

**《人工智能导论》课程项目**

**进 展 报 告**

项 目 题 目：《图像风格迁移》

所 在 学 院：计算机学院

项 目 组 长：梁添

小 组 成 员：潘宇城、张洪宇、许奕聪、袁剑钦

开 题 时 间：2020年 4月 5日

### 项目任务

本项目基于**TensorFlow**，使用**预训练的VGG19模型**构建一个Style Transfer模型，实现图片风格的迁移。我们要从预训练的VGG19模型中，获取卷积层部分的参数，用于构建我们自己的模型。我们认为：实验中最重要的是内容损失和风格损失的调整和计算，因为这影响着我们最终的图片生成效果。其次，本项目后期还会通过比较**原始图像风格迁移**和**快速风格迁移**来进一步学习研究，加深掌握这一项人工智能技术。

下面是具体的项目任务划分：

1. **实验准备：**

Tensorflow、Vgg19模型软件包下载，进行训练的电脑具备1060Ti NVIDIA显卡，实现Tensorflow 的GPU支持，降低因设备落后问题带来的时间成本。

1. **初步实施：**

查看相关论文，利用预训练模型，参考相关博客，编写优质的训练和测试代码，最后使用预选图片（风格图片和原始图片）进行实验，对初步实验结果进行分析总结。

1. **改进方法：**

尝试更多图片，对算法进行调整，小组成员跟踪每一个模块的功能实现。有不懂或者不理解的地方，小组组织开展讨论并制定解决方案。

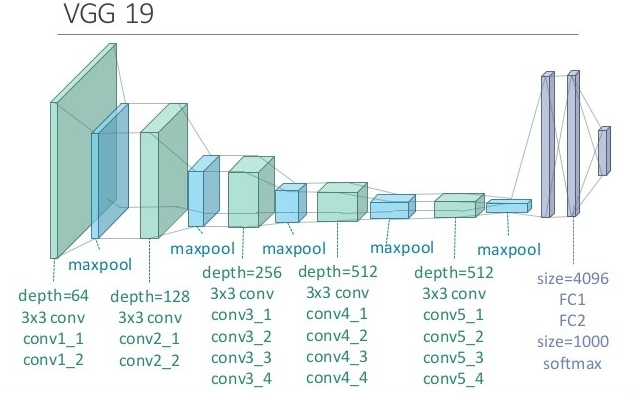
1. **整理分析得出实验结果：**

探讨损失函数的权重，得出实验结果，分析不足之处，达到进一步学习的目的。

### 技术方案

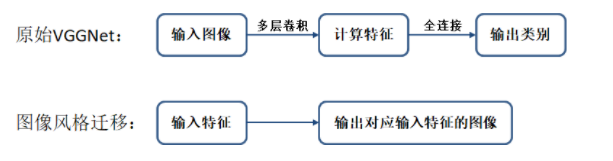
1. **模型介绍：**

我们进行图像迁移采用的是VGG19网络。VGG19可以分为5个block，每个block都是由若干卷积层及之后的池化层组成，这5个block的池化层都是最大池化，只是卷积层的层数不同，第一个block有2层卷积（conv1\_1和conv1\_2），第二个block也是2层卷积，之后的3个block都是4层卷积，最后是两个全连接层（FC1和FC2）和一个用于分类的softmax层。但是**风格迁移任务不同于物体识别**，所以我们不需要最后的两个全连接层和softmax层。

（VGG19模型图）

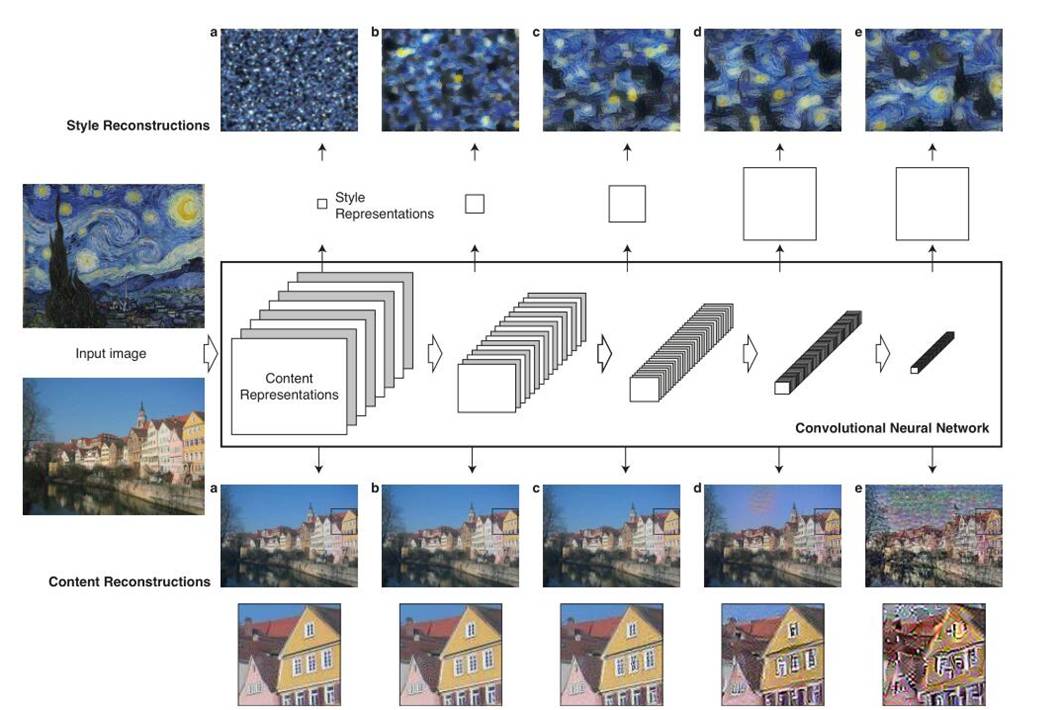
**1.1为什么说风格迁移任务不同于物体识别呢？**

**VGGNet**本意是输入图像，提取特征，并输出图像类别。图像风格迁移正好与其相反，输入的是特征，输出对应这种特征的图片，如下图所示：



具体来说，风格迁移使用卷积层的中间特征**还原**出对应这种特征的原始图像。

1. **技术介绍：**



**2.1通过模型提取风格特征和内容特征：**

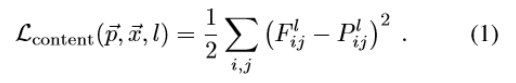
最左侧的两张图片（input image）一张是作为内容输入，一张是作为风格输入，分别经过VGG19的5个block，由浅及深可以看出，得到的特征图（feature map）的高和宽逐渐减小，但是深度是逐渐加大，我们可以看到每个block提取到的特征，对于内容图片特征的提取在很大程度上是保留了原图的信息，但是对于风格图片来说，基本上看不出原图的样貌，而是可以粗略的认为提取到了风格。

**2.2内容表示**

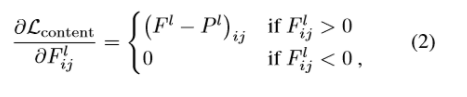
给定一张输入图片, 卷积神经网络每层使用滤波[filter]对其进行编码,  为每个特征图的; 在第  个卷积层中,由个不同的 filter产生个大小为 的特征图,所以第 层中的响应可以存储在一个矩阵中,其中表示第层的第个滤波的位置处的激活值.

**核心: 在一个带有白噪声的图像上执行梯度下降算法,寻找可以匹配原始图像特征响应的另外一个图像,更新原始图像上的噪声图像点, 是输入,而不是权重w.**

令和分别表示原始图像和生成的图像, 和分别为它们对应的第 层的特征表示;将这两个特征表示之间的平方差损失定义为



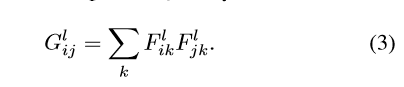
损失函数的偏导数对应的 层的激活函数为



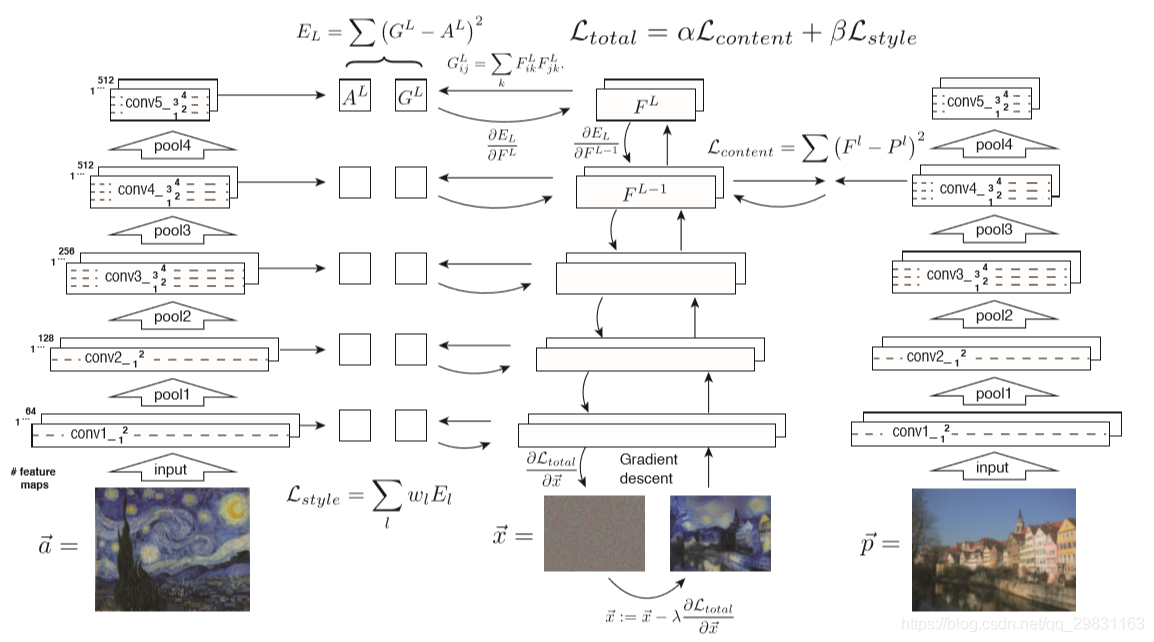
在提取content 特征时,不同卷积层的表征内容不一样,卷积神经网络在物体识别上训练，它们可以生成一个对图像的表示，在网络层级的处理过程中使得物体信息越来越明显。因此，沿着网络层级处理过程，输入图像会发生转变，对图像的真正内容越来越敏感，但会对精细的外观变得相对不变。网络的高层表示一般是关于输入图像的物体和布局信息,但不会限制重构过程中的确切像素值,底层特征一般表达图像的像素信息.将网络中高层的特征响应结果作为内容表示。

**2.2风格表示**

为了获得输入图像的风格表示，用**特征空间**获得纹理信息，它由不同滤波器响应结果的相关关系组成，其中期望值从特征地图空间上取值。特征关系用克莱姆矩阵表示：[对两个特征图求内积.为特征图的第个元素]



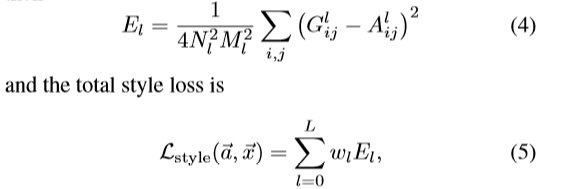
(注: Gram矩阵可表达图像的纹理特征,和位置无关.)



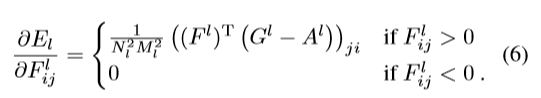
(图像风格迁移算法流程图)

(图注:上图中右上方的公式就是网络训练的总损失,包含content 和style两部分的损失,由人工设定的系数和控制权重, 图左下方的为风格图像,为内容图像.图的下方有带随机白噪声图像的更新公式,每次迭代的该变量等于总loss对求导再乘以步长.)

令和分别表示原始图像和生成的图像，和分别表示第层的风格表示。第层相对于总损失的贡献是：



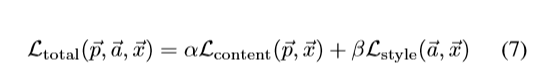
是每层对总损失函数的贡献权重因子。  
注: 内容(Content)的等于的层的和的onv4层的之间的平方差损失计算所得,参考等式(1).  
第层的损失对第层的激活函数的偏导数如下：



的梯度对于像素值的梯度可以用标准误差反向传播快速计算出来.

**2.2风格迁移**

为了将艺术照的风格转移到照片上，我们同步匹配的内容表示和的风格表示,合成一个新图像.需要最小化的损失函数如下:



### 三、实验方案

我们以预训练的模型为原型，在此基础上构造出适用于风格迁移的模型。

**1、获取预训练模型**

我们使用Tensorflow Core的API获取预训练好的VGG19模型。该模型是在imagenet数据集上预训练的模型，可用于提取图像特征，作为风格迁移模型的基础。

**2、模型构建**

由于网络的全连接层（、）和层与风格迁移无关，本次项目中不会使用到这些训练层。

我们选取VGG中选择一些特定的层作为内容特征层和风格特征层并赋予这些层在计算损失时的权重。将特征层从中提取出来，并创建层与输出之间的映射，然后初始化网络。

**3、模型训练**

将噪声图片输入到神经网络中，计算损失，以此作为反馈调整噪声图片，继续计算损失，不断对其迭代，直到指定的迭代次数。

首先将内容图片和风格图片输入到网络中计算其在某个层上的输出值用以计算损失。

内容损失的计算：内容图片在指定层上提取出的特征矩阵，与噪声图片在对应层上的特征矩阵的差值的L2范数。即求两两之间的像素差值的平方。

内容损失函数定义：

其中，是噪声图片的特征矩阵，是内容图片的特征矩阵。是的长宽，是信道数。

风格损失的计算：风格图像和噪音图像特征矩阵的格莱姆矩阵的差值的L2范数。

风格损失函数定义：

其中是特征矩阵的长宽，是特征矩阵的信道数。为噪音图像特征的Gram矩阵，为风格图片特征的GRAM矩阵。

损失函数定义为内容损失和风格损失的加权和：

### 四、目前进展

1. **下载并安装了tensorflow、Vgg等软件包，项目环境搭建完成。**

采用版本如下：

Tensorflow：2.3.1；下载网址：<https://tensorflow.google.cn/install>；

Numpy：1.18.5；参考查阅网址：

<https://numpy.org/doc/stable/reference/generated/numpy.loadtxt.html>；

Tqdm：4.54.1；

Python：3.7

1. **阅读并分析了相关文献，做好相关笔记。**

1. **参考了github上面的部分源码，搭建出自己项目的框架，并不断修改完善代码。**

参考网址：<https://github.com/AaronJny/nerual_style_change#readme>；

主要函数：

模型损失函数loss(sess, model)、

创建格莱姆矩阵的函数gram(x, size, seep)、

创建模型函数train()。

1. **项目基本可以运行，我们测试了几张图片，效果如下：**

原始图像：

****

风格图像：

****

生成图像：（训练20后）

****

1. **分析运行结果，并开始着手继续修改和测试。**

分析：使用的测试图片在自己的计算机跑了半小时左右，得到了一系列的风格迁移后的图片，基本上达到了预期效果。但是仍有不足，对于复杂一点的图片的风格迁移仍有待改进，因此，我们小组准备对模型进行微调并测试，以完善本项目。

### 五、待解决的问题

**1.基于传统图像风格迁移的人工智能算法：**

VGG网络是用来特征提取的，提取出来的这些特征可以算出图片之间的感知损失。损失函数梯度下降修正的东西，是**合成图像G（初始的白噪声图）中的像素**，使这张图片在风格和内容上的感知损失最小。其核心在于：逐步修正合成图像G的像素值，使之损失变到最小，得到的就是风格迁移后的图像。可想而知，其迭代修正的速度是很慢的，不断计算损失再迭代，以至于得到最接近输入风格的迁移图像，小组成员进行模型训练也需要漫长的等待时间。因此我们可以往**快速风格迁移**的方向去思考并尝试，但是需要大量的样本集去构建图像生成网络，现阶段我们也只能够使用他人训练好的模型（TensorFlow Hub），当然电脑的配置也是需要一定的要求的。

**2. 基于感知损失函数的图像转换网络：**

基于感知损失函数的图像转换网络不一定是最好的，在也有在感知损失的基础上，加入GAN（生成对抗网络）来生成更加效果更好的图像。这方面的改进也是一个学习的方向。

**3．训练环境版本和代码问题：**

3.1. Tensorflow、Keras版本问题，部分内置库（模块）无法导入或者下载等等。（部分组员未解决）

3.2. 运行时出现的异常，包括变量异常和数据异常。

3.3. 部分算法代码的实现出现严重错误。

### 六、参考文献

[1]《Image Style Transfer Using Convolutional Neural Networks》Leon A. Gatys

使用卷积神经网络的图像风格转移基础理论

[2]《Perceptual Losses for Real-Time Style Transfer and Super-Resolution》

Justin JohnsonEmail authorAlexandre AlahiLi Fei-Fei ECCV 2016: Computer Vision – ECCV 2016 pp 694-711

快速风格迁移，训练和应用分离，可以快速应用

[3]《Deep Photo Style Transfer》***Fujun Luan, Sylvain Paris, Eli Shechtman, Kavita Bala***; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4990-4998

这篇文章的工作集中于：

第一，风格迁移的范围扩展到转移照片的风格，抑制图片的扭曲，把transfer的过程变成颜色空间的局部仿射变换。第二：通过语义分割来解决风格溢出的问题。

[4]《A Neural Algorithm of Artistic Style》Leon A. Gatys, Alexander S. Ecker, Matthias Bethge

利用协方差矩阵可以用来反映不同kernel获取的feature是否存在相关关系，Gram作为一个特殊形式协方差矩阵，通过不减去均值来降低计算开销。

[5] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[6] Salimans T, Kingma D P. Weight normalization: A simple reparameterization to accelerate training of deep neural networks[C]//Advances in Neural Information Processing Systems. 2016: 901-909.