基于内容感知的
自动化图片风格转移方法

郝昱
复旦大学
计算机科学技术学院
16210240031@fudan.edu.cn

周云华
复旦大学
计算机科学技术学院
16210240096@fudan.edu.cn

摘要

人们通常都会对拍摄的原始图片进行调整，使之呈现出更好的视觉效果。本文实现了一个基于内容感知的自动化图片风格转移方法，该方法会对用户给出图片的语义和风格进行分析，在风格库中找出若干张与给出图片语义相近且风格相近又有所差别的图片，将这些图片的色彩、亮度等风格特征应用于用户给出的图片上，以达到图片风格转移的目的，并对处理后的图片做各种优化处理，尽量风格转移后的图片效果自然平滑，避免出现过多人工改动的痕迹。

Index Terms

深度神经网络，K-Means 聚类，图片风格转移

1. 简介

人们通常都会对拍摄的原始照片进行调整，使之呈现出更好的视觉效果。一般来说，专业的摄影师和专业的图片处理人员通常会用 Adobe Photoshop 等大型专业软件，凭借自己丰富的经验对照片的颜色、亮度分布、风格等进行调整（称为“修片”或“修图”）。而普通用户会更倾向于使用 VSCO、Snapseed 之类的简易修图工具，这些工具内置有“样式库”，用户可以从样式库内挑选样式，并应用到自己的图片上。然而这些样式库中的样式是有限且固定的，直接将这些样式应用于任意一张照片，得到的结果不一定都是令人满意的。

我们的目标是，给出一张原图，自动地在一个较大的图片风格库内选取一些“恰当”的图片做为参考图，对参考图片进行分析，得到色彩、亮度分布等风格信息，然后将这些风格信息应用到目标图片上，以达到图片风格转移的目的。同时，我们希望这个事情可以在无监督的情况下进行，而不是事先使用一些人工标注好的图片对来训练系统。

我们使用两个数据库来完成此项任务。第一个是我们自己收集的风格库，里面包含约 1500 张图片，这些图片都有较好的风格，但是语义（或称为“内容”）广度不够。用户给出一张图片，我们希望在这个风格库中自动选出几张图片作为修图参考。我们使用另外一个更大的数据库来辅助做这件事情：这个数据库包含约 10 万张图片，这些图片质量参差不齐，不能直接用来做修图参考，但此数据库内的图片有着非常丰富的语义（例如，海滩、室内家具、自拍等）。本文的关键点在于，我们可以使用这个辅助数据库来得到一个“语义-风格”映射，从而消除给出的原图与风格库中图片之间的语义鸿沟。由于我们使用无监督的方式来进行学习，不需要进行人工标注，这使得我们可以使用很大型的数据库。

我们使用高层语义特征，将辅助数据库划分成语义相同的若干个类，之后对每个类和风格库内的每张图片进行打分。在使用系统的时候，我们给出原图，将它映射到某个语义类中，并获得这个类对应的风格库中得分较高的一些图片，作为修图参考。之后，我们使用一种较为稳定的方法，分别将这些图片的风格转移到用户给出的原图上。

本文是对 Lee et al. 提出的基于内容感知的自动化图片风格转移的实现。

2. 特征提取

近些年来，深度学习（Deep Learning ）在图片分析和处理领域大放异彩，深度神经网络（Deep Neuron Networks, DNN ）在分类、特征提取方面都有广泛应用。由于深度神经网络有多个卷积层，不同的卷积层可以获取到图片从像素级到语义级的不同层次的特征，故常常被作为特征提取的工具。一幅图片的风格是一个很主观的评价标准，但一般受到亮度、色调等要素的影响。在这里，我们使用深度神经网络作为语义特征提取工具，同时使用亮度分布、颜色的协方差等统计量作为图片的风格特征。

2.1. 深度语义特征

在这里，我们使用 CaffeNet[4] 和 AlexNet[5] 来提取语义特征。这两个网络均为卷积神经网络 (Convolutional Neural Network，CNN)，均使用 ImageNet 数据集 [3] 进行训练，是目前被公认的具有较好的泛化能力的网络。

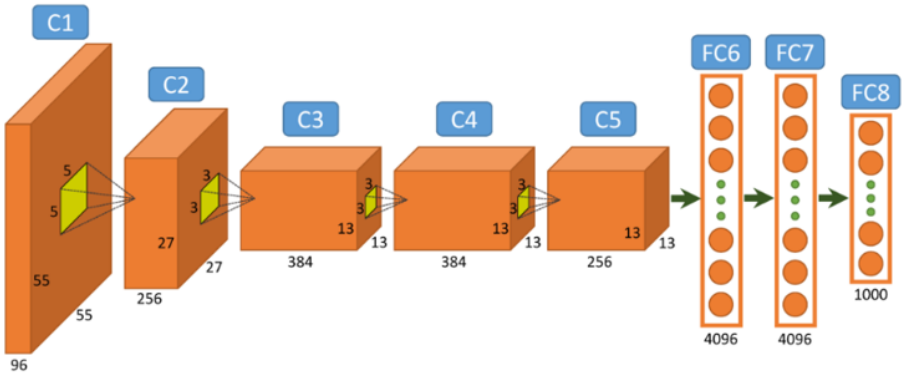


图 1. CaffeNet 的网络结构

我们希望使用该网络 FC6 层输出的 4096 维数据作为语义特征。但使用此高维度的特征会导致后续处理中过拟合的现象发生。不仅如此，由于硬件限制，如果直接使用如此高维度维的特征极有可能会导致后续处理过程中内存崩溃。为解决此问题，参考文章中去掉了网络的 FC7 层，将 FC6 层的输出改为 512 维，并对网络进行了重新训练。由于没有 GPU，重新使用 ImageNet 数据集训练深度神经网络十分耗时，所以我们采取的做法是，得到网络 FC6 层的输出后，对特征做一次主成分分析 (Principal components analysis，PCA)，将 4096 维特征转变为 512 维特征。

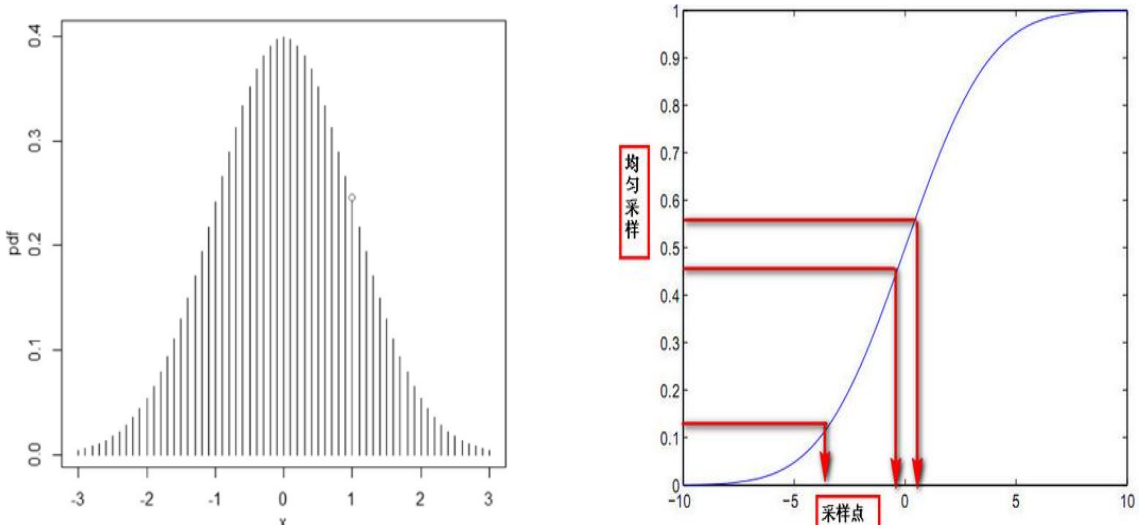
2.2. 亮度分布特征

我们对亮度通道进行累积分布函数百分位均匀采样，得到图片的亮度分布特征。首先，我们将 RGB 图片转到 CIEL*a*b* 色域 [10] 并获得图片的亮度层。之后，我们获取此亮度层的直方图，并进行累积，得到亮度累积分布函数。我们将亮度累积分布函数图像的 y 轴 N 等分（即做做均匀采样），并做与 x 轴平行的直线，这些直线和累积分布函数有一些交点，这些交点的 x 坐标即为我们想要的亮度分布特征。上述过程如图 2 所示。

这里，我们取 $N = 32$，经上述过程，即可获得一个 32 维的亮度分布特征。

2.3. 颜色特征

由先验知识可知，一幅图片上的每个像素都可以看作一个 N 维随机变量 u，这些随机变量都具有非负且连续的概率密度函数 f，这些随机变量的概率密度函数服从多元高斯分布 $f(u) \sim N(\mu_u, \Sigma_u)$。



(a) 获得图片亮度通道的直方图

(b) 获得图片亮度通道的累积分布直方图，在 y 轴上做均匀采样，得到对应的 x 的值

图 2. 对亮度通道做累计分布函数百分位均匀采样

这里，我们提取 μ_u 和 Σ_u 作为图片的颜色特征。

3. 基于内容感知的图片风格检索

给出一张图片，我们希望自动地从风格库中挑选出若干张图片，这些图片和给出的图片语义相近且风格相近。我们首先对辅助数据集中的每张图片提取深度特征，利用这些特征将辅助数据集进行语义聚类。对于每一个语义类，我们计算这个类和风格库中每一张图片的风格相似度并进行排序，以获得“语义-风格”之间的联系。在做查询的时候，我们首先提取给出图片的深度特征，并用此特征将该图片映射到某个语义类中，再根据“语义-风格”联系，得到风格库中的图片。此过程如图 3所示。

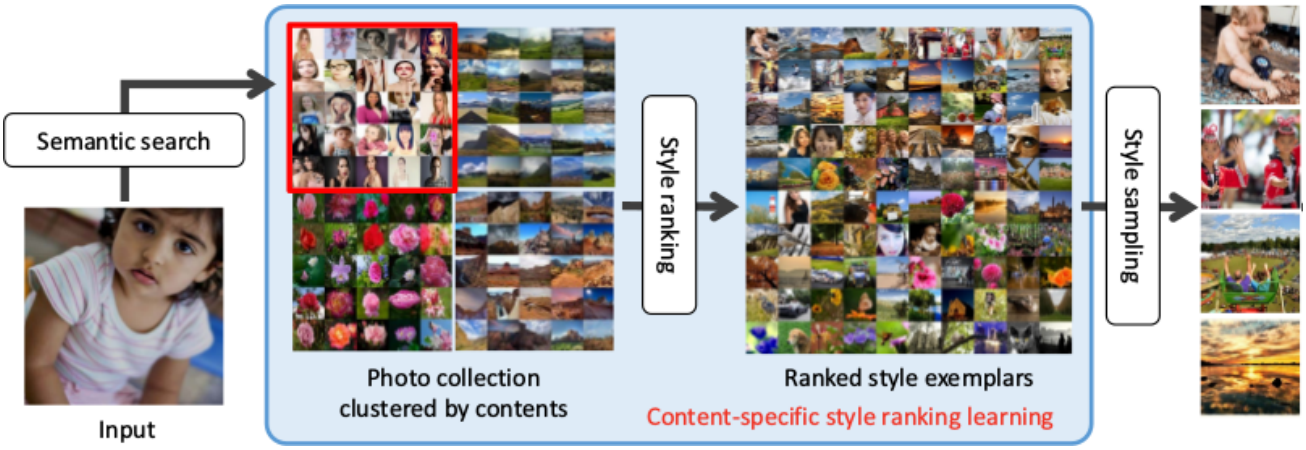


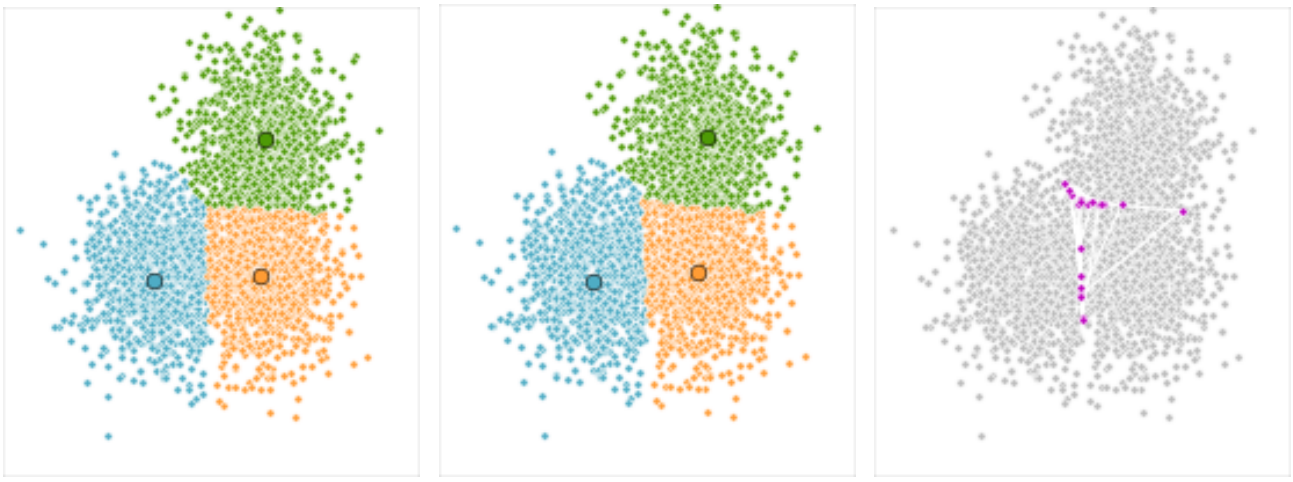
图 3. 基于内容感知的图片风格检索过程

3.1. 语义类的划分

我们将得到的深度特征，使用 K-Means 算法 [2] 对辅助数据集中的图片进行聚类，将语义相近的图片聚在一起，得到“语义类”。如果语义类的数量过少，会导致多种风格的图片都被归到同一个语义类之下，相

反，如果语义类数量过多，会导致同一个语义的图片被划分到多个语义类下。在参考论文中，作者将辅助数据集划分为 1000 个语义类。

但在实验中我们发现，由于实验硬件条件的限制，K-Means 算法聚类速度过慢。在查阅资料后，我们找到了 Mini-Batch K-Means 算法 [9]。此算法在数据规模大于 10 万的较大型数据集上的聚类速度明显优于传统 K-Means 算法，并且聚类效果和传统 K-Means 算法差距较小（如图 4 所示）。但是，在数据规模小于 100 的小数据集上，Mini-Batch K-Means 算法表现不佳。于是，当数据规模大于 10000 时，我们使用 Mini-Batch K-Means 算法以加快聚类速度；当数据规模小于 10000 时，我们使用传统 K-Means 算法以获得较为稳定的聚类结果。



(a) 使用 K-Means 做聚类　(b) 使用 Mini-Batch K-Means 做聚类　(c) 聚类结果比较

图 4. K-Means 与 Mini-Batch K-Means 在一组随机生成的数据上的聚类结果比较 [1]

3.2. 相似度的计算

我们希望使用辅助数据集作为语义和风格之间的桥梁，建立“语义-风格”之间的联系。我们定义 $Score_{i,j}$ 为第 i 个语义类与图片风格库中的第 j 张图片的风格相似度。对于第 i 个语义类，我们令其中的每一张图片与风格库中的每一张图片 j 进行风格相似度计算，并将计算结果累加到 $Score_{i,j}$ 中。此过程如算法 1 所示。

Algorithm 1 Algorithm for Content - Style Similarity

Input: Helper image data H with C clusters, Style image data S

Output: Similarity R

$R = 0$
for i in C **do**
  for j in S **do**
    for k in C_i **do**
      $R_{i,j} = R_{i,j} + S(i,j)$
    end for
  end for
end for
return R

我们使用公式 1 来计算图片 A 和 B 之间的风格相似性：

$$\mathcal{S}(A,B)=\exp\left(-\frac{\mathcal{D}_e(L_A,L_B)^2}{\lambda_l}\right)\exp\left(-\frac{\mathcal{D}_h(\mathcal{N}_A,\mathcal{N}_B)^2}{\lambda_c}\right) \tag{1}$$

其中 $\mathcal{D}_e$ 是两幅图片亮度特征的欧式距离，λ_l 和 λ_c 是两个调节参数。我们取 $\lambda_l=0.005$ ，取 $\lambda_c=0.05$ 。$\mathcal{D}_h$ 是 Hellinger 距离 [8] ，其定义为：

$$\begin{aligned}\mathcal{D}_h(\mathcal{N}_A,\mathcal{N}_B)&=1-\frac{|\Sigma_A\Sigma_B|^{\frac{1}{4}}}{|\Sigma|^{\frac{1}{2}}}\exp\left(-\frac{1}{8}\bar{\mu}^T\bar{\Sigma}^{-1}\bar{\mu}\right)\\ s.t.\quad \bar{\mu}&=|\mu_A-\mu_B|+\epsilon\\ \bar{\Sigma}&=\frac{\Sigma_A+\Sigma_B}{2}\end{aligned} \tag{2}$$

其中 $\mathcal{N}_X=(\mu_X,\Sigma_X)$ 是图片的多元高斯分布统计量，参数 $\epsilon=1$ 。使用此距离函数是因为它可以将图片之间的差异进行放大。

3.3. 查询

输入一张图片 S ，按照第 2 节中的方法提取该图片的深度特征和风格特征，经过 K-Means 算法将此图片映射到某个语义类 C 中，从“语义-风格”关系矩阵中取出 R_C 并进行排序，从风格库中取出前 k 个得分高的图片作为风格参考图。

3.4. 改进

给出一张图片 I，通过提取该图片的语义特征，我们可以将图片 I 映射到某个语义类 C 中，然后在风格库中选出相对于语义类 C 的得分最高的 k 张图片，作为修图参考。但是，这样做会导致一个后果：一旦图片 I 被映射到了某个语义类 C 中，那么得到的修图参考图就被固定下来了。

为了获取更加多样的修图参考图，我们对上述过程做一些修改：我们在获取图片 I 的语义特征后，将其最近邻的 K 个语义类都找出来，在每个类中都取风格库中得分最高的前 k 张图，并进行去重。之后，我们使用 Fréchet 距离

$$D_f(\mathcal{N}_P,\mathcal{N}_Q)=\sqrt{\|\mu_P-\mu_Q\|^2+tr[\Sigma_P+\Sigma_Q-2(\Sigma_P\Sigma_Q)^{\frac{1}{2}}]} \tag{3}$$

对图片 I 和去重后的图片再次进行风格相似度计算，并将计算结果由大到小进行排序，去除风格相似度结果大于 7.5 的图片。之后，在剩余图片中选出得分最高的 k 张图片作为修图参考。

4. 基于样例的图片风格转移

我们同时获得目标图片 I 和参考图片 S，通过对目标图片 I 做一些全局调整，使目标图片 I 的色调和风格尽可能与参考图片 S 的色调和风格相相近。为了达到此目的，我们可以对目标图片做很多处理，包括但不限于色彩混合、色调和饱和度调整，以及其他一些非线性的风格调整。

然而，直接对图片做风格转移，某些时候会导致最终结果很不自然，有很严重的人工修改的痕迹。为此，我们在风格转移的方法的基础上加入一些修正措施，以尽量减少并消除这种不自然和人工修改痕迹。

4.1. 图片预处理

首先，我们对图片做 $\gamma=2.2$ 的 Gamma 校正，以压缩图片的动态范围，从而保留更加丰富的图片细节。为了分别获取图片的亮度和色彩信息，我们将图片变换到 CIEL*a*b* 色域。

之后我们对该图片的亮度通道进行如下处理：我们首先分别去除此通道内最亮和最暗的 5% 的像素，之后再做灰度级拉伸，使得亮度覆盖整个动态范围。原文作者没有给出这样做的理由，但就实验上来看，这样可以消除一些椒盐噪声，使图片的亮度特征更加稳定。

4.2. 色彩转移

对于一张彩色图片来说，图片中的每一个像素的颜色都是一个多维随机变量。在这里我们将 I 里面的颜色记为 u ，将 S 里面的颜色记为 v 。为了简化问题，我们假设原图 I 和参考图 S 都具有非负为连续的概率密度函数 f 和 g 。于是我们的问题转化为了找到一个 C^1 连续的映射 $u \to t(u)$ ，使得颜色分布 $t(u)$ 与 g 尽可能一致。

这个问题实际上是一个换元问题，可以写成

$$f(u)du \Rightarrow f(u) = g(t(u))|\mathrm{det}J_t(u)| \tag{4}$$

的形式。其中 $J_t(u)$ 是 Jacobin 矩阵。一般来说，公式 (4) 的限制条件非常复杂，但是在这里，我们希望得到两张图之间的一个线性映射，所以可以把问题简化为

$$t(u) = Tu + t_0 \tag{5}$$

其中 N 是图片的通道数，T 是一个 $N * N$ 的矩阵。在这种形式下，Jacobian 矩阵 $J_t(u) = T$ ，并且 $|\mathrm{det}J_t(u)| = |\mathrm{det}T|$ 。这样，我们可以得到

$$f(u) \propto g(t(u)) \tag{6}$$

在这里，我们不必去寻找一个满足这个一般形式的线映射。但在 f 和 g 都服从多元高斯分布 (Multivariate Gaussian distributions，MVG) 的情况下，公式 6 总是可以写成

$$\begin{aligned} f(u) &\propto exp(-\frac{1}{2}(u-\mu_u)^T\Sigma_u^{-1}(u-\mu_u)) \\ g(u) &\propto exp(-\frac{1}{2}(v-\mu_v)^T\Sigma_v^{-1}(v-\mu_v)) \end{aligned} \tag{7}$$

的形式，其中 $f \sim \mathcal{N}(\mu_u, \Sigma_u)$，$g \sim \mathcal{N}(\mu_v, \Sigma_v)$，$\Sigma_u$ 和 Σ_v 分别是 u 和 v 的协方差矩阵。

如要满足公式 (6)，我们需要有

$$(t(u)-\mu_v)^T\Sigma_v^{-1}(t(u)-\mu_v) = (u-\mu_u)^T\Sigma_u^{-1}(u-\mu_u) \tag{8}$$

这时，转换函数 t 需要满足

$$c_O(x) = T(c_I(x)-\mu_I)+\mu_S \qquad s.t. \quad T\Sigma_I T^T = \Sigma_S \tag{9}$$

这里 T 不是唯一确定的，不同的策略可以得到不同的 T ，从而使色彩转移结果不唯一。我们使用 Pitié 和 Kokaram 的方法 [7] ，使用以下策略得到 T ：

$$T = \Sigma_I^{-\frac{1}{2}}(\Sigma_I^{\frac{1}{2}}\Sigma_S\Sigma_I^{\frac{1}{2}})^{\frac{1}{2}}\Sigma_I^{-\frac{1}{2}} \tag{10}$$

此方法可以在完成颜色转移的任务的同时，使颜色的变化量 $I[t] = \int_u \|t(u)-u\|^2 f(u)du$ 尽可能地小，并且转换结束后，原图中最亮的地方仍然保持最亮，原图中最暗的地方仍然保持最暗。但对于那些颜色比较单一的图片来说，上述方法会获得一个较小的协方差，从而导致色彩转移结果很不自然。我们可以通过加入正则化项的方法来避免此问题。我们将 Σ_I 改写为

$$\Sigma_I^{'} = \max(\Sigma_I, \lambda_r, \mathbb{I}) \tag{11}$$

并代入公式 (10) 中。这里的 $\mathbb{I}$ 是单位矩阵，参数 $\lambda_\gamma = 7.5$ 。这个公式只作用于那些有较低协方差的通道，而不会影响其他通道，可以较好地解决由于协方差值较小而导致的色彩转移结果不自然的问题。

4.3. 亮度转移

我们通过对两幅图片的亮度通道做直方图规定化操作，以达到亮度转化的目的。但在这里，我们不能直接对两幅图片的亮度通道做直方图规定化，因为这样会导致转移后亮度不平滑，从而导致转移结果不自然。为解决此问题，我们通过对两幅图片的亮度通道使用一种特殊的直方图规定化操作，来转移图片的亮度和对比度。

我们对亮度通道使用一个新的转移函数

$$l_O(x) = g(l_I(x)) = \frac{\arctan(\frac{m}{\delta}) + \arctan(\frac{l_I(x)-m}{\delta})}{\arctan(\frac{m}{\delta}) + \arctan(\frac{1-m}{\delta})} \tag{12}$$

其中 l_I 和 l_O 分别是输入和输出的亮度值，m 和 δ 是两个参数。我们对照着 GIMP(GNU Image Manipulation Program) 中的 “曲线” 工具对 m 和 δ 作出一个简单的解释：m 决定曲线的拐点，δ 决定亮度拉伸范围，如图 5 所示。

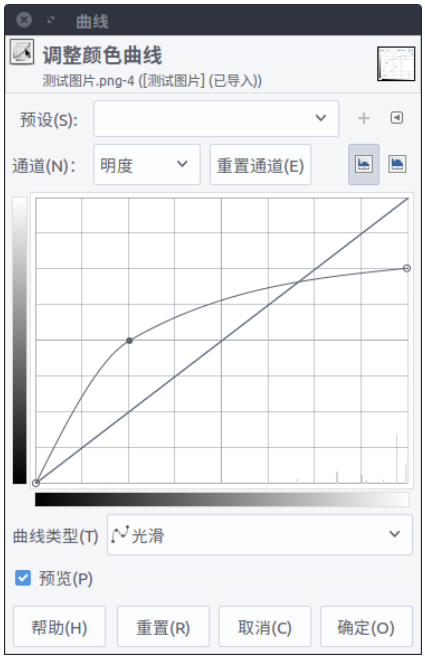


图 5. GIMP 中的 “曲线” 工具。m 决定曲线的拐点，δ 决定曲线的亮度拉伸范围

公式 12 是非负且连续的，所以可以生成一个平滑且递增的曲线。这样的曲线比较适合于亮度映射。这个通过控制 m 和 δ ，我么可以生成一系列的亮度转移曲线，继而很容易地控制亮度的转移。

那么，如何得到 m 和 δ 这两个参数的值呢？我们通过在两幅图片中提取亮度分布特征，通过尝试对亮度分布特征进行转移，同时最小化一个代价函数来获取参数 m 和 δ：

$$\begin{aligned} (m, \hat{\delta}) &= \arg\min_{m,\delta} \|g(L_I - \tilde{L})\|^2, \\ s.t. \qquad \tilde{L} &= L_I + (L_S - LI)\frac{\tau}{\min(\tau, |L_S - L_I|_\infty)} \end{aligned} \tag{13}$$

其中 L_i 和 L_s 分别是目标图和参考图的亮度分布特征。$\tilde{L}$ 用于描述分布 L_i 和 L_s 的相似程度。这里，我们令 $\tau = 0.4$。

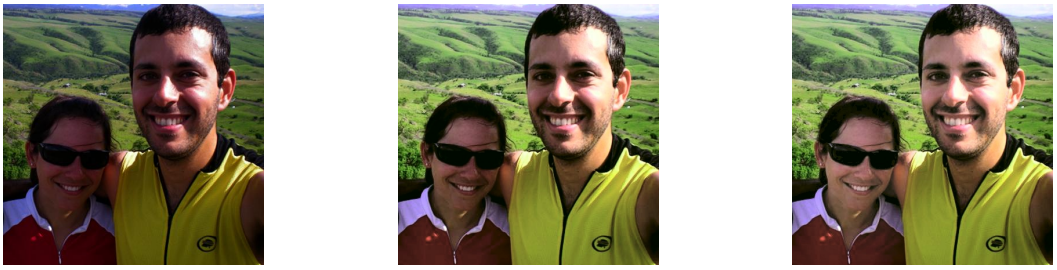
4.4. 面部亮度校正

在上述风格转移过程中，亮度转移有时会影响到人的面部，使面部亮度过暗而影响视觉效果。我们需要对过暗的面部做亮度校正。我们使用 OpenCV 内置的 Haar 级联人脸检测器来找到图片中出现的人脸，并

计算出面部中心点 p 和半径 r，同时计算出面部亮度中值 $\bar{l}$ 。如果 $\bar{l}$ 小于一个阈值 l_{th} ，那么使用这些公式对面部亮度做校正：

$$\begin{aligned} \hat{l} &= (1 - w(x)) * l(x) + w * l(x)^{\gamma} \quad if \quad \bar{l} \leq l_{th} \\ w(x) &= \exp(-\alpha_r \|(x-p)/r\|^2) * \exp(-\alpha_c \|c - \bar{c}\|^2) \\ \gamma &= \max(\gamma_{th}, \frac{\bar{l}}{l_{th}}) \end{aligned} \tag{14}$$

首先对面部做 Gamma 校正，但是直接把校正后的结果贴在原图上也会导致图片不和谐。我们利用距离和面部的平均颜色颜色生成一个权重 w，对原图和校正后的面部进行加权混合，这样混合后的结果会显得比较自然。公式 (14) 中，我们设定几个参数的值分别为 $l_{th} = 0.5$，$\gamma_{th} = 0.5$，$\alpha_r = 0.45$，$\alpha_c = 0.001$。图 6 是面部校正的实验结果。



(a) 原图　(b) 进行风格转移　(c) 进行面部亮度校正

图 6. 面部亮度校正效果对比

5. 数据集的构建

我们构建了两个数据集：一个为包含 1500 张图片的图片风格库，一个为包含 10 万张图片的辅助数据集。风格库内的图片都有较好的风格，辅助数据集内的图片比较杂乱无章。

我们使用不同的方法，分别从图片分享网站 500px 和 Filckr 上抓取图片，制作成上述两个数据集。

5.1. 风格库的构建

在这里，我们使用一种较为原始的方法构建此风格库。

我们使用 Google Chrome 浏览器浏览 500px 网站的 “最受欢迎” 页面[1]，通过不断点击键盘上的 “Page Down” 按键，利用该网页的 “无限加载” 特性，请求大量的图片。之后，我们将此页面保存，即可将该网页中的所有图片和脚本全部保存到硬盘上。我们去除无关图片和脚本，并对得到的图片进行一些人工筛选，得到最终的图片风格库。此风格库中的部分图片如图 7a所示。

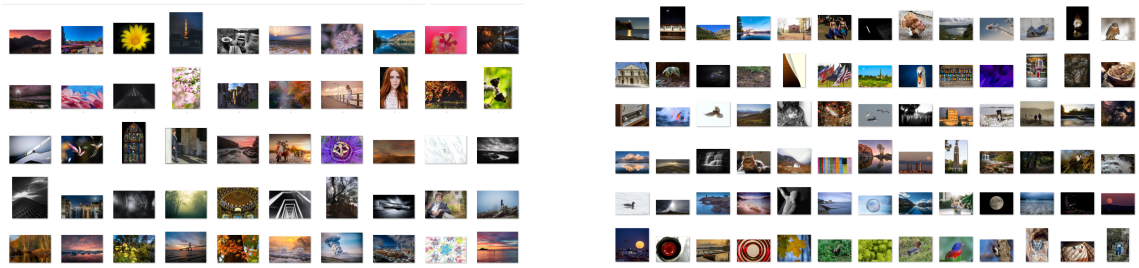
5.2. 辅助数据集的构建

由于辅助数据集内有 10 万张图片，构建风格库的方法费时费力，不适用于构建辅助数据集。这里，我们使用一个多线程爬虫，通过抓取图片网站 Flickr 上的图片来构建辅助数据集。

我们首先获取到 Flickr “每日热点” 的数据接口[2]，然后通过倒序枚举日期作为参数，不断调用此接口，即可获得需要的图片信息。之后我们对获取到的图片信息进行解析，并按照一定规则进行拼接，得到图片

1. https://500px.com/popular
2. https://www.flickr.com/services/api/flickr.interestingness.getList.html

的真实 URL 地址，并将地址放到抓取队列中。我们从抓取队列中取出每一个 URL 并进行下载，即可得到需要的图片。我们使用一个线程来完成获取图片信息和 URL 的拼接的工作，使用 Redis 作为抓取队列存储，并使用 5 个线程进行图片下载的工作。此辅助数据集中的部分图片如图 7b所示。



(a) 风格库中的部分图片示例　　(b) 辅助数据集中的部分图片示例

图 7. 两个数据集中的部分图片示例

5.3. 测试数据集的选取

在实验的过程中，我们需要构建一个小一些的数据作为测试数据集。我们从风格库中随机选出 50 张图片构成风格库子集，同时从辅助数据集中随机选出 200 张图片作为辅助数据集子集。

6. 实验

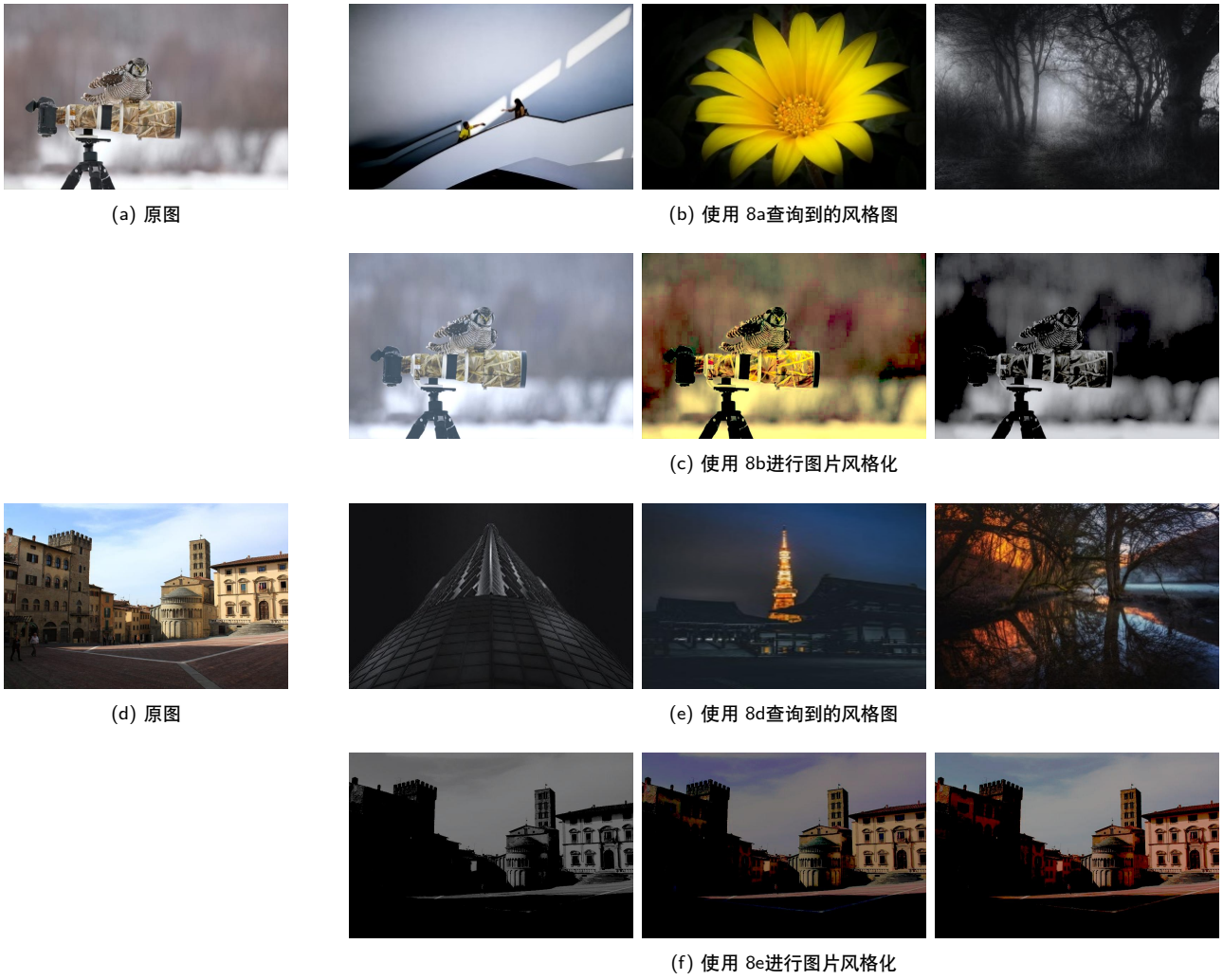
我们从网络上随机选取一些图片作为原图，使用本方法自动选取 k 张参考图片，分别对选取的原图进行风格转移。在这里，我们选用 2 张图片做为原图（如图 8a 和图 8d所示），自动选取 3 张图片做为参考图（如图 8b 和图 8e所示）。实验结果如图 8 所示。

7. 总结

本文实现了 Lee et al. 提出的基于内容感知的自动化图片风格转移方法 [6] 。通过实验我们发现，用户给出一张图片，此方法可以自动地在风格库中选出一些图片，分别将这些图片的风格应用到用户给出的图片上，并对面部亮度进行修正，同时运用各种方法尽可能消除人工痕迹，使风格转移后的图片显得比较自然。

但是此方法也有其不足之处：此方法不能动态地调整数据集。若数据集有变动，则需要重新计算整个“语义-风格”得分。此过程费时费力。这里我们提出想法：我们可以收集用户的反馈信息，同时结合图片特征，使用关联规则、人工神经网络或其他数据挖掘的方法，在线学习出能反映用户特点的“语义-风格”映射。或者可以将风格库也进行聚类，从而改“语义-单张风格图片”关系为“语义-风格类”关系。

同时我们也对此文章的假设提出质疑：文章作者认为，语义相同的图片在风格上也是相近的，我们对高校中的多名艺术学院的学生进行了调查，他们均表示此说法只能在一个很小范围内成立，而不能推广到所有的图片上。同时，文章作者认为，如果一张图片具有某个语义，这个语义和某个风格相近，那么这张图片和这个风格就应该是相近的。我们就此说法也对数学院、文学院、法学院和哲学学院的学生进行了调查，他们一致认为此推理是错误的、不合逻辑的。



(a) 原图 (b) 使用 8a查询到的风格图

(c) 使用 8b进行图片风格化

(d) 原图 (e) 使用 8d查询到的风格图

(f) 使用 8e进行图片风格化

图 8. 实验结果

参考文献

[1] "Comparison of the k-means and minibatchkmeans clustering algorithms," 2016, [Online; accessed 24-December-2016]. [Online]. Available: http://scikit-learn.org/stable/auto_examples/cluster/plot_mini_batch_kmeans.html

[2] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[4] Y. Jia, "Caffe | model zoo," 2016, [Online; accessed 24-December-2016]. [Online]. Available: http://caffe.berkeleyvision.org/model_zoo.html

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[6] J.-Y. Lee, K. Sunkavalli, Z. Lin, X. Shen, and I. S. Kweon, "Automatic content-aware color and tone stylization," *arXiv preprint arXiv:1511.03748*, 2015.

[7] F. Pitié and A. Kokaram, "The linear monge-kantorovitch linear colour mapping for example-based

colour transfer," in *Visual Media Production, 2007. IETCVMP. 4th European Conference on.* IET, 2007, pp. 1–9.

[8] D. Pollard, *A user's guide to measure theoretic probability.* Cambridge University Press, 2002, vol. 8.

[9] D. Sculley, "Web-scale k-means clustering," in *Proceedings of the 19th international conference on World wide web.* ACM, 2010, pp. 1177–1178.

[10] Wikipedia, "Lab color space — wikipedia, the free encyclopedia," 2016, [Online; accessed 14-December-2016]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Lab_color_space&oldid=754753132