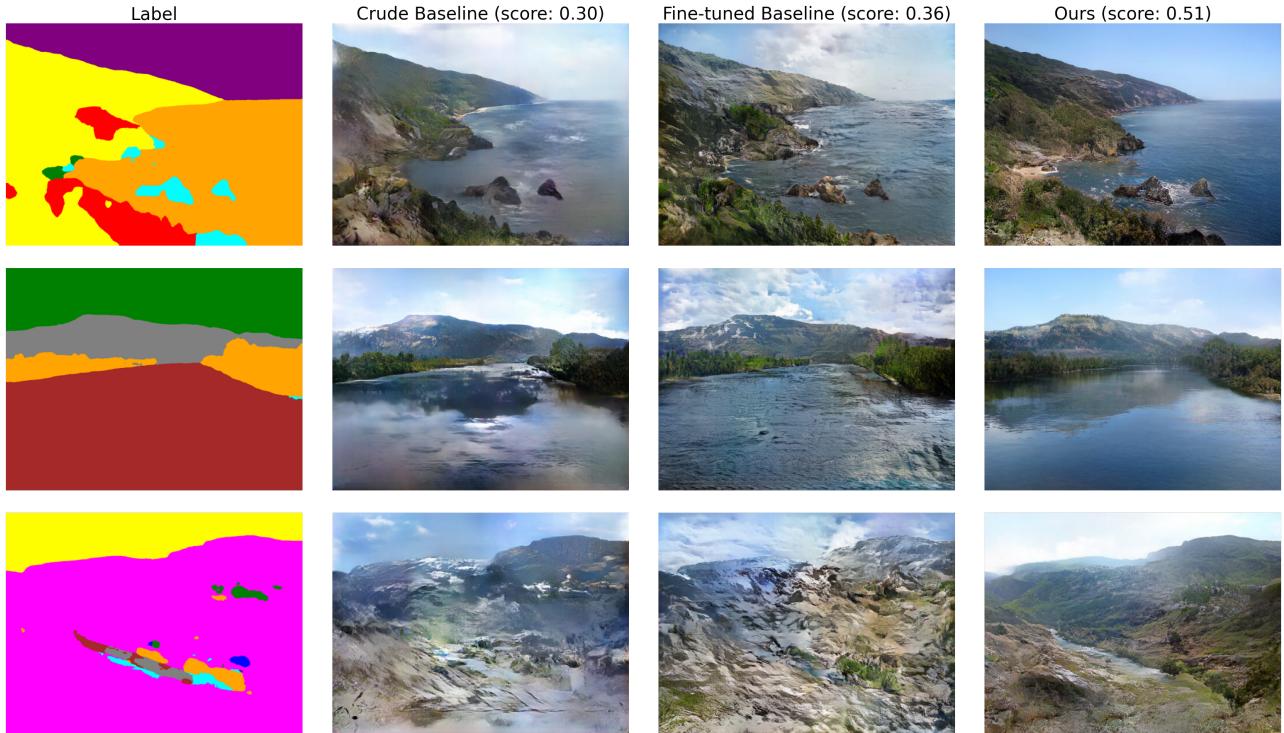


# 2022 春《计算机图形学基础》

## “风景图像生成”结题报告

高焕昂, 刘明道, 赵晨阳 @ 计{04, 02, 06}



**[摘要]** 图像生成任务一直以来都是十分具有应用场景的计算机视觉任务，在保证图片的真实性、清晰程度、多样性、美观性的前提下，如何从图像的语义分割图生成更加有意义、更加高质量的图片，仍是一个开放的议题。从传统的对抗生成网络模型（GAN）兴起至今，学界一直在不断提出更有创新性、更具规模的架构，比如 pix2pix、SPADE、TSIT 等等。我们小组对这些这个任务及有关模型架构进行了调研，并选择了其中的一些模型进行了复现，最终取得了 0.5189 的分数。我们将实验用源代码[开源](#)，供赛事方复现结果与后人参考。

## 1 题目简介 Introduction

风景图像生成任务可以形式化的表述为：寻找一个映射  $f_\theta$ ，其将标签域分布  $L$  映射为“虚拟生成”风景分布  $f(L)$ ，其中  $\theta$  可以使得  $f(L)$  与真实风景分布  $D$  最为接近。我们的题目是一个有监督的学习任务，即对于标签域分布的一个真子集  $L' \subset L$ ，我们给出其对应的真实风景  $D' \subset D$ 。于是，我们自然的想法就是，在训练时将对应的真实风景作为我们的监督信号。

**生成对抗网络模型 (GAN)** [1] 是学界目前对于不管是有条件的，还是无条件的图像生成任务的讨论热点。自从 pix2pix 模型在 MNIST 数据集上证明了其做图像生成任务的有效性以来，WGAN [6]、SPADE [14]、TSIT [16] 等模型一直在向更加高清、更加具有多样性的生成结果发起挑战，并在常见的数据集，如 CityScapes [28]、ADE20K [29] 上取得了一个又一个的进步。

我们队伍对本任务在近年来的发展历程和相关损失函数设计、模型架构等等进行了充分的调研，并选择 TSIT [16] 模型在 Jittor 深度学习框架上进行了复现。我们编写了完善的、具有扩展性的实验框架，撰写了本篇解题报告与调研报告，并在最终赛道排行榜上取得了天梯排名第 11 位的成绩。我们小组成员的贡献度列举如下：

- 高焕昂：文献调研、实验框架基础架构、第一阶段实验框架的编写与调试、报告撰写
- 刘明道：第二阶段实验框架与 TSIT 网络架构的实现与调试
- 赵晨阳：数据集清洗、输出筛选、报告撰写

## 2 相关工作与方法 Related Works & Methods

在进行我们的调研与实现时，以下提到的内容对我们有极大的启发作用。其他对我们的调研与实现有一定参考作用的资料，我们会在 [参考资料] 节中列出。

### 2.1 原始生成对抗网络

应用于图像生成的 GAN 的训练方法可以概述为“输入为一个随机向量  $z$ ，生成器  $G$  输出一幅图像  $G(z)$ ，而判别器  $D$  需要将真实图像  $x$  与合成图像  $G(z)$  区分开来。”

#### 2.1.1 随机变量的生成

基于函数逆变换的方法已经可以通过简单的均匀分布随机变量生成符合特定分布的复杂随机变量，而这一方法与 GAN 模型有着深刻的联系。

囿于计算机的计算确定性，生成真正随机的数字在理论上是不行的。但是，计算机能够使用伪随机数生成器生成大致遵循 0 和 1 之间的均匀随机分布的数字序列。通过特定数学方法，可以定义生成某些特定数字序列的算法，这些数字的分布非常接近理论随机数的分布。

假设  $X$  是某一复杂随机变量，而  $U$  是  $[0,1]$  上的均匀随机变量。随机变量完全由其[累积分布函数( CDF 定义。随机变量的 CDF 是从随机变量的定义域到区间  $[0,1]$  的函数，并且在一维中定义为：

$$CDF_X(x) = \mathbb{P}(X \leq x) \in [0, 1]$$

对于随机变量  $U$ ，我们有：

$$CDF_U(u) = \mathbb{P}(U \leq u) = u \quad \forall u \in [0, 1]$$

为简单起见，我们在这里假设函数  $CDF_X$  是可逆的，它的逆表示为  $CDF_X^{-1}$ （通过使用函数的广义逆，该方法可以很容易地扩展到不可逆的情况，但这并非需要关注的重点），接着定义：

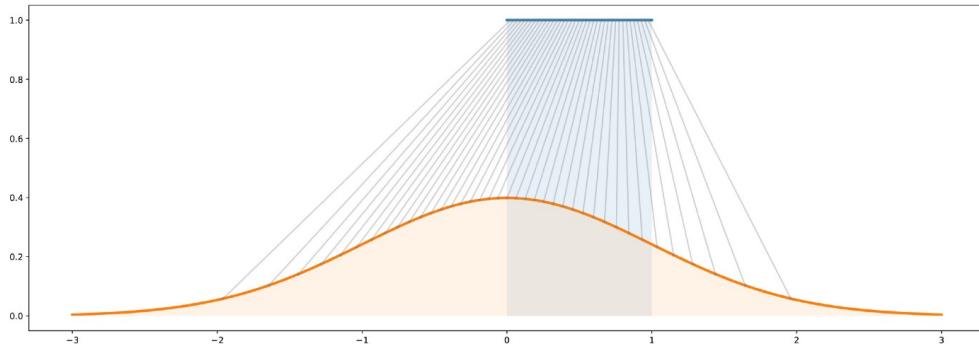
$$Y = CDF_X^{-1}(U)$$

我们有：

$$CDF_Y(y) = \mathbb{P}(Y \leq y) = \mathbb{P}(CDF_X^{-1}(U) \leq y) = \mathbb{P}(U \leq CDF_X(y)) = CDF_X(y)$$

$Y$  和  $X$  具有相同的 CDF，从而定义了相同的随机变量。因此，通过如上定义  $Y$ （作为均匀随机变量的函数），成功定义了具有目标分布的随机变量。

综上所述，逆变换方法是一种通过使均匀随机变量经过精心设计的**变换函数（逆 CDF）**来生成遵循给定分布的随机变量的方法。事实上，这个**逆变换方法**的概念可以扩展为具有一般性的**变换方法**——目标随机变量  $X$  作为某些更简单的随机变量  $Y$  的函数，而  $Y$  不一定是均匀的，然后变换函数也不一定是逆 CDF。从概念上讲，**变换函数**的目的是对初始概率分布进行重塑。



逆变换方法图示

蓝色： $[0,1]$  上的均匀分布；橙色：标准高斯分布；灰色：从均匀分布到高斯分布的映射

### 2.1.2 生成模型

假设我们有大小为  $n \times n$  像素的狗的黑白方形图像。我们可以将每个数据重塑为  $N = n \times n$  维向量（通过将列堆叠在一起），这样狗的图像就可以用向量表示。然而，这并不意味着所有的向量都代表一只狗。生成狗的新图像的问题等价于在  $N$  维向量空间上按照狗的概率分布生成新向量的问题。事实上，我们面临着一个针对特定概率分布生成随机变量的问题。

在这一点上，我们可以提到两件重要的事情。首先，“狗的概率分布”是一个非常复杂的分布在非常大的空间上的分布。其次，即使我们可以假设存在这种潜在分布，我们显然不知道如何明确地表达这种分布。前面的两点都使得从这个分布生成随机变量的过程非常困难。接下来让我们尝试解决这两个问题。

在大多数情况下，非常复杂的函数自然意味着神经网络建模。通过一个神经网络对变换函数进行建模，该神经网络将一个简单的  $N$  维均匀随机变量作为输入，并返回另一个  $N$  维随机变量作为输出，该变量在训练后应该遵循正确的“狗的概率分布”。

我们需要训练网络来表达正确的变换函数。为此，提出两种不同的训练方法。直接训练方法包括比较真实和生成的概率分布，并通过网络反向传播差异（误差）。这是规则生成匹配网络（GMN）的想法。对于间接训练方法，我们不直接比较真实分布和生成分布。相反，我们通过使这两个分布通过选择的下游任务来训练生成网络，这样生成网络相对于下游任务的优化过程将强制生成的分布接近真实分布。后一个想法即是 GAN 的基本思想。

GAN 的下游任务是真实样本和生成样本之间的区分任务。如上文所述，在 GAN 架构由两部分组成，首先是经过训练以尽可能地欺骗鉴别器的生成器，而后是鉴别器，它对真实和生成的数据进行采样，并尝试尽可能好地对它们进行分类。

具体而言，生成器是一个对变换函数进行建模的神经网络。它将一个简单的随机变量作为输入，并且必须在经过训练后返回一个遵循目标分布的随机变量。由于它非常复杂且未知，我们决定用另一个神经网络对鉴别器进行建模，该神经网络构建了一个判别函数。它将一个点（在我们的狗示例中为 N 维向量）作为输入，并以该点为真的概率作为输出。

一旦定义好，两个网络就可以以相反的目标联合训练：

- 生成器的目标是欺骗判别器，因此训练生成神经网络以最大化真实数据和生成数据之间的最终分类误差
- 判别器的目标是检测虚假生成数据，因此训练判别神经网络以最小化最终分类误差

因此，在训练过程的每次迭代中，生成网络的权重都会更新以增加分类误差，而判别网络的权重会更新以减少该误差。

相反的目标和两个网络的对抗性训练解释了对抗网络的名称：两个网络都试图互相击败。并且在这一过程中，它们都变得越来越好。从博弈论的角度来看，我们可以将此设置视为一个极小极大的两人游戏，他们之间的竞争使这两个网络在各自的目标上进步。

### 2.1.3 数学原理

GAN [1] 由两个网络组成：

- 一个生成网络  $G(\cdot)$ ，它接受一个密度为  $p_z$  的随机输入  $z$  并返回一个输出  $x_g = G(z)$ ，它应该遵循目标的概率分布
- 一个判别网络  $D(\cdot)$ ，它接受一个输入  $x$ ，该输入  $x$  可以是真实的数据  $x_t$ ，其密度表示为  $p_t$ ，或生成的数据  $x_g$ ，其密度  $p_g$  是由密度  $p_z$  生成；它的返回值  $D(x)$  为  $x$  是真实数据的概率

如果我们以相同的比例向鉴别器输入真实和生成的数据，则鉴别器的预期绝对误差可以表示为：

$$\begin{aligned} E(G, D) &= \frac{1}{2} \mathbb{E}_{x \sim p_t} [1 - D(x)] + \frac{1}{2} \mathbb{E}_{z \sim p_z} [D(G(z))] \\ &= \frac{1}{2} (\mathbb{E}_{x \sim p_t} [1 - D(x)] + \mathbb{E}_{x \sim p_g} [D(x)]) \end{aligned}$$

生成器的目标是欺骗鉴别器，鉴别器的目标是能够区分真实数据和生成的数据。因此，在训练生成器时，我们希望最大化这个误差，同时我们试图最小化判别器的误差：

$$\max_G \left( \min_D E(G, D) \right)$$

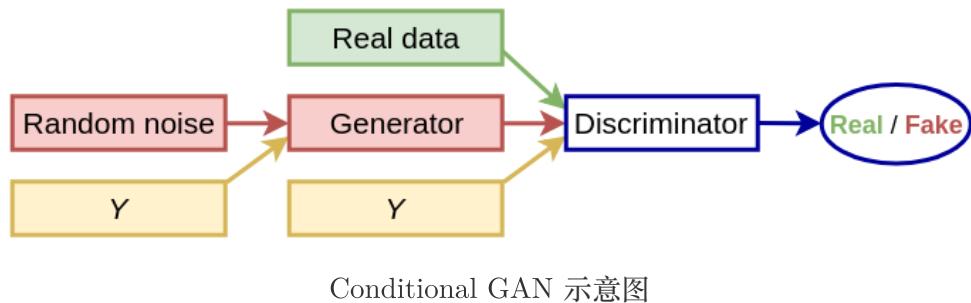
对于任何给定的生成器  $G$ （以及生成的概率密度  $p_g$ ），最好的鉴别器是能够最小化下式的鉴别器：

$$\mathbb{E}_{x \sim p_t} [1 - D(x)] + \mathbb{E}_{x \sim p_g} [D(x)] = \int_{\mathbb{R}} (1 - D(x)) p_t(x) + D(x) p_g(x) dx$$

## 2.1.4 Conditional GAN

原始 GAN 对于生成器几乎没有任何约束，使得生成过程过于自由，模型变得难以控制。CGAN (Conditional GAN) 在原始 GAN 的基础上增加了约束条件，控制了 GAN 过于自由的问题，使网络朝着既定的方向生成样本。

CGAN 的网络结构如下图所示，CGAN 生成器和判别器的输入多了一个约束项  $y$ ，约束项  $y$  可以是一个图像的类别标签，也可以是图像的部分属性数据。



## 2.2 Pix2Pix

Pix2Pix [10] 模型是 CGAN 应用于 Image Translation 的开山之作，其目标是从一张图像转换为另一张图像，也即学习从输入图像到输出图像的映射。

Pix2Pix 是 CGAN 的改进模型，在传统 GAN 和 CGAN 的基础上，网络架构有如下创新：

### 2.2.1 U-Net 生成器

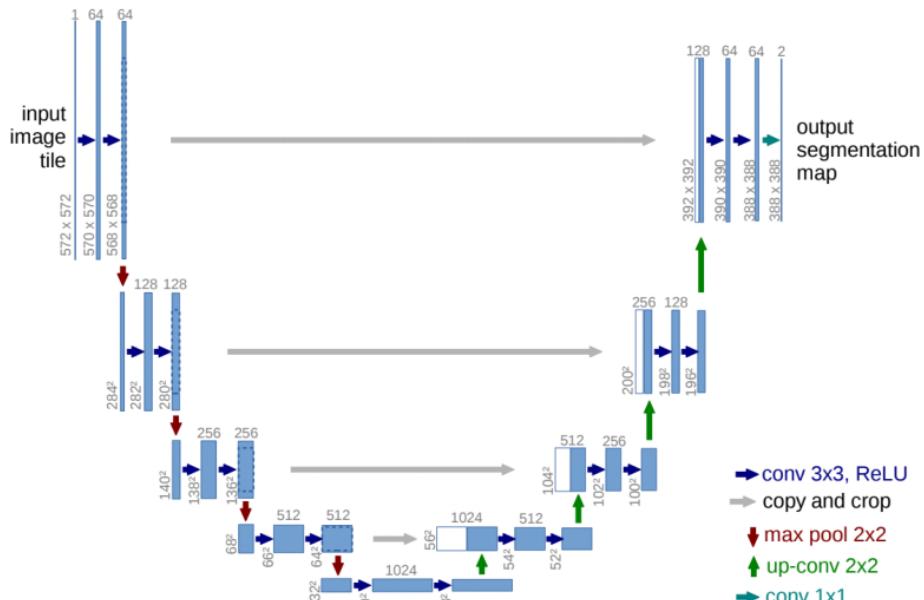
传统生成器是一个编码器-解码器网络（E-D 网络）——首先是一系列下采样层，然后是瓶颈层，然后是一系列上采样层。

在 Pix2Pix 中，作者使用带有跳跃连接的 U-Net [30] 架构作为 E-D 网络。

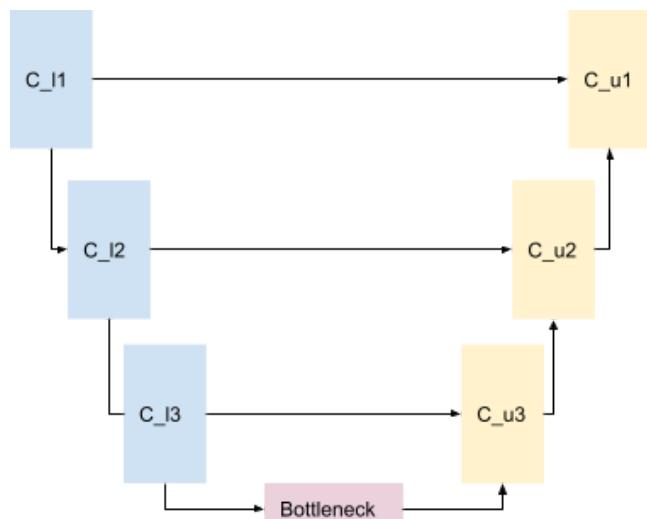
UNet [30] 由两个主要部分组成：

1. 由卷积层（左侧）组成的收缩路径，在提取信息的同时对数据进行下采样。
2. 由向上转置卷积层（右侧）组成的扩展路径，对信息进行上采样。

假设我们的下采样具有三个卷积层  $C_l(1, 2, 3)$ ，那么我们必须确保我们的上采样具有三个转置卷积层  $C_u(1, 2, 3)$ 。这是因为我们想使用跳过连接来连接相同大小的相应块。



U-Net Generator



跳过连接网络示意图

## 下采样

在下采样期间，每个卷积块提取空间信息并将信息传递给下一个卷积块以提取更多信息，直到它到达称为**瓶颈**的中间部分。上采样从瓶颈开始。

## 上采样

在上采样期间，每个转置卷积块扩展来自前一个块的信息，同时连接来自相应下采样块的信息。通过连接信息，网络可以学习根据这些信息组装更精确的输出。

该架构能够定位，也即能够逐像素地找到感兴趣的对象。此外，U-Net 还允许网络将上下文信息从较低分辨率的层传播到较高分辨率的层，使得网络生成高分辨率的样本。

## 2.2.2 PatchGAN

传统的直接机器学习采用损失函数回传梯度的方式进行参数优化，而采用的损失函数包括 L1 与 L2 两种：

$$\text{L1 Loss Function} = \sum_{i=1}^n |y_{\text{true}} - y_{\text{predicted}}|$$

$$\text{L2 Loss Function} = \sum_{i=1}^n (y_{\text{true}} - y_{\text{predicted}})^2$$

对于 AE [2] 和 VAE [3] 的相关研究表明，用 L1 loss 和 L2 loss [7] 进行图像重建会导致结果较为模糊，L1 和 L2 loss 并不能很好的恢复图像的高频部分(图像中的边缘等)，但能较好地恢复图像的低频部分(图像中的色块)。为了能更好得对图像的局部做判断，Pix2Pix 中提出 patchGAN 的结构，其思想是将图像等分成若干个 patch，分别判断每个 Patch 的真假，最后再取平均。

具体而言，PatchGAN 包含许多转置卷积块。它对图片  $N \times N$  大小的部分做卷积。 $N$  为任意大小，它可以比原始图像小，但仍然能够产生高质量的结果。PatchGAN 可以有效地将图像建模为马尔可夫随机场，其中  $N \times N$  被视为一个独立的 patch。因此，PatchGAN 可以理解为纹理损失的一种形式。由于鉴别器相比生成器具有更少的参数，因此鉴别器实际上运行更快。

## 2.2.3 目标函数

按照 CGAN 基本原理一节的内容，传统目标函数为：

$$L_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))]$$

同时定义与 ground truth 的 L1 loss：

$$L_{L1}(G) = \mathbb{E}_{x,y,z} [\|y - G(x, z)\|_1]$$

所以最终的目标函数为：

$$G^* = \arg \min_G \max_D L_{cGAN}(G, D) + \lambda L_{L1}(G)$$

GAN 本身其实是一种相对于 L1 loss 更好的判别准则或者 loss。有时候单独使用 GAN loss 效果好，有时候与 L1 loss 配合起来效果好。在 pix2pix 中，作者将 L1 loss 和 GAN 相结合使用，因为作者认为 L1 loss 可以恢复图像的低频部分，而 GAN 可以恢复图像的高频部分。

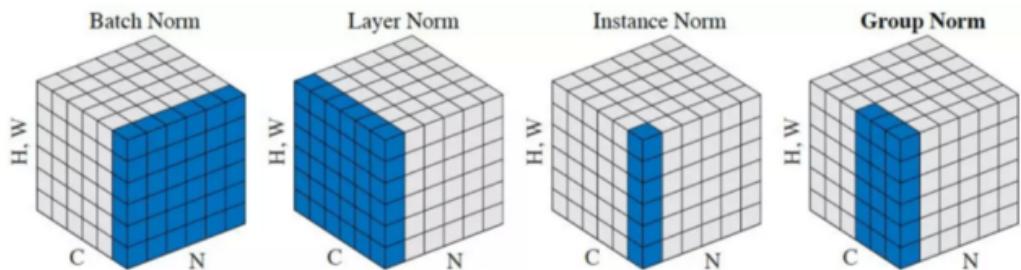
## 2.3 SPADE

从 2017 年开始， Pix2PixHD [31] 和 Vid2Vid [32] 等方法应用于语义分割图重建为图像，而 2019 年的 SPADE [14] 方法又在此领域有了显著进步。

不同于过去已有的图像合成 GAN，简单地将卷积层、归一化层、以及非线性层堆叠在一起构成生成模型，GauGAN [14] 引入了一个新的归一化层，在已有的 CGAN 基础上，GauGAN 引入了 SPADE 方法进行归一化。

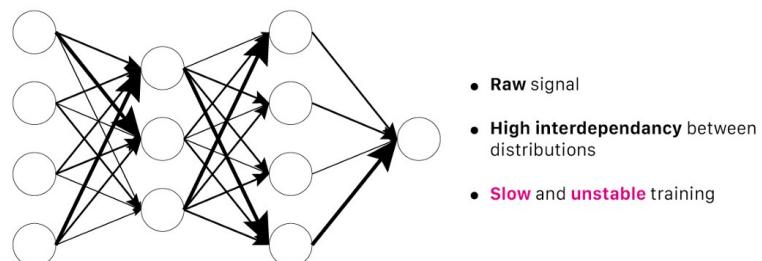
作者认为，假设语义图输入单个类别，则一整张图都是同一个数，在经过 InstanceNorm 之后会输出某个固定的 bias，语义信息完全丢失。所以 pix2pix 就无法生成有效的图片，没有任何类别信息。针对这个问题，作者提出了 SPADE (SPatially-Adaptive (DE)normalization) 方法，通过 SPADE 使用语义图来调整 normalization 输出的结果，使其更好的具有语义信息，并将语义信息贯穿整个网络。并且其方法可以应对各种使用语义图的生成任务。

### 2.3.1 无条件归一化

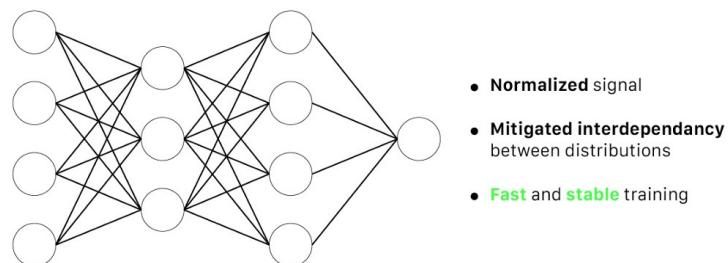


In batch normalization, the statistics are computed over feature maps across all batches. In instance normalization, the statistics are computed over feature maps across a single image.

传统的无条件归一化方法包括 batch normalization 与 instance normalization 等等。以 batch normalization(BN) 为例，其意义如下图所示：



没有批量归一化(BN)的多层感知器(MLP)



经过批量归一化(BN)的多层感知器(MLP)

深度学习中的 normalization 通常包括三个步骤：

1. 计算相关统计数据（如均值和标准差）

2. 通过减去平均值并将这个数字除以标准偏差来标准化输入
3. 重新缩放输入  $y = \gamma X + \beta$ , 其中参数  $\gamma, \beta$  可学习

具体到 Batch Normalization, 在每个隐藏层, Batch Normalization 将信号转换如下:

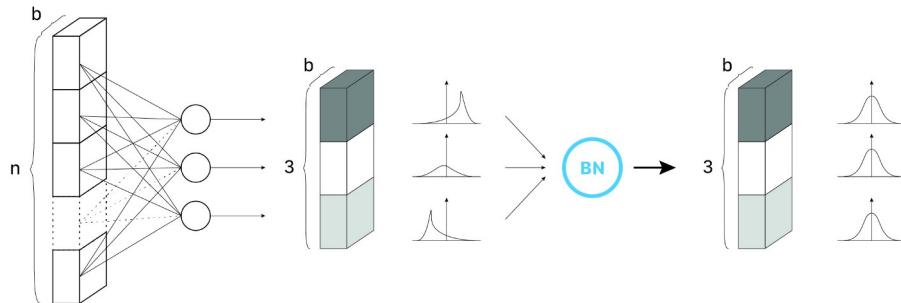
$$\mu = \frac{1}{n} \sum_i Z^{(i)}$$

$$\sigma^2 = \frac{1}{n} \sum_i (Z^{(i)} - \mu)^2$$

$$Z_{\text{norm}}^{(i)} = \frac{Z^{(i)} - \mu}{\sqrt{\sigma^2 - \epsilon}}$$

$$\tilde{Z} = \gamma * Z_{\text{norm}}^{(i)} + \beta$$

BN 层首先确定整个批次的激活值的均值  $\mu$  和方差  $\sigma^2$ 。然后它用对激活向量  $Z^{(i)}$  进行归一化。这样, 每个神经元的输出在批次中遵循标准正态分布。( $\epsilon$  是用于数值稳定性的常数)

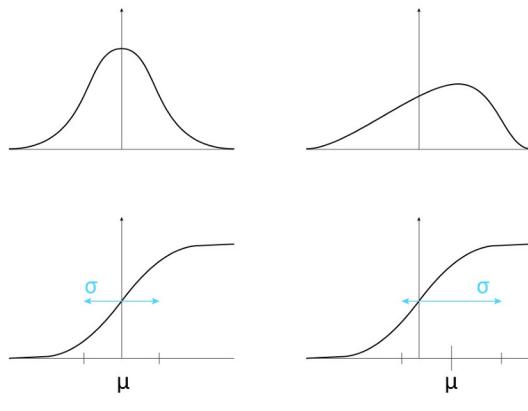


以 3 个神经元的隐藏层与 batch size 为 3 来示意 batch normalization 的第一步。

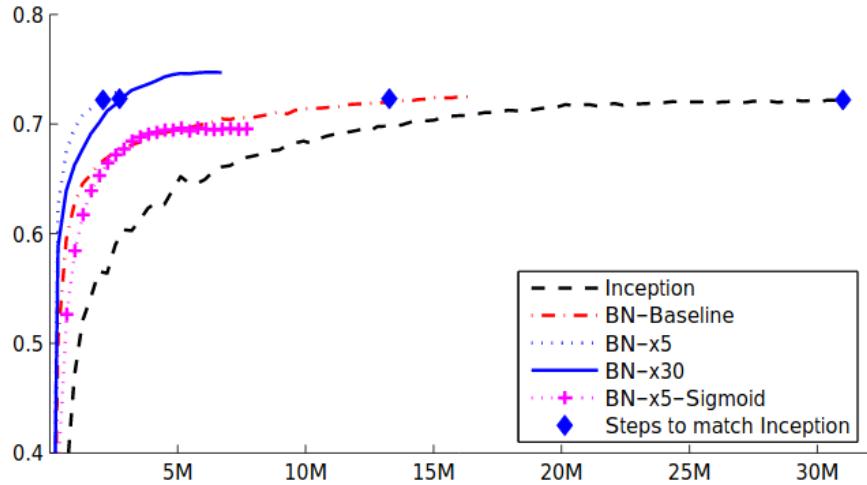
最终, BN 通过使用  $\gamma$  和  $\beta$  这两个可训练参数应用线性变换来计算层的输出  $\tilde{Z}^{(i)}$ 。这样的步骤允许模型通过调整这两个参数为每个隐藏层选择最佳分布:

可以将 BN 层视为 normalize 部分跟 denormalize 部分。normalize 部分就是 BN 的归一化部分, 计算 N 个 featuremap 某个通道的均值跟方差, 用均值跟方差进行归一化:

- $\gamma$  允许调整标准偏差。
- $\beta$  允许调整偏差, 在右侧或左侧移动曲线。



BN 取得的效果如下图所示:



在 ImageNet(2012) 上验证 BN 对训练的影响。比较了五个网络：Inception 是普通的 Inception 网络，BN-X 是具有 BN 层的 Inception 网络。x1、x5、x30 是 3 种不同的学习率。BN-X-Sigmoid 是一个带有 BN 层的 Inception 网络，所有 ReLU 都被 sigmoid 取代。

BN 层使训练更快，并允许更广泛的学习率，而不会影响训练收敛。

### 2.3.2 SPADE

作者认为无条件归一化会导致语义信息的丢失。传统的 batch normalization 为每个通道学习其参数集，也即  $\gamma$  和  $\beta$  是大小较小的向量或者就是一个数值，然而，在 SPADE 中，作者将 batch norm 的参数张量大小增加到与像素大小相同，也即  $\gamma$  和  $\beta$  是与像素大小相同的张量，从而 SPADE 能为特征图中的每个像素学习其参数集。在原文中， $\gamma$  和  $\beta$  是由分割 mask 通过卷积得到的，对语义分割 mask 来说，每个像素的值即是该像素点的类别。

### 2.3.3 SPADE Generator

作者结合残差卷积网络与 SPADE 构建了 SPADE Generator，它产生两个特征图：一个对应于逐像素的  $\gamma$ ，另一个对应逐像素的  $\beta$ ，两个特征图中的每个值代表用于重新缩放特征图中相应像素的值的参数。

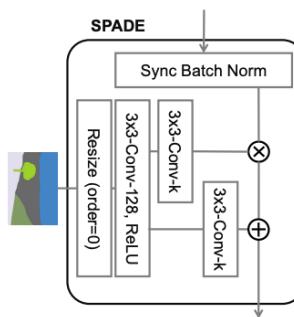


Figure 10: SPADE Design. The term  $3 \times 3\text{-Conv-}k$  denotes a 3-by-3 convolutional layer with  $k$  convolutional filters. The segmentation map is resized to match the resolution of the corresponding feature map using nearest-neighbor down-sampling.

上图通过 Batch Norm( Sync Batch Norm ) 模块来进行统计计算，这一部分的设计与 Batch Normalization 完全相同。Sync Batch Norm 是在多 GPU 系统上进行的优化。一般来说，如果你有一个批量大小，比如 32 并且你有 8 个 GPU，那么 Jittor nn.BatchNorm2d layer 将在每个 GPU 上分别计算 4 个批次的统计数据并更新参数。在 Sync Batch Norm 中，统计数据是在整个 32 个图像上计算的。当每个 GPU 批量大小较小（例如 1 或 2）时，计算小批量的统计数据可能会产生非常嘈杂的估计，从而导致训练波动。因而此时同步归一化作用显著。

从 SPADE 模块中获得的特征图逐元素相乘并添加到归一化的输入图中。其中每个卷积层都遵循限制卷积滤波器的 Lipschitz 常数的 Spectral Norm。

$$\gamma_{c,y,x}^i(\mathbf{m}) \frac{h_{n,c,y,x}^i - \mu_c^i}{\sigma_c^i} + \beta_{c,y,x}^i(\mathbf{m})$$

其中  $h_{n,c,y,x}^i$  是归一化之前的输出， $\mu_c^i$  和  $\sigma_c^i$  分别是第  $c$  个通道的均值和标准差：

$$\mu_c^i = \frac{1}{NH^iW^i} \sum_{n,y,x} h_{n,c,y,x}^i$$

$$\sigma_c^i = \sqrt{\frac{1}{NH^iW^i} \sum_{n,y,x} (h_{n,c,y,x}^i)^2 - (\mu_c^i)^2}.$$

变量  $\gamma_{c,y,x}^i(\mathbf{m})$  和  $\beta_{c,y,x}^i(\mathbf{m})$  是可通过学习来进行调节的参数。

### 2.3.4 SPADE 方法的有效性

首先，GauGAN 的输入是一个语义分割图，它被进一步独热编码（One-hot Encoding）。

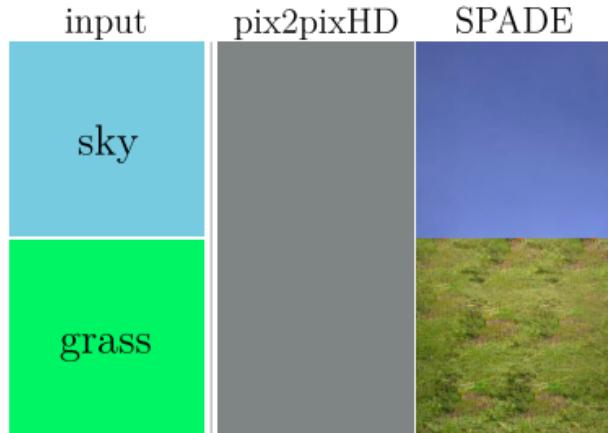
这意味着 GAN 必须输入标签值相同的区域，并生成具有不同值的像素，以使它们看起来像一个真实的图像。为每个像素设置一组不同的 Batch Norm 参数有助于解决此任务，而不是为特征图中的每个通道提供一个 Batch Norm 参数。

另外，作者认为 SPADE 会导致更具辨别力的语义信息。相同的像素类别具有相同的分割结果与近似的卷积结果，因此  $\gamma$  和  $\beta$  具有空间性，且这一空间性与分割 mask 保持一致，所以说 SPADE 是 SPatially-Adaptive 的，也即具有空间自适应性。

作者对 SPADE 的 Spatially-Adaptive 做了验证，考虑极端情况，当分割mask只有一种类别的时候，如下图的天空和草地，在经过 Instance normalization 之后，pix2pixHD 对这两种分割 mask 的输出都一样，SPADE 能对其进行区分。

而后，作者探讨了 SPADE 和 Instance Norm 的输出在给定相同标签的语义图的情况下有何不同。

作者认为语义信息在 normalization 过程中被大量删除。SPADE 和 Instance Norm 中的标准化步骤是相同的，它们不同的地方是重新缩放步骤。具体而言，在 Pix2Pix 中，Instance Norm 层的参数是不可学习的，Instance Norm 只是进行归一化（ $\gamma$  设置为 1 和  $\beta$  设置为 0）。然而，在 GauGAN 中，SPADE 具有可学习的参数。最后，Instance Norm 使用的 batch size 为 1，而 SPADE 和 Batch Norm 都可以利用更大的批量大小，可以有效减少统计噪声。



**Figure 3: Comparing results given uniform segmentation maps: while SPADE generator produces plausible textures, pix2pixHD [40] produces identical outputs due to the loss of the semantic information after the normalization layer.**

### 2.3.5 鉴别器与编码器

Gau GAN 采用了 PatchGAN 架构作为鉴别器，已在上文叙述。

在编码器部分，与普通 GAN 不同，GauGAN 不采用随机噪声向量，而仅采用语义图。这意味着给定单个输入语义图，输出将始终是确定性的。这违背了图像合成的精神，因为 GauGAN 对同一语义分割图输出重建不同图像的能力受到高度重视。

为此，作者设计了一个编码器（Encoder）。编码器取出一张图像的语义分割图，将其编码成两个向量。这两个向量用作正态高斯分布的均值和标准差。然后从这个分布中采样一个随机向量，与输入语义图一起连接作为生成器的输入。

同时，编码器输入的随机向量还可以作为生成图像的风格信息，每个随机向量将生成具有相同语义布局但不同模态特征（如颜色、亮度等）的图像。

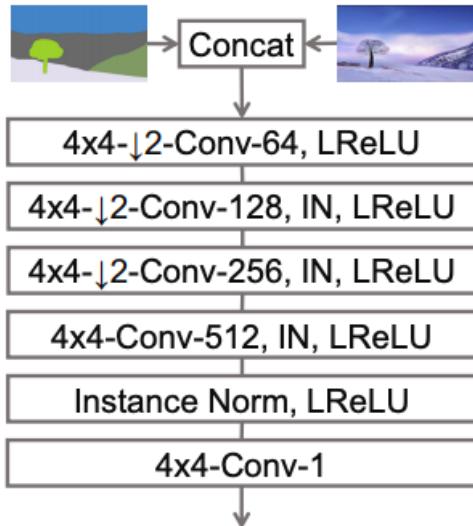


Figure 13: Our discriminator design largely follows that in the pix2pixHD [40]. It takes the concatenation the segmentation map and the image as input. It is based on the PatchGAN [20]. Hence, the last layer of the discriminator is a convolutional layer.

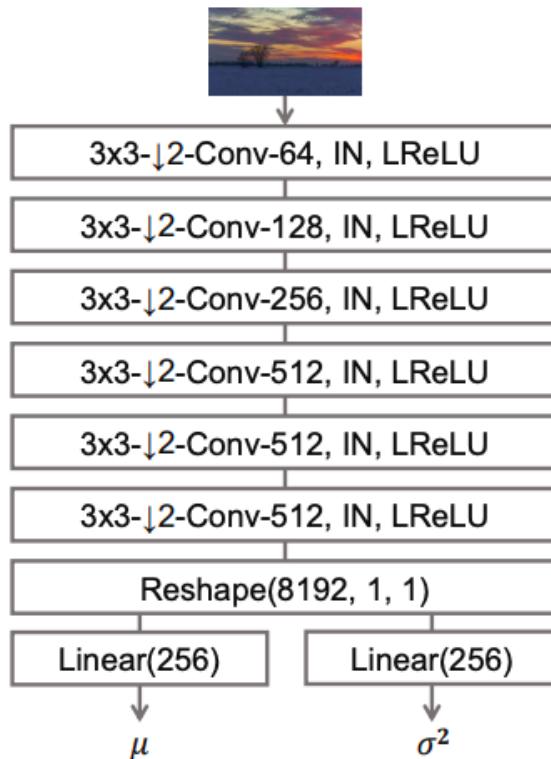


Figure 14: The image encoder consists a series of convolutional layers with stride 2 followed by two linear layers that output a mean vector  $\mu$  and a variance vector  $\sigma$ .

## 2.3.6 损失函数

### Multiscale Adversarial Loss

GauGAN 包含铰链损失 (Hinge Loss)，这在 SAGAN 和 Geometric GAN 等论文中也有所体现。下面是损失函数：

$$L_D = -\mathbb{E}_{(x,y) \sim p_{\text{data}}} [\min(0, -1 + D(x, y))] - \mathbb{E}_{z \sim p_z, y \sim p_{\text{data}}} [\min(0, -1 - D(G(z), y))]$$
$$L_G = -\mathbb{E}_{z \sim p_z, y \sim p_{\text{data}}} D(G(z), y)$$

给定生成器生成的图像，我们创建一个 image pyramid，将生成的图像调整为多个比例。然后，使用判别器计算每个尺度的真实度分数并反向传播损失。

### Feature Matching Loss

特征匹配损失鼓励 GAN 不仅能生成欺骗生成器的图像，而且生成的图像还应该具有与真实图像相同的统计特性。为了做到这一点，GauGAN 加入了真实图像与生成图像之间的 L1 距离。

如下所示，为生成图像的所有尺度计算特征匹配损失：

$$L_{FM}(G, D_k) = \mathbb{E}_{s,x} \sum_{i=1}^T \frac{1}{N_i} \left[ \|D_k^{(i)}(s, x) - D_k^{(i)}(s, G(s))\|_1 \right]$$

### VGG Loss

VGG Loss 类似于 Feature Matching Loss，唯一的区别是没有使用鉴别器来计算特征图，而是使用在 Imagenet 上预训练的 VGG-19 来计算真实图像和假图像的特征图。然后我们惩罚这些地图之间的 L1 距离。

$$L_{VGG}(G, D_k) = \mathbb{E}_{s,x} \sum_{i=1}^5 \frac{1}{2^i} [\|VGG(x, M_i) - VGG(G(s), M_i)\|_1]$$

### Encoder Loss

作者对编码器使用 KL 散度损失：

$$L_{KLD} = D_{kl}(q(z | x) || p(z))$$

在上式中， $q(z | X)$  称为变分分布，我们从中抽出随机向量  $z$  并给定真实图像  $X$ ，而  $p(z)$  是标准高斯分布。

上述损失函数可以理解为 Variational Auto-Encoder Loss 中的正则化损失项。使用编码器，GauGAN 起到了变分自动编码器中的解码器的作用，而 KL 散度损失项充当编码器的正则化项。这种损失会惩罚编码器预测的分布与零均值高斯分布之间的 KL 散度。如果缺失这种损失，编码器可以通过为数据集中的每个训练示例分配一个不同的随机向量来作弊，而不是实际学习一个捕获数据模态的分布。

## 2.4 TSIT

本次算法挑战赛，在对 baseline 代码进行修改调试与尝试 GauGAN 后，我们组又参考了 TSIT [16] 一文的方法，改进了小组的方法。

TSIT 是一个通用简洁而功能强大的框架，原作者提出了构思精妙且简单的 two-stream generative model，避免了复杂的循环连续性( cycle consistency ) 等额外约束，并将 TSIT 框架广泛应用于 **Image-to-image translation**, **Arbitrary style transfer**, **Semantic image synthesis** 等领域。

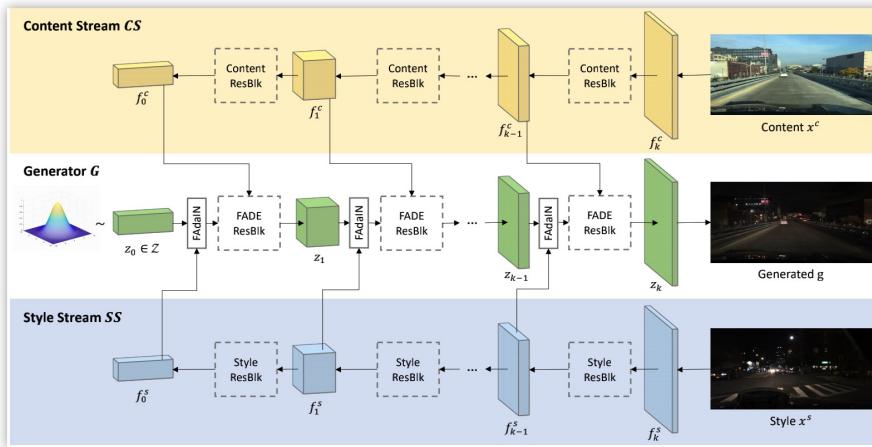
### 2.4.1 概述

我们小组将 TSIT 应用到本次风景图片的生成任务——本质上是语义图像重建。由于 ground truth 同时具有语义信息，语义图像重建任务相较无监督的风格迁移任务数据更加丰富可靠，能够将语义分割图与 ground truth 成对训练。而一般的语义重建任务需要计算生成图像的分布与 ground truth 图像的分布在高维空间的距离。

得益于完全基于特征迁移的设计，TSIT 框架显得相对简洁。在图像重建领域之前的相关工作主要侧重于语义信息本身，较少能够在语义信息与风格特征之间建立平衡。TSIT 框架通过高度对称的双数据流网络将语义信息与风格信息分解到多层次特征层面，共同影响图像的生成。具体而言，语义信息在语义数据流网络中通过 element-wise feature adaptive denormalization ( FADE )得到了保留，同时风格信息通过 feature adaptive instance denormalization( FAdaIN )得到保留。TSIT 框架只使用了简单常用的 adversarial loss 与 perceptual loss，并未使用复杂的 cycle consistency。同时，框架既可适用于无监督的风格迁移，也可用于有监督的图像重建，不必进行复杂的数据预处理。

TSIT 在相关任务上取得了惊人的成果，这主要得益于多尺度的特征归一化( FADE and FAdaIN )能够捕捉到从微观到整体的语义与风格信息，同时双数据流网络能够巧妙地将语义流信息与风格流信息进行整合。

### 2.4.2 网络架构

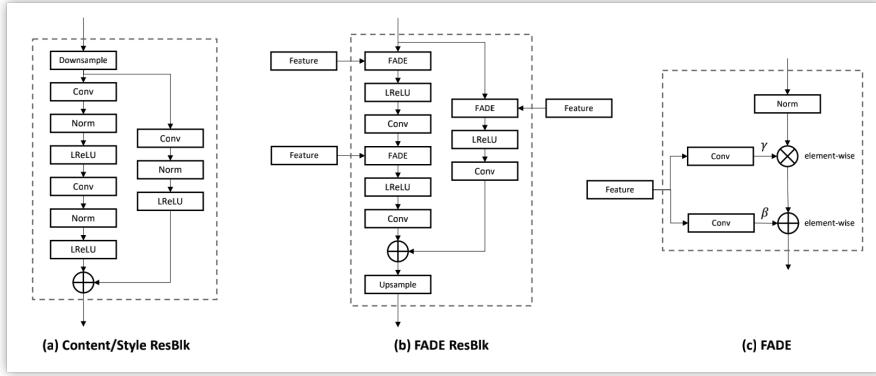


上图展示了网络架构除了 discriminators 外的部分，包括语义数据流网络、风格数据流网络与生成器网络，三者均是全卷积并且设计高度对称。内部子模块主要包括语义残差模块、风格残差模块与 FADE 残差模块。

## 语义数据流网络与风格数据流网络

与传统的 CGAN 不同，TSIT 在生成器的输入中采用了双数据流网络架构。语义数据流网络与风格数据流网络完全相同且对称，旨在多感受野分别提取对应的特征信息。

单个数据流网络主要基于标准的残差模块，作者将其分别称为语义残差块与风格残差块。如下图所示：



每个模块具有三个卷积层，使用的激活函数为 Leaky ReLU。每个数据流网络的残差块用于抽取特征并且将抽取出的结果输入给生成器中对应的特征转换层。通过不同粒度的残差块，双数据流网络能够学习到多感受野的语义特征与风格特征。

## 生成器

为了精细地刻画不同感受野的特征信息，生成器的结构与双数据流网络恰好完全相反。与此同时，为了保证图像的多样性，生成器还接受一个从高维高斯分布中采样出的随机噪声作为最底层的输入，随后在生成器的每一层接受对应层的语义数据流网络与风格数据流网络的输入。

风格的迁移由上方图片的 FADE 残差块与 FAdaIN 模块共同完成。在 FADE 残差块中，作者使用了与语义残差块和风格残差块相反的结构，并且将残差块中的 batch normalization 替代为了 FADE 模块。FADE 模块通过使用由调节参数参数  $\gamma$  和  $\beta$  定义的可学习仿射变换来逐元素去归一化。具体在 FADE 模块内部的 FAdaIN 模块能够通过 feature adaptive instance normalization 抽取风格特征。

生成器主体经过了从微观到全局的生成过程。具体而言，多感受野的语义特征与风格特征在相应生成器层不断生成的图像。通过这种方式，语义结构与风格信息能够被网络充分学习并且有效的整合进入端到端的训练过程。

## 鉴别器

作者进一步优化了前文提及的多感受野 patch discriminator。三个标准且全同的鉴别器被用于鉴别不同感受野的生成图像。虽然鉴别器之间结构相同，然而 patch discriminator 能够在全局感受野感知全局信息，同时最微观层面的 patch discriminator 能够让生成器产生更好的语义细节。

多感受野的鉴别器能够进一步提升 TIST 框架的鲁棒性，同时鉴别器还起到了为生成器提取特征以优化 feature matching loss 的功能。

### 2.4.3 特征迁移

作者提出了新的特征迁移方法，能够同时兼顾语义信息与风格信息，并且将二者有效融合。此处以  $x^c$  表示语义图像而  $x^s$  表示风格图像。 $CS, SS, G, D$  分别代表语义信息流网络，风格信息流网络，生成器，鉴别器。 $z_0 \in \mathbb{Z}$  是从高维高斯分布中采样出并输入给生成器的随机噪声。

$z_i \in \{z_0, z_1, z_2, \dots, z_k\}$  是生成器中经过了第  $i$  个残差块后的特征， $k$  表生成器中残差块的总个数，也即生成器中上采样的次数。 $f_i^c \in \{f_0^c, f_1^c, f_2^c, \dots, f_k^c\}$  代表从语义信息流网络中抽取出的相应的语义特征， $f_i^s \in \{f_0^s, f_1^s, f_2^s, \dots, f_k^s\}$  代表从风格信息流网络中抽取出的相应的风格特征。

#### Feature adaptive denormalization (FADE)

作者从 GauGAN 的 SPADE 模块中获得启发，对 SPADE 模块进行修改，构建了 FADE 方法。作者将多感受野特征  $f_i^c$  推广到语义图像  $x^c$ 。通过这种方式，TSIT 成功利用了被 CS 捕捉的语义信息。

此处约定  $N$  为 batch size， $L_i$  为每一层的特征通道的数目。 $H_i$  与  $w_i$  为高度与宽度。首先对每个特征通道  $z_i$  进行 batch normalization。而后利用学习到的感受野参数  $\gamma_i$  和  $\beta_i$  将归一化后的特征去归一化：

The denormalized activation ( $n \in N, l \in L_i, h \in H_i, w \in W_i$ ) is :

$$\gamma_i^{l,h,w} \cdot \frac{z_i^{n,l,h,w} - \mu_i^l}{\sigma_i^l} + \beta_i^{l,h,w},$$

where  $\mu_i^l$  and  $\sigma_i^l$  are the mean and standard deviation, respectively, of the generator feature map  $z_i$  before the batch normalization in channel  $l$  :

$$\mu_i^l = \frac{1}{NH_iW_i} \sum_{n,h,w} z_i^{n,l,h,w},$$

$$\sigma_i^l = \sqrt{\frac{1}{NH_iW_i} \sum_{n,h,w} (z_i^{n,l,h,w})^2 - (\mu_i^l)^2}.$$

去归一化操作是逐元素进行的，同时通过对 FADE 模块中的  $f_i^c$  进行单层卷积能够学习到参数  $\gamma_i^{l,h,w}$  与  $\beta_i^{l,h,w}$ 。相较于 SPADE 等条件归一化方法，FADE 能够对特征进行从微观到全局的进一步感知，能够更加优秀地重建语义结构信息。

#### Feature adaptive instance normalization (FAdaIN)

为了将风格信息更好地与语义信息整合，作者受到 AdaIN 方法的启发，提出了新的特征迁移方式，也即 FAdaIN 方法。具体而言，以  $z_i \in \{z_0, z_1, z_2, \dots, z_k\}$  表示生成器中经过了第  $i$  个残差块后的特征。FAdaIN 计算  $z_i$  与  $f_i^s$  之间的仿射参数，而后计算：

$$\text{FAdaIN}(z_i, f_i^s) = \sigma(f_i^s) \left( \frac{z_i - \mu(z_i)}{\sigma(z_i)} \right) + \mu(f_i^s)$$

where  $\mu(z_i)$  and  $\sigma(z_i)$  are the mean and standard deviation, respectively, of  $z_i$ .

通过 FAdaIN 方法，不同层内从微观到全局的风格特征都能够与 FADE 学习到的语义特征有效整合，允许 TSIT 框架能够被端到端训练，并且能够被应用到广泛的图像生成相关领域。

## 目标函数

TSIT 仅仅使用了基础的损失函数，这是框架简洁的一大原因。TIST 中使用了铰链损失作为 adversarial loss，对于生成器，作者使用了 perceptual loss 与 feature matching loss。对于多感受野的鉴别器，仅仅采用了基于铰链的 adversarial loss 来区分图像的真假。和 GAN 一样，生成器与鉴别器同样模拟着极大极小对抗游戏。具体生成器和鉴别器的损失如下：

$$\begin{aligned}\mathcal{L}_G &= -\mathbb{E}[D(g)] + \lambda_P \mathcal{L}_P(g, x^c) + \lambda_{FM} \mathcal{L}_{FM}(g, x^s), \\ \mathcal{L}_D &= -\mathbb{E}[\min(-1 + D(x^s), 0)] - \mathbb{E}[\min(-1 - D(g), 0)],\end{aligned}$$

其中  $g = G(z_0, x^c, x^s)$  代表生成的图像， $z_0, x^c, x^s$  分别代表输入给生成器的随机噪声、内容图像与风格图像。 $\mathcal{L}_P$  代表基于 Jittor 预训练的 VGG-19 网络的 perceptual loss。 $\mathcal{L}_{FM}$  代表 feature matching loss。 $\lambda_P$  and  $\lambda_{FM}$  是对应的参数。

简单明了的目标函数使得 TSIT 框架具有高度适应性且易于训练。尽管训练简单，但是 TSodeIT 在相应任务上取得了惊人的效果。

### 3 实现细节 Details

代码实现、预训练模型、实验结果详见 [\[Code Base\]](#)。

#### 3.1 数据集

在赛方提供数据集的基础上，我们构造了三个子集

1. **Total**：原始数据集。
2. **Selection I**：将明显具有分割错误、不同标签间深度耦合等等难以被学习的特征的图像进行了剔除，剩余 8,116 张图像。
3. **Selection II**：在 **Selection I** 的基础上，剔除了逆光严重的图像，同时对于输入相似度极高而输出相似度较低的图像，按输入相似度进行等价类划分，每个划分仅保留少数几张图片。

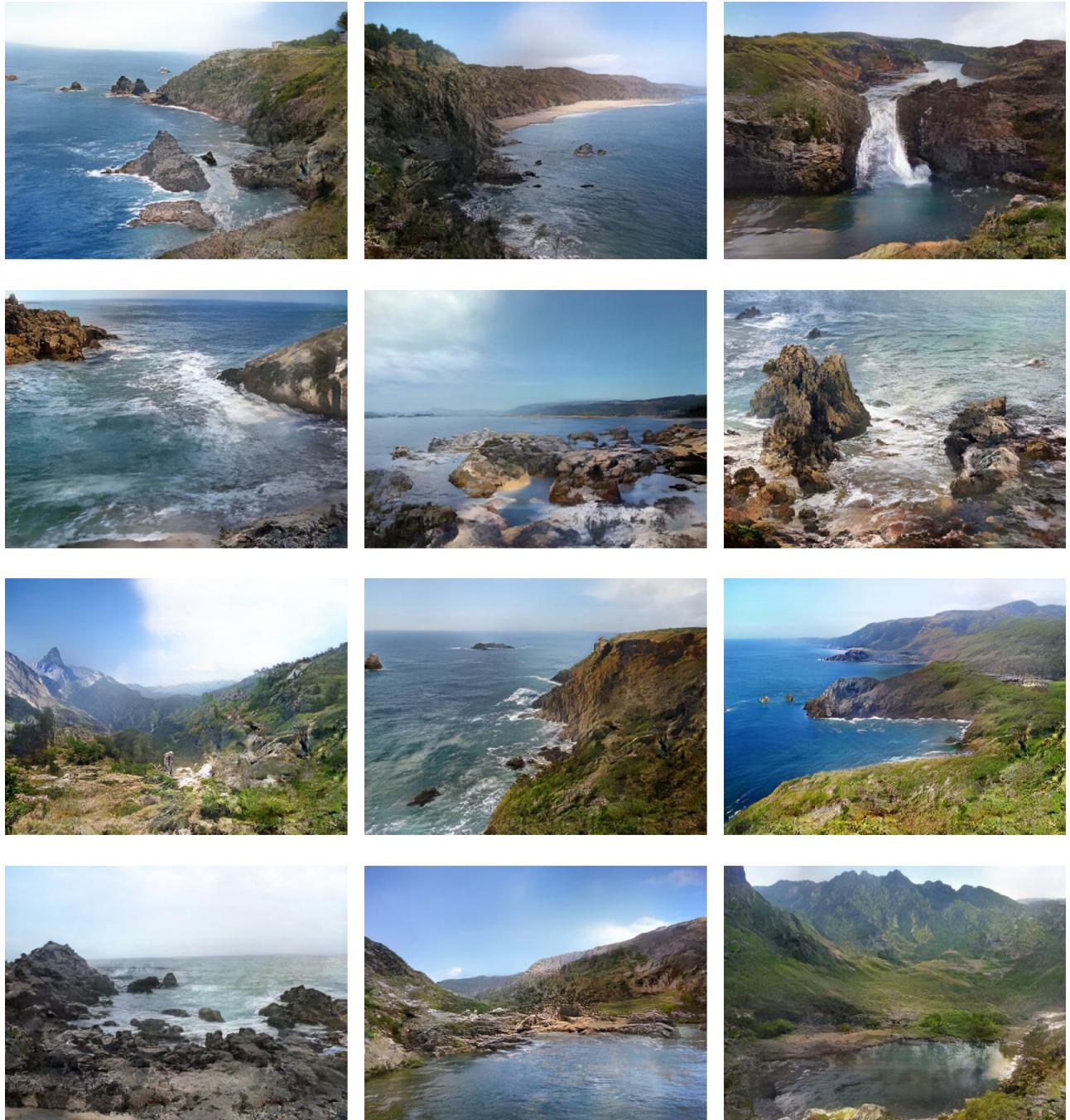
在实验时，我们对三种不同规模的数据集都进行了测试。

#### 3.2 模型选用与参数

我们实验的主体模型中，生成器采用 TSIT 论文中提到的“语义数据流网络”，而鉴别器由两部分构成。鉴别器的其中一部分是常见的带有降采样和特征混合的深度卷积神经网络，另一部分我们采用在 ImageNet 上预训练过的 VGG-19 来计算生成图像和真实图像在特征-空间中的余弦相似度。

对于我们最终提交的模型，我们使用 NVIDIA A100-SXM4-40GB 来进行训练，选择损失函数的权重为 GAN Loss : VGG Loss = 20 : 10，**main** 模型的训练过程如下

Phase	Epoch	batch_size	training set	learning rate
I	[1, 38]	2	Total	0.0002
II	(38, 71]	30	Selection I	0.0012
III	(71, 95]	5	Selection II	0.0004
IV	(95, 110]	5	Selection I	0.0002



我们的更多实验结果

### 3.3 其他思考

除了我们的最终提交版本对应的最佳结果之外，我们还进行过如下实验：

- 测试 SPADE [14] 框架的效果
- 尝试在 Generator 或 Discriminator 中引入 Transformer [19]
  - 但是由于算力的限制实验在编程完成后未能成功运作
- 尝试利用 CycleGAN [18] 的思想

- 使用在提供数据集上预训练的减量 SegFormer 模型当做语义分割器，作为 Discriminator 的一部分
- 使用基于真实图片和生成图片的对比学习尝试训练美学评分模型
- 将赛道评分公式作为 Loss 来优化
- 最后预训练后的 SegFormer [33] 语义分割器分割正确率还没有提交结果的 Mask Accuracy 高 :(

## 4 总结 Conclusion

我们对图像生成任务在近年来的发展历程和相关损失函数设计、模型架构等等进行了充分的调研，并选择 TSIT 模型在 Jittor 深度学习框架上进行了复现。我们编写了完善的、具有扩展性的实验框架，撰写了本篇解题报告与调研报告，进行参数分析与调整，并在最终赛道排行榜上取得了天梯排名前 20 名的成绩。

在报告的最后，要感谢课程组提供的这次宝贵的上手练习深度学习框架，调研学习计算机视觉领域前沿任务与研究成果的机会。也感谢赛事组提供的这次宝贵的比赛机会，让赛课结合得到了又一次的生动实践。最后，祝计图开源社区内容日益完整，日益丰富，在世界上推广出我们的影响力。

## 5 参考文献 Reference

- [1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Proc. Advances Neural Information Processing Systems Conf., 2014, pp. 2672–2680.
- [2] Kramer M A. Nonlinear principal component analysis using autoassociative neural networks[J]. AIChE journal, 1991, 37(2): 233-243.
- [3] Kingma D P, Welling M. Auto-encoding variational bayes[J]. arXiv preprint arXiv:1312.6114, 2013.
- [4] Johnson J, Alahi A, Fei-Fei L. Perceptual losses for real-time style transfer and super-resolution[C]//European conference on computer vision. Springer, Cham, 2016: 694-711.
- [5] Wu Y, Burda Y, Salakhutdinov R, et al. On the quantitative analysis of decoder-based generative models[J]. arXiv preprint arXiv:1611.04273, 2016.
- [6] Gulrajani I, Ahmed F, Arjovsky M, et al. Improved training of wasserstein gans[J]. Advances in neural information processing systems, 2017, 30.
- [7] Mao X, Li Q, Xie H, et al. Least squares generative adversarial networks[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2794-2802.

- [8] Creswell A, White T, Dumoulin V, et al. Generative adversarial networks: An overview[J]. IEEE signal processing magazine, 2018, 35(1): 53-65.
- [9] Heusel M, Ramsauer H, Unterthiner T, et al. Gans trained by a two time-scale update rule converge to a local nash equilibrium[J]. Advances in neural information processing systems, 2017, 30.
- [10] 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. 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.
- [11] Talebi H, Milanfar P. NIMA: Neural image assessment[J]. IEEE transactions on image processing, 2018, 27(8): 3998-4011.
- [12] Zhang R, Isola P, Efros A A, et al. The unreasonable effectiveness of deep features as a perceptual metric[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 586-595.
- [13] Lee J T, Kim C S. Image aesthetic assessment based on pairwise comparison a unified approach to score regression, binary classification, and personalization[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 1191-1200.
- [14] Park T, Liu M Y, Wang T C, et al. Semantic image synthesis with spatially-adaptive normalization[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 2337-2346.
- [15] Sun K, Xiao B, Liu D, et al. Deep high-resolution representation learning for human pose estimation[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 5693-5703.
- [16] Jiang L, Zhang C, Huang M, et al. Tsit: A simple and versatile framework for image-to-image translation[C]//European Conference on Computer Vision. Springer, Cham, 2020: 206-222.
- [17] Yang Y, Soatto S. Fda: Fourier domain adaptation for semantic segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 4085-4095.
- [18] Zhu J Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2223-2232.
- [19] Esser P, Rombach R, Ommer B. Taming transformers for high-resolution image synthesis[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 12873-12883.

- [20] Liu M Y, Huang X, Yu J, et al. Generative adversarial networks for image and video synthesis: Algorithms and applications[J]. Proceedings of the IEEE, 2021, 109(5): 839-862.
- [21] She D, Lai Y K, Yi G, et al. Hierarchical layout-aware graph convolutional network for unified aesthetics assessment[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 8475-8484.
- [22] Sushko V, Schönfeld E, Zhang D, et al. You only need adversarial supervision for semantic image synthesis[J]. arXiv preprint arXiv:2012.04781, 2020.
- [23] Wang Y, Qi L, Chen Y C, et al. Image synthesis via semantic composition[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 13749-13758.
- [24] Yu N, Liu G, Dundar A, et al. Dual contrastive loss and attention for gans[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 6731-6742.
- [25] Zhou X, Zhang B, Zhang T, et al. Cocosnet v2: Full-resolution correspondence learning for image translation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 11465-11475.
- [26] Richter S R, Al Hajja H A, Koltun V. Enhancing photorealism enhancement[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.
- [27] Zhang Y, Li D, Law K L, et al. IDR: Self-Supervised Image Denoising via Iterative Data Refinement[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 2098-2107.
- [28] Cordts M, Omran M, Ramos S, et al. The cityscapes dataset for semantic urban scene understanding[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 3213-3223.
- [29] Zhou B, Zhao H, Puig X, et al. Scene parsing through ade20k dataset[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 633-641.
- [30] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]//International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015: 234-241.
- [31] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
- [32] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro. Video-to-video synthesis. In Advances in Neural Information Processing Systems, 2018.
- [33] Xie E, Wang W, Yu Z, et al. SegFormer: Simple and efficient design for semantic segmentation with transformers[J]. Advances in Neural Information Processing Systems, 2021, 34: 12077-12090.