Disentangling the Style for Multi-domain ImageTranslation（多领域图像转化）

概述

图像向图像的转换中，对于一个输入图像，我们可能期望有多个输出，如转化现实人物为动漫人物的过程中，一个人可以有多种颜色的头发，这种细节我们可以通过初始化高斯噪声作为解决方法，但这种方法无法实现跨领域（跨度比较大）的转化

> 如将现实人物转化为动漫人物与油画人物图片的过程中，一个模型不够，得要多个解码器（N(N-1)）

现在有的方法具体如下：

> 1．使用多个生成器转换不同的领域
> 2．StarGAN v2—使用多分支风格编码器，输入风格图片/高斯噪声后后能得到多个分支的风格编码，让生成器依据风格编码调整图像，并在此基础上提出了一些改进
>
> 1．提高风格间紧凑性，改进风格表示，让语义相似的风格距离相近以增加图片变化的连贯性
> 2．改进判别器，让它判断风格对不对，增加准确性

但我们可以注意到，第一种方法比较落后（麻烦），第二种方法有两种风格编码方式，他们被分开对待了，这是两种不同空间下的编码，却让同一个生成器来使用，影响模型上限，而且我们不知道图片优化过程中具体被怎么改变了，无法解释，还有就是由于训练用图片等问题，不同的特会相互影响，导致我们很难单独修改某一个部分

因此，作者提出了一个组件映射网络，让潜在风格与参考风格分布在相同的空间，两种正则化（对抗与三元组），分别使其在同一空间，以及保证编码器输出的平滑性

相关工作

图像转化

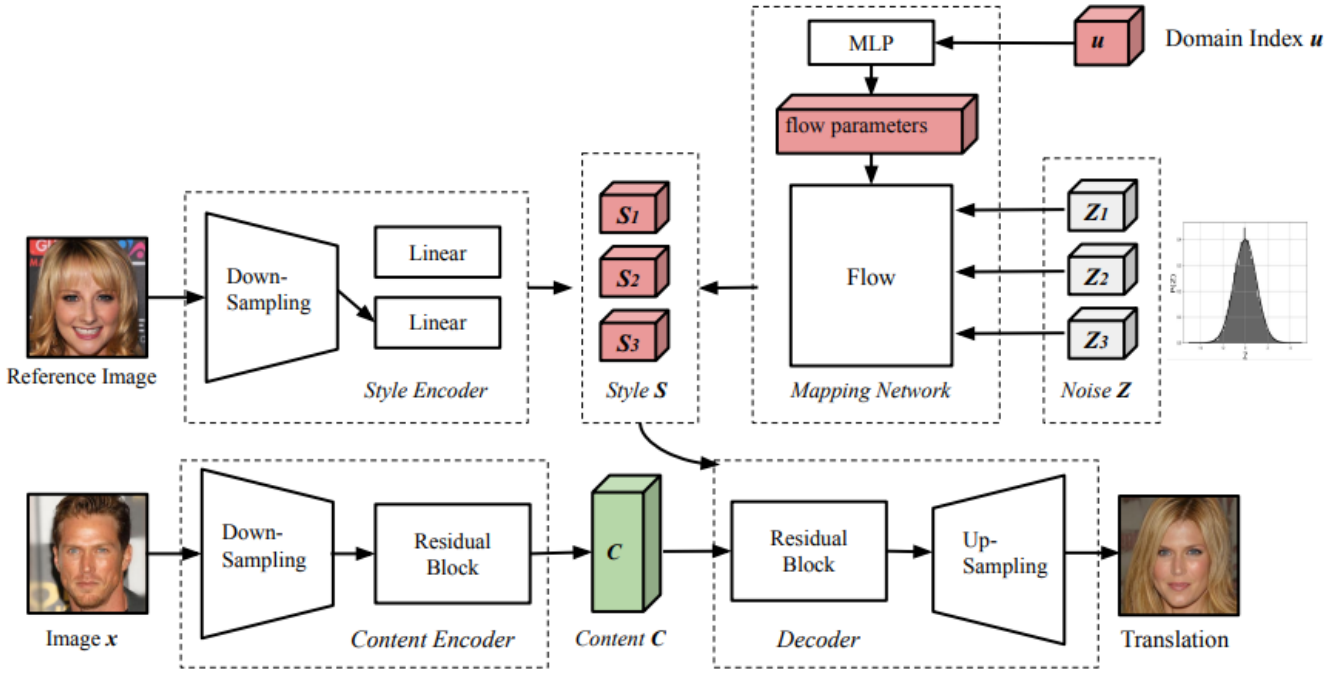
> Pix2pix通过惩罚翻译输出与真实配对图像之间的L1距离来实现内容保留。
> 但这要求配对数据，因此我们提出一系列假设：
>
> 1．循环一致性：X->Y后也能有Y->X
>
> 2．关系保持进行图像翻译时，尽管风格或某些属性发生变化，但图像间固有的关系（如对象间的相对位置和尺度、图像内部的几何结构等）应保持不变。
>
> 3．潜在空间：将图像表现为其他维度的向量，这样可以将其不同部分分开与组合，方便转换

文章发行期间比较火的事stargan（从图像或者要求中转换出风格编码—潜在或参考风格编码）与SmoothGAN
文章的工作在于统一了基于潜在和参考的风格空间且风格组件相互独立以及使用三元组损失鼓励同一领域的风格靠近

解纠缠和非线性独立成分分析

目标是学习世界独立数据生成因素的可解释分解表示—就是将各个因素的修改分开，比如我修改你嘴唇的红润程度但不要修改你嘴唇大小
作者这里的创新是它解纠缠大的范围更大，能在多个领域中尝试解纠缠，通过假设潜在变量在给定辅助变量的条件下是独立的，使用领域标签作为辅助变量，并提出一种使风格表示在条件上独立的方法。（不过内容给算成相关的了）

方法



现有方法的问题

现在的主流方法是用两种特定领域的编码器分别编码随机高斯分布与参考图像，并选取其中一种作为解码器的输入，但这样编码得到的两种风格向量并不是位于相同的向量空间中，一方面影响生成器的性能，另一方面，风格被特定领域的MLP或CNN纠缠（？）

如何构建解纠缠的风格空间

1. 高斯分布风格编码器MLP

   由于我们希望能保留领域特征，我们不希望用VAE直接最小化风格与高斯分布之间的KL散度，因此直接用现有的方法是不可行的，所以作者使用NSF（因为NSF在变换过程中保持输入数据各个组件的相互独立性，而且我们的变换不引入其他的内容，这使得原本就独立者不受其他影响），首先将领域索引转化成参数θ，再结合θ将随机采样的得到的数据转化为领域中的风格编码

$$\theta_u = \mathrm{MLP}(u), \qquad s^u_{\text{latent}} = f(z; \theta_u).$$

2. 图像风格提取器CNN

   这部分和starGAN一样，没咋变，但为了生成在统一的空间中，我们使用了对抗训练来匹配基于潜在变量和参考的风格分布——用一个判别器D判定风格编码属于哪边，这样可以让CNN生成的编码接近MLP，而且正因为它接近MLP，CNN得到的数据也是独立的

$$\mathcal{L}_{\text{uni}} = \mathbb{E}_{x,z}[\log D^u_s(s^u_{\text{latent}})] + \mathbb{E}_{x,x^u}[\log(1 - D^u_s(s^u_{\text{reference}}))],$$

3. 平滑风格编码器

   尽管我们得到的内容已经放到了同一个空间，但还有一个问题——我们如何保证我们编码器的结果足够平滑（指不会因为微小的改变出现大的变动）？
   因此提出了一种正则化方法就是选一张图片，增强他，再选一张负样本，编码他们，最小化原图编码结果和增强版的编码结果距离，最大化和原图和负样本的编码结果距离，其中负样本y是同一批次中和原图编码距离最小的那个

$$\mathcal{L}_{\text{smooth}} = \mathbb{E}_{x^u, x^u_a, x^u_n} \max(\alpha + \|E^u(x^u) - E^u(x^u_a)\| - \|E^u(x^u) - E^u(x^u_n)\|, 0),$$

$$x^u_n = \min_y \|E^u(x^u) - E^u(y^u)\|$$

总结

除此以外作者还用了其他四种损失来优化模型：

1. 对抗损失——用各个域上的判别器D判断土拍你是否是该域上的真实图片

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{x^u}[\log D^u(x^u)] + \mathbb{E}_{x,s^u}[\log(1 - D^u(G(c, s^u))]$$

2. 风格重建损失——最小化提取出的风格编码和依据这个风格编码得到的新生成图片的风格编码之间的差

$$\mathcal{L}_{\text{sty}} = \mathbb{E}_{x,s^u} \left\| E^u\big(G(c, s^u)\big) - s^u \right\|.$$

3. 循环一致性损失——将内容与风格结合得到新图片，再提取新的图片的内容和原本的风格组装重新变回原本的图片，最小化变回去的图片没完成的部分的差距

$$\mathcal{L}_{\text{cyc}} = \mathbb{E}_{x^{u_1},s^u} \left\| \big(G(\hat{c}, s^{u_1})\big) - x^{u_1} \right\|, \text{ where } \hat{c} = E_c(G(c, s^u)), c = E_c(x^{u_1}).$$

4. 多样性损失——鼓励不同输出间的差异

$$\text{; } \mathcal{L}_{\text{ds}} = -\mathbb{E}_{s_1^u,s_2^u} \left\| G(c, s_1^u) - G(c, s_2^u) \right\|.$$

再加上之前所说的两种损失，我们得到了最终的损失值函数

$$\mathcal{L}_{\text{full}} = \mathcal{L}_{\text{adv}} + \lambda_{\text{sty}}\mathcal{L}_{\text{sty}} + \lambda_{\text{cyc}}\mathcal{L}_{\text{cyc}} + \lambda_{\text{ds}}\mathcal{L}_{\text{ds}} + \lambda_{\text{uni}}\mathcal{L}_{\text{uni}} + \lambda_{\text{smooth}}\mathcal{L}_{\text{smooth}},$$

理论分析

接纠虽然做了出来，但有个问题，无监督接纠缠和现实因素对不上，但本文确实是没有给属性标签，因此作者证明了一个定理，如果满足这个定理，我们的模型分析出来的风格编码中解纠缠出来的风格编码点就可以找到一种变换，让接纠缠的内容分解（si）和实际的风格分解（具体事务，如线条粗细）相对应

定理首先假设模型已经满足如下条件：
1. 内容编码随机变量与风格编码随机变量的概率密度都是正的而且对数密度有二阶导
2. 任意风格编码随机向量，它都存在一组u（u的数量为2n-1，n为风格编码这个向量的长度），使得这2n个向量w(s,ui)-w(s,u0)能够组成一组坐标系（线性独立）（w是映射出一个向量，向量每一维的值是对应s的维度在u的条件下的对数密度的导数/二阶导）—要求有多个领域而且领域间差距足够大
此时，只要模型在各领域都能匹配边缘分布，我们就能找到一个可逆变换，让真实s*和我们的组件对应上