) - D_k^{(i)}(\mathbf{s}, G(\mathbf{s}))||_1], \tag{4}$$



(a) Semantic labels (b) Boundary map

Figure 3: Using instance maps: (a) a typical semantic label map. Note that all connected cars have the same label, which makes it hard to tell them apart. (b) The extracted instance boundary map. With this information, separating different objects becomes much easier.



(a) Using labels only (b) Using label + instance map

Figure 4: Comparison between results without and with instance maps. It can be seen that when instance boundary information is added, adjacent cars have sharper boundaries.

where T is the total number of layers and N_i denotes the number of elements in each layer. Our GAN discriminator feature matching loss is related to the perceptual loss [11, 13, 22], which has been shown to be useful for image super-resolution [31] and style transfer [22]. In our experiments, we discuss how the discriminator feature matching loss and the perceptual loss can be jointly used for further improving the performance. We note that a similar loss is used for training VAE-GANs [29].

Our full objective combines both GAN loss and feature matching loss as:

$$\min_G \left(\left(\max_{D_1,D_2,D_3} \sum_{k=1,2,3} \mathcal{L}_{\text{GAN}}(G, D_k) \right) + \lambda \sum_{k=1,2,3} \mathcal{L}_{\text{FM}}(G, D_k) \right) \tag{5}$$

where λ controls the importance of the two terms. Note that for the feature matching loss $\mathcal{L}_{\text{FM}}$, D_k only serves as a feature extractor and does not maximize the loss $\mathcal{L}_{\text{FM}}$.

3.3. Using Instance Maps

Existing image synthesis methods only utilize semantic label maps [5, 21, 25], an image where each pixel value represents the object class that the pixel belongs to. This map does not differentiate objects of the same class. On the other hand, an instance-level semantic label map contains a unique object ID for each individual object. To incorporate the instance map, a simple way would be to directly pass it into the network, or encode it into a one-hot vector. However, both approaches are difficult to implement in practice,

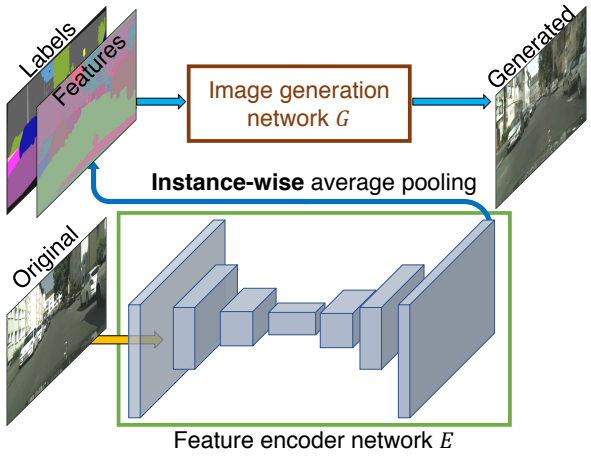


Figure 5: Using instance-wise features in addition to labels for generating images.

since different images may contain different numbers of objects of the same category. A simple solution would be to pre-allocate a fixed number of channels (e.g. 10) for each class, but it fails when the number is set too small, and wastes memory when the number is too large.

Instead, we argue that the most important information the instance map provides, which is not available in the semantic label map, is the object boundary. For example, when a number of same-class objects are next to one another, looking at the semantic label map alone cannot tell them apart. This is especially true for the street scene since many parked cars or walking pedestrians are often next to one another, as shown in Fig. 3a. However, with the instance map, separating these objects becomes an easier task.

Therefore, to extract this information, we first compute the instance boundary map (Fig. 3b). In our implementation, a pixel in the instance boundary map is 1 if its object ID is different from any of its 4-neighbors, and 0 otherwise. The instance boundary map is then concatenated with the one-hot vector representation of the semantic label map, and fed into the generator network. Similarly, the input to the discriminator is the channel-wise concatenation of instance boundary map, semantic label map, and the real/synthesized image. Figure 4b shows an example demonstrating the improvement by using object boundaries. Our user study in Sec. 4 also shows the model trained with instance boundary maps renders more photo-realistic object boundaries.

3.4. Learning an Instance-level Feature Embedding

Image synthesis from semantic label maps is a one-to-many mapping problem. An ideal image synthesis algorithm should be able to generate diverse realistic images using the same semantic label map. Recently, several works learn to produce a fixed number of discrete outputs given the same input [5, 15] or synthesize diverse modes controlled

by a latent code that encodes the entire image [64]. Although these approaches tackle the multi-modal image synthesis problem, they are unsuitable for our image manipulation task mainly for two reasons. First, the user has no intuitive control on which kinds of images the model would produce [5, 15]. Second, these methods focus on global color and texture changes and allow no object-level control on the generated contents.

To generate diverse images and allow instance-level control, we propose adding additional low-dimensional feature channels as the input to the generator network. We show that, by manipulating these features, we can have flexible control over the image synthesis process. Furthermore, note that since the feature channels are continuous quantities, our model is, in principle, capable of generating infinitely many images.

To generate the low-dimensional features, we train an encoder network E to find a low-dimensional feature vector that corresponds to the ground truth target for each instance in the image. Our feature encoder architecture is a standard encoder-decoder network. To ensure the features are consistent within each instance, we add an instance-wise average pooling layer to the output of the encoder to compute the average feature for the instance. The average feature is then broadcasted to all the pixel locations of the instance. Figure 5 visualizes an example of the encoded features.

We replace $G(\mathbf{s})$ with $G(\mathbf{s}, E(\mathbf{x}))$ in Eq. (5) and train the encoder jointly with the generators and discriminators. After the encoder is trained, we run it on all instances in the training images and record the obtained features. Then we perform a K-means clustering on these features for each semantic category. Each cluster thus encodes the features for a specific style, for example, the asphalt or cobblestone texture for a road. At inference time, we randomly pick one of the cluster centers and use it as the encoded features. These features are concatenated with the label map and used as the input to our generator. We tried to enforce the Kullback-Leibler loss [27] on the feature space for better test-time sampling as used in the recent work [64], but found it quite involved for users to adjust the latent vectors for each object directly. Instead, for each object instance, we simply present K modes for users to choose from.

4. Results

We first provide a quantitative comparison against leading methods in Sec. 4.1. We then report a subjective human perceptual study in Sec. 4.2. Finally, we show a few examples of interactive object editing results in Sec. 4.3.

Implementation details We use LSGANs [36] for stable training. In all experiments, we set the weight $\lambda = 10$ (Eq. (5)) and $K = 10$ for K-means. We use 3-dimensional vectors to encode features for each object instance. We experimented with adding a perceptual loss $\lambda \sum_{i=1}^{N} \frac{1}{M_i}[||F^{(i)}(\mathbf{x}) - F^{(i)}(G(\mathbf{s}))||_1]$ to our objective (Eq. (5)), where $\lambda = 10$ and $F^{(i)}$ denotes the i-th layer with M_i elements of the VGG network. We observe that this loss slightly improves the results. We name these two variants as **ours** and **ours (w/o VGG loss)**. Please find more training and architecture details in the appendix (Sec. 6).

	pix2pix [21]	CRN [5]	Ours	Oracle
Pixel acc	78.34	70.55	**83.78**	84.29
Mean IoU	0.3948	0.3483	**0.6389**	0.6857

Table 1: Semantic segmentation scores on results by different methods on the Cityscapes dataset [7]. Our result outperforms the other methods by a large margin and is very close to the accuracy on original images (i.e., the oracle).

Datasets We conduct extensive comparisons and ablation studies on Cityscapes dataset [7] and NYU Indoor RGBD dataset [39]. We report additional qualitative results on ADE20K dataset [61] and Helen Face dataset [30, 48].

Baselines We compare our method with two state-of-the-art algorithms: pix2pix [21] and CRN [5]. We train pix2pix models on high-res images with the default setting as only 256×256 images are provided. We produce the high-res CRN images via the authors' publicly available model.

4.1. Quantitative Comparisons

We adopt the same evaluation protocol from previous image-to-image translation works [21, 63]. To quantify the quality of our results, we perform semantic segmentation on the synthesized images, and compare how well the predicted segments match the input. The intuition is that if we can produce realistic images that correspond to the input label map, an off-the-shelf semantic segmentation model (e.g. PSPNet [59] that we use) should be able to predict the ground truth label. The obtained segmentation accuracy is reported in Table 1. As can be seen, for both pixel-wise accuracy and mean intersection-over-union (IoU), our method outperforms the other methods by a large margin. Moreover, our result is very close to the result on the original images, which is theoretically the "upper bound" of the realism we can achieve. This justifies the superiority of our algorithm.

4.2. Human Perceptual Study

We further evaluate our algorithm via a human subjective study. We perform pairwise A/B tests deployed on the Amazon Mechanical Turk (MTurk) platform on the Cityscapes dataset [7]. We follow the same experiment procedure as described in Chen and Koltun [5]. More specifically, two different kinds of experiments are conducted: unlimited time and limited time, as explained below.

	pix2pix [21]	CRN [5]
Ours	93.8%	86.2%
Ours (w/o VGG)	94.6%	85.2%

Table 2: Pairwise comparison results on the Cityscapes dataset [7] (unlimited time). Each cell lists the percentage where our result is preferred over the other method. Chance is at 50%.

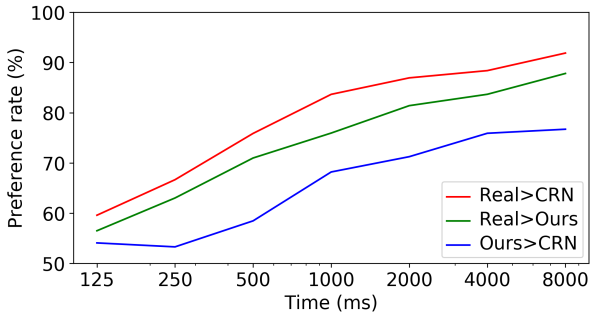


Figure 6: Limited time comparison results. Each line shows the percentage when one method is preferred over the other.

Unlimited time For this task, the workers are given two images at once, each of which is synthesized by a different method for the same label map. They are then given unlimited time to select which image looks more natural. The left-right order and the image order are randomized to ensure unbiased comparisons. All 500 Cityscapes test images are compared 10 times, resulting in $5,000$ human judgments for each method. In this experiment, we use the model trained on labels only (without instance maps) to ensure a fair comparison. Table 2 shows that both variants of our method outperform the other methods significantly.

Limited time Next, for the limited time experiment, we compare our result with CRN and the original image (ground truth). In each comparison, two of the three images are picked, and they are shown for a short period of time. We randomly select a duration between $1/8$ seconds and 8 seconds, as adopted by prior work [5]. This evaluates how quickly the difference between the images can be perceived. The comparison results at different time intervals are shown in Fig. 6. It can be seen that as the given time becomes longer and longer, the differences between these three images become more apparent and easy to observe. Figures 7 and 9 show example synthesized results.

Analysis of the loss function We also study the importance of each term in our objective function using the *unlimited time* experiment. Specifically, our final loss contains three components: GAN loss, discriminator-based feature matching loss, and VGG perceptual loss. We compare our final implementation to the results using (1) only GAN loss, and (2) GAN + feature matching loss (i.e., without VGG loss). The obtained preference rates are 68.55% and 58.90%, respectively. As can be seen, adding the feature matching loss substantially improves the performance, while adding perceptual loss further enhances the results. However, note that using the perceptual loss is not critical, and we are still able to generate visually appealing results even without it (e.g. Figs. 7c, 9b).

	U-Net [21, 42]	CRN [5]	Our generator
Pixel acc (%)	77.86	78.96	**83.78**
Mean IoU	0.3905	0.3994	**0.6389**

Table 3: Semantic segmentation scores on results using different generators on the Cityscapes dataset [7]. Our generator obtains the highest scores.

	U-Net [21, 42]	CRN [5]
Our generator	80.0%	76.6%

Table 4: Pairwise comparison results on the Cityscapes dataset [7]. Each cell lists the percentage where our result is preferred over the other method. Chance is at 50%.

Using instance maps We compare results using instance maps to results without using them. We highlight the car regions in the images, and ask the participants to choose which region looks more realistic. We obtain a preference rate of 64.34%, which indicates that using instance maps improves the realism of our results, especially around the object boundaries.

Analysis of the generator We compare results of different generators with all the other components fixed. In particular, we compare our generator with two state-of-the-art generator architectures: U-Net [21, 42] and CRN [5]. We evaluate the performance regarding both semantic segmentation scores and human perceptual study results. Table 3 and Table 4 show that our coarse-to-fine generator outperforms other networks by a large margin.

Analysis of the discriminator Next, we also compare results using our multi-scale discriminators and results using only one discriminator while we keep the generator and the loss function fixed. The segmentation scores on Cityscapes [7] (Table 5) demonstrate that using multi-scale discriminators helps produce higher quality results as well as stabilize the adversarial training. We also perform pairwise A/B tests on Amazon Mechanical Turk platform. 69.2% of the participants prefer our results with multi-scale discriminators over the results trained with a single-scale discriminator (Chance is 50%).

	single D	multi-scale Ds
Pixel acc (%)	82.87	**83.78**
Mean IoU	0.5775	**0.6389**

Table 5: Semantic segmentation scores on results using either a single discriminator (`single D`) or multi-scale discriminators (`multi-scale Ds`) on the Cityscapes dataset [7]. Using multi-scale discriminators helps improve the segmentation scores.

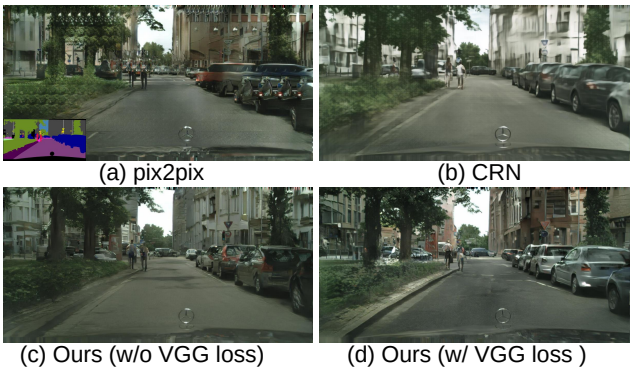


Figure 7: Comparison on the Cityscapes dataset [7] (label maps shown at the lower left corner in (a)). For both without and with VGG loss, our results are more realistic than the other two methods. Please zoom in for details.

Additional datasets To further evaluate our method, we perform unlimited time comparisons on the NYU dataset. We obtain 86.7% and 63.7% against pix2pix and CRN, respectively. Example images are shown in Fig. 8. Finally, we show results on the ADE20K [61] dataset (Fig. 10).

4.3. Interactive Object Editing

Our feature encoder allows us to perform interactive instance editing on the resulting images. For example, we can change the object labels in the image to quickly create novel scenes, such as replacing trees with buildings (Fig. 1b). We can also change the colors of individual cars, or the textures of the road (Fig. 1c). Please check out our interactive demos on our website.

In addition, we implement our interactive object editing feature on the Helen Face dataset where labels for different facial parts are available [48] (Fig. 11). This makes it easy to edit human portraits, e.g. changing the face color to mimic different make-up effects or adding beards to a face.

5. Discussion and Conclusion

The results in this paper suggest that conditional GANs are able to synthesize high-resolution photo-realistic imagery without any hand-crafted losses or pre-trained networks. We have observed that incorporating a perceptual loss [22] can slightly improve the results. Our method allows many applications and will be potentially useful for domains where high-resolution results are in demand but pre-trained networks are not available (e.g., medical imaging [17] and biology [8]).

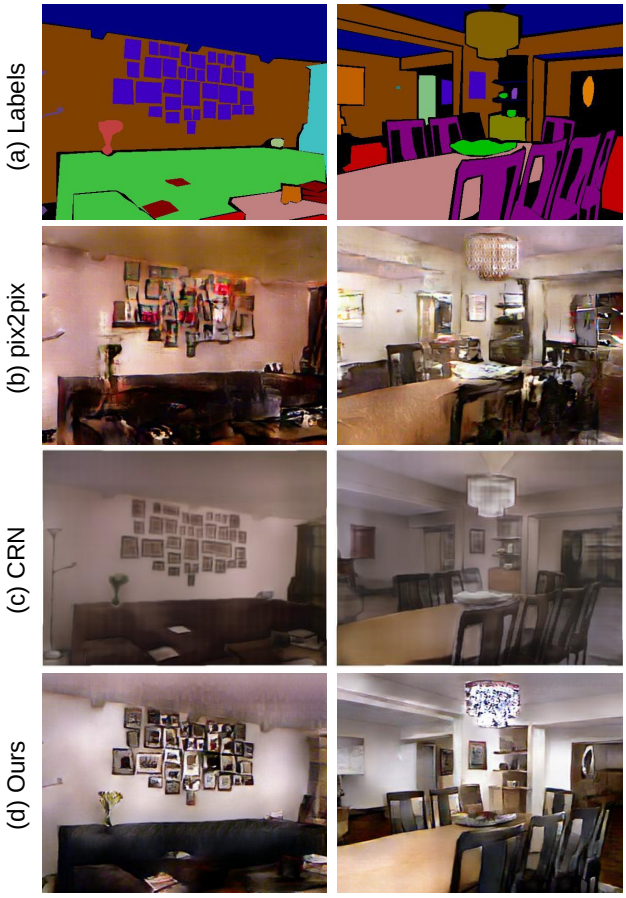


Figure 8: Comparison on the NYU dataset [39]. Our method generates more realistic and colorful images than the other methods.

This paper also shows that an image-to-image synthesis pipeline can be extended to produce diverse outputs, and enable interactive image manipulation given appropriate training input-output pairs (e.g., instance maps in our case). Without ever been told what a "texture" is, our model learns to stylize different objects, which may be generalized to other datasets as well (i.e., using textures in one dataset to synthesize images in another dataset). We believe these contributions broaden the area of image synthesis and can be applied to many other related research fields.

Acknowledgements We thank Taesung Park, Phillip Isola, Tinghui Zhou, Richard Zhang, Rafael Valle and Alexei A. Efros for helpful comments. We also thank Chen and Koltun [5] and Isola et al. [21] for sharing their code. JYZ is supported by a Facebook graduate fellowship.

Figure 9: Additional comparison results with CRN [5] on the Cityscapes dataset. Again, both our results have finer details in the synthesized cars, the trees, the buildings, etc. Please zoom in for details.

Figure 10: Results on the ADE20K dataset [61] (label maps shown at lower left corners in (a)). Our method generates images at similar level of realism as the original images.

Figure 11: Diverse results on the Helen Face dataset [48] (label maps shown at lower left corners). With our interface, a user can edit the attributes of individual facial parts in real-time, such as changing the skin color or adding eyebrows and beards. See our video for more details.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. In *International Conference on Machine Learning (ICML)*, 2017. 2

[2] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[3] M. Brown, D. G. Lowe, et al. Recognising panoramas. In *IEEE International Conference on Computer Vision (ICCV)*, 2003. 4

[4] P. Burt and E. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983. 2, 4

[5] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 2, 3, 4, 5, 6, 7, 8, 9

[6] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu. Sketch2photo: Internet image montage. *ACM Transactions on Graphics (TOG)*, 28(5):124, 2009. 3

[7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3, 6, 7, 8, 12

[8] P. Costa, A. Galdran, M. I. Meyer, M. Niemeijer, M. Abràmoff, A. M. Mendonça, and A. Campilho. End-to-end adversarial retinal image synthesis. *IEEE Transactions on Medical Imaging*, 2017. 8

[9] E. 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 (NIPS)*, 2015. 2, 4

[10] H. Dong, S. Yu, C. Wu, and Y. Guo. Semantic image synthesis via adversarial learning. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[11] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2, 3, 5

[12] I. Durugkar, I. Gemp, and S. Mahadevan. Generative multi-adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2016. 4

[13] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3, 5

[14] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand. Deep joint demosaicking and denoising. *ACM Transactions on Graphics (TOG)*, 35(6):191, 2016. 3

[15] A. Ghosh, V. Kulharia, V. Namboodiri, P. H. Torr, and P. K. Dokania. Multi-agent diverse generative adversarial networks. *arXiv preprint arXiv:1704.02906*, 2017. 5, 6

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 2

[17] J. T. Guibas, T. S. Virdi, and P. S. Li. Synthetic medical images from dual generative adversarial networks. *arXiv preprint arXiv:1709.01872*, 2017. 8

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[19] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie. Stacked generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 4

[20] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017. 4

[21] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3, 5, 6, 7, 8, 13

[22] 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. 2, 3, 5, 8, 12

[23] M. Johnson, G. J. Brostow, J. Shotton, O. Arandjelovic, V. Kwatra, and R. Cipolla. Semantic photo synthesis. In *Computer Graphics Forum*, volume 25, pages 407–413. Wiley Online Library, 2006. 3

[24] T. Kaneko, K. Hiramatsu, and K. Kashino. Generative attribute controller with conditional filtered generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[25] L. Karacan, Z. Akata, A. Erdem, and E. Erdem. Learning to generate images of outdoor scenes from attributes and semantic layouts. *arXiv preprint arXiv:1612.00215*, 2016. 2, 5

[26] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 12

[27] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2013. 6

[28] J.-F. Lalonde, D. Hoiem, A. A. Efros, C. Rother, J. Winn, and A. Criminisi. Photo clip art. In *ACM Transactions on Graphics (TOG)*, volume 26, page 3. ACM, 2007. 3

[29] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. In *International Conference on Machine Learning (ICML)*, 2016. 5

[30] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *European Conference on Computer Vision (ECCV)*, 2012. 6, 12

[31] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 5

[32] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan. Perceptual generative adversarial networks for small object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[33] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 2

[34] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[35] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[36] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 6

[37] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *International Conference on Learning Representations (ICLR)*, 2016. 2

[38] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[39] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision (ECCV)*, 2012. 6, 8, 12

[40] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3

[41] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2015. 2

[42] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015. 3, 7

[43] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992. 2

[44] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[45] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays. Scribbler: Controlling deep image synthesis with sketch and color. *arXiv preprint arXiv:1612.00835*, 2016. 2, 3

[46] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[47] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[48] B. M. Smith, L. Zhang, J. Brandt, Z. Lin, and J. Yang. Exemplar-based face parsing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 6, 8, 9, 12

[49] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. In *International Conference on Learning Representations (ICLR)*, 2017. 2

[50] H.-Y. F. 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 *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[51] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 12

[52] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[53] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *European Conference on Computer Vision (ECCV)*, 2016. 2

[54] Z. Yi, H. Zhang, P. T. Gong, et al. DualGAN: Unsupervised dual learning for image-to-image translation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[55] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 4

[56] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European Conference on Computer Vision (ECCV)*, 2016. 3

[57] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros. Real-time user-guided image colorization with learned deep priors. In *ACM SIGGRAPH*, 2017. 3

[58] Z. Zhang, Y. Song, and H. Qi. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[59] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6

[60] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. In *International Conference on Learning Representations (ICLR)*, 2017. 2

[61] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ADE20K dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6, 8, 9, 12

[62] 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. 2, 3

[63] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 6, 12

[64] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 2, 6

6. Appendix

6.1. Training details

All the networks were trained from scratch, using the Adam solver [26] and a learning rate of 0.0002. We keep the same learning rate for the first 100 epochs and linearly decay the rate to zero over the next 100 epochs. Weights were initialized from a Gaussian distribution with mean 0 and standard deviation 0.02. We train all our models on an NVIDIA Quadro M6000 GPU with 24GB GPU memory.

The inference time is between $20 \sim 30$ milliseconds per 2048×1024 input image on a NVIDIA 1080Ti GPU with 11GB GPU memory. This real-time performance allows us to develop interactive image editing applications.

Below we discuss the details of the datasets we used.

Cityscapes dataset [7]: 2975 training images from the Cityscapes training set with image size 2048×1024. We use the Cityscapes validation set for testing, which consists of 500 images. The training time is around one week.

NYU Indoor RGBD dataset [39]: 1200 training images and 249 test images, all at resolution of 561×427.

ADE20K dataset [61]: 20210 training images and 2000 test images with varying image sizes. We scale the width of all images to 512 before training and inference.

Helen Face dataset [30, 48]: 2000 training images and 330 test images with varying image sizes. We resize all images to 1024×1024 before training and inference.

6.2. Generator architectures

Our generator consists of a global generator network and a local enhancer network. we follow the naming convention used in the Johnson el al. [22] and CycleGAN [63]. Let `c7s1-k` denote a 7×7 Convolution-InstanceNorm [51]-ReLU layer with k filters and stride 1. `dk` denotes a 3×3 Convolution-InstanceNorm-ReLU layer with k filters, and stride 2. We uee reflection padding to reduce boundary artifacts. `Rk` denotes a residual block that contains two 3×3 convolutional layers with the same number of filters on both layer. `uk` denotes a 3×3 fractional-strided-Convolution-InstanceNorm-ReLU layer with k filters, and stride $\frac{1}{2}$.

Recall that we have two generators: our global generator and our local enhancer.

Our global network

`c7s1-64,d128,d256,d512,d1024,R1024,R1024,R1024,R1024,R1024,R1024,R1024,R1024,R1024,u512,u256,u128,u64,c7s1-3`

Our local enhancer:

`c7s1-32,d64`[2]`,R64,R64,R64,u32,c7s1-3`

[2]We add the last feature map (`u64`) in our global network to the output of this layer.

6.3. Discriminator architectures

For discriminator networks, we use 70×70 PatchGAN [21]. Let `Ck` denote a 4×4 Convolution-InstanceNorm-LeakyReLU layer with k filters and stride 2. After the last layer, we apply a convolution to produce a 1 dimensional output. We do not use InstanceNorm for the first `C64` layer. We use leaky ReLUs with slope 0.2. All our three discriminator have the identical architecture as follows:

```
C64-C128-C256-C512
```