

VToonify: Controllable High-Resolution Portrait Video Style Transfer

Yang S, Jiang L, Liu Z, et al.

摘要

生成高质量的艺术人像视频时计算机图形学和计算机视觉中的一个重要任务。虽然基于强大的StyleGAN，研究者们已经提出了一系列成功的人像卡通模型，但这些面向图像的方法在应用于视频时存在明显的局限性，如固定帧大小、人脸对齐要求、非面部细节缺失和时序信息丢失等。VToonify框架是专门用于视频卡通化的混合框架，结合了基于StyleGAN的框架和图像转换框架的优点，实现了可控的高分辨率人像视频风格转换，还进一步提出了基于模拟单一合成数据上的相机运动的闪烁抑制损失来消除闪烁。

关键词：人脸卡通化；模型蒸馏；StyleGAN；VToonify

1 引言

本次课程的论文复现工作拟通过合成数据和网络架构两个方面对基于StyleGAN的图像模型进行提炼，将高级样式代码和多层内容特征与空间分辨率相结合，更好地重构图像细节，克服固定帧大小、人脸对齐要求、非面部细节缺失和时序信息丢失等局限性，生成运动自然的高分辨率视频，用gradio构建一个简单实用的用户交互系统，提供灵活的风格控制，让用户调整并选择自己喜欢的风格。最终实现真人视频转动画的功能。

2 相关工作

2.1 面部图像风格迁移

2.1.1 图像到图像的转换框架

Image-to-Image Translation主要采用图像转换框架进行人脸风格的转换和学习结构风格。Pix2pix首次提出监督图像转换框架，将图像从源域映射到目标域，主要通过配对数据进行训练。然而用于面部风格迁移的数据收集有一定的困难。因此，后续相关工作通过生成用于训练的合成面部卡通数据对提取现有的风格转移模型，其中配对的面孔和漫画具有相同的身份，但是不具有像素级的对应关系。

为了支持未配对的数据，CycleGAN提出了用于无监督图像转换的周期一致性。在此基础上，通过数据增强和匹配多尺度统计特征等方法可以加强域不变性和稳定训练。循环一致性有时会限制不平衡的领域，例如，将抽象的动画面孔转换回真实的面孔。为了解决这个问题，CouncilGAN利用多个生成器之间的协作取代周期一致性，而AniGAN用多尺度风格损失约束转换。未配对图像转换框架的一个主要缺点是需要从头开始学习复杂的转换，使得该框架仅限于低分辨率的图像转换。

2.1.2 基于 StyleGAN 的风格迁移

StyleGAN^[1]可以生成照片级逼真的面部图像，同时提供分层风格控制，适于作为面部风格化的大骨干。Toonify 在卡通数据集上对预训练的 StyleGAN 进行微调，并将微调模型的浅层与原始模型的深层结合起来，以生成卡通结构中的面部，同时保持逼真的面部颜色和纹理。pSp 方法通过训练编码器将真实的人脸图像投射到经过微调的潜在空间中最近的卡通面孔中，达到加速 Toonify 的效果。

AgileGAN 和 ReStyle 分别通过变分编码器和迭代细化机制改进了 pSp。StyleCariGAN 通过显式学习循环翻译的结构迁移，将 StyleGAN 与图像翻译结合起来。DualStyleGAN 扩展了 StyleGAN 与外部风格路径，以接受来自风格图像的条件，进行基于范例的风格迁移。StyleGAN-NADA 在 CLIP 的指导下，将 StyleGAN 转移到新的艺术领域，而不使用任何真正的卡通数据集，实现文本驱动的卡通化。上述方法均取得了较高质量的风格转换，但前提是人脸高度对齐，这对于视频来说是不现实的。

2.2 StyleGAN 反演

StyleGAN 反演的目的是将真实的人脸图像投影到 StyleGAN 的潜在空间中进行编辑。PSP^[2]训练编码器来加速投影。E4e 预测扩展的潜在空间 W^+ 中靠近潜在空间 W 的潜在代码，以提高可编辑性。PTI 不是搜索最优潜伏码，而是根据其预测的空间潜伏码对 StyleGAN 进行微调以接近目标图像。基于这一想法，HyperInverter 和 HyperStyle 提出训练一个超网络来直接预测 StyleGAN 参数的偏移量，以更有效的方式模拟微调。最近，新的方法提出将真实图像反演为 StyleGAN 的特征空间 (F)，可以更好重建图像细节，适用于空间编辑。

2.3 视频风格迁移

2.3.1 基于光流的视频风格转换方法

视频风格转换特别注重时间一致性，通常通过光流实现的。将神经风格转移应用于视频，通过使用输入视频的光流将之前的程式化帧应用到当前帧，以进行帧初始化。

使用时间一致性损失来约束连续程式化帧的对应像素之间的一致性。通过复合正则化以更好地适应光流和局部抖动的时间变化性质。上述方法假设输入帧之间和程式化帧之间的光流相同，这并不适用于面部经常变形的 toonification。与以前的研究不同，本文通过模拟相机在单帧上的运动，制定了一个有效且简单的闪烁抑制损失。

2.3.2 基于图像动画的视频风格转换方法

该合成艺术人像视频的方法如一阶运动，以视频驱动的运动来动画艺术人像。最近，DaGAN 通过估计面部视频中的密集 3D 几何形状来约束输出，使其与 3D 面部结构更加一致，从而展示了良好的结果。然而，这样的方法只使用单个程式化框架的信息，不可避免地丢失了重要的细节。且即使考虑了 3D 几何，面部无纹理区域的运动仍然难以预测，从而导致闪烁。

2.3.3 基于 stylegan 的视频风格转换方法

将视频帧投射到 StyleGAN 潜在代码的序列中，并训练一个网络从这些低维代码中捕获时间相关性。

训练一个潜在的转换网络用于分离身份和面部属性，以更好地保存身份来编辑面部。

结合 StyleGAN 的草图分支，用于基于草图的视频编辑。为了无缝拼接裁剪后的人脸和背景，STIT 建议调整 StyleGAN 以提供空间一致的过渡。

以上方法都需要人脸对齐和裁剪作为预处理。

3 本文方法

3.1 本文方法概述

StyleGAN2^[3]的最后 11 层主要为面部渲染颜色和纹理，对几何转换具有鲁棒性，可以用预训练的 StyleGAN2 生成不同尺寸的图像以及未对齐的人脸。

删除 StyleGAN2 的固定输入并添加另一个编码器来提供可变大小的特征（编码器直接从未对齐的人脸上提取特征），构建一个完全卷积的编码器-生成器架构，用于不同的输入大小。

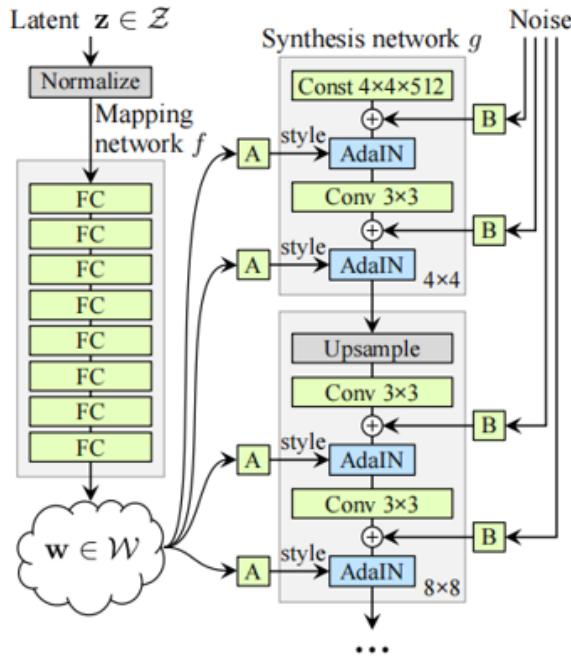


图 1: StyleGAN 框架

如图 1 所示，StyleGAN 的每个卷积层之后都会加上高斯噪声，当我们去除第 13 层以及之后的高斯噪声时，如图 2(f) 所示，噪声输入对高分辨率层几乎没有影响。因此使用低分辨率的视频作为输入来生成高分辨率的视频，提高效率，降低对输入分辨率的要求。

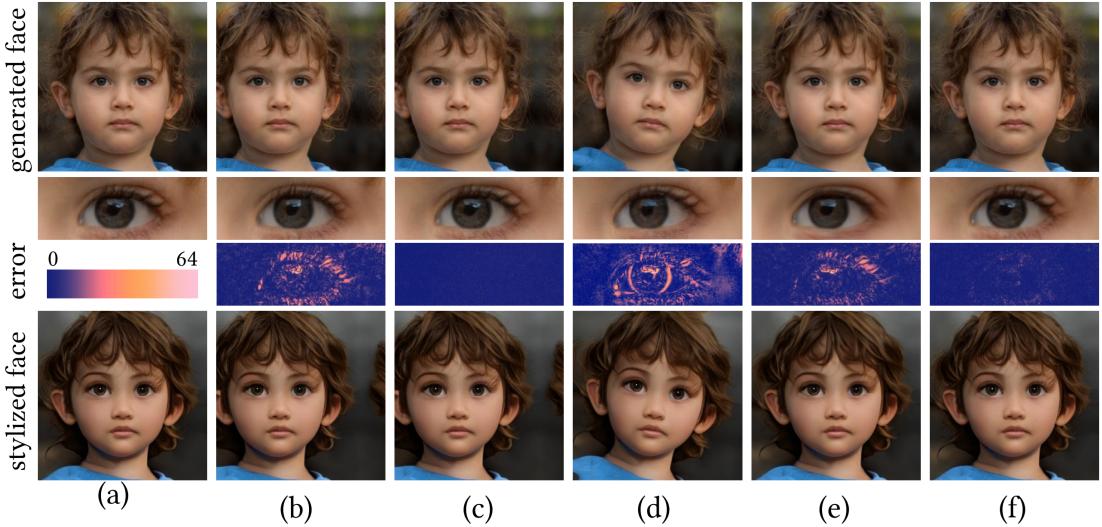


图 2: StyleGAN 分析

本文实现两种视频风格迁移框架，分别是基于集合的视频风格迁移(以 toonify^[4]为 backbone)和基于样例的视频风格迁移(以 DualStyleGAN^[5]为 backbone)。

3.2 模型框架

3.2.1 基于集合的视频风格迁移

如图 3 所示，基于集合的视频风格迁移框架，首先通过 BiSeNet 获得人脸语义映射，再作为需要训练的解码器的输入。生成器则由 Toonify 通过加入可训练的融合模块得到。

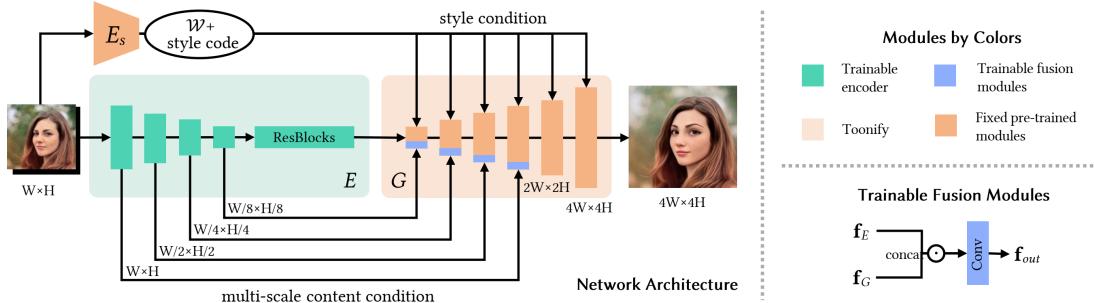


图 3: 基于集合的视频风格迁移框架

3.2.2 基于样例的视频风格迁移

如图 4 所示，基于样例的视频风格迁移框架比起基于集合的视频风格迁移框架多了一个额外的风格编码和风格迁移程度。生成器则由 DualStyleGAN 通过加入可训练的融合模块和一个训练好的 ModRes 模块得到。

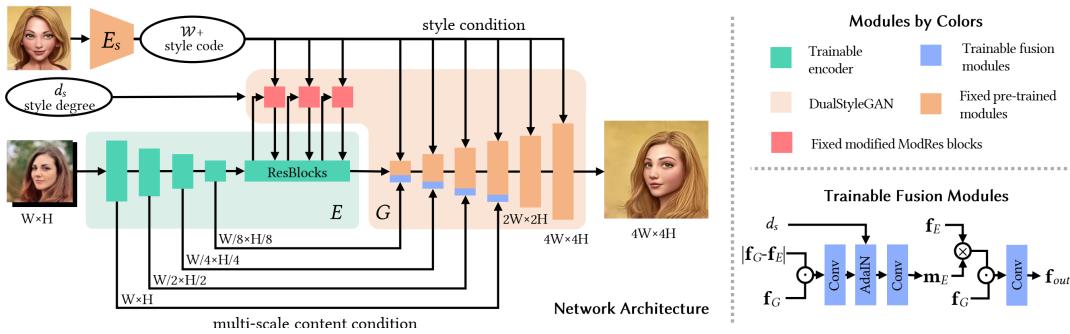


图 4: 基于样例的视频风格迁移框架

3.3 损失函数定义

两个框架在训练部分，只有编码器需要重新训练。

其中基于集合的方式训练 Encoder 的损失函数如下：

$$\mathcal{L}_E = \|f_E^{(last)}(x'_\downarrow) - f_{G_1}^{(8)}(w'', d_s)\|_2$$

基于样例的方式训练 Encoder 的损失函数如下：

$$\mathcal{L}_E = \|f_E^{(last)}(x'_\downarrow, w'', d_s) - f_{G_1}^{(8)}(w', w'', d_s)\|_2$$

4 复现细节

4.1 数据生成

在生成用于训练的人脸时，会随机从向量 \$\mathbf{n}\$ 中（不做处理、移除微笑、闭眼、睁眼和张嘴五个动作）选取一个加入到初始图像，整体流程如下：

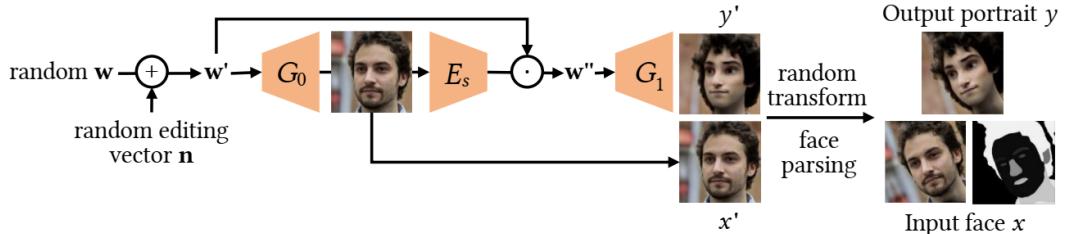


图 5: 基于收集的方法数据生成

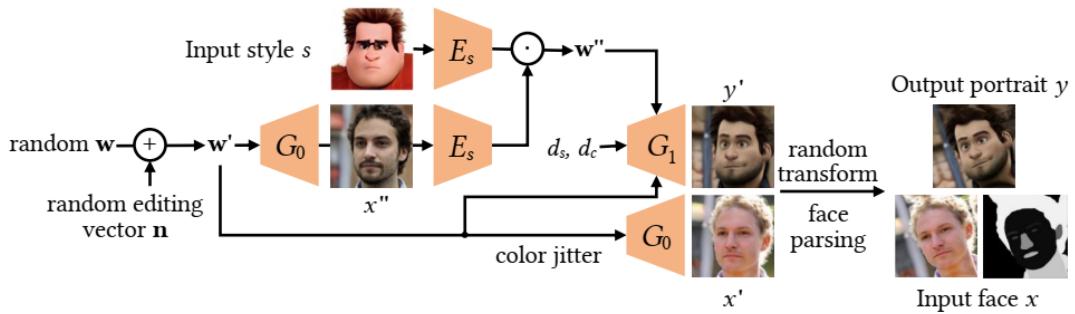


图 6: 基于样例的方法数据生成

4.2 与已有开源代码对比

为了让视频风格迁移有更好的时序性，加入了平滑算法，减少人物和一些背景细节的突兀变化，公式及代码如下：

```


$$\hat{p}_i = \sum_{j \in [i-k, i+k]} \frac{w_{j,i} \otimes \theta(p_j, f_{j,i})}{w_i}$$


$$w_{j,i} = \exp\left(-\frac{(i-j)^2}{2\sigma_t^2} - \frac{\|I_i - \theta(I_j, f_{j,i})\|^2}{2\sigma_s^2}\right) \otimes m_{j,i}$$


# temporal weights of the (2*args.window_size+1) frames
wt = torch.exp(-(torch.arange(2*window+1).float()-window)**2/(2*((window+0.5)**2))).reshape(2*window+1,1,1,1).to(device)

parse = []
for ii in tqdm(range(len(Is))):
    i = ii + window
    image2 = Is_[i-window:i+window+1].to(device)
    image1 = Is_[i].repeat(2*window+1,1,1,1).to(device)
    padder = InputPadder(image1.shape)
    image1, image2 = padder.pad(image1, image2)
    with torch.no_grad():
        flow_low, flow_up = raft_model((image1+1)*255.0/2, (image2+1)*255.0/2, iters=20, test_mode=True)
        output, mask = warp(torch.cat((image2, Ps_[i-window:i+window+1].to(device)), dim=1), flow_up)
        aligned_Is = output[:,0:3].detach()
        aligned_Ps = output[:,3:4].detach()
        # the spatial weight
        ws = torch.exp(-((aligned_Is-image1)**2).mean(dim=1, keepdims=True)/(2*(0.2**2))) * mask[:,0:1]
        aligned_Ps[window] = Ps_[i].to(device)
        # the weight between i and i should be 1.0
        ws[window,:,:,:] = 1.0
        weights = ws*wt
        weights = weights / weights.sum(dim=(0, 1, 2), keepdims=True)
        fused_Ps = (aligned_Ps * weights).sum(dim=0, keepdims=True)
        parse += [down(fused_Ps).detach().cpu()]
    parse = torch.cat(parse, dim=0)

basename = os.path.basename(args.video_path).split('.')[0]
np.save(os.path.join(args.output_path, basename+'_parsingmap.npy'), parse.numpy())

```

图 7: parsing map smoothing

4.3 界面分析与使用说明

在用户交互上，一共分为三步：

1. 首先是风格的选择，界面如图 8所示，选择完后加载相应的训练好的模型。

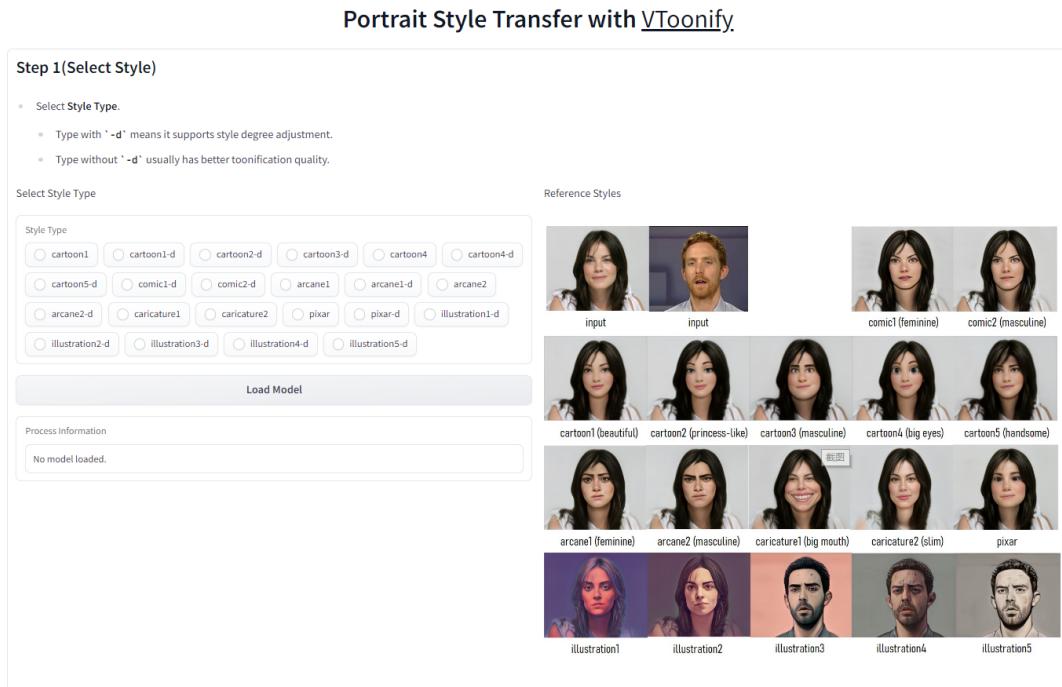


图 8: step 1

2. 第二步中，如图 9所示接收用户上传地文件，进行以眼睛为中心锚点、上下左右各保留 200 像素（用户可自行调节）的裁剪，并展示裁剪结果。

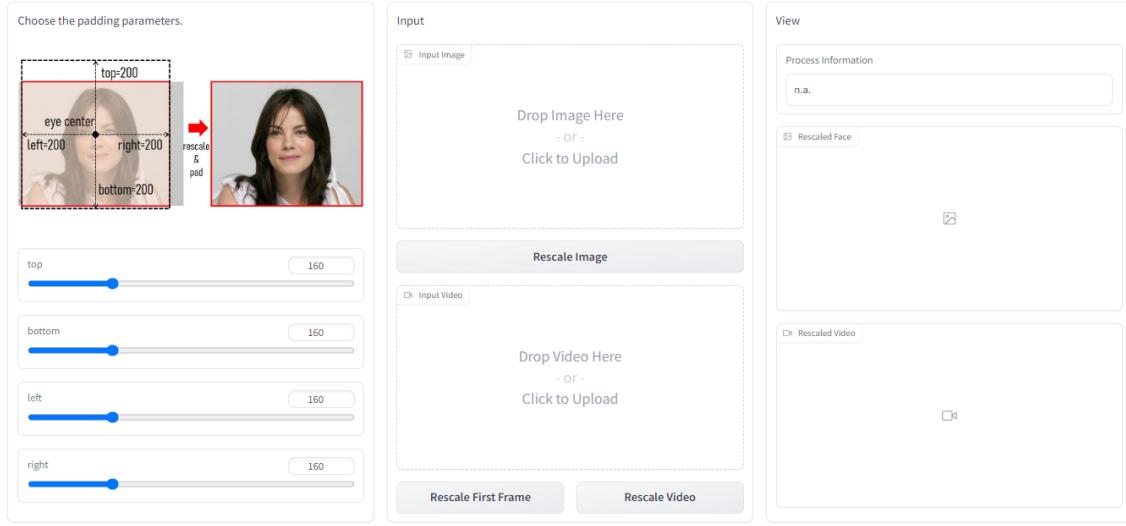


图 9: step 2

3. 最后是风格化，用户可调节风格化的程度，如图 10:

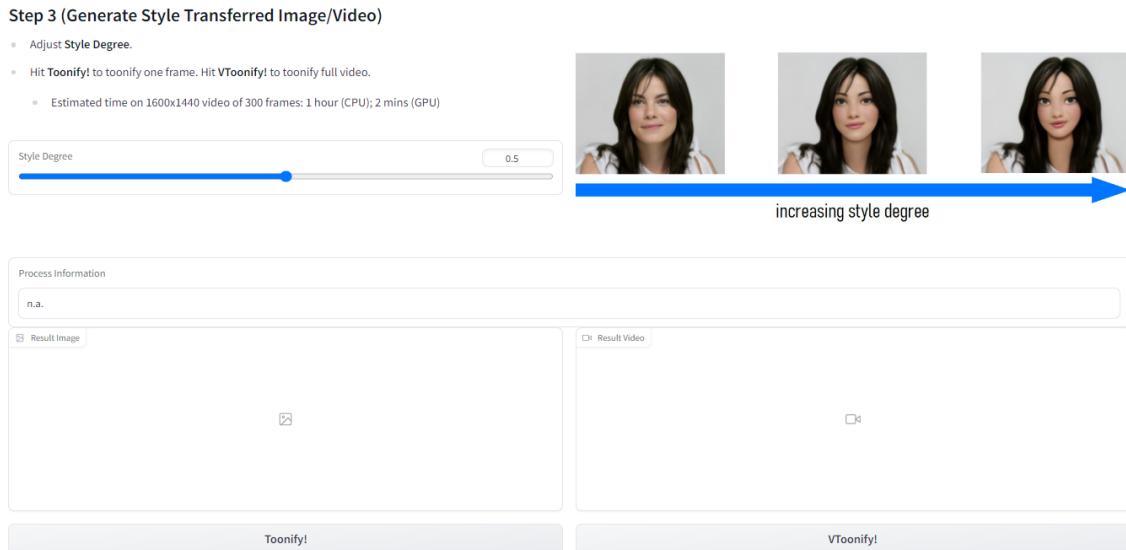


图 10: step 3

4.4 创新点与不足

创新点在于加入的语义映射平滑算法能有效避免非人脸部分的物体的缺失，在物体与人脸有重叠的部分也能保留物体的轮廓特征。而不足之处在于当人脸被遮挡时，虽然保留了物体地轮廓特征，但会透明化，丢失了与人脸地前后空间关系。

5 实验结果展示

首先是该框架对图片风格化的效果，见图 11，左边是原图像，右边是风格化的结果，中间是所选择的风格。



图 11: image

其次是对视频的效果，如图 12:



图 12: image

6 总结与展望

在这项工作中，对于一些比较极端的角度（例如侧面）的人脸风格迁移存在眼睛注视方向错误的问题，还有不能很好地保留非人脸区域内容以及光线不自然等问题，这些都有进一步提高。

参考文献

- [1] KARRAS T, LAINE S, AILA T. A style-based generator architecture for generative adversarial networks [J]., 2019: 4401-4410.
- [2] RICHARDSON E, ALALUF Y, PATASHNIK O, et al. Encoding in style: a stylegan encoder for image-to-image translation[J]., 2021: 2287-2296.
- [3] KARRAS T, LAINE S, AITTALA M, et al. Analyzing and improving the image quality of stylegan[J]., 2020: 8110-8119.
- [4] DADE K. Toonify: Cartoon Photo Effect Application[Z].
- [5] YANG S, JIANG L, LIU Z, et al. Pastiche Master: Exemplar-Based High-Resolution Portrait Style Transfer[J]., 2022: 7693-7702.