High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs

Wenjie Niu

August 24, 2018

1. Improving Photorealism and Resolution

The paper improves 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 The paper decomposes 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. 1. The global generator network operates at a resolution of 1024 $\times$ 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 $\times$ 1024, and the output image resolution of $G = \{G_1, G_2, G_3\}$ is 4096 $\times$ 2048.

The global generator is built on the architecture proposed by Johnson *et al.* [2], which consists of 3 components: a convolutional front-end $G_1^{(F)}$, a set of residual blocks $G_1^{(R)}$ [1], and a transposed convolutional back-end $G_1^{(B)}$. A semantic label map of resolution 1024 $\times$ 512 is passed through the 3 components sequentially to output an image of resolution 1024 $\times$ 512. The local enhancer network also consists of 3 components: a convolutional front-end $G_2^{(F)}$, a set of residual blocks $G_2^{(B)}$, and a transposed convolutional back-end $G_2^{(B)}$. The resolution of the input label map to G_2 is 2048 $\times$ 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, they choose to 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.

Multi-scale discriminators They use 3 discriminators that have an identical network structure but operate at different image scales. They will refer to the discriminators as D_1, D_2 and D_3. Specifically, they 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.

With the discriminators, the learning problem then becomes a multi-task learning problem of

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

Improved adversarial loss The 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 $\mathcal{L}_{GAN}(G, D_k)$ is then calculated as:

$$\mathcal{L}_{FM}(G, D_k) = \mathbb{E} \sum_{i=1}^{T} [\|D_k(s, x) D_k(s, G(s))\|_1] \tag{2}$$

where T is the total number of layers and N_i denotes the number of elements in each layer.

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

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

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

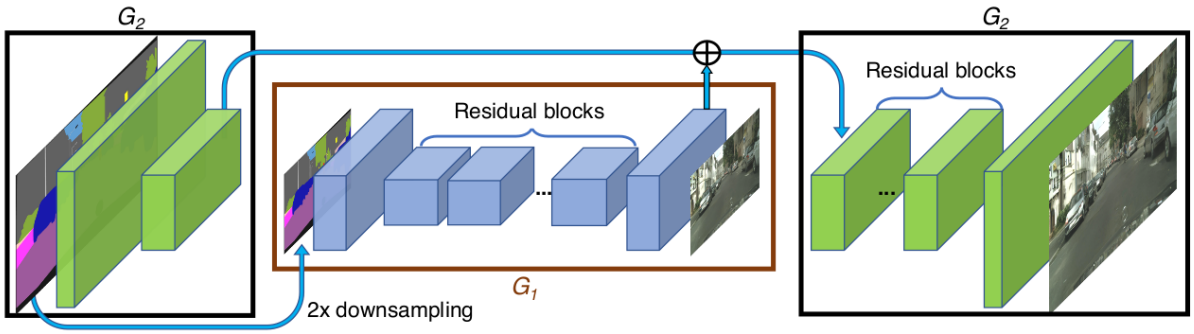

Figure 1. Network architecture of our generator. We first train a residual network G_1 on lower resolution images. Then, another residual network G_2 is appended to G_1 and the two networks are trained jointly on high resolution images. Specifically, the input to the residual blocks in G_2 is the element-wise sum of the feature map from G_2 and the last feature map from G_1 . [3]

2. Using Instance Maps

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, 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.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2015. 1

[2] J. Johnson, A. Alahi, and F. F. Li. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 1

[3] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 2