Submission Template

1. Team information

Team name:
(If you have participated as an individual, please use your Grand Challenge username as your team name)

Team member1
Name:
Institution:
E-mail:

Team member2(optional)
Name:
Institution:
E-mail:

Team member3(optional)
Name:
Institution:
E-mail:

2. Method description

2.1. Framework

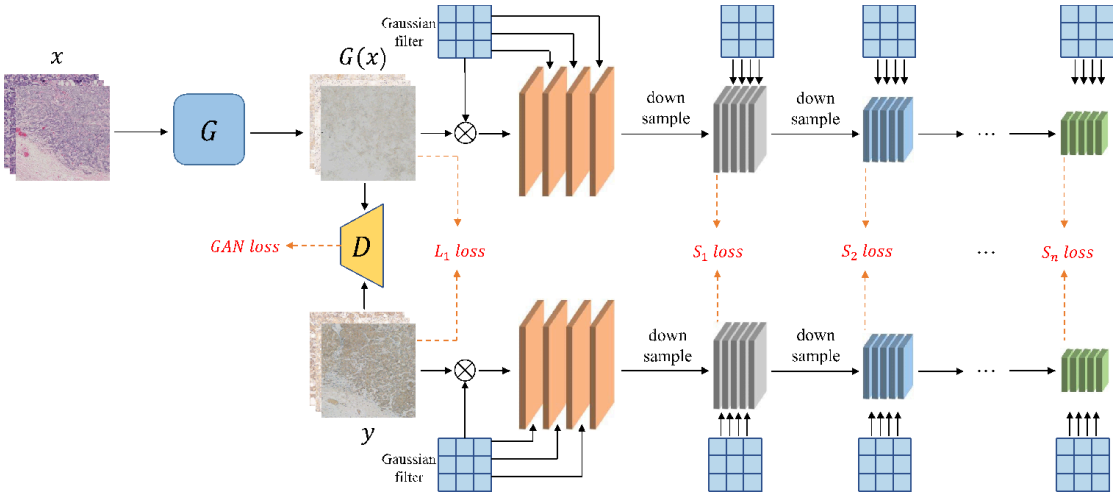

Figure 1. The framework of the proposed pyramid pix2pix. The image of each scale is obtained from the image of the previous scale after four Gaussian convolutions and one downsampling.

2.2. **Description**(including motivation, detailed introduction to the framework, loss function etc.)

The L_1 loss in pix2pix[2] algorithm directly calculates the difference between the generated image and the ground truth, which is too restrictive on the generated image. For our BCI dataset, we need to weaken the constraints L_1 loss, while aligning the generated image and ground truth at other scales. Inspired by scale-space theory[1], we will perform the same scale transformation on the generated image and ground truth. The scale transformation consists of two steps: 1) Using a low-pass filter to smooth the image. 2) Downsampling the smooth image. Since the Gaussian kernel is the only linear kernel that realizes the image scale transformation, our low-pass filter

uniformly uses the Gaussian kernel with a standard deviation of 1. With the progress of Gaussian filtering, the image becomes more and more blurred, and we reduce the resolution by downsampling to remove redundant pixels. For each resolution level (octave), multiple Gaussian convolutions are performed to achieve scale transformation.

Our pix2pix pyramid has several octaves, the first layer of each octave is obtained by down-sampling the last image of the previous octave; each octave has 5 layers and performs 4 Gaussian blurrings. For each output of octave, we define it as a scale.

In our Gaussian Pyramid, we extract the first layer of images in each octave to calculate the loss. The loss for each scale is denoted as $S_i (i = 1, 2, 3...)$:

$$S_i = E_{x,y,z}[\|F_i(y) - F_i(G(x,z))\|_1$$

where F_i and G represent the Gaussian filtering operation and the generator, respectively. x, y and z represent the input image, the ground truth, and random noise, respectively. Even if we cannot make the generated image highly consistent with the ground truth in the first octave, we can still make the generated image close to the ground truth on a higher-dimensional scale. Our multi-scale loss is recorded as:

$$L_{multi-scale} = \sum_i \lambda_i S_i$$

where λ_i represents the weight of scale i. Our adversarial loss $L_{cGAN}(G, D)$ and L_1 loss are still consistent with pix2pix.

Our overall objective function is:

$$G^* = arg L_{cGAN}(G, D) \quad + \lambda_1 L_1 + L_{multi-scale}$$

3. Experiment implementation

In our proposed method, we use the generator structure of resnet-9blocks as the baseline, while the discriminator structure uses the default patchGAN; the Gaussian kernel used is 3×3 with a standard deviation of 1; the input images are not preprocessed; the batch size is set to 2; the optimizer used is Adam; the total number of training epochs is set to 100: the learning rate of first 50 epochs is set to 0.0002 and the learning rate of the remaining 50 epochs gradually drops to 0.

References

[1] Lowe D G. Distinctive image features from scale-invariant keypoints[J]. International journal of computer vision, 2004, 60(2): 91-110.

[2] Isola P, Zhu J Y, Zhou T, et al. Image-to-image translation with conditional adversarial networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1125-1134.