**自适应实例归一化实时任意风格传输**

黄勋 塞尔吉·贝隆吉

康奈尔大学计算机科学与康奈尔技术系

*{*xh258,sjb344*}*@cornell.edu

# 摘 要

### Gatys等最近提出了一种神经算法，可以以另一幅图像的风格渲染内容图像，从而实现所谓的风格迁移。但是，他们的框架要求缓慢的迭代优化过程，这限制了其实际应用。后面提出使用前馈神经网络进行快速近似，以加快神经风格转换。不幸的是， 提高速度是有代价的：网络通常绑定到一组固定的风格，并且无法适应任意新风格。在本文中，我们提出了一种简单而有效的方法，该方法首次实现了实时的任意风格迁移。我们方法的核心是新颖的自适应实例规范化（AdaIN）层，该层将内容特征的均值和方差与风格特征的均值和方差对齐。我们的方法实现的速度可与现有最快方法相媲美，而不受限于一组预定义的风格。此外，我们的方法允许灵活的用户控制，例如内容和风格的权衡（注：就是谁的影响强），风格插值，颜色和空间控制，所有这些都使用单个前馈神经网络。

1. **介绍**

Gatys等人的开创性工作[16]表示，深度神经网络

（DNN）不仅编码图像的内容，还编码图像的风格信息。 此外，图像风格和内容在某种程度上是可分离的：可以在保留图像内容的同时更改其风格。[[16]足够灵活，可以组合任意图像的内容和风格。但是，它依赖于过慢的化过程。已经致力于加速神经风格的转移。[24, 51, 31]尝试训练前馈神经网络，这些神经网络通过单个前向传我们的方法可以实时转换任意新风格[16]，并且速度媲美最快前馈方法的速度[24, 52].我们的方法受到实例规范化（IN）的启发[52, 11]层，这在前馈风格迁移中非常有效。为了说明实例规范化的成功，我们提出了一种新的解释，即实例规范化通过对特征统计量进行归一化来执行风格归一化，发现这些特征统计量可以承载图像的风格信息[16, 30, 33].基于我们的解释，我们引入了对IN的简单扩展，即自适应实例规范化（AdaIN）。给定内容输入 和风格输入，AdaIN只需调整内容输入的均值和方差以 匹配风格输入的均值和方差即可。通过实验，我们发现AdaIN通过传递特征统计信息有效地结合了前者和后者的内容。然后，通过将AdaIN输出反相回到图像空间， 学习解码器网络以生成最终的风格化图像。我们的方法比[16]快，而不会牺牲将输入转换为任意新风格的灵活性。此外，我们的方法在运行时提供了丰富的用户控制，而无需对训练过程进行任何修改。

# 相关工作

**风格**转移。风格迁移的问题源自非真实感渲染[28]， 并且与纹理合成和转移[13, 12, 14]紧密相关.一些早期的方法包括对线性滤波器响应进行直方图匹配[19]和非参数采样[12, [15](#_bookmark22)].这些方法通常依赖于低级统计信息，并且通常无法捕获语义结构。Gatys等[16]首次通过在DNN的卷积层中匹配特征统计信息展示了令人印象深刻的风格迁移结果。最近，若干对[16]的改进被提出提出 [30] 在深度特征空间中引入了基于马尔可夫随机场（MRF） 的框架，以实施局部模式。Gatys[[17](#_bookmark23)]提出了控制颜色保留，空间位置和风格迁移规模的方法。Ruder等。[45]通过施加时间限制来实现视频风格迁移质量。

比单风格传输方法慢得多[6].

arXiv：1703.06868v2 [cs.CV] 2017年

在这项工作中，我们提出了第一个神经风格迁移算 法，该算法解决了这种基本的灵活性-速度困境。

1

Gatys等的框架。[16]基于慢速优化过程，该过程 会迭代更新图像以最大程度地减少内容损失和损失网 络计算出的风格损失。即使使用现代GPU，收敛也可能 需要几分钟。因此，移动应用程序中的设备上处理太 慢而无法实用。常见的解决方法是用经过训练的前馈 神经网络替换优化过程，以最小化同一目标[24, 51, 31].这些前馈风格传递方法比基于优化的替代方法快 大约三个数量级，这为实时应用打开了大门。Wang等。

[[53](#_bookmark32)]通过多分辨率体系结构提高了前馈风格传输的粒度。Ulyanov等[52]提出了提高生成样本质量和多样

**深度生成图像建模**

有几个图像生成的替代框架，包括可变自动编码器[27]，自回归模型[[40](#_bookmark28)]和生成对抗网络（GAN）[18].值得注意的是，GAN的视觉质量令人印象深刻。对GAN框架进行了各种改进也被提出，比如条件生成，多阶段生成，以及更好的训练目标。GAN也被用在了风格迁移和跨领域生成。

跨域图像生成[50, [3,](#_bookmark19) 23, 38, 37, [25](#_bookmark26)].

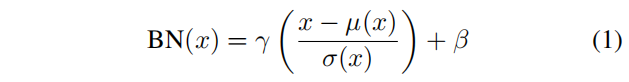
# 背景

## 批量标准化

艾菲和塞格迪的开创性工作[[22](#_bookmark25)]引入了批量标准化

（BN）层，该层通过标准化特征统计信息大大简化了 前馈网络的训练。BN层最初旨在加速判别网络的训练， 但也已发现在生成图像建模中有效[[42](#_bookmark29)].给定输入批次 x RN *×*C*×*H*×*W，BN标准化每个特征通道的平均值和标准

偏差：



性的方法。但是，在每个网络都绑定到固定风格的意 义上，上述前馈方法受到限制。为了解决这个问题， Dumoulin等[11]引入了一个单一网络，该网络能够

编码32种风格及其插值。与我们同期的工作，Li等。[32]提出了前馈其中γ，βRC∈是从数据中获悉的仿射参数；

*µ*（x），σ∈（x）RC是均值和标准差，是针对每个特征

通道独立于批处理大小和空间尺寸计算得出的：

可以合成多达300种纹理和

16种风格。尽管如此，以上两种方法仍无法适应训练

期间未观察到的任意风格。

最近，Chen和Schmidt [6]引入了一种前馈方法，该方法可以传递任意风格，这要归功于

风格交换层。给定内容的特征激活值

和风格图片，风格交换层会以逐个补丁的方式将内容

特征替换为最匹配的风格特征。尽管如此，它们的风格交换层仍创建了新的计算瓶颈：95％以上的计算都

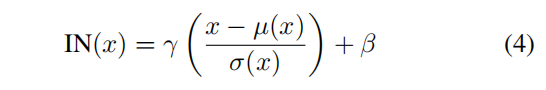
花在512\*512个输入图像的风格交换上。我们的方法还允许任意风格迁移，而速度比[6]块1-2个数量级.

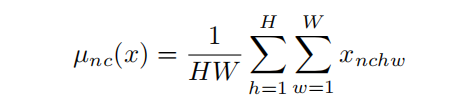
风格迁移的另一个主要问题是要使用哪种风格损失函数。Gatys等人的原始框架。[16]通过匹配由Gram矩阵捕获的特征激活之间的二阶统计信息来匹配风格。已经提出了其他有效的损失函数，例如MRF损失[30]， 对抗损失[31]，直方图损失[54]，珊瑚损失[41]，MMD 损失[33]，以及通道平均值和方差[33].注意，以上所有损失函数旨在匹配风格图像和合成图像之间的某些特征统计量。

BN在训练过程中使用小批量统计，并在推理过程中将

其替换为流行的统计信息，从而在训练和推理之间引入差异。批量重新规范化[[21](#_bookmark24)最近有人提议通过在训练过程中逐渐使用流行的统计数据来解决这个问题。Li 等人作为BN的另一个有趣应用。[[34](#_bookmark27)]发现BN可以通过重新计算目标域中的流行统计信息来减轻域移位。最近，已经提出了几种替代的标准化方案，以将BN的有效性扩展到循环架构[35, [2,](#_bookmark18) 47, 8, 29, [44](#_bookmark30)].

## 实例规范化

在原始的前馈风格化方法[51]，风格传输网络在每 个卷积层之后都包含一个BN层。令人惊讶地，Ulyanov 等。[52]发现，仅用IN层替换BN层即可实现显着改善：

与BN层不同的是，此处的µ（x）和σ（x）是针对 每个通道和每个样本跨空间维度独立计算的：

建议的，例如条件生成[43, 23]，多

阶段处理[[9,](#_bookmark21) 20]，以及更好的训练目标[46, [1](#_bookmark17)].GAN也已应用于风格迁移[31]和

10 10 10

规范

实例

rm

批号

批号 rm

实例 规范

规范

实例

rm

批号

8 8 8

6 6 6

4 4 4

风格损失

风格损失

风格损失

2

0

0 1000 2000 3000 4000 5000

迭代

* + 1. 训练有原始图像。

2

0

0 1000 2000 3000 4000 5000

迭代

* + 1. 用对比度归一化图像训练。

2

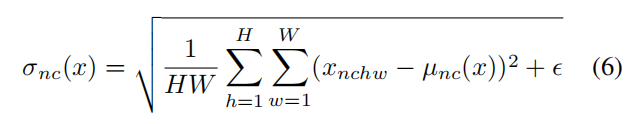
0

0 1000 2000 3000 4000 5000

迭代

* + 1. 训练过风格归一化图像。

图1.为了了解IN在风格迁移中有效的原因，我们使用（a）MS-COCO中的原始图像训练了一个IN模型和一个BN模型[36]，（b） 对比度归一化图像和（c）使用预先训练的风格迁移网络[风格]归一化图像[24].即使将所有训练图像均归一化为相同的对比度， IN带来的改进仍然显着，但是当（近似）所有图像均归一化为相同风格时，IN所带来的改进仍然很小。我们的结果表明IN执行 一种风格规范化。

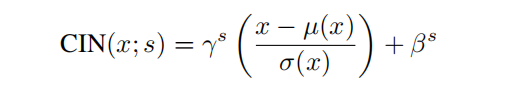


IN的成功在于其对内容图像对比度的不变性。但是，IN 发生在特征空间中，

另一个区别是IN层在测试时间不变的情况下应用， 而BN层通常用总体统计量代替小批量统计量。

## 条件实例规范化

Dumoulin等人[11]提出了一个条件实例规范化（CIN） 层，该层为每种风格s学习一组不同的参数γs和βs， ，而不是学习一组仿射参数γ和β，而是研究了它们。



训练期间，风格图像及其索引从固定的风格s集合s ∈

{1, 2, ..., S}中随机选择的

（他们的实验中S = 32）。然后，通过风格传输网络处理内容图像，其中在CIN层中使用

了相应的γs和βs。出乎意料的是，网络可以使用IN 层中相同的卷积参数但使用不同的仿射参数来生成完全不同风格的图像。

与没有归一化层的网络相比，具有CIN层的网络需 要2FS附加参数，其中F是网络中特征图的总数[11].由于附加参数的数量与风格的数量呈线性比例关系，因此扩展其方法以对大量风格（例如，数万种）建模具有挑战性。而且，他们的方法如果不重新训练网络就无法适应任意新风格。

# 解释实例规范化

尽管（有条件的）实例规范化取得了巨大的成功， 但是它们在风格迁移中表现特别出色的原因仍然难以 捉摸。Ulyanov [52]归因

它成功地实现了它对内容图像的对比度的不变性。然而，IN发生在特征空间中，因此它应该比在像素空间中进行简单的对比度规范化有更深远的影响。也许更令人惊讶的是，IN中的仿射参数可以完全改变输出图像的样式。

据知道，DNN的卷积特征统计可以捕捉图像[16,30,33]的风格。而盖蒂斯[16]等人则这样认为利用二阶统计量作为其优化目标。Li等人。[33]最近表明，匹配许多其他统计数据，包括通道平均均值和方差，对样式传输也是有效的。受这些观察结果的激励，**我们认为实例归一化通过规范化特征统计量来实现一种风格归一化**，即均值和方差。虽然DNN在[16,33]中作为图像描述符，但我们相信生成器网络的特征统计量也可以控制生成的图像的样式。

我们运行改进的纹理网络的代码[52]执行单样式传输，与IN或BN层。 正如预期的那样，具有IN的模型比BN模型收敛得更快(图1。 1(a))。 为了测试[52]中的解释，我们然后通过在亮度通道上执行直方图均衡化，将所有训练图像归一化为相同的对比度。 如图所示 1(b)IN仍然有效，表明[52]解释不完整。 为了验证我们的假设，我们使用[24]提供的预先训练的样式传输网络将所有训练图像归一化为相同的样式(不同于目标样式。 根据图。 1(c)，当图像样式已经标准化时，IN带来的改进就会小得多。 剩下的差距可以解释为，与[24]的风格正常化并不完美。 此外，在样式归一化图像上训练BN的模型可以像在原始图像上训练IN的模型一样快速收敛。 我们的结果表明，IN确实执行了一种样式规范化。因为B类规范会一批样本的特征统计量

而不是单个样本，**可以直观地理解为将一批样本标准化为以单个风格为中心。**但是，每个样本可能仍具有不同的风格。当我们想要将所有图像都转移到相同风格时，这是不希望的，就像原始前馈风格迁移算法[51].尽管卷积层可能会学习补偿批内风格差异，但它给训练带来了其他挑战。另一方面，IN可以将每个样本的风格归一化为目标风格。因为网络的其余部分可以专注于内容操作，同时丢弃原始风格信息，因此可以简化训练。CIN成功背后的原因也很清楚：不同的仿射参数可以将特征统计量标准化为不同的值，从而将输出图像标准化为不同的风格。

# 自适应实例规范化

如果IN将输入归一化为仿射参数指定的单个风格， 是否可以通过使用自适应仿射变换将其适应于任意给 定的风格？在这里，我们提出了对IN的简单扩展，我们称之为自适应实例规范化（AdaIN）。AdaIN接收内 容输入x和风格输入y，并简单地将x的通道平均和方差 对齐以匹配y。与BN，IN或CIN不同，AdaIN没有可学习 的仿射参数。而是从风格输入中自适应计算仿射参数：

AdaIN(*x, y*) = *σ*(*y* ( *x* − *µ*(*x*)) + *µ*(*y*) (8)

)

*σ*(*x*)

其中，我们只需使用σ（y）缩放输入的归一化内容， 然后使用µ（y）对其进行平移即可。与IN相似，这些统计信息是跨空间位置计算的。

**直观地，让我们考虑一个特征通道，该通道可以检测特定风格的笔触。具有这种笔触的风格图像将为此功能产生较高的平均激活度。AdaIN产生的输出将为此功能具有相同的高平均激活，同时保留内容图像的空间结构。可以使用前馈解码器将笔触功能反转到图像空间，类似于[10].此功能通道的变化可以编码更多细微的风格信息，这些信息也将传输到AdaIN输出和最终输出图像。**

简而言之，AdaIN通过传递特征统计信息（特别是通道平均值和方差）在特征空间中执行风格传递。尽管我们的AdaIN层扮演的角色与[6]类似.但风格交换操作非常耗时且占用内存，但我们的AdaIN层与IN层一样简单， 几乎不增加计算成本。

**L**𝑠

图2.我们的风格迁移算法的概述。我们使用固定VGG-19网络 的前几层来编码内容和风格图像。AdaIN层用于在特征空间 中执行风格迁移。学习解码器以将AdaIN输出反相到图像空 间。我们使用相同的VGG编码器来计算内容损失Lc（公式[12](#_bookmark9)） 和风格损失Ls（等式[13](#_bookmark10)).

# 实验装置

图[2](#_bookmark5) 显示了基于建议的AdaIN层的风格迁移网络的概述。代码和预训练模型（在Torch 7中[[7](#_bookmark20)]）可在以下位置获得：[https:](https://github.com/xunhuang1995/AdaIN-style)

[//github.com/xunhuang1995/AdaIN-style](https://github.com/xunhuang1995/AdaIN-style)

## 架构

我们的风格传送网络T将内容图像c和任意风格图像 s作为输入，并合成将前者和后者的内容重新组合的输出图像。我们采用简单的编码器-解码器体系结构，其中，编码器f固定到预训练的VGG-19 [48]的少量层中.在对特征空间中的内容和风格图像进行编码之后，我们将两个特征图都馈入AdaIN图层，该图层将内容特征图的均值和方差与风格特征图的均值和方差对齐，从而生成目标特征图t：

*t* = AdaIN(*f*(*c*)*, f*(*s*)) (9)

训练一个随机初始化的解码器g以将t映射回图像空 间，从而生成风格化图像T（c，s）：

*T* (*c, s*) = *g*(*t*) (10)

x解码器主要镜像编码器，所有池化层都替换为最近的上采样，以减少棋盘效应。我们在f和g中都使用了反射填充，以避免出现边界伪像。另一个重要的体系结构选择是解码器是否应使用实例，批处理或不使用规范化层。如第4节所述，IN将每个样本归一化为单一风格，而BN则将一批样本归一化为以单一风格为中心。当我们希望解码器生成截然不同的风格的图像时，两者都不可取。因此，我们不在解码器中使用归一化层。在[7.1](#_bookmark11) 我们将证明解码器中的IN / BN层确实会损害性能。

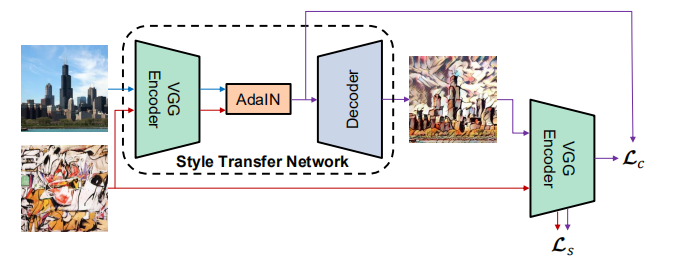


图2。对我们的风格转移算法的概述。我们使用固定的VGG-19网络的前几层对内容和样式图像进行编码。添加层用于在特征空间中执行样式转移。解码器将输出转换到图像空间。我们使用相同的VGG编码器来计算一个内容损失。12)和一个风格损失ls。13).

## 训练

我们使用MS-COCO训练我们的网络[36]作为内容图像和绘画数据集，这些图像大部分是从WikiArt [39按照[]的设置作为风格图像6].每个数据集包含大约80， 000个训练示例。我们使用亚当优化器[26]和8个内容 风格图像对的批处理大小。在训练期间，我们首先调整大小

6

5

4

ln(L*s*)

3

2

1

00 100 200 300 400 500

迭代

* + 1. 风格损失

3.0

2.8

Gatys等。

Ulyanov等。

我们的

内容图片

2.6

ln(L*c*)

2.4

2.2

2.00 100 200 300 400 500

迭代

* + 1. 内容损失

保留长宽比时，两个图像的最小尺寸为512，然后随机裁剪大小为256× 256.由于我们的网络是完全卷积的，因此可以在测试过程中将其应用于任何大小的图像。

相似 [51, 11, 52]，我们使用预先训练的VGG-19 [48]计算损失函数来训练解码器：

l = lc + λls (11)

它是内容损失c和风格损失s与风格损失权重λ的L加权组合。内容损失L是目标特征与输出图像的特征之间的欧式距离。我们使用AdaIN输出t作为内容目标，而

不是内容图像的常用特征响应。我们发现这会导致收 敛速度稍快，也符合我们反转AdaIN输出t的目标。

Lc = If (g(t)) − tI2 (12)

由于我们的AdaIN层仅传递风格特征的均值和标准偏差，因此风格损失仅与这些统计信息匹配。尽管我们发现常用的Gram矩阵损失可以产生相似的结果，但我们匹配IN统计量，因为它在概念上更干净。Li等人 也探讨了这种风格的丧失。[33].

L*L*

Ls = Iµ(φi(g(t))) − µ(φi(s))I2

+

*i*=1

L*L*

Iσ（φi（g（t）））-σ（φi

（s））I2 (13)

*i*=1

其中每个φi表示VGG-19中用于计算风格损失的一层。在我们的实验中， 我们使用相等权重的relu1 1 ， relu2 1，relu3 1，relu4 1层。

# 结果

## 与其他方法的比较

在本小节中，我们将我们的方法与三种类型的风格 转移方法进行比较：1）灵活但缓慢的基于优化的方法[16]，2）限于单一风格的快速前馈方法[52]，以及3） 灵活的基于补丁的中速方法[6].除非另有说明，否则通过在默认配置下运行它们的代码来获得比较方法的 结果。1 对于

1我们运行了500次[16]使用Johnson的公开实现：

<https://github.com/jcjohnson/neural-style>

图3.就风格和内容损失而言，不同方法的定量比较。从我们的测试集中随机选择的10幅风格图片和50幅内容图片中计算出的平均值。

[6]，我们使用作者提供的预训练逆网络。所有测试图像的尺寸均为512×512。

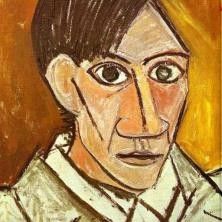
**定性的例子。在图**4 我们展示了通过比较方法生成的示例风格迁移结果。请注意，在训练模型期间，从未观察到所有测试风格的图像，而[52通过将一种网络拟

合到每种测试风格来获得。即使这样，我们的风格化 图像在质量上还是[52]和[16]用于许多图像（例如第1、2、3行）。在其他某些情况下（例如第5行），我们的 方法的质量略低于[52]和[16].这并不出乎意料，因为 我们认为在速度，灵活性和质量之间需要三权衡。和....相比 [6]，对于大多数比较的图像，我们的方法似乎可以更忠实地传递风格。最后一个示例清楚地 说明了[6]，尝试将每个内容补丁与最匹配风格的补丁 匹配。但是，如果大多数内容补丁与少数不代表目标 风格的风格补丁匹配，则风格传输将失败。因此，尽 管在某些情况下（例如第3行），[6]也可以产生吸引 人的结果。

**定**量评估。我们的算法是否会为了提高速度和灵活性 而牺牲一些质量，如果可以，要付出多少呢？为了定 量地回答这个问题，我们将我们的方法与基于优化的

方法进行比较[16]和快速的单风格传输方法[52]就内 容和风格而言。由于我们的方法使用的是基于IN统计信息的风格损失，因此我们还修改了[16]和[52]进行 公平比较（其结果在4 仍使用默认的Gram矩阵损失获得）。此处显示的内容丢失与[52, 16].所报告的数字是从WikiArt数据集的测试集中随机选择的10个风格图像和50个内容图像的平均值[39]和MS-COCO [36].

如图。3，我们合成图像的平均内容和风格损失略 高，但与Ulyanov等人的单风格迁移方法相当。[52]. 特别是，我们的方法和[52]获得与[[16]在50到100次迭代之间进行优化-



风格 内容 我们的 陈和施密特 Ulyanov等。 Gatys等。

图4.示例风格迁移结果。在培训期间，我们的网络永远不会观察到所有经过测试的内容和风格图像。

tion.考虑到我们的网络在训练期间从未见过测试风格， 而每个[52]经过测试风格的专门培训。另外，请注意， 我们的风格损失比原始内容图片小得多。

**速**度分析。我们的大部分计算都花在内容编码，风格 编码和解码上，每种计算大约花费三分之一的时间。 在某些应用场景中，例如视频处理，风格图像仅需要

编码一次，AdaIN可以使用存储的风格统计信息来处理所有后续图像。在其他一些情况下（例如，将相同的内容传输到不同的风格），可以共享在内容编码上花费的计算。

在标签页中。1 我们将我们的方法的速度与预

以前的[16, 52, 11, 6].不包括风格验证时间

编码，我们的算法分别针对256 256和512 512图×像以56和15 ×FPS运行，从而可以实时处理任意用户上传的风格。在适用于任意风格的算法中，我们的方法比

[16]，比[]快1-2个数量级6].速度比[6因为[[]中的样 式交换层6]不能很好地缩放到高分辨率风格的图像。此外，我们的方法可以达到与前馈方法（仅限于几种 风格）相当的速度[52, 11].我们方法的处理时间稍长， 主要是由于我们基于VGG的网络更大，而不是方法上的 局限性。使用更高效的体系结构，我们的速度可以进 一步提高。



表1. 256的速度比较（以秒为单位）

*×* 256和512512*×*

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 时间（256像素） | 时间（512像素） | ＃风格 |
| Gatys 等 。 Chen和Schmidt Ulyanov等。Dumoulin等。 | 14*.*17 (14*.*19)  0*.*171 (0*.*407)  **0.011** (N/A)  **0.011** (N/A)  **0.018** ( *.* ) | 46*.*75 (46*.*79)  3*.*214 (4*.*144)  **0.038** (N/A)  **0.038** (N/A)  **0.065** ( *.* ) | *∞*  *∞*  1  32 |

(a) 风格 （b）内容 (c) Enc-AdaIN-Dec

图像。我们的方法可以达到与限于少数风格的方法相当的速度[52, 11]，但比其他适用于任意风格的现有算法[16, 6].我们显示了排除和包括（用括号括起来）风格编码过程的处理时间。使用Pascal Titan X GPU获得的结果平均超过100张图像。



## 其他实验。

在本小节中，我们进行实验以证明我们重要的架构选择是合理的。我们表示本节中描述的方法。[6](#_bookmark6) 作为Enc-AdaIN-Dec。我们尝试了一个名为Enc-Concat-Dec 的模型，该模型将AdaIN替换为

(d) Enc-Concat-Dec (e) Enc-AdaIN-BNDec (f) Enc-AdaIN-INDec

图5.与基线的比较。在融合内容和风格信息方面，AdaIN比 串联更有效。同样，重要的是不要在解码器中使用BN或IN层。

级联，这是组合的自然基线策略 4

从内容和风格图像中提取信息。在广告中- 3

在解码器中，我们运行具有BN / IN层的模型，

失利

表示为Enc-AdaIN-BNDec和Enc-AdaIN-INDec re- 2

Enc-AdaIN-Dec（风格）

Enc-AdaIN-Dec（内容） Enc-Concat-Dec（风格） Enc-Concat-Dec（内容）

Enc-AdaIN-BNDec（风格）

分别地。其他训练设置保持不变。

在图5 和6，我们展示了以下示例和训练曲线

比较的方法。在由Enc-Concat-Dec基线生成的图像中5

（d）），可以清晰地观察到风格图像的对象轮廓，这

1

00 5000 10000 15000 20000 25000 30000

迭代

Enc-AdaIN-BNDec（内容）

Enc-AdaIN-INDec（风格） Enc-AdaIN-INDec（内容）

表明网络无法将风格信息与风格图像的内容区分开。这也与图一致。6，其中Enc-Concat-Dec可以减少风格损失，但无法减少内容损失。具有BN / IN层的模型在

图6.风格和内容丢失的训练曲线。

在AdaIN的仿射参数之间进行插值。

质量上也较差，而损耗则始终较高。IN层的结果特别

T（c，s，α）= g（（1-α）f（c）+αAdaIN（f（c），f（s））

差。这再次验证了我们的主张：IN层倾向于将输出规 范化为单个风格，因此当我们要生成不同风格的图像 时应避免使用。

## 运行时控制

为了进一步强调我们方法的灵活性，我们证明了样 式转移网络允许用户控制风格化程度，在不同风格之 间进行插值，在保留颜色的同时转移风格以及在不同 的空间区域中使用不同的风格。请注意，所有这些控 件仅在使用相同网络的运行时应用，而无需对培训过 程进行任何修改。

**内**容风格的权衡。可以在训练过程中通过调整Eqa中的风格权重λ来控制风格迁移的程度。[11](#_bookmark8).另外，我们的方法通过在馈送到解码器的特征图之间进行插值，允

许在测试时权衡内容风格。请注意，这等效于

（14）

网络尝试在α= 0时忠实地重建内容图像，并在α

= 1时合成最风格化的图像。7，可以通过将α从0更改为1来观察内容相似度和风格相似度之间的平滑过渡。

**风格**插值。在具有相应权重w1，w2，...，wK的一组K风格图像s ，s ，...，s 之间进行插值Kw = 1，我们类似地在特征图之间进行插值（*k*=结1 果如图2所示。8):

1 2 K k

L*K*

T（c，s1,2,...K，w1,2,...K）= g（wkAdaIN（f（c），f（sk））））

*k*=1

(15)

**空间和色彩控制。Gatys等。[**[17](#_bookmark23)]最近引入了对颜色信 息和风格迁移的空间位置的用户控制，这些控制可以

轻松地并入我们的框架中。为了保留内容图像的颜色， 我们首先将风格图像的颜色分布与



*α* = 0 *α* = 0*.*25 *α* = 0*.*5 *α* = 0*.*75 *α* = 1 风格

图7.内容风格的权衡。在运行时，我们可以通过更改Equ中的权重α来控制内容和风格之间的平衡。[14.](#_bookmark14)





图9.颜色控制。左：内容和风格图像。右：保留颜色的风格迁移结果。

图8.风格插值。通过向解码器提供通过AdaIN转换为不同风格的特征图的凸组合（Equ。[15](#_bookmark15)），我们可以在任意新风格之间进行插值。



内容图片的图片（类似于[[17](#_bookmark23)]），然后使用颜色对齐 的风格图像作为风格输入执行常规风格迁移。实例结果如图2所示。9.

在图10 我们证明了我们的方法可以将内容图像的不同区域转换为不同的风格。这是通过使用来自不同风格输入的统计信息对内容特征图中的不同区域分别执行AdaIN来实现的，类似于[4, [17](#_bookmark23)]，但以完全前馈的方式进行。尽管我们的解码器仅接受具有同质风格的输入的训练，但它自然地可以推广到不同区域具有不同风格的输入。

# 讨论和结论

在本文中，我们提出了一个简单的自适应实例规范化（AdaIN）层，该层首次实现了实时的任意风格迁移。除了引人入胜的应用之外，我们相信这项工作也为我 们对一般深层图像表示的理解提供了启示。

考虑我们的方法与基于特征统计的先前神经风格迁移方法之间的概念差异很有趣。Gatys等。[16]采用优化过程来操纵像素值以匹配特征统计信息。优化过程由[24, 51, 52].不过，

图10.空间控制。左：内容图像。中：两个带有相应蒙版的风格图像。右：风格迁移结果。

训练了作品以修改像素值以间接匹配特征统计信息。 我们采用了一种非常不同的方法，即一次将特征空间 中的统计信息直接对齐，然后将特征反转回到像素空 间。

考虑到我们方法的简单性，我们认为仍有很大的改进空间。在未来的工作中，我们计划探索更高级的网络架构，例如残留架构[24]或具有来自编码器的其他 跳过连接的架构[23].我们还计划研究更复杂的培训计划，例如增量培训[32].此外，我们的AdaIN层仅对齐最基本的特征统计信息（均值和方差）。有可能用相关比对替换AdaIN [[49](#_bookmark31)]或直方图匹配[54]可以通过传输高阶统计信息来进一步提高质量。另一个有趣的方向是将AdaIN应用于纹理合成。

#### 致谢

我们要感谢Andreas Veit的有益讨论。这项工作得到了 Google 重 点 研 究 奖 ， AWS Cloud Credits for Research和Facebook设备捐赠的部分支持。

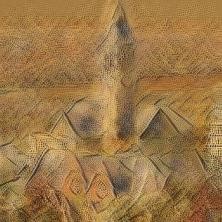


图11.更多风格迁移示例。每行共享相同的风格，而每列表示相同的内容。和以前一样，网络从未见过测试风格和内容图像。

# 参考文献

1. M. Arjovsky，S。Chintala和L. Bottou。Wasserstein 甘。

arXiv预印本arXiv：1701.07875，2017。[2](#_bookmark1)

1. JL Ba，JR Kiros和GE Hinton。层归一化。

arXiv预印本arXiv：1607.06450，2016年。[2](#_bookmark1)

1. K. Bousmalis，N.Silberman，D.Dohan，D.Erhan和

D.克里希南生成对抗网络的无监督像素级域自适应。arXiv预印本arXiv：1612.05424，2016。[2](#_bookmark1)

1. AJ Champandard.语义风格转换并将两位涂鸦转换成精美的艺术品。arXiv预印本arXiv：1603.01768，2016。[8](#_bookmark16)
2. D. Chen，L。Yuan，Liao J，Y。Yu，和G. Hua。风格库：神经图像风格迁移的显式表示。在CVPR中，2017年。[1](#_bookmark0)
3. TQ Chen和M. Schmidt。基于补丁的快速风格迁移任意风格。arXiv预印本arXiv：1612.04337，2016。[1](#_bookmark0)[, 2,](#_bookmark1) [4,](#_bookmark4) [5,](#_bookmark7) [6,](#_bookmark12) [7](#_bookmark13)
4. R. Collobert，K。Kavukcuoglu和C. Farabet。Torch7： 类似于Matlab的机器学习环境。在NIPS研讨会上，2011 年。[4](#_bookmark4)
5. T. Cooijmans, N. Ballas, C. Laurent, C¸ .居尔谢尔， 以及

答：库维尔。循环批标准化。在ICLR，2017年。 [2](#_bookmark1)

1. EL Denton，S。Chintala，R。Fergus等。使用对抗网络的拉普拉斯金字塔的深度生成图像模型。在NIPS中， 2015年。[2](#_bookmark1)
2. A. Dosovitskiy和T. Brox。用卷积网络反转视觉表示。在CVPR中，2016年。[4](#_bookmark4)
3. V. Dumoulin，J。Shlens和M. Kudlur。学到的艺术风格表现形式。在ICLR，2017年。[1,](#_bookmark0) [2,](#_bookmark1) [3,](#_bookmark2) [5,](#_bookmark7) [6,](#_bookmark12) [7](#_bookmark13)
4. AA Efros和WT Freeman。图像缝，用于纹理合成和转移。在SIGGRAPH中，2001年。[1](#_bookmark0)
5. AA Efros和TK梁。通过非参数采样进行纹理合成。在ICCV中，1999年。[1](#_bookmark0)
6. M. Elad和P. Milanfar。通过纹理合成进行风格迁移。

arXiv预印本arXiv：1609.03057，2016。[1](#_bookmark0)

1. O. Frigo，N。Sabater，J。Delon和P. Hellier。拆分和匹配：基于示例的自适应补丁采样，用于无监督的风格迁移。在CVPR中，2016年。[1](#_bookmark0)
2. LA Gatys，AS Ecker和M. Bethge。使用卷积神经网络进行图像风格迁移。在CVPR中，2016年。[1,](#_bookmark0) [2](#_bookmark1)[, 3,](#_bookmark2) [5,](#_bookmark7) [6,](#_bookmark12) [7,](#_bookmark13) [8](#_bookmark16)
3. LA Gatys，AS Ecker，M.Bethge，A.Hertzmann和

E. Shechtman. 控制神经风格转移中的感知因素。在CVPR中，2017年。[1,](#_bookmark0) [7,](#_bookmark13) [8](#_bookmark16)

1. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu,

D. Warde-Farley ， S 。 Ozair ， A 。 Courville 和Y.

Bengio。生成对抗网络。在NIPS中，2014年。[2](#_bookmark1)

1. DJ Heeger和JR Bergen。基于金字塔的纹理分析/合成。在SIGGRAPH中，1995年。[1](#_bookmark0)
2. X. Huang， Y。Li， O。Poursaeed， J。Hopcroft和S.

Belongie。

堆叠式生成对抗网络。在CVPR中，2017年。[2](#_bookmark1)

1. S.艾菲。批量重新归一化：在批量归一化的模型中减少

最小批量依赖。arXiv预印本arXiv：1702.03275，2017。[2](#_bookmark1)

1. S. Ioffe和C. Szegedy。批量归一化： 通过减少内部协变量偏移来加速深度 网络训练。在JMLR中，2015年。[2](#_bookmark1)
2. P. 伊索拉（ J.-Y.Zhu ， T。Zhou和AA Efros。使用条件对抗网络进行图像到图像的翻译。在CVPR中，2017年。[2,](#_bookmark1) [8](#_bookmark16)
3. J. Johnson， A。Alahi和L. Fei-Fei。实时风格传输和超分辨率的感知损失。在ECCV中，2016年。 [1,](#_bookmark0) [2,](#_bookmark1) [3,](#_bookmark2) [8](#_bookmark16)
4. T. Kim，M。Cha，H。Kim，J。Lee和J. Kim。学习发现与生成对抗网络的跨域关 系 。 arXiv 预 印 本 arXiv ： 1703.05192，2017。[2](#_bookmark1)
5. D.金马和J.Ba.亚当：一种随机优化方法。在ICLR中，2015年。[5](#_bookmark7)
6. DP Kingma和M. Welling。自动编码可变贝叶斯。在ICLR中，2014年。[2](#_bookmark1)
7. JE Kyprianidis， J。Collomosse， T。Wang和T. Isenberg。“艺术现状：图像和视频的艺术风格化技术分类。TVCG，2013年。[1](#_bookmark0)
8. C. Laurent，G。Pereyra，P。Brakel， Y。Zhang和Y. Bengio。批量归一化递归神经网络。在ICASSP中，2016年。[2](#_bookmark1)
9. C. Li和M. Wand。结合马尔可夫随机场和卷积神经网络进行图像合成。在CVPR中，2016年。[1,](#_bookmark0) [2,](#_bookmark1)[3](#_bookmark2)
10. C. Li和M. Wand。使用马尔可夫生成对抗网络预先计算的实时纹理合成。在ECCV中，2016年。[1,](#_bookmark0) [2](#_bookmark1)
11. Li Y， C. Fang，J. Yang， Wang.Z， Lu X，and M.-H.杨具有前馈网络的多样化纹理合成。在CVPR中，2017年。[1,](#_bookmark0) [2,](#_bookmark1)[8](#_bookmark16)
12. 李Y，王N.刘J和侯X.揭开神经风格转移的神秘面纱。arXiv预印本arXiv： 1701.01036，2017。[1,](#_bookmark0) [2](#_bookmark1)[, 3,](#_bookmark2) [5](#_bookmark7)
13. 李Y，王N.石J.刘J和侯X.重新讨论批量归一化以进行实际领域调整。arXiv 预印本arXiv：1603.04779，2016。[2](#_bookmark1)
14. Q. Liao， K。Kawaguchi和T. Poggio。流式归一化：寻求在线和循环学习的 更简单且生物学上更合理的归一化。arXiv 预印本 arXiv ： 1610.06160 ， 2016。[2](#_bookmark1)
15. T.-Y.Lin，M。Maire，S。Belongie，

J。Hays，P。Perona，D。Ramanan，P。Dolla´r 和 CL Zitnick 。 Microsoft coco：上下文中的常见对象。在ECCV 中，2014年。[3,](#_bookmark2) [5](#_bookmark7)

1. M.-Y.Liu，T。Breuel和J. Kautz。无监督的图像到图像翻译网络。arXiv预印本arXiv：1703.00848，2017。[2](#_bookmark1)
2. M.-Y.刘和O.图泽尔。耦合生成对抗网络。在NIPS中，2016年。[2](#_bookmark1)
3. 尼科尔画家的数字， 维基百科。 [https://www.](http://www/)

kaggle.com/c/painter-by-numbers，2016年。[5](#_bookmark7)

1. A. vd Oord，N。Kalchbrenner和K. Kavukcuoglu。像素递归神经网络。在ICML中，2016年。[2](#_bookmark1)
2. X. Peng和K. Saenko。使用深度生成相关性比对网络合成到实际的适应。arXiv预印本arXiv：1701.05524， 2017。[2](#_bookmark1)
3. A. Radford，L。Metz和S. Chintala。深度卷积生成对抗网络的无监督表示学习。在ICLR，2016年。[2](#_bookmark1)
4. S.Reed，Z.Akata，X.Yan，L.Logeswaran，B.Schiele和

H.李生成对抗性文字以进行图像合成。在

icml，2016年。[2](#_bookmark1)

1. M. Ren，R。Liao，R。Urtasun，FH Sinz和RS Zemel。规范化规范化器：比较和扩展网络规范化方案。在ICLR， 2017年。[2](#_bookmark1)
2. M. Ruder，A。Dosovitskiy和T. Brox。视频的艺术风格转移。在GCPR中，2016年。[1](#_bookmark0)
3. T. Salimans ， I 。 Goodfellow ， W 。 Zaremba ， V 。Cheung，A。Radford和X. Chen。训练gans的改进技术。在NIPS中，2016年。[2](#_bookmark1)
4. T. Salimans和DP Kingma。权重归一化：一个简单的重新参数化，可加快深度神经网络的训练速度。在NIPS中， 2016年。[2](#_bookmark1)
5. K. Simonyan和A. Zisserman。用于大规模图像识别的非常深的卷积网络。在ICLR中，2015年。 [4,](#_bookmark4) [5](#_bookmark7)
6. B.Sun，J.Feng和K.Saenko。令人沮丧的领域适应性回归。在AAAI中，2016年。[8](#_bookmark16)
7. Y. Taigman，A。Polyak和L. Wolf。无监督的跨域图像生成。在ICLR，2017年。[2](#_bookmark1)
8. D. 乌 里 扬 诺 夫 ， V 。 列 别 杰 夫 ， A.Vedaldi 和V.Lempitsky。纹理网络：纹理和风格化图像的前馈合成。在ICML中，2016年。[1,](#_bookmark0) [2,](#_bookmark1) [4,](#_bookmark4) [5,](#_bookmark7) [8](#_bookmark16)
9. D.乌里扬诺夫，A.Vedaldi和V.Lempitsky。改进的纹理 网络：在前馈风格化和纹理合成中最大化质量和多样性。在CVPR中，2017年。[1,](#_bookmark0) [2,](#_bookmark1) [3,](#_bookmark2) [5](#_bookmark7)[, 6,](#_bookmark12) [7,](#_bookmark13) [8](#_bookmark16)
10. X.Wang，G.Oxholm，D.Zhang和Y.-F.王多峰传输：一种用于快速艺术风格转换的分层深度卷积神经网络。arXiv预印本arXiv：1612.01895，2016。[2](#_bookmark1)
11. P. Wilmot，E。Risser和C. Barnes。使用直方图损失进行稳定且可控制的神经纹理合成和风格迁移。arXiv 预印本arXiv：1701.08893，2017。[2,](#_bookmark1) [8](#_bookmark16)
12. H. Zhang和K. Dana。用于实时传输的多风格生成网络。arXiv预印本arXiv：1703.06953，2017。[1](#_bookmark0)