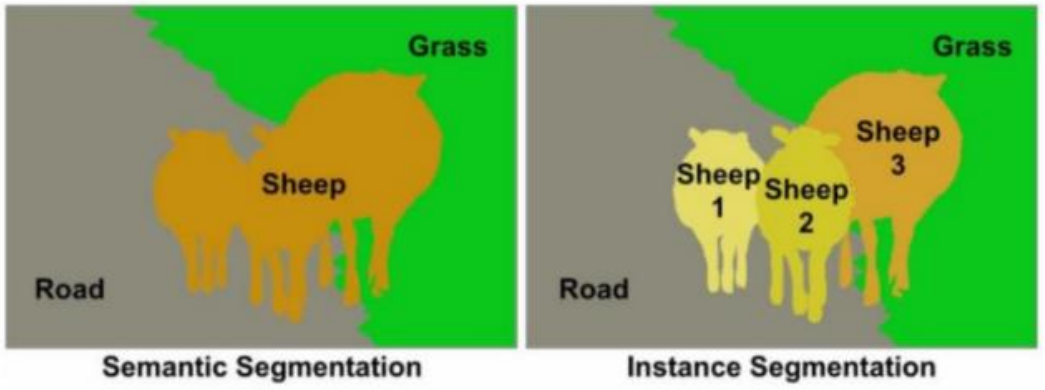
pix2pixhd

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.

- cGAN기반 새로운 loss, multi scale discriminator

- Interactive visual manipulation
  - instance segmentation 사용 (add/remove object)
  - feature embedding (generate diverse results in same input)
  - =>유저가 이미지 생성에 개입



1. Introduction

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 [44] or pre-trained networks (e.g. VGGNet [48]) for perceptual losses [11, 22] (Figs. 9c, 10b). Then we show that adding perceptual losses from pre-trained networks [48] can slightly improve the results in some circumstances (Figs. 9d, 10c), if a pre-trained network is available.

- SOTA의 문제점
  - High resolution 이미지 생성의 어려움[21]
  - 생성해도 디테일이 부족함 [5]

- [5]처럼 perceptual loss(VGG loss) 추가하여 성능 향상

2. Related Work

Image-to-image translation

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. 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, 9, 8, 10).

- CRN[5](Cascaded refinemen network)은 1024*2048 Semantic label로 부터 이미지를 생성
- perceptual loss를 사용
- Pix2pix와 비교해서는 더 좋은 성능을 가짐

CRN architecture

:Up sample

Semantic Map (1024*2048)

Down sample

Down sample 4*8 resolution

Refinement module 0 (4*8)

Refinement module 1 (8*16)

Refinement module 8 (1024*2048)

output

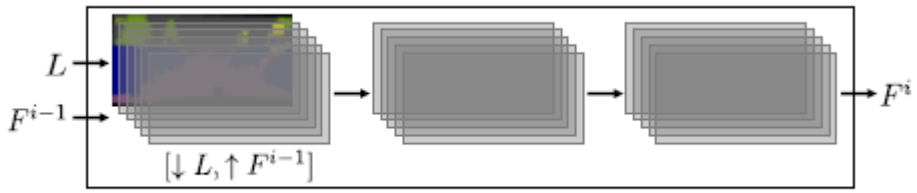


Figure 3. A single refinement module.

3*3 conv,
normalization
Leaky relu

CRN vs pix2pix

CRN[5]

pix2pix[21]

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 photo-realism 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

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.

1. Loss 개선, 네트워크 구조 변경

2. instance-level object semantic information 사용

3. Instance-level feature embedding

- 고해상도에서 디테일 향상
  - Coarse-to-fine generator
  - Multi scale discriminator
  - Robust adversarial learning

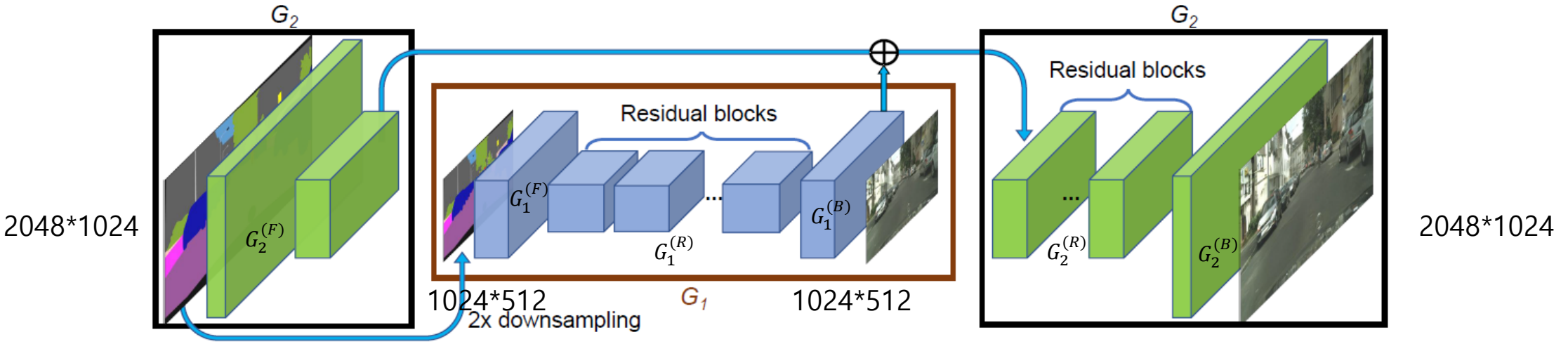


Figure 3: Network architecture of our generator. We first train a residual network G_1 on lower resolution images. Then, another residual network G_2 is appended to G_1 and the two networks are trained jointly on high resolution images. Specifically, the input to the residual blocks in G_2 is the element-wise sum of the feature map from G_2 and the last feature map from G_1.

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. 3. 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.

- Global generator(1024*512): G1
- Local enhancer(2048*1024): G2

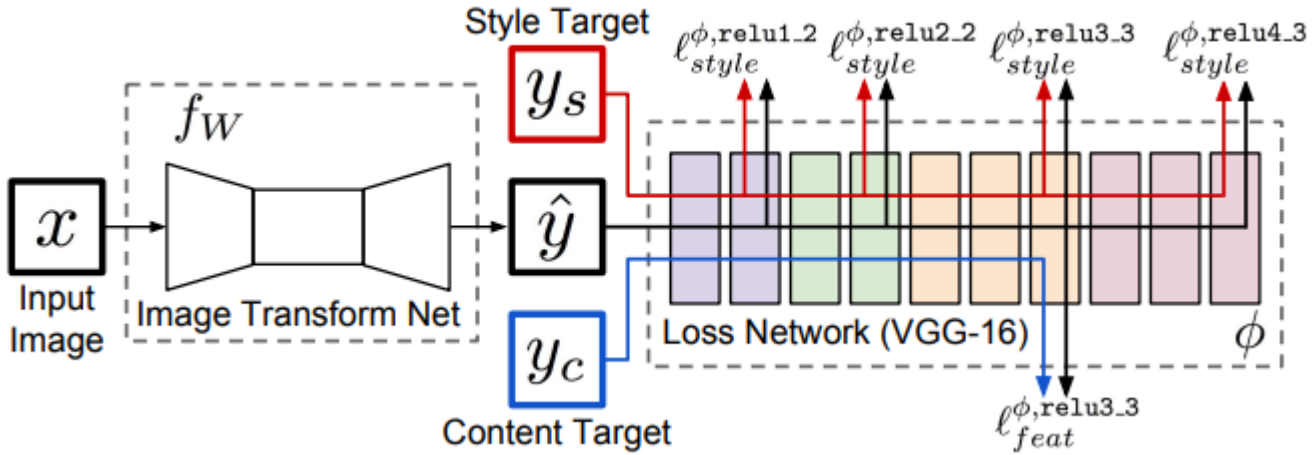
Our global generator is built on the architecture proposed by Johnson *et al.* [22], which has been proven successful for neural style transfer on images up to 512×512. It consists of 3 components: a convolutional front-end $G_1^{(F)}$, a set of residual blocks $G_1^{(R)}$ [18], and a transposed convolutional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

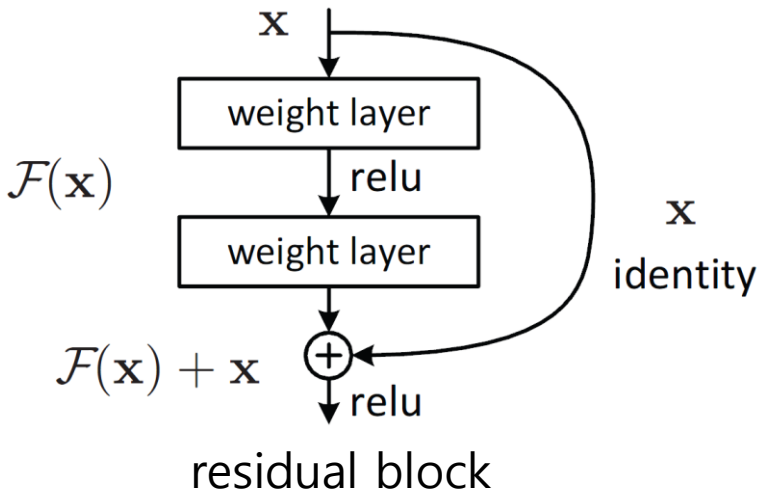
The local enhancer network also consists of 3 components: a convolutional front-end $G_2^{(F)}$, a set of residual blocks $G_2^{(R)}$, and a transposed convolutional back-end $G_2^{(B)}$. The resolution of the input label map to G_2 is 2048×1024. Different from the global generator network, the input 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.

- Generator는 각각 convolution block, residual block, transposed convolution block 로 이뤄져 있다.
- Local enhancer network(G2)의 경우 G2의 convolution block 출력과 G1의 transposed convolution block의 출력이 elementwise sum되어 G2의 residual block의 입력으로 들어간다.

[22] Perceptual Losses for Real-Time Style Transfer and Super-Resolution

- Per-pixel -> +perceptual loss(high quality image in super resolution)





residual block

(skip connection, resnet block)

- Conv*3 + residual block*5 + deconv*3

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, 57].

- 1. global generator 따로 학습
- 2. local enhancer 따로 학습
- 3. 결합하여 fine tune
- =>global, local information을 효과적으로 결합

Multi-scale discriminators High-resolution image synthesis poses a significant 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, both of which would increase the network capacity and potentially cause overfitting. Also, both choices demand a larger memory footprint for training, which is already a scarce resource for high-resolution image generation.

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

- 고해상도 이미지 생성에서 discriminator는 더 큰 receptive field를 필요로 한다.
- 이는 layer 수가 늘어나거나 kernel width가 늘어나므로 학습에 필요한 파라미터 수가 늘어나 오버피팅을 야기할 수 있다.
- 더 큰 메모리를 필요로 함

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 at the finest scale encourages 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 a 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.

- Multi-scale discriminator는 동일한 구조를 가지며 각기 다른 scale의 이미지를 입력으로 받는 discriminator들이다.
- 작은(coarsest) scale은 원본에서 봤을 때 가장 큰 receptive field를 가지며 큰(finest) scale은 작은 receptive field를 가진다.
- 각각 global 정보와 fine detail 정보에 특화되어 판별하여 generator에 피드백을 준다.

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:

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

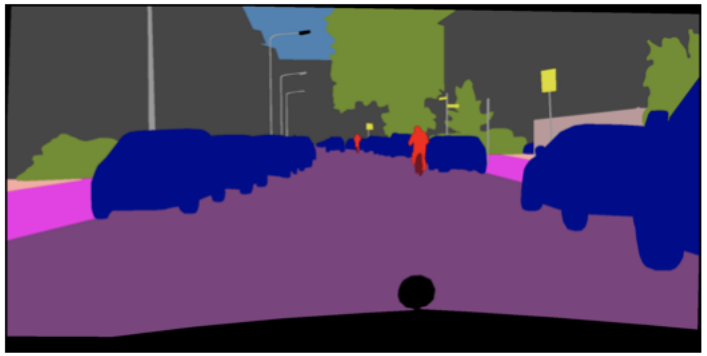
where T is the total number of layers and N_i denotes the number of elements in each layer. Our GAN discriminator feature matching loss is related to the perceptual loss [11, 13, 22], which has been shown to be useful for image super-resolution [32] and style transfer [22]. In our experiments, we discuss 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 in VAE-GANs [30].

- 논문에서는 feature loss를 적용하여 여러 scale의 input에서도 안정적인 결과를 얻어낼 수 있다고 소개

- Feature loss는 perceptual loss와 유사하며 discriminator의 각 layer에서의 값들의 차를 계산하는 것이다.

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 of the pixel. This map does not differentiate objects of the same category. On the other hand, an instance-level semantic label map contains a unique object ID for each individual object. To incorporate the instance map, one can directly pass it into the network, or encode it into a one-hot vector. However, both approaches are difficult to implement in practice, since different images may contain different numbers of objects of the same category. Alternatively, one can pre-allocate a fixed number of channels (e.g., 10) for each class, but this method fails when the number is set too small, and wastes memory when the number is too large.



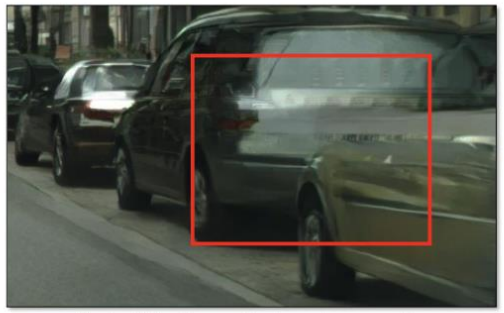
(a) Semantic labels

- 기존의 GAN은 semantic 정보만 사용하여 각 객체를 구분하지 못하는 문제점이 있음
- Instance map을 사용하려 함

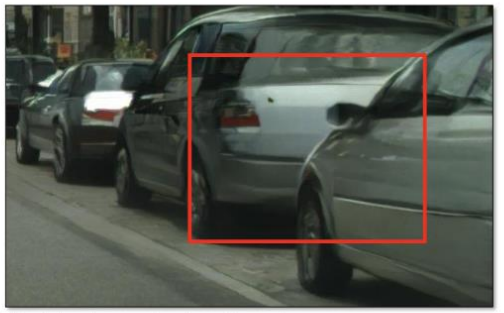
Instead, we argue that the most critical information the instance map provides, which is not available in the semantic label map, is the object boundary. For example, when objects of the same class are next to one another, looking at the semantic label map alone cannot tell them apart. This is especially true for the street scene since many parked cars or walking pedestrians are often next to one another, as shown in Fig. 4a. 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. 4b). 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 5b 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.

- Instance map을 쓰는 대신 object간의 경계인 boundary map을 구함
- Boundary map을 semantic map과 concatenate하여 사용



(a) Using labels only



(b) Using label + instance map

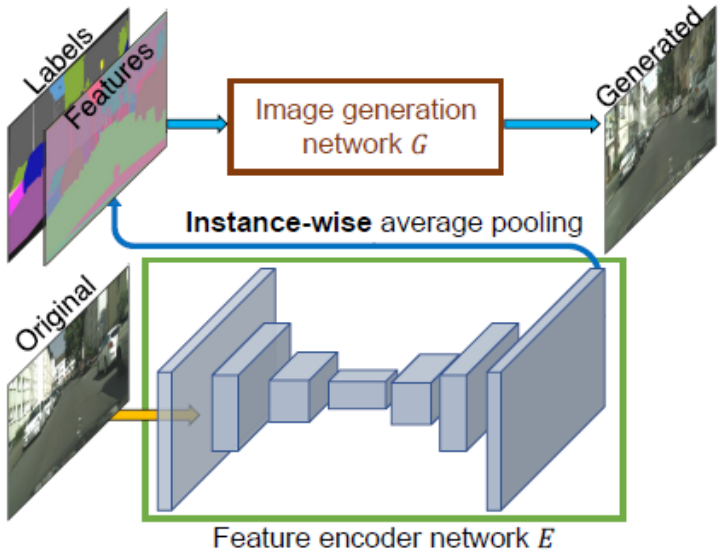
3.4. Learning an Instance-level Feature Embedding

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 [66]. 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 over 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.

- 이상적인 image synthesis는 동일한 입력에서 다양하고 현실적인 이미지를 생성해야 한다.
- 이전의 연구들은 유저가 직접 제어 할 수 없고 object-level에서만 작동하였다

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 object instance. The average feature is then broadcast to all the pixel locations of the instance. Figure 6 visualizes an example of the encoded features.



- 다양한 이미지 생성과 Instance-level에서 제어 하기 위해 encoder로 부터 얻어진 low-dimension feature를 입력의 한 채널로 연결한다.

- Feature가 각 instance에 일괄적으로 적용되기 위해 instance-wise average pulling을 encoder의 output에서 적용한다.

We replace $G(\mathrm{s})$ with $G(\mathrm{s}, E(\mathrm{x}))$ 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

- encoder와 generator, discriminator 동시 학습
- Encoder에서 얻어진 feature들을 가지고 각 semantic category에 대해 k-mean clustering 적용
- 얻어진 cluster는 특징을 가진다

Feature embedding

- Label에 적용하고 싶은 특징 이미지를 선택하면 그 이미지의 특징(cluster)에 해당하는 latent vector가 입력에 연결됨

4. Results

Implementation details We use LSGANs [37] 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 $\lambda \sum_{i=1}^{N} \frac{1}{M_i}[||F^{(i)}(\mathrm{x}) - F^{(i)}(G(\mathrm{s}))||_1]$ 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.

4.1. Quantitative Comparisons

	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).

- LSGAN 사용
- VGG network을 사용한 perceptual loss 적용
- 정량적 평가 방법
  - IoU

    overlapping Region/combined region

GAN

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))]$$

LSGAN

$$\min_D V_{LSGAN}(D) = \frac{1}{2}\mathbb{E}_{x \sim p_{data}(x)}[(D(x) - b)^2] + \frac{1}{2}\mathbb{E}_{z \sim p_z(z)}[(D(G(z)) - a)^2] \quad (1)$$

$$\min_G V_{LSGAN}(G) = \frac{1}{2}\mathbb{E}_{z \sim p_z(z)}[(D(G(z)) - c)^2] \quad (2)$$