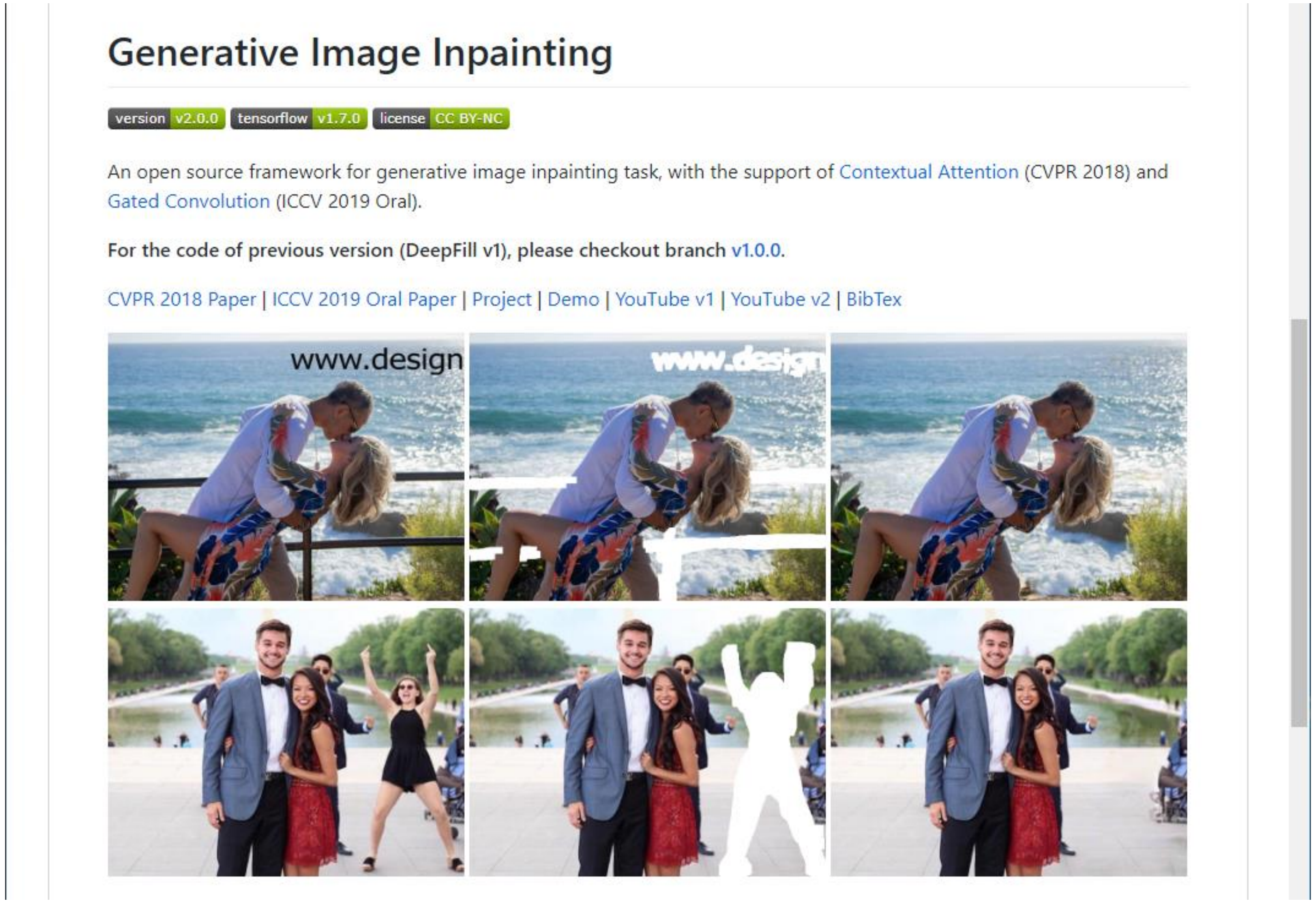
fast campus

Fast Campus **Online** | Life Changing **Education**

Chapter 05. 이미지 복원 (Image Reconstruction)

복원할 부분에 집중하는 기법

Generative Image Inpainting



Generative Image Inpainting

version v2.0.0 | tensorflow v1.7.0 | license CC BY-NC

An open source framework for generative image inpainting task, with the support of Contextual Attention (CVPR 2018) and Gated Convolution (ICCV 2019 Oral).

For the code of previous version (DeepFill v1), please checkout branch v1.0.0.

CVPR 2018 Paper | ICCV 2019 Oral Paper | Project | Demo | YouTube v1 | YouTube v2 | BibTex

Adobe에서 발표한 Generative Image Inpainting 방법. 2018년 CVPR 버전을 살펴보자.

https://github.com/JiahuiYu/generative_inpainting

Contextual Attention

Generative Image Inpainting with Contextual Attention

Jiahui Yu[1] Zhe Lin[2] Jimei Yang[2] Xiaohui Shen[2] Xin Lu[2] Thomas S. Huang[1]

[1]University of Illinois at Urbana-Champaign
[2]Adobe Research

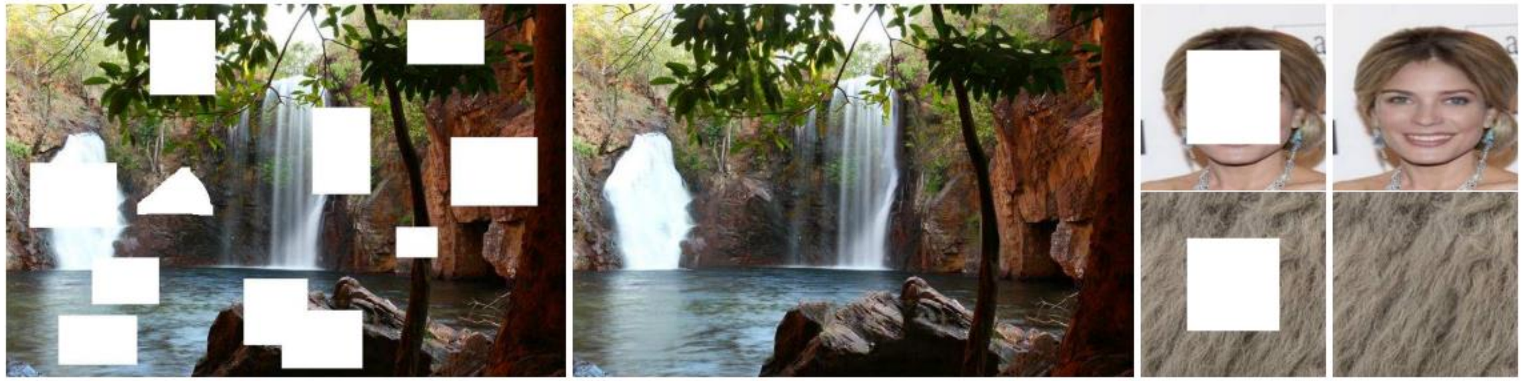


Figure 1: Example inpainting results of our method on images of natural scene, face and texture. Missing regions are shown in white. In each pair, the left is input image and right is the direct output of our trained generative neural networks without any post-processing.

빈 공간을 채우는 Inpainting 알고리즘. 주변 영상의 정보를 끌어와서 내용을 채우는 특징이 있다.

https://arxiv.org/pdf/1801.07892.pdf

Overall Architecture

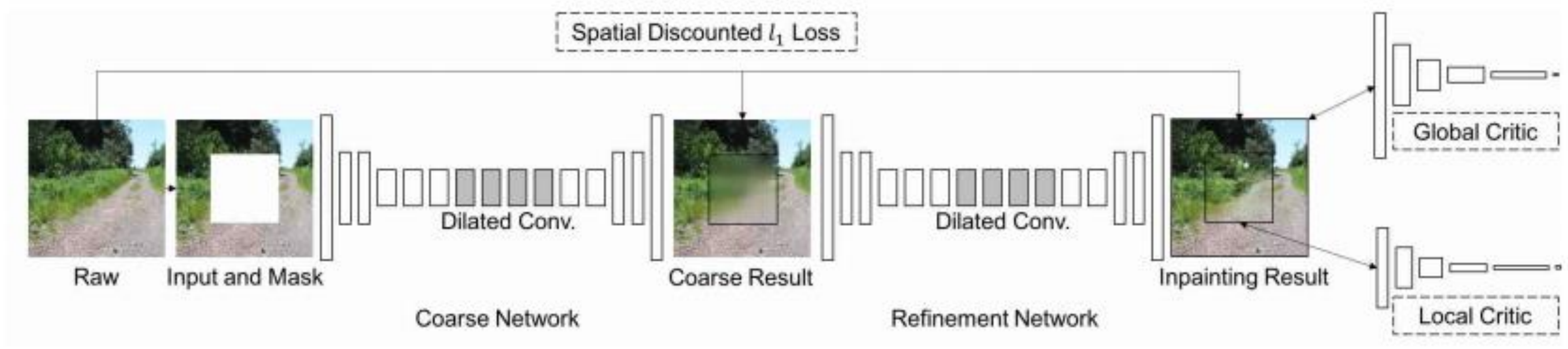
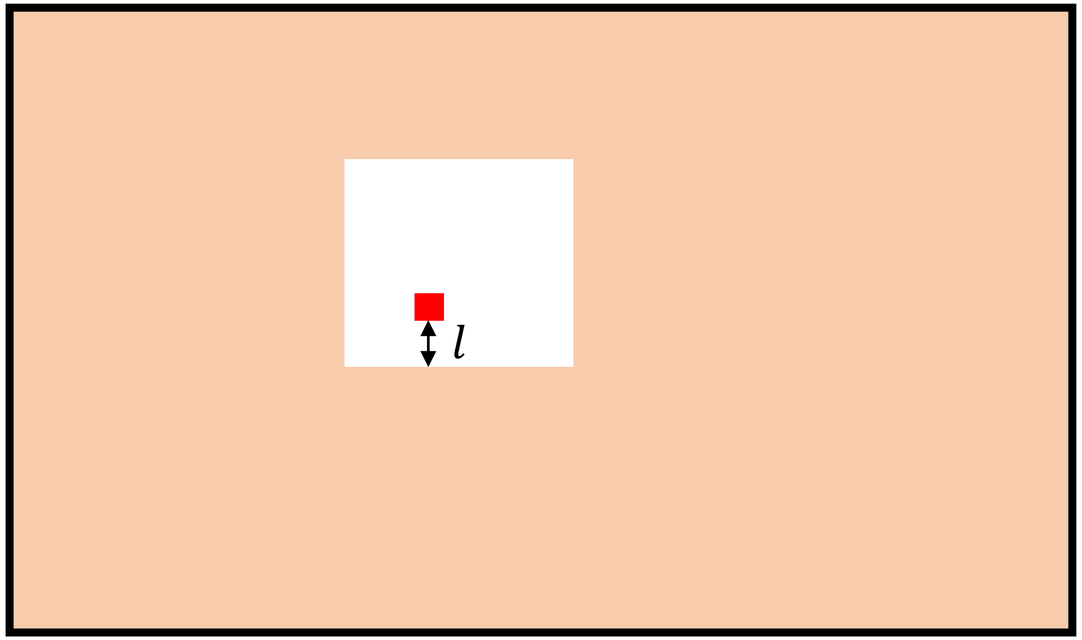


Figure 2: Overview of our improved generative inpainting framework. The coarse network is trained with reconstruction loss explicitly, while the refinement network is trained with reconstruction loss, global and local WGAN-GP adversarial loss.

전체 구조를 보면, 크게 2-Stage로 되어 있으며 Fully Convolutional한 특징이 있다.

L-1 Loss와 더불어 Global과 Local GAN Loss를 사용한다 (WGAN-GP)

Spatial Discounted Loss



$w = \gamma^l \ (\gamma = 0.99)$

Known Pixel로부터 거리가 멀 수록 L-1 loss의 신뢰도가 낮다. 이것을 수학적으로 정의한 것!

Contextual Attention

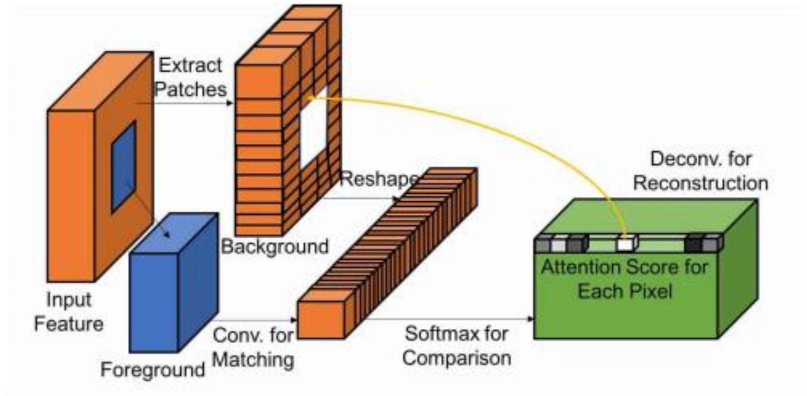


Figure 3: Illustration of the contextual attention layer. Firstly we use convolution to compute matching score of foreground patches with background patches (as convolutional filters). Then we apply softmax to compare and get attention score for each pixel. Finally we reconstruct foreground patches with background patches by performing deconvolution on attention score. The contextual attention layer is differentiable and fully-convolutional.

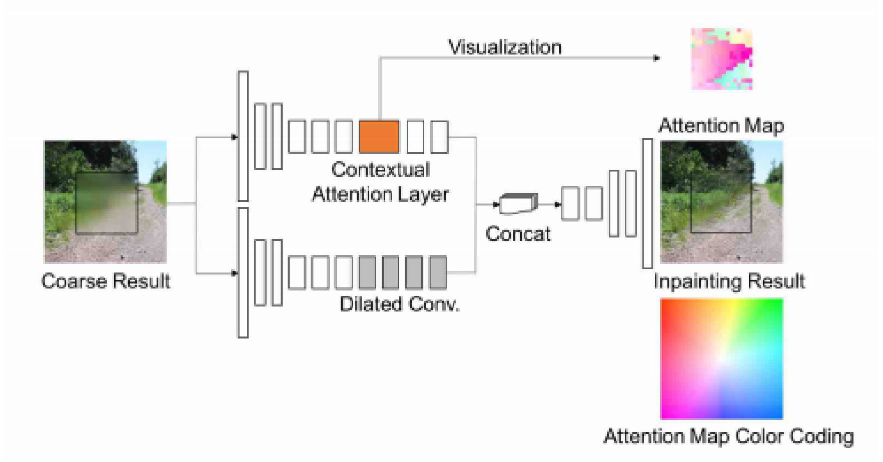


Figure 4: Based on coarse result from the first encoder-decoder network, two parallel encoders are introduced and then merged to single decoder to get inpainting result. For visualization of attention map, color indicates relative location of the most interested background patch for each pixel in foreground. For examples, white (center of color coding map) means the pixel attends on itself, pink on bottom-left, green means on top-right.

Hole 영역(Foreground)와 유사한 영역을 Cosine similarity를 이용해 찾아낸다.

해당 영역의 Feature를 이용해 영상을 Deconvolution하여 복원한다.

Interesting Results

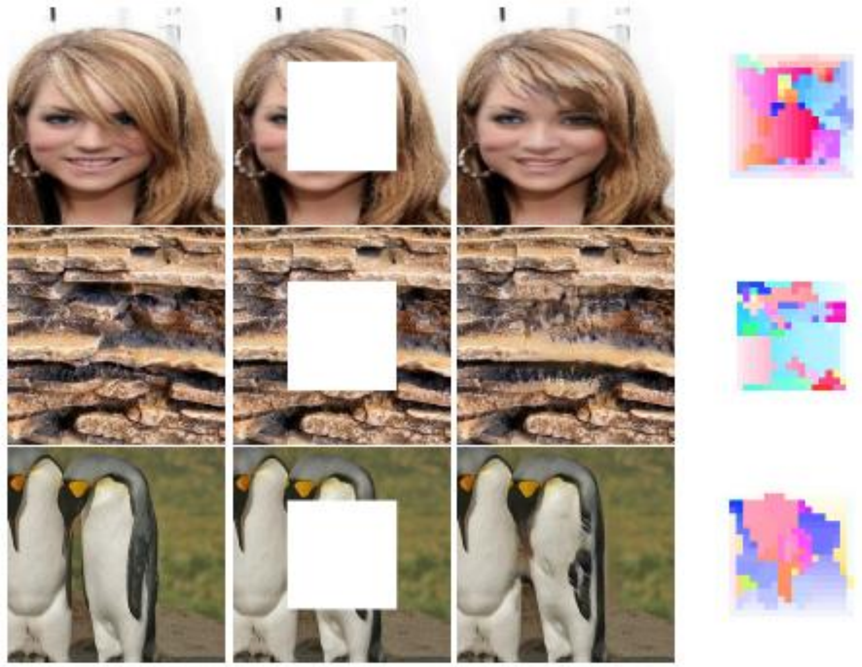
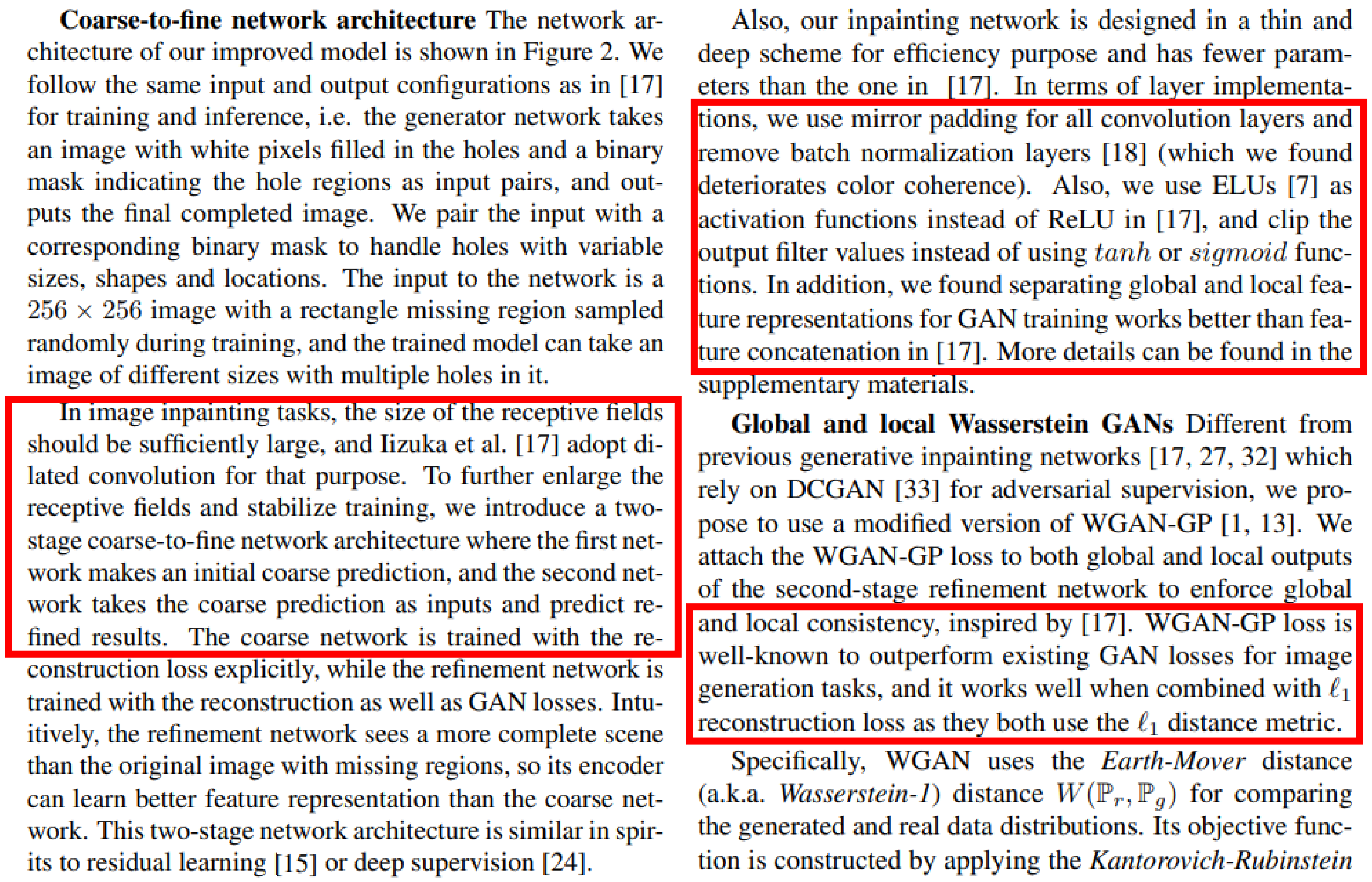


Figure 7: Sample results of our model on CelebA faces, DTD textures and ImageNet from top to bottom. Each row, from left to right, shows original image, input image, result and attention map (upscaled 4×), respectively.

주변 영역을 이용하여 충실하게 복원하는 경우도 있고, '얼굴'을 이해해 눈을 복원하기도 한다.

Insights



Coarse-to-fine network architecture The network architecture of our improved model is shown in Figure 2. We follow the same input and output configurations as in [17] for training and inference, i.e. the generator network takes an image with white pixels filled in the holes and a binary mask indicating the hole regions as input pairs, and outputs the final completed image. We pair the input with a corresponding binary mask to handle holes with variable sizes, shapes and locations. The input to the network is a 256×256 image with a rectangle missing region sampled randomly during training, and the trained model can take an image of different sizes with multiple holes in it.

In image inpainting tasks, the size of the receptive fields should be sufficiently large, and Iizuka et al. [17] adopt dilated convolution for that purpose. To further enlarge the receptive fields and stabilize training, we introduce a two-stage coarse-to-fine network architecture where the first network makes an initial coarse prediction, and the second network takes the coarse prediction as inputs and predict refined results. The coarse network is trained with the reconstruction loss explicitly, while the refinement network is trained with the reconstruction as well as GAN losses. Intuitively, the refinement network sees a more complete scene than the original image with missing regions, so its encoder can learn better feature representation than the coarse network. This two-stage network architecture is similar in spirits to residual learning [15] or deep supervision [24].

Also, our inpainting network is designed in a thin and deep scheme for efficiency purpose and has fewer parameters than the one in [17]. In terms of layer implementations, we use mirror padding for all convolution layers and remove batch normalization layers [18] (which we found deteriorates color coherence). Also, we use ELUs [7] as activation functions instead of ReLU in [17], and clip the output filter values instead of using $tanh$ or $sigmoid$ functions. In addition, we found separating global and local feature representations for GAN training works better than feature concatenation in [17]. More details can be found in the supplementary materials.

Global and local Wasserstein GANs Different from previous generative inpainting networks [17, 27, 32] which rely on DCGAN [33] for adversarial supervision, we propose to use a modified version of WGAN-GP [1, 13]. We attach the WGAN-GP loss to both global and local outputs of the second-stage refinement network to enforce global and local consistency, inspired by [17]. WGAN-GP loss is well-known to outperform existing GAN losses for image generation tasks, and it works well when combined with ℓ_1 reconstruction loss as they both use the ℓ_1 distance metric.

Specifically, WGAN uses the *Earth-Mover* distance (a.k.a. *Wasserstein-1*) distance $W(\mathbb{P}_r, \mathbb{P}_g)$ for comparing the generated and real data distributions. Its objective function is constructed by applying the *Kantorovich-Rubinstein*

좋은 논문은 꼭꼭 자세히 읽어보세요!