题　　目：　基于卷积神经网络的图像风格迁移

目　录

1 风格迁移的现实意义

风格迁移或者说神经风格迁移（Neural Style Transfer）是一种应用于计算机视觉和图像处理领域的技术，它基于卷积神经网络（Convolutional Neural Network，CNN），一种受到生物视觉系统启发的人工神经网络类型，使用神经表征来分离和重新组合任意图像的内容和风格，可以在保留原始图像信息内容的同时将另一幅图像的艺术风格（画风）转移到另一幅图像上，从而创造出新的图像[1]。

1.1 人文艺术领域

在艺术领域，人类已经掌握了通过在图像内容和风格之间构成复杂的相互作用来创造独特视觉体验的技能，但真正拥有这种技能的人往往十分难得。

神经风格迁移技术将以接近于人类的表现为人类提供一种新的创作手段，使人们能够探索、融合和创造不同的艺术风格，从而创作出独特而富有创意的艺术作品。

这样，个人可以根据自己的审美，将特定的艺术风格（如抽象派）应用于个人照片或图像中，实现个性化设计。影视制作和娱乐行业，也可以为电影、电视节目和游戏创造出更多独特的视觉效果，丰富作品的艺术表现形式。此外，在艺术史研究中，也可以用于模拟艺术家的风格，分析和研究不同艺术作品之间的联系。

1.2 科学研究领域

由于卷积神经网络在视觉感知领域表现出的强大处理能力，神经风格迁移技术将为科学研究中影像数据的呈现和可视化提供更具可理解性的手段。

例如，可以改善医学影像的可视化效果，帮助医生更轻松地分析和诊断影像数据；可以改善地理图像和地球观测数据的可视化效果，提高图像的清晰度、对比度和视觉吸引力，有助于研究人员更好地分析和理解地球表面的地形、气候和环境变化等等。只有是关乎视觉感知方面的，其实都可以应用神经风格迁移技术在科学研究中发挥一定作用。

2 核心原理介绍

风格迁移技术是先基于卷积神经网络提取图像的多层次级别特征从而才可以组成表示所需的内容与特征风格特征。所以真正的理解卷积神经网络以及图像内容与风格特征的本质才可以进行风格迁移的实现。

2.1 卷积本质

卷积神经网络是神经网络模型之一，“卷积”一词是其与其它神经网络的本质区别。

在早期的神经网络和卷积神经网络的发展阶段，特征提取的过程并没有被详细地解释和理解。如在最早的卷积神经网络论文，尽管卷积操作的效果被观察到并应用于手写数字识别等任务，但并没有详细解释卷积核为何能提取图像中的特征的，而是关注于展示神经网络在特定任务上的性能，对于卷积核的工作原理并没有进行深入的解释[2]。

在数学中，卷积是两个函数之间的一种运算，描述了两个函数（或信号）之间的交互，表示一个函数经过另一个函数影响后的结果。形式上数学中的连续卷积定义为两个函数的

积分运算：

$$(f \cdot g)(t) = \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau$$

其公式可以理解为:每一时刻可变系统 f 在受其它所以时刻影响 g 下后的状态的叠加。

而在 CNN 中，假设有一个输入图像 I，卷积核表示为 K（其大小为 M*N），进行卷积操作后生成的特征映射为 F，卷积操作用数学公式则表示为：

$$F(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(i+m, j+n) \cdot K(m,n)$$

其中，F(i,j)是在卷积操作后图像位置(i,j)处的值,K(m,n)为卷积核在位置(m,n)处的值。这类似于在离散情况下，数学中对卷积的定义。

上面说到，数学中卷积是两个函数之间的一种运算，可以运算出可变系统 f 经影响 g 后的值。实际上卷积神经网络中的卷积核就好比是一组“影响因子”所构成的“影响”g，某一时刻系统 f 的值就是图像的某一像素点。在卷积神经网络中，卷积操作通过滑动窗口的方式应用于整个输入图像（叠加），从而算出每个像素在卷积核范围内的像素所对应的影响下所对应的值，以此得到一张特征图。

特定的卷积核（权重序列）可以提取以特定的特征为基准的特征置信图，这其实就是所谓的权重共享即“拉取周围影响”可以对输入图像的局部区域一环接着一环进行感知所带来的能力。

2.2　特征提取

尽管如今的卷积神经网络可以高度的抽象化图像不同层次的特征信息，但如果无法通过这些特征信息进一步分离和捕捉图像的内容和风格并定义出所谓的内容特征和风格特征，风格迁移任务依然无法实现。所以找到正确的方式，分别将图像的内容和风格两者在卷积神经网络不同层的相关映射以数值化的方式量化出来[3]，是任务的首要前提。

2.2.1　内容特征

内容特征是指图像中的高级语义信息，它代表了图像中物体的物体的整体形状、轮廓、位置和结构等。在卷积神经网络中，较深层次的特征表示通常能够捕捉到图像的内容信息，因为这些层次的特征较为抽象，能够捕捉到图像的整体语义信息。所以可以选择卷积神经网络当中较深卷积层的激活值（Feature Maps）来表示图像的内容特征。

2.2.2　风格特征

风格特征则代表了图像的纹理、色彩、笔触等艺术风格的信息。这些特征不涉及图像中具体物体的信息，而是表现出一种艺术风格或者图像的整体外观特征。风格特征在卷积神经网络中则是通过浅层次的卷积层或池化层的激活值和它们之间的相关性来定义的，因

为这些层次的特征更多地捕捉的是图像的纹理、颜色等局部细节信息。

2.3 损失函数

损失函数（Loss function）是在机器学习和深度学习中衡量模型预测值与真实值之间差异的函数。当明确了图像的内容和风格与卷积神经网络提取出来的特征的联系后，就可以将风格与内容特征数值化定义以设计合适的损失函数，并将其作为优化算法的目标函数，以使用优化器调整参数最小化损失，从而达到风格迁移的效果。

同时，图像又由任意个像素点组成，而不同格式的图像（以及视频）可以使用不同类型的通道和颜色模型来表示图像的像素信息。如常见的图像格式（如 JPEG、PNG、BMP 等）以及视频格式（如 MP4、AVI 等）可以采用不同的通道表示图像的颜色信息。最常见的 RGB 通道（Red, Green, Blue），其图像中的每个像素就由红色（R）、绿色（G）、蓝色（B）三个通道的强度值组成。依据这些数值，就可以定义风格与内容特征设计合适的损失函数。

2.3.1 内容损失函数

为量化生成图像与原始内容图像之间的内容差异，使用预处理的卷积神经网络处理生成图像与原始内容图像并提取两者在卷积神经网络当中较深对应层的特征信息（假设图像为 RGB 格式），一组卷积核处理对应的则是三组通道信息即三组像素矩阵，将两者每组对应矩阵中的像素值进行对比就可以表示出内容差异。通常采用的是均方误差（Mean Squared Error, MSE）来衡量这种差异，转换为数学公式即：

$$L_{content}=\frac{1}{2}\sum\left(C_c-G_c\right)^2$$

其中括号内分别是内容（Content）图像和生成（Generate）图像在对应位置的值。

2.3.2 风格损失函数

风格损失函数对比的是用户所提供的风格图像以及生成图像之间的差异，相比于内容损失函数，风格损失函数则是基于图像的统计特征将卷积神经网络当中相同层在不同通道的特征之间的相关性、协方差用数值的方式表示出来。一种常用的方法是使用格拉姆矩阵（Gram Matrix），该矩阵能够捕捉不同特征之间的相关性。其运算的过程是将每个通道的像素矩阵信息拉平为一个向量作为一行并将这些向量放入一个矩阵当中，该矩阵与本身的转置进行数量积，就可以算出两两通道之间的数量积：

$$G_{ij}=\sum_k F_{ik}F_{jk}$$

其中 i 和 j 分别代表某个通道，k 为总通道数。

这种数量积的数值大小可以表现出两个通道之间的‘共现相关性’，将风格（Style）图像以及生成（Generate）图像在对应层的特征信息当中所蕴含的这种‘共现相关性’进

行差值比较，就可以定义出风格损失函数：

$$E=\frac{1}{4N^2M^2}\sum_{i,j}(G_{ij}-S_{ij})^2$$

其中 N 为通道数 M（特征图数量）为信息矩阵大小（特征图尺寸）。由于该公式计算的是某一层的损失值，而每一层的特征图数量 N 和特征图的尺寸 M 不一定相同，所以需要前面的系数确保不同层级之间的损失贡献能够平等地影响总的风格损失。

往往风格损失函数需要多个层级 L，且对应不同的权值：

$$L_{style}=\sum_{l=0}^{L}w_lE_l$$

2.3.3　总损失函数

以上当风格损失函数以及内容损失函数都定义好之后，可以根据个人的喜好给内容损失函数和风格损失函数设置权重满足生成需求，并合并组成总损失函数以供优化器进行参数（像素值）优化：

$$L_{total}=\alpha L_{content}+\beta L_{style}$$

2.3.4　总变差损失

除了以上这些主要的损失之外，其实还可以引入总变差损失（Total Variation Loss）用来衡量生成图像像素之间的差异，以减少图像中的噪声，促进最终生成的图像具有更加的平滑。

如果在更新像素值的时候，只关注于风格损失和内容损失，那么各个像素值参数之间更偏向于独立的个体。为了改善生成图像的视觉质量使图像看起来更加清晰和平滑，引入总变差损失（TV）对图像中每个像素的水平和垂直方向上的差异进行了求和，将其引入损失函数当中进行降低就可以缩小像素值之间的差异。其损失函数的公式可以表示为：

$$TV(I)=\sum_{i,j}\sqrt{\left(I_{i+1,j}-I_{i,j}\right)^2+\left(I_{i,j+1}-I_{i,j}\right)^2}$$

表示二维图像 I，其像素表示为矩阵，对图像中每个像素与其相邻像素之间的差异进行了求和。

2.4　优化器

优化器是深度学习中用于更新和调整神经网络模型参数以最小化损失函数的算法。为了满足更多的保留内容且更完整的迁移风格，就需要不断的更新和调整生成图像的像素值

以最小化损失函数。不同的优化器可能对同一个模型在不同的任务上表现出不同的性能，在设备硬件性能满足的前提下，最小化损失就需要挑选合适的优化器。

Adam（Adaptive Moment Estimation）是一种自适应学习率的优化算法，它结合了梯度的一阶矩估计（均值）和二阶矩估计（方差）的指数移动平均来更新参数，即动量方法和自适应学习率（RMSprop）方法，以更高效地更新神经网络模型参数且该方法易于实现，计算效率高，对内存要求还低[4]。

在风格迁移任务中，损失函数是由内容损失和风格损失组成，这些损失函数会涉及大量的特征表示，导致梯度比较稀疏这会导致某些参数的梯度值相对较小，甚至为零。对于一些常规的优化器（如标准的随机梯度下降算法）在梯度较小或接近于零的情况下，梯度更新会很缓慢，使得模型收敛速度变慢，甚至停滞不前，这个时候选择具有一定的自适应性的优化算法（如 Adam、RMSprop）会使训练任务更加高效。RMSprop 会在历史数据的支持下，自动调整学习率,而 Adam 是 RMSprop 和动量法的结合,动量法则会在历史数据下调整梯度。

2.4.1　动量法

关于梯度下降算法的优化一方面是减少计算量，另一方面则是优化下降路径为了加速收敛速度，并且克服典型的梯度下降方法中可能出现的问题，如收敛速度慢或陷入局部最优解。这里一般会优化学习率和梯度，动量法则是为了优化梯度而设计出来的一种方法。

动量法引入了一个额外的概念：动量（momentum）。这一般是物理上的概念，在这里可以理解为给梯度下降过程一个“加速度”，以便更快地前进到一个相对更好的方向以到达损失函数的最优解处。一般参数的更新形式为：

$$w_{(t+1)i}=w_{(t)i}-\eta\frac{\partial J\left(w_{(t)i}\right)}{\partial w_i}$$

其中 i 代表的是某一维度，对应参数所在的维度。动量法依据的是历史数据，也就是之前的梯度信息，令

$$\nabla w_{(t+1)i}=\frac{\partial J\left(w_{(t)i}\right)}{\partial w_i}$$

有

$$V_{(t+1)}=\nabla w_{(t+1)i}+V_t$$

以上公式根据之前的梯度信息进行“累加”，会依据之前的梯度信息平等的将影响施加到参数更新上，但实际上历史越久远的信息在直觉上影响应更小，此时应用“指数加权移动平均法”（Exponential Weighted Moving Average, EWMA）累加方式应该为：

$$V_{(t+1)}=(1-\beta)\nabla w_{(t+1)i}+\beta V_t$$

它通过对最近的观测值赋予较高的权重，较早的观测值赋予较低的权重，从而计算出平滑后的序列。β一般设置的值为0.9,实际上历史久远的数据在数值上将会指级数级减小。

动量法的完整参数更新公式为：

$$w_{(t+1)i}=w_{(t)i}-\eta V_{(t+1)}$$

2.4.2 自适应学习率

RMSprop 是深度学习梯度下降中用于优化学习率的一种优化算法。它会在梯度更新时学习率也能自适应地进行调整，在处理不同特征的梯度变化幅度不同的情况时，避免了学习率在训练过程中衰减过慢或者由于固定学习率衰减而产生的震荡问题。

该方法参数更新的完整公式为：

$$w_{(t+1)i}=w_{(t)i}-\frac{n}{\sqrt{s_{(t+1)}+\varepsilon}}.\nabla w_{(t+1)i}$$

其中，ϵ代表的是某个极小值，只是为了避免分母为零

$$s_{(t+1)}=\nabla w^2_{(t+1)i}+s_{(t)}$$

以上公式可以看到，对于不同的参数，如果其历史梯度方差（范围波动）较大，会降低其学习率；如果梯度方差较小，会增加学习率。这种自适应性甚至有助于在鞍点附近更快地越过局部最小值。

此时的自适应算法其实还叫做 AdaGrad，同样使用指数加权移动平均法，使得

$$s_{(t+1)}=(1-\beta)\nabla w^2_{(t+1)i}+\beta s_{(t)}$$

此时才叫完整的 RMSprop（Root Mean Square Propagation）。

3 环境搭建

当对基本的原理理解之后，就可以了进行项目的代码实现。首先需要选择合适的卷积神经网络，利用它提取出来的图像特征。然后就是对于深度学习框架的选择，当已经定义好了明确的损失函数之后，没有必要从零开始构建优化算法，可以使用深度学习框架中已经提供的优化器，自己调整参数，实现生成图像的优化。

3.1　卷积神经网络的选择

常用的卷积神经网络模型包括 VGG、ResNet、Inception、MobileNet 等，对于此次任务选择 VGG 网络。因为 VGG 模型是在大规模图像数据集上进行预训练的，因此对于大多数图像的内容和风格有很好的特征提取能力。

VGG 网络的特点是其简单却具有深度性，因此在不同层次可以捕捉到图像不同抽象级别的特征。通常，在其浅层网络的特征更多是图像的低级结构（如边缘、纹理等），而深层网络的特征更多地关注于图像的高级抽象内容（如物体、整体结构等）[5]，再加上它对于硬件的资源要求很低且在常见的深度学习框架中也可以使用，所以选择 VGG 网络去实现风格迁移任务是不错的选择。

3.2　深度学习框架的选择

TensorFlow、PyTorch、Keras 等都有庞大的社区支持和丰富的文档，便于学习和解决问题，这些框架之间各有优缺点，但随着不断变化的需求和技术发展，其实都在不断的更新迭代。

这里选择由 Google 开发的 TensorFlow 框架，TensorFlow 已经更新到第二代版本，即 TensorFlow 2.x。它引入了许多改进和新特性，其中的即刻执行模式可以立即执行代码当中的每个操作，无需构建计算图，并且 TensorFlow 2.x 移除了 TensorFlow 1.x 中一些复杂的符号版本化系统使得 TensorFlow 更适合初学者。

TensorFlow 2.x 框架本身也提供了许多常用的对图像以及计算的方法，可以直接使用，这很重要。

4　项目实现

当构建好了基础的项目运行环境，就可以进行项目的代码实现。以下将展示整个风格迁移项目的思路流程以及实验结果。

4.1　项目总流程

1.加载 VGG19 卷积神经网络选取层中权重参数，构建网络。

2.加载风格以及内容图像的像素矩阵信息并进行标准化，将信息导入网络当中在选定层进行卷积，池化以及激活操作得到抽象出来的内容以及风格特征信息并进行标准化。将这些像素信息作为各损失函数的传入参数进行计算。

3.正常情况下是使用噪声图像作为初始生成图像与内容图像和风格图像的特征信息计算得出初始损失值。在此基础上开始更新生成图像。

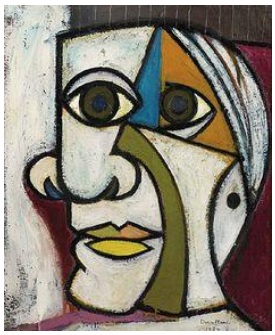
4.根据每次更新好的生成图像使用定义好权重的内容与风格损失和总变差损失计算函数得到总损失函数，以此为目标函数并结合 TensorFlow 2.x 框架下的方法设置好 Adam 优化器开始优化生成图像参数。

5.保存每一次迭代的损失值，绘制成图。观察三类损失值在总迭代过程下的变化，逐步调整超参数，得到最好的参数集以生成风格迁移图。

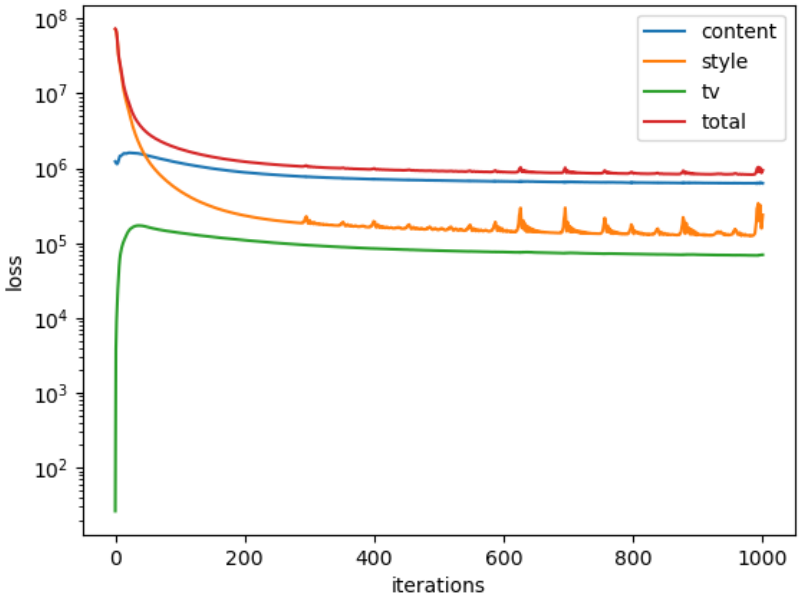
4.2　结果展示与分析

内容图像：

风格图像：

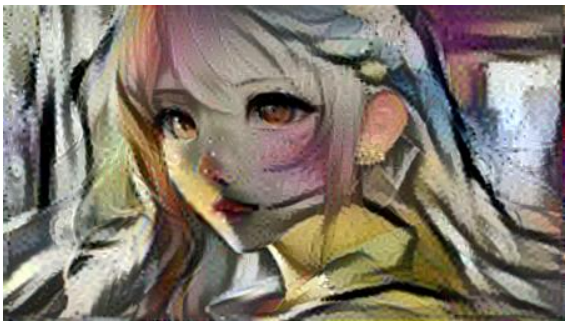


单个像素值的范围为 0-255,调整合适的超参数值（如将学习率设为 6，权衡风格与内容权重等），得到以下损失变化图：



从以上的变化图可以看出损失函数随着迭代次数下降，并逐渐的收敛趋于稳定。据此

可以对不同的超参数组合进行尝试，比较不同参数设置下生成的图像质量，选择效果最佳的参数组合。最终得出效果符合风格迁移要求的风格生成图：

5　结论（结束语）

本次风格迁移任务，旨在将一幅图像的内容与另一幅图像的艺术风格相结合，重新创造出一幅图像。通过将卷积神经网络和损失函数组合，经过数轮迭代，我们成功地实现了内容和风格之间的融合，生成了具有独特画风的图像。在训练过程中我们重点关注图像内容和风格的保留，并通过逐渐调整超参数优化出了最终的生成图像。

在此过程中，我们遇到了一些挑战，如调整超参数、确保损失函数的平衡性。通过不断尝试和调整，我们逐步改善了生成图像的质量和效果。并从中收获到了深度学习的乐趣。这次风格迁移任务为我们提供了深入了解深度学习技术以及框架的机会，为未来的工作和学习提供了宝贵的经验。虽然任务完成了，但图像风格迁移作为一个持续发展的领域，仍然有许多有待探索和改进的地方。在未来，可以进一步研究改进模型、探索更多的损失函数组合以及优化训练方法，以产生更加逼真和艺术化的图像。

最后，我们感谢所使用的工具和库，以及老师和相关研究人员，为我们提供了实现这一目标的框架和技术理论支持。

主要参考文献：

[1] GATYS L, ECKER A, BETHGE M. A Neural Algorithm of Artistic Style[J]. Journal of Vision, 2016, 16(12):326-326.

[2] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-Based Learning Applied to Document Recognition.pdf[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.

[3] GATYS L A, ECKER A S, BETHGE M. Image Style Transfer Using Convolutional Neural Networks[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016:2414-2423.

[4] KINGMA D P, BA J. Adam: A Method for Stochastic Optimization[J]. CoRR, 2014, abs/1412.6980.

[5] SIMONYAN K, ZISSERMAN A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. CoRR, 2014, abs/1409.1556.