

# Extracted the style from the image with Deep Neural Network

吴秉哲

2015 年 12 月 21 日

## 摘要

在视觉艺术的领域，尤其是在绘画方面，画家已经能通过自己的经验和  
技术进行创作出一副独特的复杂的艺术作品，这些作品的内容和风格的联  
系十分复杂。在计算机视觉领域，由于卷积神经网络的兴起，一些传统的任  
务比如物体定位和物体识别已经取得了很大的进步。尤其是物体识别已  
经有算法在大规模图片识别 (imageNet 图片数据集) 的任务上达到了类人  
的表现 [5]，其他方面，由于 GPU 通用计算 [2] 的发展，人们已经有能力进  
行深度人工神经网络的训练 (2014 年 ImageNet 挑战赛的冠军使用的网络  
模型已经到了 19 层)。在这次的期末项目中，我会使用深层的卷积神经网  
络来提取一副图像的物体和另一副绘画的艺术风格，并将它们结合在一起，  
形成一副全新的图片。项目中使用的算法也提供一个途径帮助计算机理解  
人类是如何完成复杂的艺术创作的。

## 1 Introduction

在这次的项目中，我们主要使用一种特殊的人工神经网络，卷积神经网络。  
卷积神经网络可以理解为一个特征提取的网络，区别与传统的计算机视觉的方  
法，卷积神经网络会自动完成对图片特征的提取，在训练一个多层的卷积神经  
网络的时候，随着层数的增加，图片中物体的轮廓会越来越清晰，但是是一些图  
片的细节会损失。根据卷积神经网络这个特点，我们在提取一张图片的具体内  
容（就是图片里面的一些物体，比如建筑，动物等）的时候，我们只需要根据  
每一个卷积层得到的特征图（即上一层图片进过这层卷积得到的图片），对特征  
图进行重建，我们就可以提取图片中内容的信息。在这次的期末项目中，我们  
主要利用网络中深层所得到的特征图进行重建得到图片的内容的信息。

另一方面，为了提取一幅绘画作品的风格特征，在项目中我们设计了一个  
基于原始图片纹理信息的特征空间。我们使用的这一个特征空间建立在卷积神  
经网络每一层的特征图上，它们由这些特征图之间的相关性组成，通过这种取  
特征图之间相关性的做法（详细见方法篇），我们可以获得原始绘画作品的风格

特征而不是画中具体的物体信息和物体之间的布局信息。为了方便，在下文中，我们将这样构建的特征空间称为风格特征空间。

这样，有了上面的风格特征空间，我们可以在风格特征空间上通过一些方法重建一幅图片，使得这幅图片可以与原始的绘画作品的风格匹配。

在这次的期末项目中，我们可以通过上面的方法分别获得一张图片的内容以及一张绘画作品的艺术风格，将它们混合在一起得到一张全新的图片。在项目中我们设计了一个基于原始图片纹理信息的特征空间中，为了展示这个效果，我们会将自己拍摄的一些图片和一些著名画家的油画作品的风格融合，具体的效果见实验结果。

上面提到的图像的合成  $i$  (即一张图片的内容结合绘画作品的风格) 是通过这样的方法：初始化一张白噪声图片，然后定义误差函数，即给定一张图片，这张图片与原始图片的内容的误差，以及和原始绘画作品的风格误差函数，训练的时候通过梯度下降法，最小化所定义的误差函数，这样初始化的那张白噪声图片就可以通过迭代形成最后我们想要得到的两种图片的融合。

当然一幅图像的风格和内容是不可能完全分离开的，比如名画呐喊中的风格和图中出现的那个个人。当我们进行上述的图片合成的时候，我们不会得到一张图片，它既完美匹配一张图片的内容，又和给定的绘画作品的风格匹配得很好。这里就存在一个平衡点，在项目中为了寻找这个平衡点，我们主要通过改变我们的误差函数中风格误差和内容误差的系数得到。

在这次项目中，我们基于一台带 GPU (Titan X) 的实现了一个自动的智能系统，输入两张图片（一张是你想要的内容和另一张你想要的风格），得到一张融合两种图片的全新的图片。我们使用的神经网络模型是 2014 年获得 ImageNet 挑战赛冠军使用的模型 VGGNET。

这个报告的其余部分由下述的部分组成：第二个小节介绍关于卷积神经网络的基本理论和 VGGNET 这个模型的基本结构，在第二节中也会对我们使用的数据集 (ImageNet DataSet) 进行介绍。第三小节会对我们使用的图片生成方法进行严格的数学描述。第四小节会介绍我们实现这个智能系统的具体实现细节（包括网络的训练和合成图片的训练）。第四小节展示了最后实验的一些结果。

## 2 背景

### 2.1 数据集

ImageNet 图片数据库由 1500,000 张标注图片组成，这些图片一共有 22000 个类别。这些图片通过亚马逊的标注系统人工标注。2010 年，一项基于这个数据集的大规模图像识别比赛 (ILSVRC) 开始举办，这个比赛选取 ImageNet 图片库的一个子集，这个子集有 1000 类图片，120000 张训练图片我们项目中使用的神经网络模型就是基于这个图片库进行训练。

这个数据库的图片有各种大小，所以在训练之前，我们对图片进行了预处理，基于线性插值的方法将每张训练图片下采样到  $256 \times 256$ 。除了下采样，我们不对图片做任何其他的预处理。完成图片预处理之后，我们使用开源的深度学习框架 Caffe[4] 训练我们的 VGG 网络模型。

## 2.2 CNN Basics

卷积神经网络最早是受神经科学的启发而产生的（猫的大脑皮层的感受野是基于局部感知的）[3]。1998 年 Lecun 提出了最早的卷积神经网络的结构 LeNet 用于手写字体的识别上，达到了商用标准。随着 21 世纪后 GPU 通用计算的发展，以及一些大型标注图片库的出现，CNN 出现在了计算机视觉的领域内，在 2012 年的 ImageNet 挑战赛上，多伦多大学的一个组使用了一个深层的卷积神经网络结构获得了第一名，与使用传统计算机视觉方法的第二名拉出了很大距离。自此卷积神经网络开始应用到一些传统的领域（比如模式识别，自然语言处理）。

与传统的人工神经网络的结构相比，卷积神经网络采用权值共享和稀疏连接的策略大大减小了反向传播算法的计算量，也降低了模型的复杂度。一个典型的卷积神经网络有两部分构成：一个特征提取器和一个分类器。特征提取器即为多层的卷积层，它将输入的图片通过卷积和下采样等方式得到一系列原始图片的特征（保存在一系列特征图中），训练卷积神经网络就是完成计算机对图片的特征的提取，而不需要人工去构造特征。这些提取的特征将会被送到分类器当中（通用的有 SVM 和多层感知器），达到对图片识别的效果。

一个卷积神经网络通常有多层结构。特征提取器就由多层卷积层和采样层构成，每一层的输入均为上一层通过卷积所得到的特征图。图一展示了一个典型的卷积神经网络，卷积层接受  $N$  个特征图作为输入，每一个输入特征都与一个卷积核  $S$  做卷积，总共输出  $M$  ( $M$  对应与卷积核的个数) 个特征图作为下层卷积层的输入。

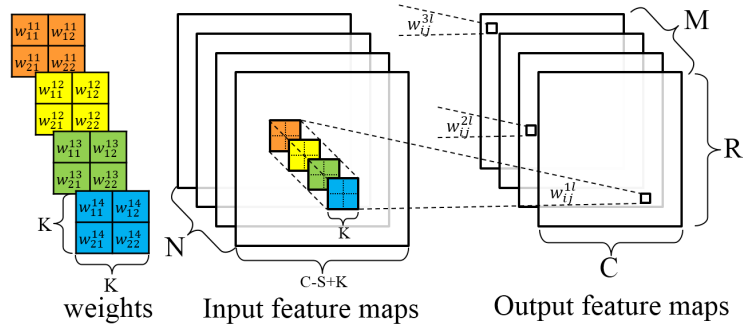


图 1: Graph of a convolutional layer

## 2.3 VGG Model

### 2.3.1 ARCHITECTURE

VGG 网络模型是一个卷积神经网络模型，他在一项图片识别的任务中达到了类人的表现，它的详细介绍可以参 [6].

在这次的期末项目中，我们使用 [6] 中介绍的网络模型，将其应用于对我们数据集的训练。在训练的时候，输入图片的大小固定在  $225 \times 225$ ，图片为 RGB 格式，即有三个颜色通道。我们对图片进行如下的预处理：

将图片通过线性插值的方法固定在网络输入所要求的大小，然后将图片每一个像素点的数值做归一化处理。

不同于一些其他的卷积神经网络模型（比如 AlexNet, LeNet），在 VGG 的网络结构中，它的卷积核的维数很小，每一层卷积层的卷积核均为  $3 \times 3$ （这已经是一个二维卷积所能达到的最小了，再小就是对图片的线性变化了）。在 VGG 的卷积层中，卷积核滑动的步长固定，在项目中我们取了符合二维卷积数学定义的 1。模型的下采样层我选择了局部最大化采样，即在一个图像的局部选取局部像素值最大的一点代表这个局部的情况，这样可以达到对特征图降维的目的，下采样的范围定为  $2 \times 2$ ，即在特征图的每个  $2 \times 2$  的像素块进行下采样。在 [6] 描述的结构中，除了上述的卷积层和下采样层，之后还有三个全连接层（即传统的人工神经网络结构）进行对之前提取特征的分类，但是在我们的项目中，由于只需要将原始输入图片的特征提取出来即可，不需要分类，所以我们去掉了网络中的全连接层。

所有隐含层使用的激活函数都是 ReLU[1]，即  $\max(0, x)$ 。（也可以使用传统的比如 sigmoid 函数，通过实验发现不同的激活函数差别不大）

整个 VGG 一共有 16 层卷积层，5 个下采样层。整个网络的结构可以在宏观上展现为图二中的结构：

## 3 方法

我们实验最后展现出来的结果是基于 VGG 的网络结构（见第二小节），在项目中我们会使用 16 个卷积层提供的特征来得到图片的内容和风格。我们不使用网络最后的全连接层。我们还会改变一些原来网络中下采样层的一些结构。下面分两个方向，即模型的训练和图片的合成两个方向来进行说明。

### 3.1 模型训练

我们在项目中使用深度学习的开源框架 Caffe 对数据集进行训练，针对大数据量的训练（我们最后训练使用的图片的存储大小为 168 个 G），我们使用了 GPU 进行计算，在训练之前，也将所有的图片创建为一种高性能的存储格式（lmdb）。在这里列出我们训练的一些主要参数：初始学习率为 0.01，初始动

量为 0.9, 初始的权重分布选取为高斯分布, 使用的优化算法为随机梯度下降法。最后训练完成的模型以 google protobuf 的协议存储在二进制文件当中, 之后通过 lua 中的 loadcaffe 模块调用模型文件。

### 3.2 合成图片

我们可以将卷积神经网络中的每一层的卷积核看为一个滤波器, 图片与每个卷积核的卷积运算就可以当成是对图像的滤波下面引入一些记号。我们对网络其中一层卷积层 (第  $l$  层卷积) 引入记号即可, 其他层类似。  $N_l$  个滤波器 (即有  $N_l$  个卷积核, 每个卷积核分别可以获得一种图像的特征), 记这些卷积核展开为一维向量过后大小为  $M_l$ , 其中  $M_l$  就是二维卷积核的高乘上宽

这样  $l$  层的经过滤波过后的图片可以存储在矩阵  $F^l \in R^{N_l \times M_l}$  中其中  $F_{ij}^l$  是第  $i^{th}$  个滤波器的响应函数。为了将图片的每一层特征图通过数值显式的表达出来, 我们将对一个初始的白噪声图片通过梯度下降法迭代, 从而寻找一副既能和一副照片的内容, 也和一副绘画作品的风格匹配的图片出来。在这次的期末项目中我采取的算法是分别将这三张图片 (即需要迭代的图片, 原始的需要获得其内容的图片, 原始的需要获得其风格的图片) 传入之前训练好的神经网络模型中然后在神经网络的每一层我们会分别获得关于这些图片的一些特征图, 我们在通过对他们特征图形成的特征空间进行误差的评估。

下面分别讨论如何评价要迭代的图片和原始图片的内容的误差, 与要迭代图片与原始绘画作品风格的误差。

#### 3.2.1 内容误差

我们分别记  $\vec{p}$  和  $\vec{x}$  是原始的我们想要获得其内容的照片和我们想要获取其内容的照片, 记  $P^l$  和  $F^l$  分别是它们通过 VGG 网络第  $l$  层时的全体特征图, 这样我们就可以定义两张图片在这一层的特征之间的均方误差函数, 定义式如下:

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

有上面的表达式, 我们可以得到误差函数 (第  $l$  层, 总体的误差函数会在本节最后定义) 梯度的解析表达式:

$$\frac{\partial L_{content}}{\partial F_{ij}^l} = \begin{cases} (F^l - P^l)_{ij}, & F_{ij}^l > 0 \\ 0, & otherwise \end{cases} \quad (2)$$

通过上面的梯度表达式, 我们就可以根据误差的反向传播算法对要迭代的图片进行更新了。这样我们就可以先初始化一张随机图片 (如这次项目中所用的白噪声图片), 然后根据反向传播算法对其迭代, 直到这张图片在这一层的经过卷积过后的特征图和我们想要获得图片类容的图片的特征图匹配即可。

### 3.2.2 风格误差

为了获得一幅图片的风格误差，我们首先需要定义一张图它的不同特征图之间的相关性，在这次项目中，我们采用如下的定义计算。这些特征图之间的相关被一个格拉姆矩阵  $G^l \in R^{N_l \times N_l}$  给出，其中  $G_{ij}^l$  是第  $l$  层第  $i$  个特征图与第  $j$  个特征图之间的内积，定义如下：

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l \quad (3)$$

为了生成一幅给定绘画作品的纹理特征，在这次项目中，我们类似上面寻求图片内容的方法，同样通过迭代一张初始化的白噪声图片来寻求一张匹配所给绘画作品风格的图片。迭代也是被错误驱动的，通过梯度下降法最小化我们下面定义的均方误差函数。

衡量风格是否相似的误差函数被定义为被迭代图片在第  $l$  层的 Gram 矩阵和所想要获得其风格的绘画作品的 Gram 矩阵之间的二次距离。记  $\vec{a}$  和  $\vec{x}$  分别为原始绘画作品和要进行迭代的图片  $A^l$  和  $G^l$  分别为他们第  $l$  层的 Gram 矩阵（分别代表两幅图像的风格）。那么上述的误差函数就可以如下定义（每一层的误差合在一起）：

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

and the total loss is

$$L_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^L \omega_l E_l \quad (5)$$

其中  $\omega_l$  表示这一层误差函数在总体风格误差中所占的比重。有上述表达式，我们可以直接求得误差函数的梯度的解析表达式：

$$\frac{\partial E_l}{\partial F_{ij}^l} = \frac{1}{N_l^2 M_l^2} ((F^l)^T (G^l - A^l))_{ji} \quad (6)$$

于是我们可以根据上述的表达式，使用反向传播算法迭代原始图片，使得它更加匹配绘画作品的风格。

### 3.2.3 总体误差

有了上述两种误差的定义，我们需要把它们结合起来。如果过于追求获得的图片更符合原始照片的内容，那么绘画作品的风格就会体现不出来，反之，迭代的图片就会很混乱，没有内容。所以，我们需要寻求两者之间的平衡点。

类似于之前的思想，我们依旧初始化一张白噪声图片，然后通过最小化误差函数来迭代图片。下面来定义总体误差函数。于是记  $\vec{p}$  为想获取其内容的图片， $\vec{a}$  为想要获得其风格的绘画作品。我们要极小化的误差函数如下定义：

$$L_{total}(\vec{a}, \vec{p}, \vec{x}) = \alpha L_{content}(\vec{p}, \vec{x}) + \beta L_{style}(\vec{a}, \vec{x}) \quad (7)$$

其中  $\alpha$  和  $\beta$  就可以表示风格和内容分别对最后的影响因子，用户可以根据自己的需求调整。整个过程的流程图如图 3 所示 (tex 的图片排版是自动的，我暂时还没有找到方法调整图片的位置所以有些图片自动排到了文章末尾)：

如图所示，代表风格和内容的两张图片，分别送入之前训练好的神经网络，用上文提到的方法，分别提取他们的风格和内容，然后迭代初始的白噪声图片，得到最后的结果。

图中展示的绘画作品为梵高的星空（我们要提取它的风格），要提取的内容为河边的建筑，它们分别经过我们提前训练好的神经网络，在每一层卷积层我们都会得到它们的一系列的特征图，然后根据方法章节介绍的误差函数，我们对一张初始化的白噪声图片迭代，就可以得到我们最后的图片。在这次项目中，我们提取内容时，选择的误差函数来自第四层卷积层，而在提取绘画作品的风格的时候我们选择第一到五层的误差函数结合。然后将两种误差函数按上一个章节的总体误差函数结合，就可以得到总体的误差函数。最后实验的效果会在下一小节展示。

## 4 实验结果

在这次期末项目中，为了展示这个算法的效果，我选取四张照片分别融合凡高的星空和挪威画家蒙克的呐喊的风格。通过修改代码，我们还可以让一幅画融合不同风格的照片。

首先展示的是一张图片（我们想要获取其内容）和一幅绘画作品（我们想要获取其风格）的融合，绘画作品选取的是上文提到的星空和呐喊。效果如图四所示（在报告末尾）。我们还尝试了融合多种风格，得到了图 5 所示的效果：

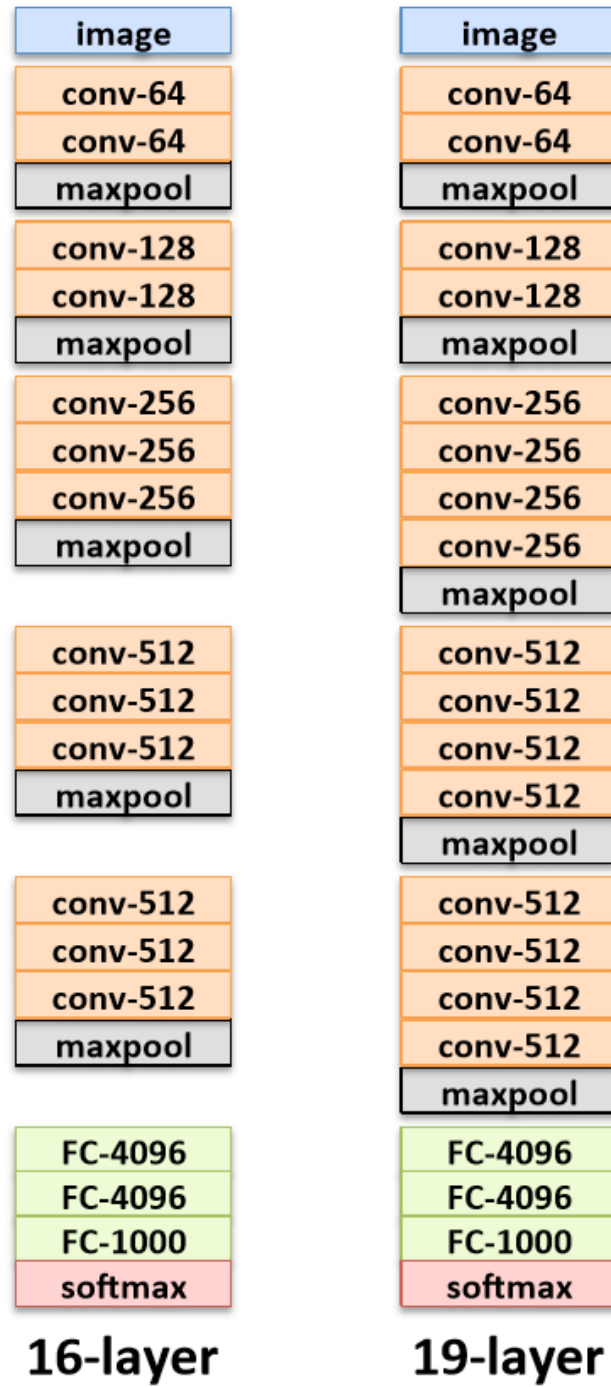
我们还尝试了融合多种风格，得到了如下所示的效果：

## 参考文献

- [1] G. E. Dahl, T. N. Sainath, and G. E. Hinton. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *IEEE International Conference on Acoustics, Speech Signal Processing*, pages 8609–8613, 2013.
- [2] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and Fei-Fei. Large scale visual recognition challenge. In *In <http://www.image-net.org/challenges/LSVRC/2012/index>*, 2012.
- [3] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160(1):106–154, 1962.

- [4] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.
- [5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:2012, 2012.
- [6] K. Simonyan, A. Zisserman, K. Simonyan, and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Eprint Arxiv*, 2014.





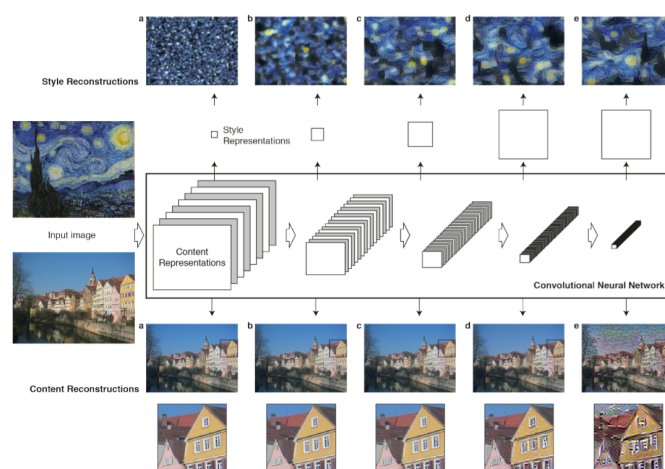


图 3: Graph of a convolutional layer

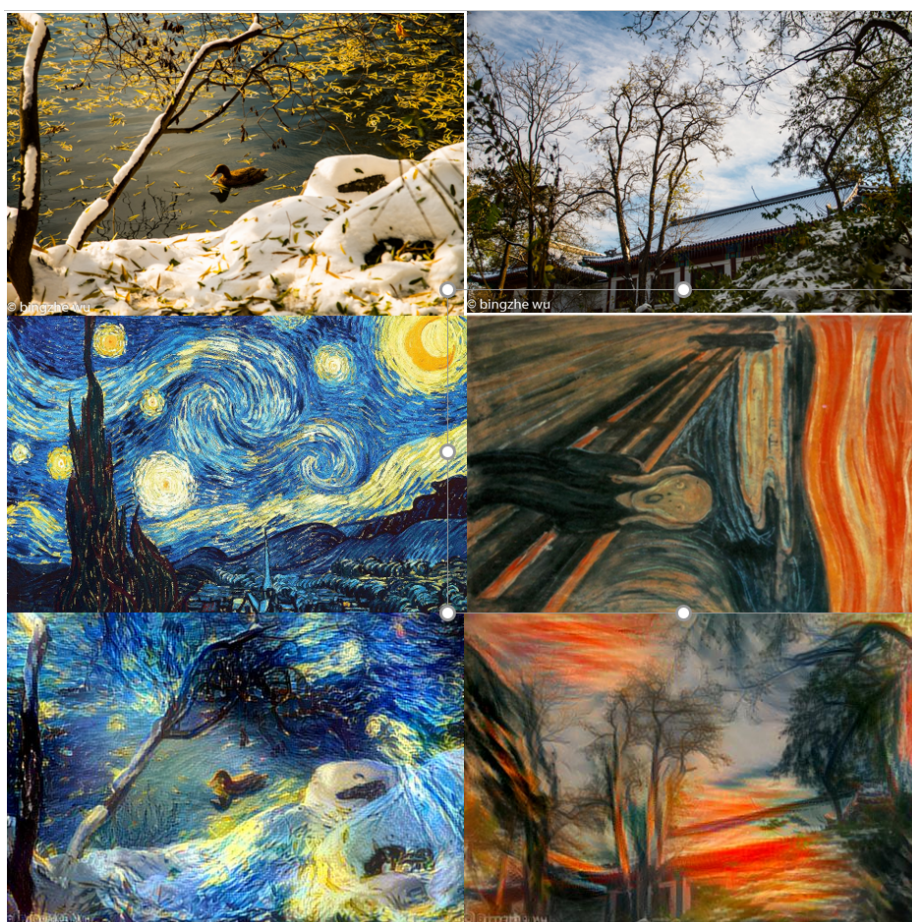


图 4: Graph of results



图 5: Graph of results