

创建

模型参数说明

```
opt.input_nc = 3          # number of input image channels: 3 for RGB and 1 for grayscale
opt.output_nc = 3         # number of output image channels: 3 for RGB and 1 for grayscale
opt.ngf = 64              # number of gen filters in the last conv layer
opt.netG = 'resnet_9blocks' # specify generator architecture \
                           # [resnet_9blocks | resnet_6blocks | unet_256 | unet_128] \
                           # (根据论文内容, 6blocks 对应 128*128; 9blocks 对应 256*256 及以上)

opt.norm = 'instance'     # instance normalization or batch normalization [instance | batch | none]
opt.no_dropout = False    # no dropout for the generator
opt.init_type = normal    # network initialization [normal | xavier | kaiming | orthogonal]
opt.init_gain = 0.02      # scaling factor for normal, xavier and orthogonal
```

命令

```
net_G = networks.define_G(opt.input_nc, opt.output_nc, opt.ngf, opt.netG, opt.norm, \
                           not opt.no_dropout, opt.init_type, opt.init_gain)
```

模型结构

```
ResnetGenerator(
(model): Sequential(
  (0): ReflectionPad2d((3, 3, 3))
  (1): Conv2d(3, 64, kernel_size=(7, 7), stride=(1, 1))
  (2): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
  (3): ReLU(inplace)
  (4): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
  (5): InstanceNorm2d(128, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
  (6): ReLU(inplace)
  (7): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
  (8): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
  (9): ReLU(inplace)
  (10): ResnetBlock(
    (conv_block): Sequential(
      (0): ReflectionPad2d((1, 1, 1, 1))
      (1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
      (2): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
      (3): ReLU(inplace)
      (4): ReflectionPad2d((1, 1, 1, 1))
      (5): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
      (6): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
    )
  )
  (11): ResnetBlock(
    (conv_block): Sequential(
      (0): ReflectionPad2d((1, 1, 1, 1))
      (1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
      (2): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
      (3): ReLU(inplace)
      (4): ReflectionPad2d((1, 1, 1, 1))
```

```

(5): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
(6): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
)
)
(12): ResnetBlock(
  (conv_block): Sequential(
    (0): ReflectionPad2d((1, 1, 1, 1))
    (1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
    (2): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
    (3): ReLU(inplace)
    (4): ReflectionPad2d((1, 1, 1, 1))
    (5): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
    (6): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
  )
)
(13): ResnetBlock(
  (conv_block): Sequential(
    (0): ReflectionPad2d((1, 1, 1, 1))
    (1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
    (2): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
    (3): ReLU(inplace)
    (4): ReflectionPad2d((1, 1, 1, 1))
    (5): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
    (6): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
  )
)
(14): ResnetBlock(
  (conv_block): Sequential(
    (0): ReflectionPad2d((1, 1, 1, 1))
    (1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
    (2): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
    (3): ReLU(inplace)
    (4): ReflectionPad2d((1, 1, 1, 1))
    (5): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
    (6): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
  )
)
(15): ResnetBlock(
  (conv_block): Sequential(
    (0): ReflectionPad2d((1, 1, 1, 1))
    (1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
    (2): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
    (3): ReLU(inplace)
    (4): ReflectionPad2d((1, 1, 1, 1))
    (5): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
    (6): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
  )
)
(16): ResnetBlock(

```

```

(conv_block): Sequential(
  (0): ReflectionPad2d((1, 1, 1, 1))
  (1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
  (2): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
  (3): ReLU(inplace)
  (4): ReflectionPad2d((1, 1, 1, 1))
  (5): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
  (6): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
)
)
(17): ResnetBlock(
  (conv_block): Sequential(
    (0): ReflectionPad2d((1, 1, 1, 1))
    (1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
    (2): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
    (3): ReLU(inplace)
    (4): ReflectionPad2d((1, 1, 1, 1))
    (5): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
    (6): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
  )
)
(18): ResnetBlock(
  (conv_block): Sequential(
    (0): ReflectionPad2d((1, 1, 1, 1))
    (1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
    (2): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
    (3): ReLU(inplace)
    (4): ReflectionPad2d((1, 1, 1, 1))
    (5): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
    (6): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
  )
)
(19): ConvTranspose2d(256, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), output_padding=(1, 1))
(20): InstanceNorm2d(128, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
(21): ReLU(inplace)
(22): ConvTranspose2d(128, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), output_padding=(1, 1))
(23): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
(24): ReLU(inplace)
(25): ReflectionPad2d((3, 3, 3, 3))
(26): Conv2d(64, 3, kernel_size=(7, 7), stride=(1, 1))
(27): Tanh()
)
)

```