风格迁移综述

徐帅
21821174
计算机技术
xushuai100@gmail.com

摘要（Abstract）

风格迁移是将一幅图片以特定风格重新绘制的过程。Gatys 等人的工作开创性地利用卷积神经网络将图片的内容和风格分离出来，从而可以生成很理想的风格化图片。这项工作引起了风格迁移在学术研究和工业应用的热潮，在原始方法的基础上，研究者又提出了很多改进和拓展的方法。

本文对风格迁移的技术进展进行了综述，并实现了几种流行的风格迁移方法，对它们的效果进行了比较和分析，同时对未来的研究重点和方向进行了展望。

1. 引言

绘画是艺术的一种表现形式，非常熟练的画家可以模仿名画的风格进行重新创作，而这通常会耗费几天甚至几个月的时间。近些年来，风格迁移算法成为计算机研究的热点领域，算法能从目标图片中提取内容的结构和语义信息，并以指定的风格对内容进行重新绘制[1]。然而，如何独立地提取并表示图片的内容和风格是一个非常困难的问题。

Gatys 等人 [2] 受到深度卷积神经网络（CNN）的启发，他们发现利用 CNN 可以提取图片的高层次的语义信息，而且图片的内容和风格的特征表示是可以分离的。基于这些发现，他们提出了 Neural Style Transfer 算法，该算法可以将给定照片的内容和名画的风格融为一体，并且生成的图片具有很理想的视觉效果。这种方法的本质是将一个随机噪声图片作为初始结果，然后以梯度下降的方式迭代地更新每个像素值，直到它的统计特征分布与内容和风格完成匹配。

这项研究成果在学术界和工业界引起了广泛关注，一些后续的工作在这个算法的基础上进行了补充和完善，从而产生了很多新的基于深度神经网络的风格迁移方法。在本文中，我们首先对 Neural Style Transfer 这个领域近期的研究成果进行概述，并自己实现了几种流行的风格迁移方法，对不同方法的效果和性能展开了讨论。本文接下来的内容组织如下：第二节对已有的风格迁移方法进行分类并概述。第三节中详细介绍了几种风格迁移算法的主要思想以及实现细节。第四节对实验结果进行了分析和讨论。最后一节总结了这个领域目前面临的挑战以及可能的研究方向。

2. 相关工作

目前的基于深度神经网络的风格迁移方法可以大致分为两类：基于图片迭代的方法和基于模型的方法。第一类方法直接迭代目标图片的像素值来生成风格化图片，而另一类方法先训练一个模型，然后就可以利用一次前向传播获得风格化图片。

2.1. 基于图片迭代的方法

基于图片迭代的方法利用后向传播迭代更新目标图片的像素值，目的是使得生成的风格化图片同时匹配内容图片的内容信息以及风格图片的风格信息。一个关键的挑战是如何表示图片的风格信息，即如何定义风格损失函数用于匹配风格化特征。

Gatys 等人最先提出了基于深度神经网络的风格迁移方法 [2]，利用 VGG 网络的中间层特征输出对图片进行重建，他们发现深度卷积神经网络可以提取任意照片的语义内容以及画作中的风格特征。依据这一观察，他们利用目标图片与内容图片在网络高层次的特征差异来重构语义内容结构，并通过匹配目标图片与风格图片特征的统计信息来重构风格样式。经过在目标图片上的迭代，最终可以获得比较理想的风格化图片。

然而，Risser 等人 [3] 发现这个方法在迭代过程中具有不稳定性，原因在于两个具有不同均值和方差的特征也可能产生相同的 Gram 矩阵，为了解决这个问题，他们在之前方法的整体损失函数中加入一个直方图损失，这种直方图损失使得优化过程趋向于保持风格特征的均值和方差，因此更加稳定。

由于上面提到的方法要计算网络多层特征的像素级别的差异，因此计算开销比较大。Chen 等人 [4] 认为局部级别的信息就可以捕捉绘画的风格特征，他们提出了一种基于 patch 的局部优化目标，并作用在网络中的单层特征上，从而使得目标函数的收敛更快，但存在的不足是丢失了全局风格的信息。

2.2. 基于模型的方法

基于图片迭代的方法可以生成很理想的风格化图片，但这类方法还是存在一些限制。其中一个就是效率的问题，而基于模型的方法以牺牲灵活性为代价，解决了之前方法的速度问题。主要的想法是利用大量某种风格的图片训练一个前向传播网络，利用梯度下降法逐步优化更新网络的参数。Johnson 等人 [5] 首先对某个具体的风格训练一个前向传播网络，这样只需一次传播过程就可以将内容图片转换成风格化图片。这个方法由两部分组成，即生成网络和损失网络，利用损失网络作为优化生成网络的目标函数。

尽管基于模型的方法要比基于图片迭代的方法快很多，但是缺点也比较明显，对于每种风格都要训练一个生成网络，这样既耗时又不灵活。Dumoulin 等人 [6] 提出了一个可以同时学习多个风格的方法，他们发现很多画作使用的是相同的画笔，而只是调色板的颜色不同，因此，他们认为很多风格图片可以共享相同的计算，如果单独地为每一个风格图片训练一个前向网络是多余的。

3. 基于神经网络的风格迁移

3.1. Neural Style Transfer

Gatys 等人提出的风格迁移方法实际上是结合了图片重构策略和纹理合成方法，他们发现网络每个层的输出可以视为对输入图片的编码。为了可视化图片在网络中被编码后包含的信息，我们可以对一个空白噪声图片执行梯度下降，使之在网络中某层的输出与内容图片的输出相匹配。我们定义 和 为原始图片和将要生成的图片，它们在网络某层 中特征的损失函数为：

$$L_{content}(p,x,l)=\frac{1}{2}\sum_{i,j}(F_{ij}^{l}-P_{ij}^{l}).$$

随着网络层次的增加，提取的特征越来越趋向于描述图片的内容信息，而不再限制具体的像素值。因此，我们利用网络中的高层特征作为图片的内容表示。对于风格特征，可以使用特征的不同通道间的相关系数进行表示，相关系数可以用 Gram 矩阵 $G^{l}\in R^{N_l\times N_l}$ 的形式进行表示，其中 G_{ij}^{l} 是第 l 层的特征向量 i 和 j 间的内积：

$$G_{ij}^{l}=\sum_{k}F_{ik}^{l}F_{jk}^{l}$$

利用多个层中特征的相关系数，我们得到图片在多个维度上的统计信息，可以用于表示图片的纹理特征。利用梯度下降，逐步更新一个随机噪声图片，使之在某些层输出特征的 Gram 矩阵与风格图片特征的 Gram 矩阵之间的均方误差最小化。我们定义 A^{l} 和 G^{l} 为原始图片与目标图片在网络 l 层的 Gram 矩阵，对于第 l 层的风格损失函数定义为：

$$E_l=\frac{1}{4N_l^2M_l^2}\sum_{i,j}(G_{ij}^{l}-A_{ij}^{l})^2$$

然后总的风格损失定义为：

$$L_{style}(a,x)=\sum_{l=0}^{L}w_lE_l$$

我们可以通过改变 的值控制每一层特征的重要性。为了将画作 a 的风格迁移到照片 p 上，我们可以合成一张新的图片同时匹配 p 的内容特征和 a 的风格特征，因此我们定义总的损失函数为：

$$L_{total}(p,a,x)=\alpha L_{content}(p,x)+\beta L_{style}(a,x)$$

其中，参数 α 和 β 用于控制风格和内容的比重。Gatys 等人使用 L-BFGS 优化算法对目标图片进行优化，而我在实验中使用了 Adam 优化方法，可以有更快的收敛速度。

3.2. 保持原始颜色

上面的方法生成的风格化图片的颜色分布与风格图片相同，但是人们可能更希望保持原来内容图片的颜色，即只迁移风格特征而不迁移颜色。Gatys 等人[7] 在之后的论文中给出了两种保持内容图片颜色的方法，其中一种方法是在风格迁移之前，将风格图片的颜色匹配到内容的颜色。他们使用了一个线性的颜色迁移方法，即对图片中的每一个像素值 $x_i=(R,G,B)^T$ 进行线性转换：

$$x_{i'}\leftarrow Ax_S+b$$

其中， A 是一个 3×3 的矩阵， b 是一个 3 维向量。我们期望这个转换可以使得新的风格图片与内容图片具有相同的均值和协方差，这样对 A 的取值有两种常用的计算方法，第一种是对协方差矩阵做 Cholesky 分解，第二种是求协方差矩阵的平方根，在实验中发现两种计算方法的区别并不明显。

另一种保持颜色的方法是只在图片的亮度通道上进行风格迁移，因为人类视觉对亮度比颜色更敏感。该方法将图片转换为 YIQ 颜色空间，然后只在 Y 通道上进行样式迁移，然后与原始内容图片的 I 和 Q 通道组合在一起，生成目标的风格化图片。

3.3. 部分风格迁移

通常我们不会只使用网络中低层的特征作为风格的表示，因为低层的特征无法捕获图片的全局风格。Novak 和 Nikulin [8] 发现图片中的颜色特征主要是由

网络的较低层表示，因此他们使用低层的内容特征替代之前方法的低层风格特征，保持了内容图片的颜色和一些结构细节，只对中高层的风格进行迁移，获得了比较理想的效果。

3.4. 基于patch的快速风格迁移

之前我们提到基于模型的方法具有很高的效率，但是缺点是对于每种风格都要训练一个网络，因此无法适用于任意风格的迁移。Chen 和 Schmidt [4] 提出了一种快速的基于 patch 的风格迁移方法，他们定义了一个只依赖于网络中单层特征的优化目标。这种方法的主要部分是利用一个基于 patch 的操作生成在单个层的目标特征，他们将这个过程称为“swapping the style”，即将内容图片里的每一个小块用风格图片进行替换。使用 $\phi(C)$ 和 $\phi(S)$ 分别表示网络中某一层的内容和风格的特征，一个 style swap 过程如下：

1. 从内容和风格特征中分别提取一个 patch 集合，表示为 $\{\phi_i(C)\}_{i\in n_c}$ 和 $\{\phi_j(S)\}_{j\in n_s}$ ，提取的 patch 间要有足够的重叠。
2. 对于每一个内容的 patch，找到与之最接近的风格的 patch，其中利用相关系数作为差异的度量：

$$\phi_i^{ss}(C,S) := \underset{\phi_j(S),\, j=1,\ldots,n_s}{\arg\max} \frac{\langle \phi_i(C), \phi_j(S) \rangle}{\| \phi_i(C) \| \cdot \| \phi_j(S) \|}$$

3. 将每个内容 patch $\phi_i(C)$ ，用与它最接近的风格的 patch $\phi_i^{ss}(C,S)$ 进行替换，并重构完整的内容特征 $\phi^{ss}(C,S)$ ，并对重叠的部分取平均值。

上面的操作过程可以用卷积网络实现，网络结构如图所示，一个二维卷积层，一个通道间取最大值，一个二维反卷积层。

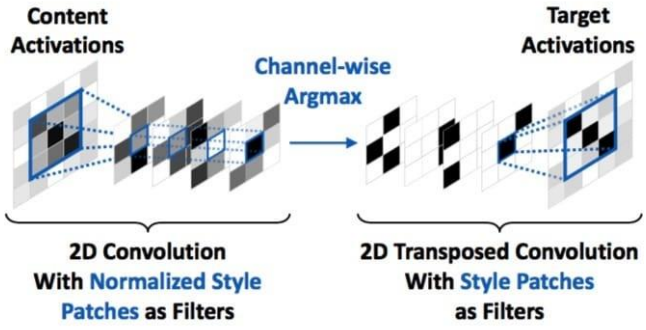

Figure 1 style swap的卷积网络实现方法

对于风格化图片的每个像素值的计算，可以定义它与目标特征 $\phi^{ss}(C,S)$ 的损失函数，与之前的方法一样使用平方误差作为优化目标函数：

$$L_{stylized}(C,S) = \frac{1}{2}\sum_{i,j}(\phi(I) - \phi^{ss}(C,S))^2$$

我们可以选择网络的不同层的特征，但我们直接在原图的 RGB 通道上做 style swap 操作时，就相当于给图片进行了重新上色。另外，我们可以通过调整 patch 的大小来控制风格化的程度，当 patch 的尺寸增大时，越来越多的内容图片中的结构会被丢失，并被风格化的纹理所替代。

4. 实验结果

在本节中，我们使用 MXNet 框架以及预先训练的 VGG-19 网络，实现了之前介绍的几种风格迁移方法，并对实验结果进行了分析讨论。

4.1. 图片特征重构

我们通过匹配网络中多个层的特征的 Gram 矩阵来重构图片的风格，因此选取层的个数和位置就决定了风格化的效果，下图所示是风格重构的结果，我们发现匹配高层的风格特征可以获得更连续更平滑的视觉体验，因此我们匹配从低到高多个层的风格特征。当使用低层的内容特征进行匹配时，算法主要匹配了像素级别的信息，生成的图片几乎无法看出风格化的特征。而使用高层特征匹配内容时，细节的像素信息就不会被严格限制，而只是保持了内容图片的结构信息，可以与风格图片很好的进行融合。

4.2. 内容与风格的权重

图片中的内容和风格信息显然无法完全分离，因此生成的图片就无法同时完全匹配风格和内容特征。因为我们定义的损失函数是内容损失和风格损失的线性组合，我们可以手动地调整风格和内容的权重，来获得较好的视觉效果。我们使用一个随机白噪声作为初始图片，并设置 $\alpha/\beta = 1\times10^{-4}$ 为风格化权重，通过训练得到风格化图片，如下图所示，我们观察到风格特征很好的转移到了内容图片上，并且内容的结构没有被完全破坏。

4.3. 保持内容的颜色

在实验中，我们采用了两类方法对内容的颜色进行保持。首先我们在风格迁移之前，用线性转换方法将风格图片的颜色分布进行调整，然后将新的风格图片作为网络的输入，我们看到两种线性转换的方法并没有太大的区别。另外，第二个图是使用部分样式迁移的效果，这种方法通过使用内容的低层特征来保持颜色，同时具有更清晰的内容细节。

4.4. Style Swap结果

基于 patch 的方法只在单层网络特征上计算损失，因此选取不同层的特征会对结果产生影响。我们直接在原图的 RGB 通道上做 style swap 操作时，就相当于给图片进行了重新上色。我们发现在 “relu3_1” 层的特征上进行操作可以保持比较连续的内容结构，并且具有比较理想的视觉效果。

可以通过设置不同的 patch 大小来控制风格化的程度，patch 的尺寸越大，内容图片中的结构就越容易丢失，取而代之的是风格化的纹理特征。下图展示了不同 patch 大小对风格化程度的影响，可以看到，当 patch 比较大的时候，内容的结构特征已经非常模糊了。

下表是 Neural Style Transfer 方法和 Style Swap 方法的平均计算时间对比。由于 Style Swap 方法只使用网络中的单层输出，因此在单次迭代上比 Neural Style Transfer 方法要快一些。同时，我们看到在总迭代次数上 Style Swap 方法也更具优势，因为计算 patch 损失的方法相比于计算每个像素的损失具有更好的稳定性，因此也更容易收敛。

Figure 2 计算时间对比

方法	迭代次数	单次迭代时间(s)	总时间(s)
Neural Style Transfer	500	0.325	162.5
Fast Style Swap	100	0.104	10.4

5. 总结和展望

风格迁移在过去的这几年里成为了一项热门的研究领域，本文对风格迁移方法进行了分类，概述了这些方法的原理和拓展，通过自己实现几种方法，对它们的优缺点进行了比较。我们发现原始的 Neural Style Transfer 具有优化过程不稳定的缺点，导致计算代价比较大，而基于 patch 的方法降低了计算代价，更易于收敛，但是该方法无法获得全局的风格特征，因此生成风格化的效果并不是很理想。

风格迁移方法未来的研究可能会着重于两个方向，其中第一个方面就是对现有方法的缺点进行改进，例如，如何自动地调整参数[9]，以及如何做到更快的风格迁移。而另一方面，可以对风格迁移的方法进行扩展，一些有趣的延伸会成为未来风格迁移领域的热门话题。

References

[1] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, and Mingli Song. Neural styletransfer: A review. arXiv preprint arXiv:1705.04058, 2017.

[2] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer usingconvolutional neural networks, 2016.

[3] Pierre Wilmot, Eric Risser, and Connelly Barnes. Stable and controllable neural texturesynthesis and style transfer using histogram losses. arXiv preprint arXiv:1701.08893,2017.

[4] Tian Qi Chen and Mark Schmidt. Fast patch-based style transfer of arbitrary style. arXivpreprint arXiv:1612.04337, 2016.

[5] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time styletransfer and super-resolution. In European Conference on Computer Vision, pages 694–711. Springer, 2016.

[6] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representationfor artistic style. CoRR, abs/1610.07629, 2(4):5, 2016.

[7] Leon A Gatys, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Preserving colorin neural artistic style transfer. arXiv preprint arXiv:1606.05897, 2016.

[8] Yaroslav Nikulin and Roman Novak. Exploring the neural algorithm of artistic style.arXiv preprint arXiv:1602.07188, 2016.

[9] Lu Sheng, Ziyi Lin, Jing Shao and Xiaogang Wang, "Avatar-Net: Multi-scale Zero-shot Style Transfer by Feature Decoration", in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.