

1、项目概览

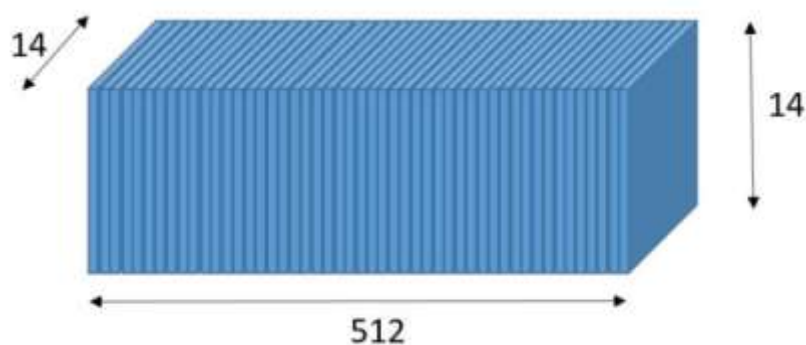


上图左方是 content image，即提供内容的图片，右方是 style image，即提供风格的图片。所要形成的图片如下方所示，怎么样，感觉是不是很高大上，但要实现这个项目，需要做很多准备工作，先来学习一下基本知识吧。



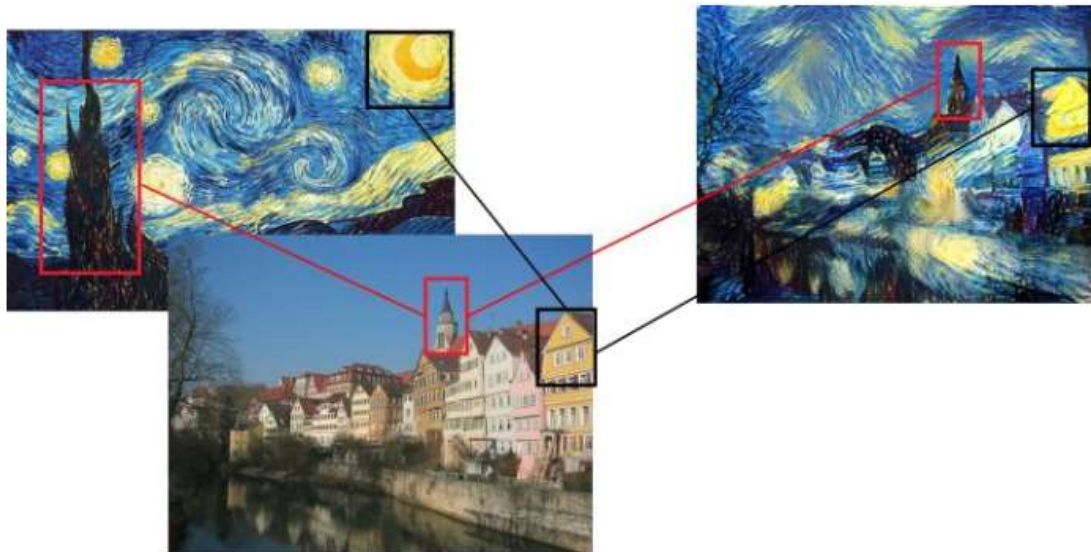
2、项目原理

CNN 对图像处理的本质是逐层抽象表达，浅层提取处理的对象是简单的图像特征（垂直边缘、水平边缘、线条等），而深层处理的是更加抽象具有语义化的信息，越往后，特征越加高级和稳定。比如分类问题中，最后一层的输出就是稳定的具有语义化的输出标签；而在风格迁移例子中，内容上的相似即是深层的特征向量的二范数差异值较小，风格上的相似即是某一层或某几层的格拉姆矩阵（Gram metric）差异值较小。格拉姆矩阵的定义如下：



这是某一层的输出 a ，将其展开为 512×196 的矩阵，这个矩阵自己跟自己相乘，得到了 512×512 的矩阵，这个矩阵称为 Gram 矩阵。Gram 矩阵表达了向量组中任意特征

之间的相关性，两张图的风格表达，即为原图和新图的 Gram 矩阵的二范数。至于为什么它能代表，其实从算法原理上讲，目前我是理解不了的，不过通过可视化的方法感性的认识一下也是容易的：



譬如说，某一个滤波器专门检查尖状物，另一个专门检测黑色，又有两个专门检测圆形和金黄色，而在风格图像上，这两者成对出现的概率比较大，或者说，这样是一种独特的风格，所以两个滤波器的相关性较大，因此在风格转换上，常常需要运用这样的滤波器形成目标风格，那么 Gram 矩阵做这个就再合适不过了。

好了，基本的思路已经清晰了，内容和风格的代价函数定义我们已经了解了，现在已经提供的材料有 content image、style image、target image（噪声图）、loss 函数，还需要的材料有神经网络模型、梯度求解算法，以及优化器。

神经网络模型：这里直接导入 keras 库中的 vgg19，vgg19 原理还未了解。

图片预处理：调用 keras 的 image 库，将三通道图片转化为能够处理的矩阵模型，这里其实还有很多细节需要说明。

图片后处理：形成的图片是无法直接查看的，需要在调用 image 库转化为正常的三通道 RGB 图片，才能供人查看。

初始化目标图片占位符：这里可以直接初始化为噪声图片，也可以用原图作为目标图片，其中的差异我还不了解。最后将三张图片组装成一个四阶 tensor（张量）供模型优化。

损失函数：之前已经了解到内容和风格损失函数的定义了，这里不再赘述代码内容了。

(自然图片)正则项：这防止过拟合，感性上认识，是防止图片锐化，使得目标图片自然润滑。

梯度值：这是神经网络的输出张量，作为优化器的目标参数。

优化：最后一步就是调用优化器，进行梯度下降算法，通过迭代运算获得最佳图

片特征。

以上便是该项目中重要的几个模块。