**集美大学计算机工程学院**

**计算机视觉课程论文**

**题 目** 基于VGG-19模型的图像风格迁移方法及系统

**专 业：**  信息管理与信息系统 **班 级：**  信管1611

**学生姓名：**  邱 蕾 **学 号：** 201621124002

2019年7月12日

**基于VGG-19模型的图像风格迁移方法及系统**

[摘要] 图像风格转换是最近新兴起的一种基于深度学习的技术，卷积神经网络所带来的对图像特征的高层特征的抽取使得风格和内容的分离成为了可能。本文是基于VGG-19模型，使用已经训练好的VGG-19模型的各层的权重信息， 通过加载VGG参数提取出图像的特征图谱即feature map，计算content loss，计算style loss，计算总体loss，进行优化函数，构建模型图，训练模型，通过训练好的模型，最终完成内容图像和风格图像的风格迁移。系统实现指定一幅输入图像作为基础图像，也被称做内容图像；同时指定另一幅或多幅图像作为希望得到的图像风格，算法在保证内容图像的结构的同时，将图像风格进行转换，使得最终输出的合成图像呈现出输入图像内容和风格的结合。

[关键词] 图像风格迁移； VGG； 深度学习

# 目录

[摘要 II](#_Toc13779911)

[第1章 引言 1](#_Toc13779913)

[1.1 背景介绍与研究意义 1](#_Toc13779914)

[1.2 研究内容与目标 2](#_Toc13779915)

[第2章 VGG网络原理 3](#_Toc13779916)

[2.1 深度学习与卷积神经网络简介 3](#_Toc13779917)

[2.2 卷积神经网络原理 3](#_Toc13779918)

[2.2.1 输入层 4](#_Toc13779919)

[2.2.2 卷积层 4](#_Toc13779920)

[2.2.3 激活函数层 6](#_Toc13779921)

[2.2.4 池化层 7](#_Toc13779922)

[2.2.5 全连接层 7](#_Toc13779923)

[2.3 VGG网络模型 8](#_Toc13779924)

[2.3.1 VGG网络模型简介 8](#_Toc13779925)

[2.3.2 VGG网络模型结构 8](#_Toc13779926)

[2.3.1 VGG网络模型特点 10](#_Toc13779927)

[2.4 小结 11](#_Toc13779928)

[第3章 图像风格迁移模型原理 12](#_Toc13779929)

[3.1 图像风格迁移模型 12](#_Toc13779930)

[3.1.1 图像风格迁移模型思路 12](#_Toc13779931)

[3.1.2 图像风格迁移模型结构 12](#_Toc13779932)

[3.2 感知损失函数 13](#_Toc13779933)

[3.2.1 内容损失（content loss） 13](#_Toc13779934)

[3.2.2 风格损失（style loss） 13](#_Toc13779935)

[3.2.3 总损失total loss 14](#_Toc13779936)

[3.3 小结 14](#_Toc13779937)

[第4章 系统设计与实验结果分析 15](#_Toc13779938)

[4.1系统框架设计 15](#_Toc13779939)

[4.2前端界面设计 15](#_Toc13779940)

[4.3后端逻辑设计 16](#_Toc13779941)

[4.4系统实现 16](#_Toc13779942)

[4.4.1 环境配置 16](#_Toc13779943)

[4.4.2 前端界面实现 16](#_Toc13779944)

[4.4.3功能逻辑实现 19](#_Toc13779945)

[4.5实验结果分析 19](#_Toc13779946)

[4.5.1 实验设置 19](#_Toc13779947)

[4.5.2 数据集 19](#_Toc13779948)

[4.5.3 实验结果分析 20](#_Toc13779949)

[4.6 小结 23](#_Toc13779950)

[结 论 1](#_Toc13779951)

[参考文献 2](#_Toc13779952)

# 第1章 引言

## 1.1 背景介绍与研究意义

近年来，由深度学习所引领的人工智能技术浪潮，开始越来越广泛地应用到社会各个领域。这其中，人工智能与艺术的交叉碰撞，不仅在相关的技术领域和艺术领域引起了高度关注。以相关技术为基础而开发的各种图像处理软件和滤镜应用更是一推出就立刻吸引了海量用户，风头一时无两。在这各种神奇的背后，最核心的就是基于深度学习的图像风格迁移(style transfer)。

2014年，牛津大学计算机视觉组（Visual Geometry Group）和Google DeepMind公司的研究员一起研发出了新的深度卷积神经网络：VGGNet，并取得了ILSVRC2014比赛分类项目的第二名（第一名是GoogLeNet，也是同年提出的) .

在神经网络之前，图像风格迁移的程序有一个共同的思路，分析某一种风格的图像，给那种图像建立一个数学或者统计模型，再改变要做迁移的图像让它能更好的符合建立的模型，但这个方法有很大的缺点，就是一个程序基本只能做某一个风格或某一个场景，由此基于传统风格迁移研究应用有着非常的的困难。

在2015年前所有关于图像纹理的论文都是手动建模的，其中一个最重要的思想：纹理可以用图像局部特征的统计模型来描述。

另一个重要基础就是显卡（GPU），显卡最大功能是处理和显示图像，不同于CPU的是，CPU早期是单线程的，也就是一次只能处理一个任务，GPU可以一次同时处理多个任务，虽然单个任务的处理能力和速度差于CPU，显卡在过去20年内也变得越来越好，显卡计算能力的爆炸性增长直接导致了被放置了十几年的神经网络的复活和深度学习的崛起。

基于神经网络的风格迁移算法 “A Neural Algorithm of Artistic Style” 最早由 Gatys 等人在 2015 年提出，随后发表在 CVPR 2016 上。斯坦福大学的 Justin Johnson（cs231n 课程的主讲人之一）给出了 Torch 实现 neural-style。除此之外，这篇文章的作者另外还建立了一个在线艺术风格迁移的网站，deepart.io。

## 1.2 研究内容与目标

Style Transfer属于迁移学习的一种，就是通过将一个图片的风格（style）运用到另一个图片上，类似现在各种P图软件中滤镜的功能。艺术风格迁移的核心思想就是，可以从一副图像中提取出“风格 style”（比如梵高的夜空风格）以及“内容 content”（比如你能在画中看出河边有匹马）。你可以告诉机器，把 A 用 B 的风格再画一遍。

本课题研究内容是图像风格迁移算法，可以解决如下问题：指定一幅输入图像作为基础图像，也被称做内容图像；同时指定另一幅或多幅图像作为希望得到的图像风格，算法在保证内容图像的结构的同时，将图像风格进行转换，使得最终输出的合成图像呈现出输入图像内容和风格的完美结合。其中，图像的风格可以是某一艺术家的作品、也可以是由个人拍摄的图像所呈现出来的风格。研究目标主要是基于VGG-19网络建立图像风格迁移模型，实现图像风格变换。

# 第2章 VGG网络原理

## 2.1 深度学习与卷积神经网络简介

深度学习[1-6] (Deep Learning) 的概念由Hinton等人于2006年提出。基于深信度网(DBN)提出非监督贪心逐层训练算法，为解决深层结构相关的优化难题带来希望，随后提出多层自动编码器深层结构。此外Lecun等人提出的卷积神经网络是第一个真正多层结构学习算法，它利用空间相对关系减少参数数目以提高训练性能。

卷积神经网络(convolutional neural networks, CNN)由纽约大学的Yann LeCun于1998年提出。CNN本质上是一个多层感知机，其成功的原因关键在于它所采用的局部连接和共享权值的方式，一方面减少了的权值的数量使得网络易于优化，另一方面降低了过拟合的风险。CNN是神经网络中的一种，它的权值共享网络结构使之更类似于生物神经网络，降低了网络模型的复杂度，减少了权值的数量。该优点在网络的输入是多维图像时表现的更为明显，使图像可以直接作为网络的输入，避免了传统识别算法中复杂的特征提取和数据重建过程。在二维图像处理上有众多优势，如网络能自行抽取图像特征包括颜色、纹理、形状及图像的拓扑结构；在处理二维图像问题上，特别是识别位移、缩放及其它形式扭曲不变性的应用上具有良好的鲁棒性和运算效率等。

## 2.2 卷积神经网络原理

一个卷积神经网络由若干卷积层、Pooling层、全连接层组成。你可以构建各种不同的卷积神经网络，它的常用架构模式为：

INPUT -> [[CONV]\*N -> POOL?]\*M -> [FC]\*K

也就是N个卷积层叠加，然后(可选)叠加一个Pooling层，重复这个结构M次，最后叠加K个全连接层。

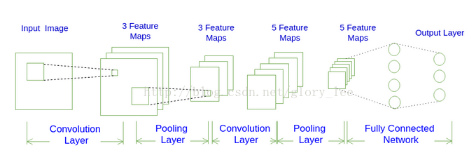


图 1 卷积神经网络结构图

### 2.2.1 输入层

卷积神经网络的输入层的输入格式保留了图片本身的结构。对于黑白的的图片，卷积神经网络的输入是一个二维神经元。

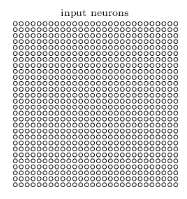


图 2 黑白图片输入

而对于RGB格式的图片，卷积神经网络的输入则是一个三维神经元（RGB中的每一个颜色通道都有一个的矩阵）。

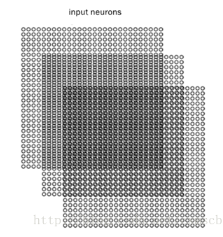


图 3 RGB格式图片输入

### 2.2.2 卷积层

一个感受视野带有一个卷积核，我们将感受视野中的权重矩阵称为卷积核 ；将感受视野对输入的扫描间隔称为步长（stride）；当步长比较大时（stride>1），为了扫描到边缘的一些特征，感受视野可能会“出界”，这时需要对边界扩充(pad)，边界扩充可以设为0或 其他值。步长 和 边界扩充值的大小由用户来定义。

卷积核的大小由用户来定义，即定义的感受视野的大小；卷积核的权重矩阵的值，便是卷积神经网络的参数，为了有一个偏移项 ，卷积核可附带一个偏移项，它们的初值可以随机来生成，可通过训练进行变化。

将通过一个带有卷积核的感受视野 扫描生成的下一层神经元矩阵称为一feature map (特征映射图)，如下图的右边便是一个 feature map：

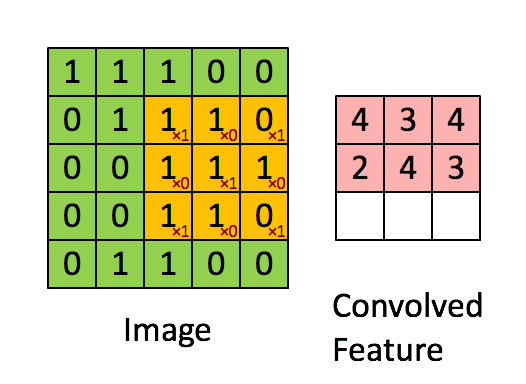


图 4 输入为5\*5的图像，3\*3的filter，步长=1卷积操作

每个卷积层可以有多个filter。每个filter和原始图像进行卷积后，都可以得到一个Feature Map。因此，卷积后Feature Map的深度(个数)和卷积层的filter个数是相同的。

下图显示了包含两个filter的卷积层的计算。我们可以看到7\*7\*3输入，经过两个3\*3\*3filter的卷积(步幅为2)，得到了3\*3\*2的输出。Bias0是1，也就是在输入元素的周围补了一圈0。

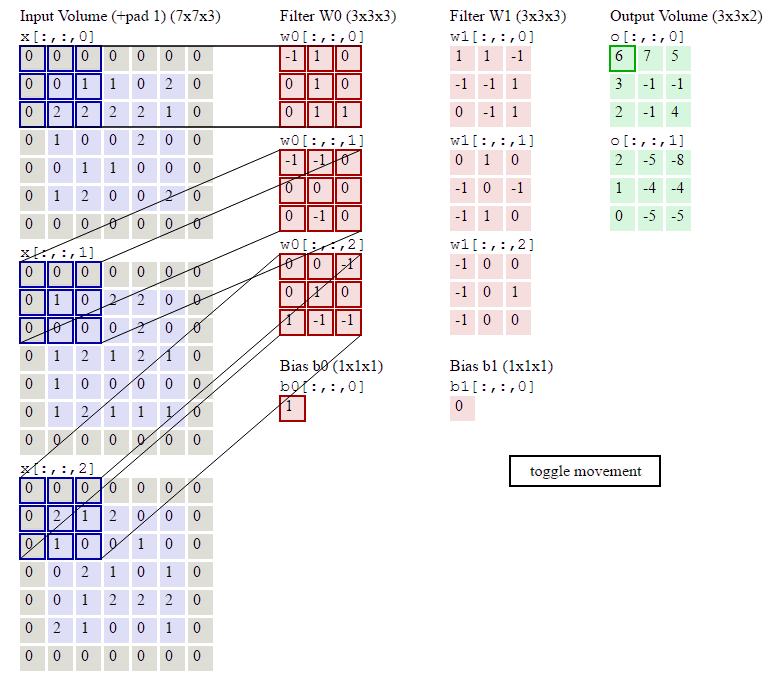


图 5 输入3\*7\*7 filter=2\*3\*3 biasb0=1 b1=0的卷积操作

卷积计算公式如下：



其中D是深度；F是filter的大小(宽度或高度，两者相同)；Wd,m,n表示filter的第层第m行第n列权重；ad,I,j表示图像的第d层第i行第j列像素；

### 2.2.3 激活函数层

激励层主要对卷积层的输出进行一个非线性映射，因为卷积层的计算还是一种线性计算。最近几年卷积神经网络中，激活函数往往不选择sigmoid或tanh函数，而是选择relu函数。使用的激励函数一般为ReLu函数：



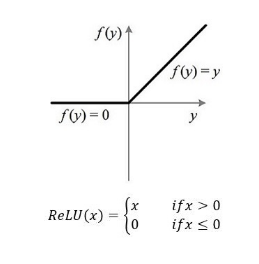


图 6 relu激活函数

卷积层和激励层通常合并在一起称为“卷积层”。

### 2.2.4 池化层

Pooling层主要的作用是下采样，通过去掉Feature Map中不重要的样本，进一步减少参数数量。Pooling的方法很多，最常用的是Max Pooling。Max Pooling实际上就是在n\*n的样本中取最大值，作为采样后的样本值。下图是2\*2 max pooling：



图 7 2\*2maxpooling

### 2.2.5 全连接层

全连接层是连接所有的特征，将输出值送给分类器（如softmax分类器）。

右图池化层输出了20个12\*12的图像然后通过了一个全连接层变成了1\*100的向量，就是用100个20\*12\*12（20指的是吃化层的厚度）的卷积核卷积出来的。

****

图 8 全连接层

## 2.3 VGG网络模型

### 2.3.1 VGG网络模型简介

VGG是牛津大学计算机视觉组（VisualGeometry Group）和GoogleDeepMind公司的研究员一起研发的的深度卷积神经网络。VGG探索了卷积神经网络的深度与其性能之间的关系，通过反复堆叠3\*3的小型卷积核和2\*2的最大池化层，VGG成功地构筑了16~19层深的卷积神经网络。VGG相比之前state-of-the-art的网络结构，错误率大幅下降，并取得了ILSVRC 2014比赛分类项目的第2名和定位项目的第1名。同时VGG的拓展性很强，迁移到其他图片数据上的泛化性非常好。VGG的结构非常简洁，整个网络都使用了同样大小的卷积核尺寸（3\*3）和最大池化尺寸（2\*2）。到目前为止，VGG依然经常被用来提取图像特征。VGG训练后的模型参数在其官方网站上开源了，可用来在特定的图像分类任务上进行再训练（相当于提供了非常好的初始化权重），因此被用在了很多地方。VGG可以看成是加深版的AlexNet，整个网络由卷积层和全连接层叠加而成，和AlexNet不同的是，VGG中使用的都是小尺寸的卷积核(3×33×3)。

### 2.3.2 VGG网络模型结构

VGG研究卷积网络深度的初衷是想搞清楚卷积网络深度是如何影响大规模图像分类与识别的精度和准确率的，最初是VGG-16号称非常深的卷积网络全称为（GG-Very-Deep-16 CNN），VGG在加深网络层数同时为了避免参数过多，在所有层都采用3x3的小卷积核，卷积层步长被设置为1。VGG的输入被设置为224x244大小的RGB图像，在训练集图像上对所有图像计算RGB均值，然后把图像作为输入传入VGG卷积网络，使用3x3或者1x1的filter，卷积步长被固定1。VGG全连接层有3层，根据卷积层+全连接层总数目的不同可以从VGG11 ～ VGG19，最少的VGG11有8个卷积层与3个全连接层，最多的VGG19有16个卷积层+3个全连接层，此外VGG网络并不是在每个卷积层后面跟上一个池化层，还是总数5个池化层，分布在不同的卷积层之下，下图是VGG11 ～GVV19的结构图：

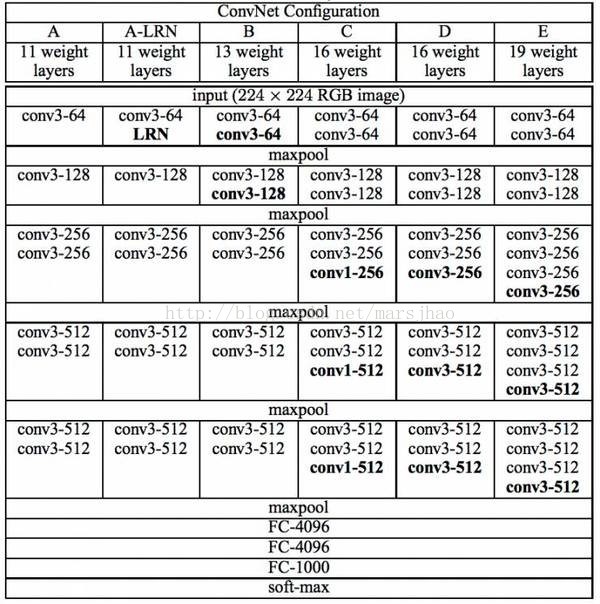


图 9 VGG11 ～VGG19的结构图

* conv表示卷积层
* FC表示全连接层
* conv3表示卷积层使用3x3 filters
* conv3-64表示 深度64
* max pool表示最大池化

https://img-blog.csdn.net/20170609113147653?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvbWFyc2poYW8=/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center

图 10 VGG11 ～ VGG19参数总数

以上是整个模型的参数总量，单位是millions，参数来自卷积层和全连接层

### 2.3.1 VGG网络模型特点

1. 小卷积核和连续的卷积层

VGG中使用的都是3×33×3卷积核，并且使用了连续多个卷积层。这样做的好处：

使用连续的的多个小卷积核(3×3)，来代替一个大的卷积核（例如(5×5)。

使用小的卷积核的问题是，其感受野必然变小。所以，VGG中就使用连续的3×3卷积核，来增大感受野。VGG认为2个连续的3×3卷积核能够替代一个5×5卷积核，三个连续的3×3能够代替一个7×7。

小卷积核的参数较少。3个3×3的卷积核参数为3×3×=27，而一个7×7的卷积核参数为7×7=49。

由于每个卷积层都有一个非线性的激活函数，多个卷积层增加了非线性映射。

1. 小池化核，使用的是2×22×2。
2. 通道数更多，特征度更宽。

每个通道代表着一个FeatureMap，更多的通道数表示更丰富的图像特征。VGG网络第一层的通道数为64，后面每层都进行了翻倍，最多到512个通道，通道数的增加，使得更多的信息可以被提取出来。

1. 层数更深。

使用连续的小卷积核代替大的卷积核，网络的深度更深，并且对边缘进行填充，卷积的过程并不会降低图像尺寸。仅使用小的池化单元，降低图像的尺寸。

1. 全连接转卷积（测试阶段）

这也是VGG的一个特点，在网络测试阶段将训练阶段的三个全连接替换为三个卷积，使得测试得到的全卷积网络因为没有全连接的限制，因而可以接收任意宽或高为的输入，这在测试阶段很重要。

如本节第一个图所示，输入图像是224x224x3，如果后面三个层都是全连接，那么在测试阶段就只能将测试的图像全部都要缩放大小到224x224x3，才能符合后面全连接层的输入数量要求，这样就不便于测试工作的开展。

全连接转卷积的替换过程如下：

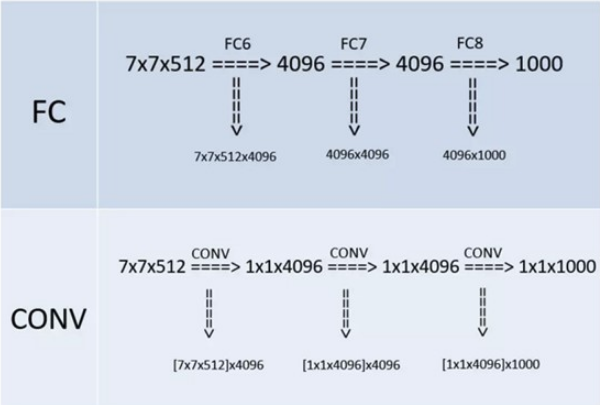


图 11 全连接转卷积

## 2.4 小结

通过上文对卷积神经网络基础知识和VGG网络的深入理解和研究。可以看出深度学习在计算机视觉领域有突出表现，可以根据不同任务用多种层构造自己的网络模型。不同激活函数有不同效果，根据需求选择。VGG-16和VGG-19，使用很小的卷积核(3×33×3)构建各种深度的卷积神经网络结构，并对这些网络结构进行了评估，最终证明16-19层的网络深度，能够取得较好的识别精度，常常被用来提取图像特征。接下来的章节将详细介绍基于VGG网络的图像风格迁移模型。

# 第3章 图像风格迁移模型原理

## 3.1 图像风格迁移模型

### 3.1.1 图像风格迁移模型思路

1. 初始化合成图（synthetic image）；
2. 从Content Image中提取内容信息，最小化合成图内容和Content Image内容的差异；
3. 从Style Image中提取风格信息，最小化合成图风格和Style Image风格的差异；
4. 更新合成图的信息，得到最终synthetic image。

### 3.1.2 图像风格迁移模型结构

该网络用了VGG19网络（16 convolutional and 5 pooling layers），a为风格图像，x为生成图片，p为内容图片。VGG19网络是训练好的，用来提取特征，并计算网络的损失函数，需要训练的是x，他是随机初始化，通过训练生成想要的图片。

风格损失在每一层都进行计算，最后分别乘上权重因子 ω 再加起来。而内容损失只在conv\_4进行。



图 12 图像风格迁移模型结构图

## 3.2 感知损失函数

### 3.2.1 内容损失（content loss）

对于内容损失，只取conv4\_2层的特征，计算内容图片特征和噪声图片特征之间的欧式距离，公式为内容图片的损失计算公式：

其中 p代表content image， x代表我们的合成图片。 l 代表content image在第l 层的feature representations，同样的，Fl代表合成图片在第l 层的feature representations。在不考虑batch size的情况下， ，其中Nl代表第l层的feature maps数量， Ml代表第l层的feature maps的

由于上面这个content loss公式很难收敛，本文采用替代1/2。

### 3.2.2 风格损失（style loss）

1. Gram 矩阵计算公式：

Gram矩阵：feature之间的偏心协方差矩阵（即没有减去均值的协方差矩阵）

Gram矩阵用于度量各个维度自己的特性以及各个维度之间的关系。内积之后得到的多尺度矩阵中，对角线元素提供了不同特征图各自的信息，其余元素提供了不同特征图之间的相关信息。这样一个矩阵，既能体现出有哪些特征，又能体现出不同特征间的紧密程度。

1. 第l层Gram矩阵的平方差计算公式（即第l层的style loss）

在上述公式中， Gl代表style image在第l层feature maps的Gram Matrix（格雷姆矩阵）， A代表合成图片在第l 层feature maps的Gram Matrix。 Nl代表第 l 层feature maps的数量，Ml等于feature maps的。Gram Matrix可以获取到当前l层中不同feature maps的feature correlations。

1. 总的style loss计算公式：

上述公式中 a代表style image， x代表合成图像，wl代表第l层的权重

### 3.2.3 总损失total loss

总损失计算公式：

对合成图片的content loss和style loss进行加权求和，一般 =0.0001或 0.001 ，可以自己调整。

当a很小时，内容损失相对大，风格损失相对较小，所以生成的图片更凸显风格；当a较大时，内容损失相对小，风格损失相对较大，生成的图片更凸显内容。

## 3.3 小结

通过对图像风格迁移模型算法思路和结构的研究理解，可以看出要通过一个模型分别捕捉到content image的内容信息和style image的风格信息，在处理图像问题中，CNN（卷积神经网络）被证明是一种提取图像特征的有效模型。

本文将VGG19模型作为预训练模型，将其中池化层maxpooling改成averpooling，风格损失在每一层都进行计算，最后分别乘上权重因子 ω 再加起来。而内容损失只在conv\_4进行，通过训练得到我们想要的图像风格迁移模型。

# 第4章 系统设计与实验结果分析

## 4.1系统框架设计

系统分为前端界面和后端部分，前端部分为web页面，使用Bootstrap 框架是基于 HTML、CSS、JAVASCRIPT 的，它简洁灵活，使得 Web 开发更加快捷。Flask是一个使用 Python 编写的轻量级 Web 应用框架。其 WSGI 工具箱采用 Werkzeug ，模板引擎则使用 Jinja2 。Flask使用 BSD 授权。Flask也被称为 "microframework" ，因为它使用简单的核心，用 extension 增加其他功能。Flask没有默认使用的数据库、窗体验证工具。后端算法部分基于tensorflow进行数据训练，前端界面通过选择图像文件和风格选择，通过后端进行调用，实现图片的风格转换。

## 4.2前端界面设计

本次实验前端界面分为登入页面和主页面，主页面分为风格图片展示界面和图像风格转换操作界面。前端主要使用了bootstrap框架， Bootstrap，来自 Twitter，是目前最受欢迎的前端框架。Bootstrap 是基于 HTML、CSS、JAVASCRIPT 的，它简洁灵活，使得 Web 开发更加快捷。Bootstrap 包含了一个响应式的、移动设备优先的、不固定的网格系统，可以随着设备或视口大小的增加而适当地扩展到 12 列。它包含了用于简单的布局选项的预定义类，也包含了用于生成更多语义布局的功能强大的混合类。Bootstrap 轮播（Carousel）插件是一种灵活的响应式的向站点添加滑块的方式。除此之外，内容也是足够灵活的，可以是图像、内嵌框架、视频或者其他您想要放置的任何类型的内容。

登录界面背景为 点击进入系统即可进入到图像风格转换系统

主界面背景的风格设计为油画风格，页面顶端为导航栏，包括两个界面的进入按钮，为各图片风格展示及图片风格转换选项，第一个页面内含选择图片按钮和不同风格选项菜单，页面左侧内有已经进行不同风格转换的各个图像，采用轮播的方式进行展示，页面右侧为不同风格的图像及名称。第二个页面可选择内容图片和风格图片进行图片风格转换，输出转换后的图像。

## 4.3后端逻辑设计

后端逻辑是数据源读取并预处理和算法调用的核心部分，将处理结果传输给前端界面。是基于Flask是一个使用 Python 编写的轻量级 Web 应用框架。模板引擎则使用 Jinja2 。Flask使用 BSD 授权。Flask也被称为 "microframework" ，因为它使用简单的核心，用 extension 增加其他功能。Flask没有默认使用的数据库、窗体验证工具。后端主要实现从web前端页面获取图片信息，再进行模型训练计算，将计算的结果显示到前端页面中。

首先将内容图像加入白噪声得到图G，然后G与内容内容图像在内容特征上的差别，记为J(C,G)；计算G与风格图像在风格特征上的差别，记为J(S,G)。通过计算这两种损失（根据先前给出的损失函数），然后用梯度下降法修正G的像素值，使得它越来越接近最后的风格迁移图像G。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。

## 4.4系统实现

### 4.4.1 环境配置

开发环境和训练环境不同

表4-4开发环境

|  |  |
| --- | --- |
| 名称 | 型号 |
| 系统 | Window10 |
| CPU | Intel E3-1225 v6 |
| Python  JetBrains PyCharm | 3.7.1  2018.3.5 |

### 4.4.2 前端界面实现

前端页面使用html5语言和bootstrap框架编写。系统整体风格为油画艺术风格，更好的与我们的图像风格迁移主题相切合。

登入页面的背景为梵高先生的作品星空，点击进入系统即可进入系统主页面。



图 13 登入页面

主界面：下图为风格展示页面，顶端为导航栏，左侧为之前所训练得到的图片，通过轮播的方式将同张图片不同风格展示出来，图片来源主要为集美大学建筑物图像。右侧为我们转换的风格原图片，依次用椭圆展示出来，下端为风格名称，这样排列更加简洁美观。

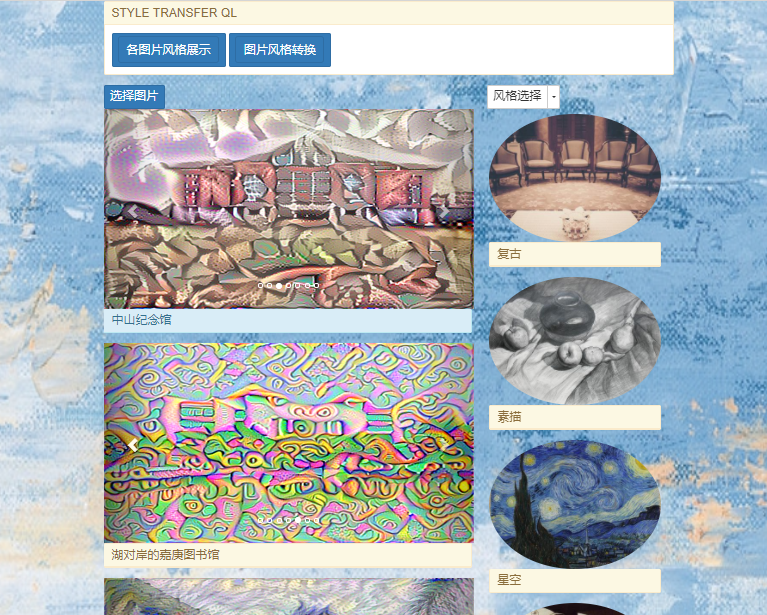


图 14 各风格展示页面

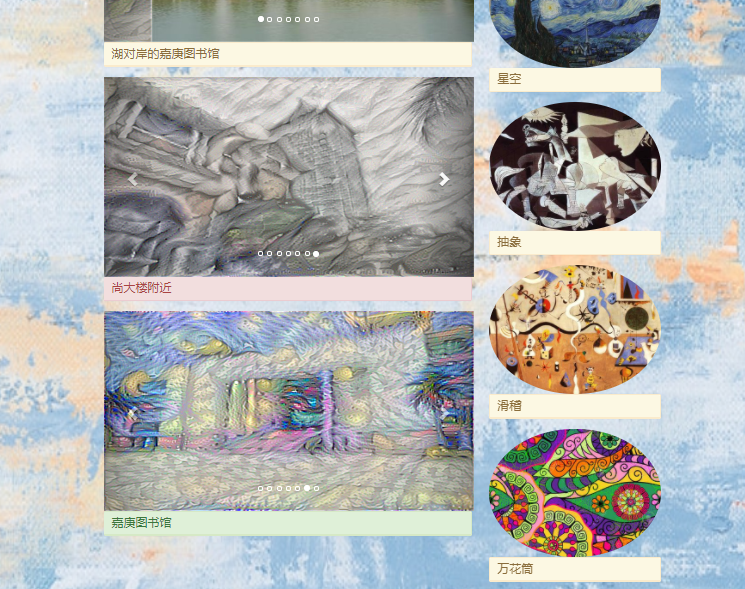


图 15 风格展示页面

图像风格变换页面也包括和前面一样的导航栏，左侧为选择内容图片选择项，选择后显示在页面，右侧为风格图片选择项，进行transfer后，可以得到风格转换后的图像，显示于两者底部。

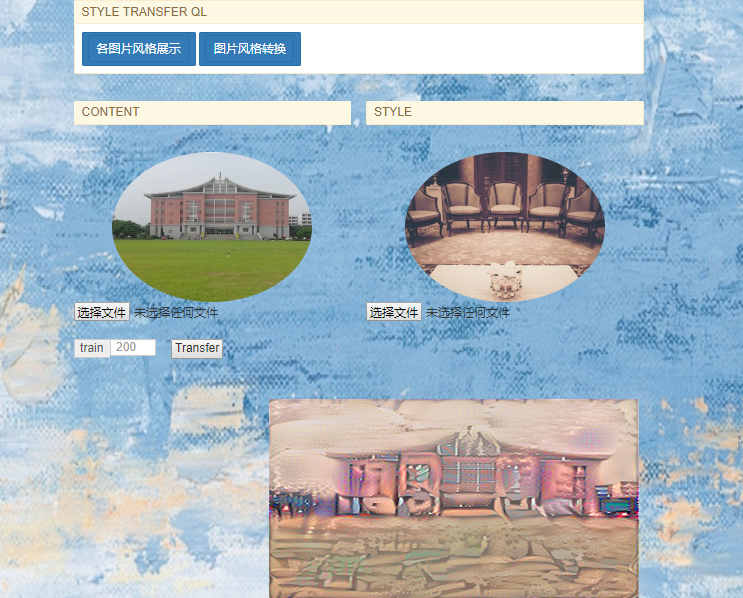


图 16 图像风格转换

### 4.4.3功能逻辑实现

本次课设功能主要实现图像风格变换和展示已经训练好的图片。

图像风格变换模块通过前端读取两张图片，使用网络模型，通过不断迭代合成最终效果的风格转换图像。

不同风格展示模块调取已经训练好的数据，通过轮播方式展示出来。

4.5实验结果分析

### 4.5.1 实验设置

本次实验训练模型环境为实验室提供的服务器上

表4-5 开发环境

|  |  |
| --- | --- |
| 名称 | 型号 |
| 系统 | Linux |
| CPU | Intel E3-1225 v6 |
| GPU | NVIDIA GTX1080ti,11G |
| Tensorflow | 1.0 |
| Python | 3.6.5 |

### 4.5.2 数据集

实验数据由内容图像集和风格图像集组成，其中内容图像为集美大学中山纪念馆，嘉庚图书馆，尚大楼等多处图片，风格图像集为不同风格，如素描风格，抽象风格，复古风格，星空风格，万花筒风格，滑稽风格等。



图 17 内容图像

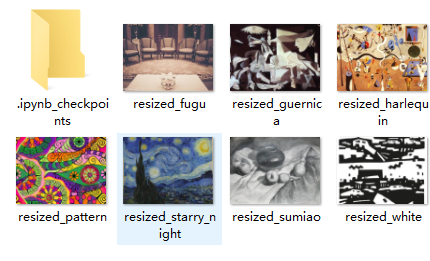


图 18 风格图像

### 4.5.3 实验结果分析

1、各损失曲线图

1. contents loss

下图为内容图片在200次迭代中的contents loss变化曲线图，content loss为经过第五层卷积层的损失值。对于内容损失，只取conv4\_2层的特征，计算内容图片特征和噪声图片特征之间的欧式距离。在计算风格损失时，5个block提取的特征都用来计算了，而计算内容损失，实际上只用了第四个block提取的特征。这是因为每个block提取到的风格特征都是不一样的，都参与计算可以增加了风格的多样性。而内容图片每个block提取到特征相差不大，所以只取一个就好。

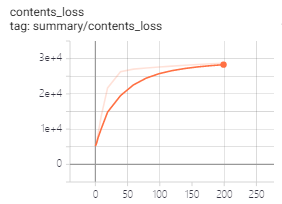


图 19 contents loss

1. style loss

下图为风格图片在200次迭代中的style loss变化曲线图，风格图像每次都经过五层卷积，其损失值为每层损失的加权平均值。

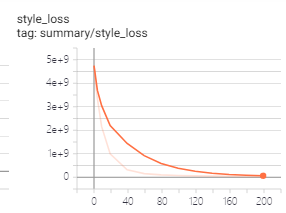


图 20 style loss

1. total loss

下图在200次迭代中的total loss变化曲线图，为style loss和content loss的加权损失值和。 实验中我们取a/b为0.001.当a很小时，内容损失相对大，风格损失相对较小，所以生成的图片更凸显风格；当a较大时，内容损失相对小，风格损失相对较大，生成的图片更凸显内容。

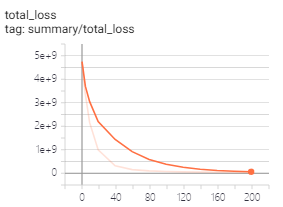


图 21 total loss

2、目标图片迭代过程

下图为尚大楼图片进行素描风格变迁，在迭代过程所得到的目标图像，我们可以看到随着迭代次数的增加，图片的风格变化越加明显，经过200次的迭代我们可以明显的看出素描风格，但同时也能够看出原先的图片的内容。



图 22 风格迭代过程

3、多张图像不同风格图

下图为多张原始图像及多个风格变换后的图像，我们可以看出不同风格的图像实现效果。且可用统一个模型，比之前

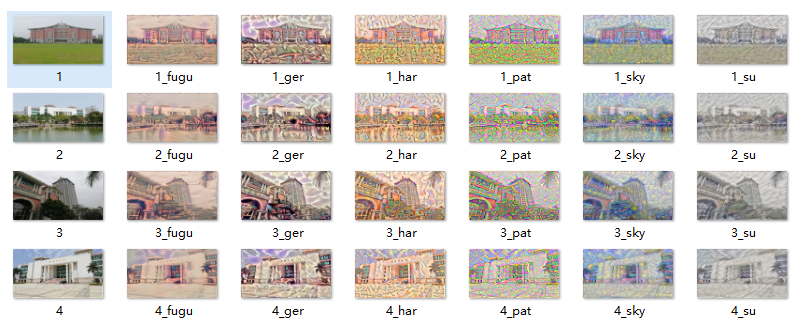


图 23 原始图像及多个风格变换后的图像

## 4.6 小结

根据上面的实验结果，我们可以看出我们所训练的模型对于图像风格变迁有着很好的效果，迭代200次所获得的图片，能够明显的看出style图片的风格，而且计算时长较短，我们能够较好完成图像风格迁移。

这边我们对内容图像进行白噪声处理，得到带有白噪声的内容图像，初始化合成图。

在实验中我们采用gram矩阵来衡量图片风格差异性，Gram Matrices抽取到的是：在图片的同一位置下，不同特征之间的组合。Gram矩阵实际上是矩阵的内积运算，在风格迁移算法中，其计算的是feature map之间的偏心协方差，在feature map 包含着图像的特征，每个数字表示特征的强度，Gram矩阵代表着特征之间的相关性，因此可以通过Gram矩阵衡量风格的差异性。

# 结 论

本文从基于深度卷积网络VGG模型建立图像风格迁移模型，实现了不同的图片的风格迁移，即保留原图的内容，将其风格改变成想要迁移的风格图像的风格，最后通过web前端界面可视化将不同图片风格展示出来和实现图片转换风格功能。

深度学习技术在计算机视觉领域尤为明显，图像分类、识别、定位、超分辨率、转换、迁移、描述等等都已经可以使用深度学习技术实现。其背后的技术可以一言以蔽之：卷积网络具有超强的图像特征提取能力。

其中，风格迁移算法的成功，其主要基于以下两点：1、两张图像经过预训练好的分类网络，若提取出的高维特征(high−levelhigh−level)之间的欧氏距离越小，则这两张图像内容越相似。2、两张图像经过与训练好的分类网络，若提取出的低维特征(low−levellow−level)在数值上基本相等，则这两张图像越相似，换句话说，两张图像相似等价于二者特征的GramGram矩阵具有较小的弗罗贝尼乌斯范数。

建立图像风格迁移模型，style tranfer通过白噪声初始化(就是高斯分布)一个输出的结果，然后通过优化损失对这个结果进行风格和内容两方面的约束修正。采用VGG网络模型作为预训练模型，加载VGG模型，获取指定层的权重数据，这里重新构造了池化层，将原有的Maxing-pooling换成average-pooling，应用不同loss公式计算content图片和style图片的损失，最终计算出总损失， 使用作为优化函数建立图像风格迁移模型，得出进行风格迁移后的图像。

# 参考文献

1. Leon A. Gatys. A Neural Algorithm of Artistic Style [J]., 2015.
2. Leon A. Gatys. Image Style Transfer Using Convolutional Neural Networks. 2016.CVPR
3. Deng, L.; Yu, D.Deep Learning: Methods and Applications . Foundations and Trends in Signal Processing[J],2014, **7**: 3–4.
4. Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives[J]. IEEE transactions on pattern analysis and machine intelligence, 2013, 35(8): 1798-1828.
5. LeCun Y, Bengio Y, Hinton G. Deep learning[J]. nature, 2015, 521(7553): 436.
6. Schmidhuber J. Deep learning in neural networks: An overview[J]. Neural networks, 2015, 61: 85-117.
7. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
8. Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1-9.
9. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
10. Deng J, Dong W, Socher R, et al. Imagenet: A large-scale hierarchical image database[C]//2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009: 248-255.
11. Neubeck A, Van Gool L. Efficient non-maximum suppression[C]//18th International Conference on Pattern Recognition (ICPR'06). IEEE, 2006, 3: 850-855.
12. De Boer P T, Kroese D P, Mannor S, et al. A tutorial on the cross-entropy method[J]. Annals of operations research, 2005, 134(1): 19-67.