**基于CycleGAN的图像风格迁移**

**小组成员：陈伦豪、江琳、戴珊珊**

1. [**绪论**](#_Toc51230631)

图像风格迁移是一种计算机视觉任务，旨在将一幅图像的内容与另一幅图像的风格进行合成，生成一幅新的图像，既保留了原始图像的内容信息，又呈现了目标图像的风格特征。这项技术的出现为图像编辑和艺术创作提供了有趣的可能性。

在传统的图像编辑方法中，修改图像的风格往往需要手动操作和艺术设计技巧。而图像风格迁移技术利用深度学习的方法，通过训练一个神经网络模型来自动学习图像的内容和风格表示，从而实现图像的自动合成。

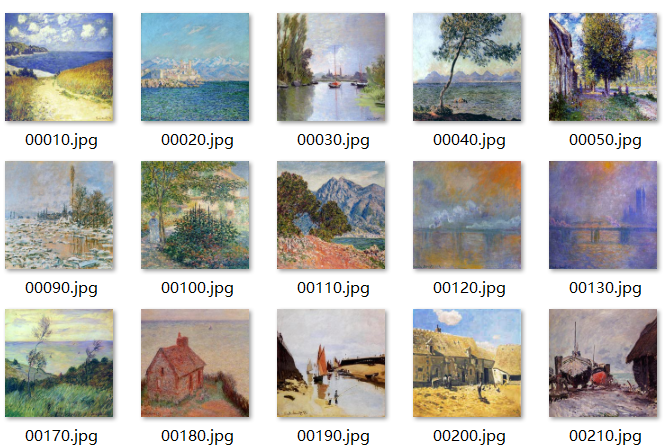
而本实验旨在使用CycleGAN（循环一致性生成对抗网络）来实现图像风格迁移。CycleGAN是一种无监督的图像转换模型，能够将图像从一个领域转换到另一个领域，而无需配对的训练数据。

1. **实验内容**

**2.1 数据集介绍以及处理**

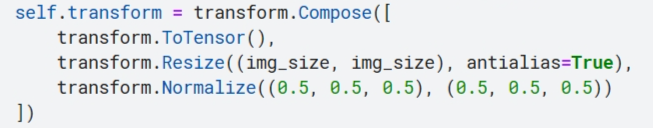
数据集使用CycleGAN论文提供的Monet2Photo数据集，它由1193张莫奈绘画和7038张自然照片组成，其中，使用1072张莫奈图片和3287张自然照片当作训练集。每张图片的规格为256×256。

Monet：



在数据预处理阶段，我们将数据传入CreateDatasets 类中，并调用transform.Compe函数用于图像转换。转换包括将图像转换为张量，将图像大小调整为指定的维度，以防止他训练出错，并使用每个通道的平均值和标准偏差值（0.5，0.5，0.5）对图像的像素值进行归一化。

相关代码如下：

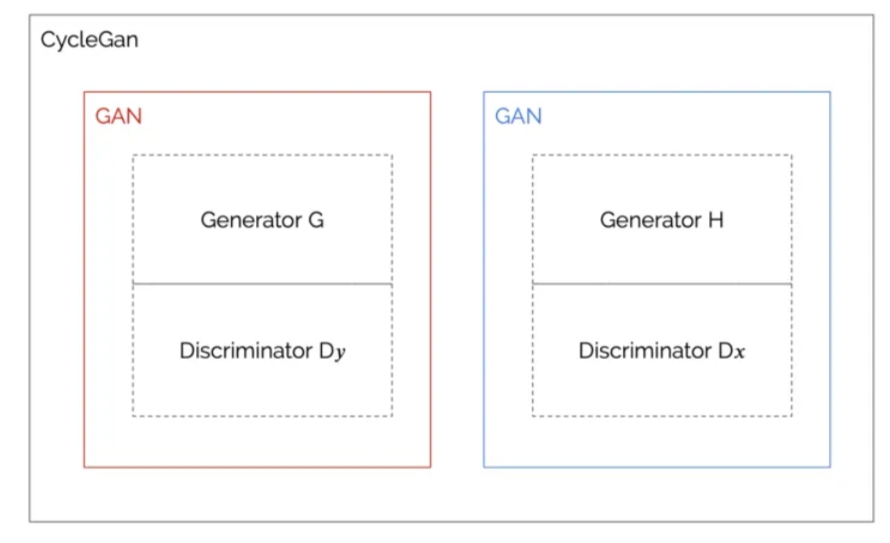


**2.2 模型原理**

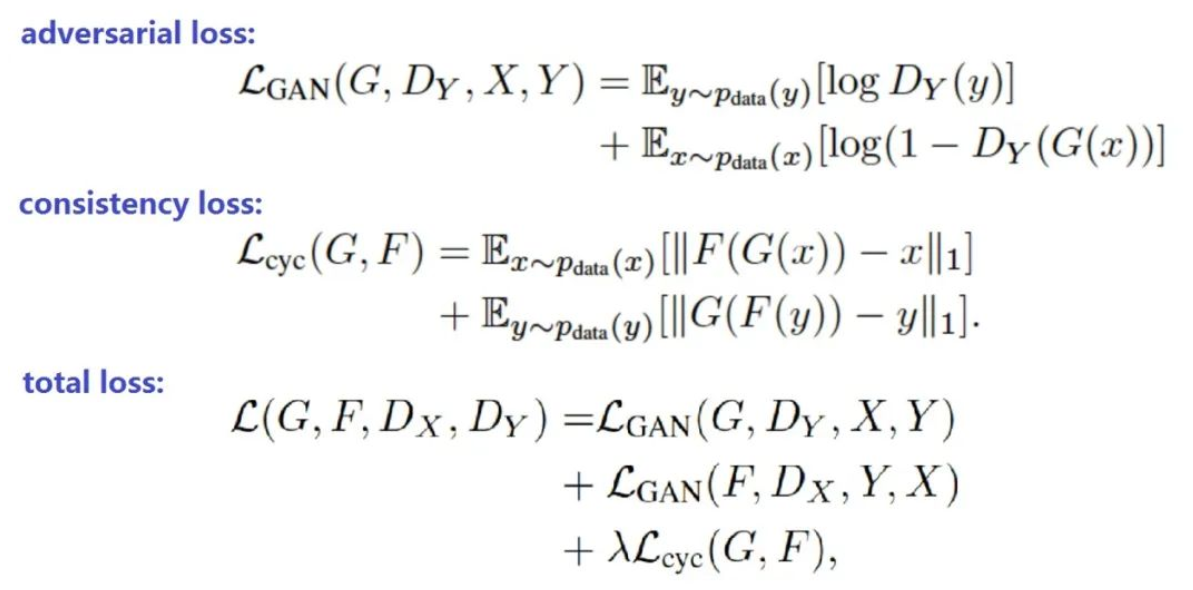
CycleGAN神经网络是一种用于图像转换的神经网络模型，它可以在两个不同的领域之间进行无监督的图像转换。CycleGAN的目标是学习两个领域之间的映射，而无需配对的训练数据。它由两个生成器（Generator）和两个判别器（Discriminator）组成。

生成器网络的目标是将输入图像映射到目标领域中的图像。它接受输入图像，并通过一系列的卷积层、批量归一化层（Batch Normalization）和激活函数（如ReLU）来处理输入，以逐渐生成与目标领域图像相似的输出图像。生成器网络通常包含编码器和解码器结构，编码器用于提取输入图像的特征，解码器用于生成输出图像。

判别器网络的目标是通过对输入图像进行分类，判断输入图像是真实的目标图像还是生成器生成的图像。判别器接受输入图像，并通过一系列的卷积层、批量归一化层和激活函数处理输入，最后输出一个概率值来表示输入图像是真实的目标图像的概率。



而在CycleGAN网络中，最核心的部分当属他的loss函数，CycleGAN的loss由对抗损失（称为gan loss或adversarial loss）和循环一致性损失（consitency loss）组成。前者的作用是为了让生成的图像更能称之为图像，也就是生成的图像更真实。但它不保证能生成到我们想要的图像。而后者则通常称为L1loss，因为其计算方式通常是用l1范数计算的。当我们把目标图像X送入到生成器G中后，得到的是假的风格图像y，再把这张假的y图送入到另一个生成器F中，得到更假的x图。理想情况下，此时的更假的x图应该与原始的x图相差无几。这样也便构成了一个循环，因此叫做循环一致性损失。



除此之外，在CycleGAN论文中还有一个loss没有被提及，那就是Identity loss，这是一个很重要的loss，主要用于防止生成器自主地修改图像的色调，使得整体的颜色产生变化。



通常来说，对抗损失采用LSGAN的方式，所以是MSE Loss；循环一致损失按照论文采用L1 Loss；identity损失同样采用L1 Loss。

**2.3 实验设计**

CycleGAN有两个结构一样的判别器和两个结构一样的生成器因此只需要定义一个判别器和一个生成器，后面train过程使用时实例化成不同对象就可以了。

生成器（Generator）部分包括以下组件：

① 三层卷积层，用于将输入图像进行编码和提取特征。

② 六个残差块（Resnet Block），用于增加网络的深度和提升图像转换的质量。

③ 上采样层，通过反卷积操作将特征图的尺寸恢复到原始输入图像的尺寸。

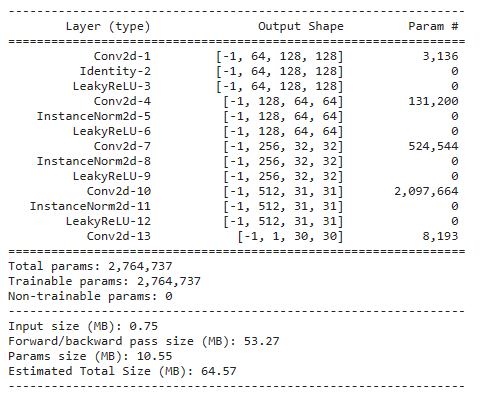
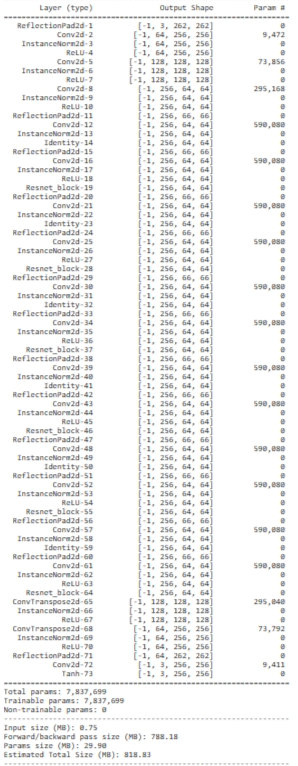
④ 最后一层卷积层，用于生成最终的输出图像。

判别器（Discriminator）部分包括以下组件：

① 四层卷积层，用于对输入图像进行分类和判别。

② 最后一层卷积层，输出一个概率值，表示输入图像是真实的目标图像的概率。

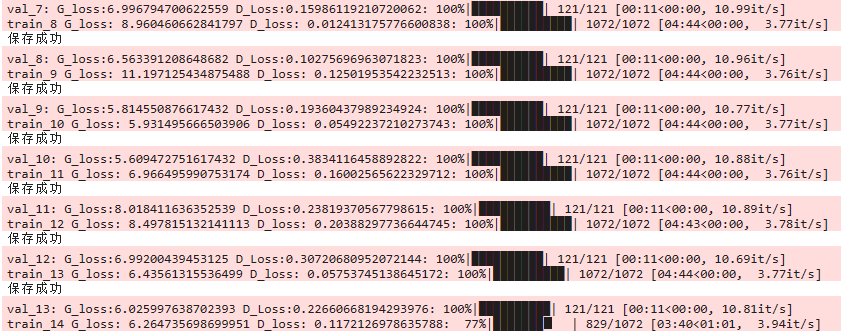
使用torchsummary打印出模型视图如下：

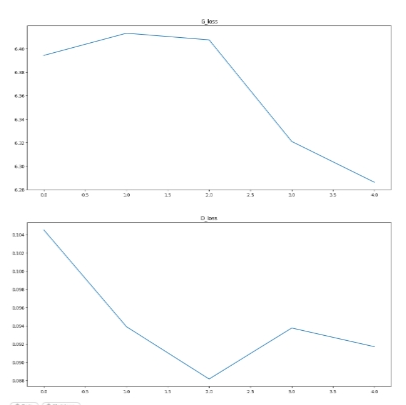


模型的训练则是通过kaggle平台进行训练，相对应的参数设置如下所示：



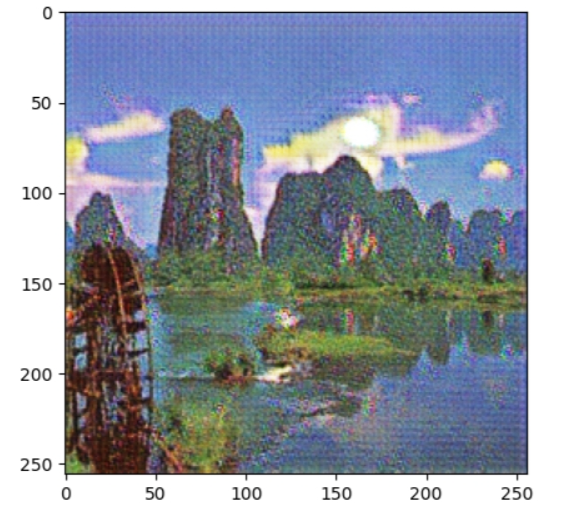
在模型训练过程中，为了防止网络或平台问题，导致训练中断，因此将，每次epoch产生的模型权重参数保存下来，以备后续重新加载训练。另外CycleGAN模型由于有两个生成器两个判别器，因此需要对这个网络同时进行训练，因此这里将训练的函数进行封装，并返回总的loss损失。模型的训练过程以及对应loss变化图如下：





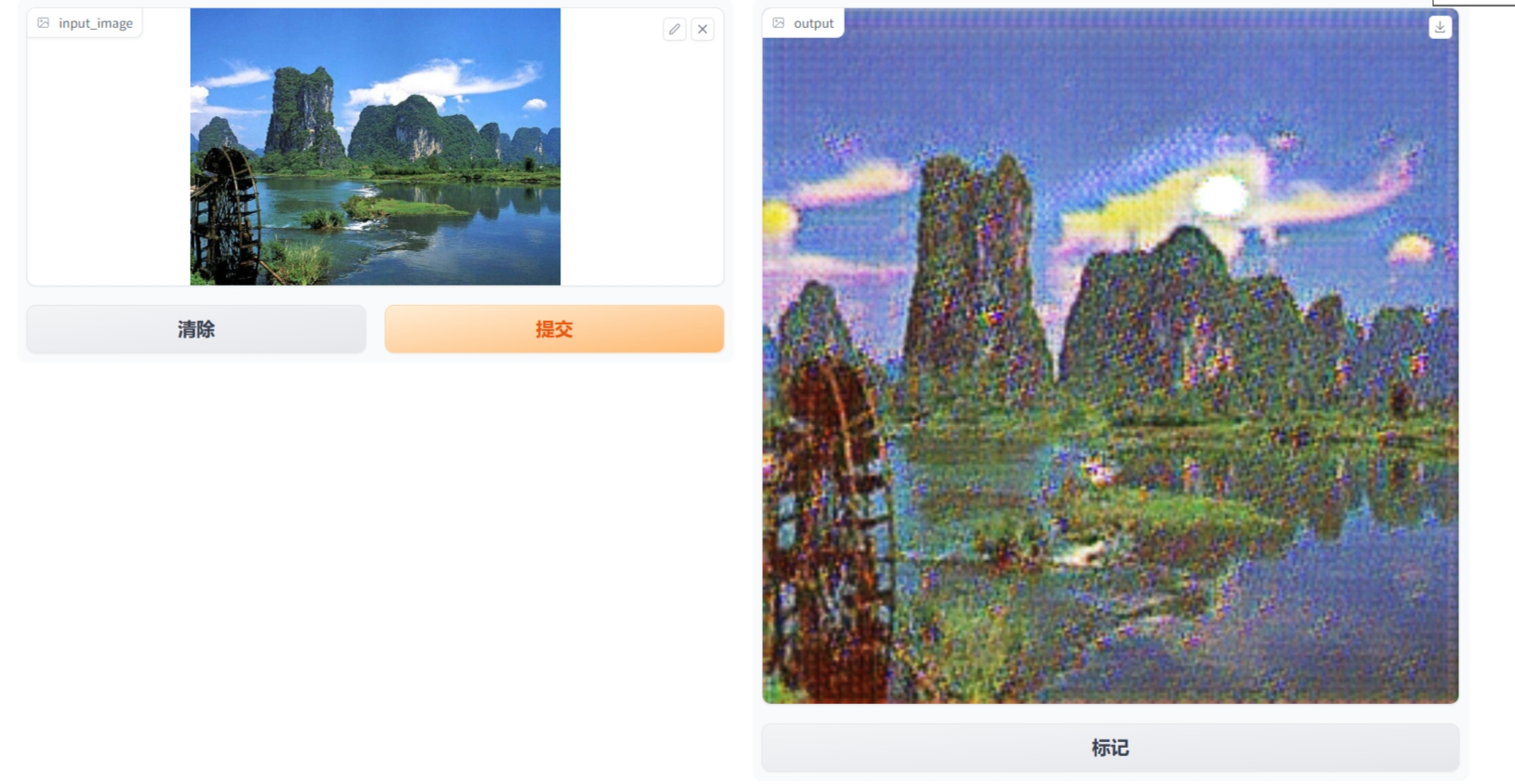
**2.4 结果测试并部署**

在经过30次迭代训练之后，模型已经初步具备风格迁移的能力。加载保存下来的模型权重文件，并对图片123进行测试，这里并没有示原图。而是将生成后的图像展示出来，并计算它们之间的内容损失，如下图所示：





Gradio是一个开源的Python库，用于构建机器学习和数据科学演示应用。通过使用gradio开源库，可以将训练好的模型部署到web上，供用户使用，其对应代码以及示例图如下：



核心代码：

inputs = gr.inputs.Image()

outputs = gr.outputs.Image(type='numpy')

interface = gr.Interface(fn=generate\_image, inputs=inputs, outputs=outputs)

1. **实验总结**

本实验展示了使用CycleGAN实现图像风格迁移的能力，展示了该模型在不同风格之间转换图像的能力。通过训练生成器和判别器，CycleGAN能够学习两个风格之间的映射关系，并将图像从一个风格转换为另一个风格。然而，该方法仍存在一些局限性，就按照上面测试结果显示，生成的图片还是具备光影效果，并没有完美的生成目标风格图像。此外，对于没有足够训练样本的风格，模型的转换效果可能会受到限制。