

项目报告

一、Introduction

- 1. GAN
- 2. cartoonGAN
- 3. animeGAN

二、Configuration

- Requirements
- Structure

三、Process and Results

- Model construction
- Picture stylized
- Video stylized
- Face stylized

四、Conclusion and Distribution

五、References

一、Introduction

1. GAN

生成对抗网络（GAN）是一种深度学习模型，由生成器（Generator）和判别器（Discriminator）两个部分组成。GAN的目标是通过对抗训练的方式，让生成器生成的样本与真实样本无法被判别器区分出来，从而生成逼真的数据样本。

生成器是一个神经网络模型，它接收一个随机噪声向量作为输入，并输出一个与真实样本相似的数据样本。生成器的目标是尽可能地生成逼真的样本，以欺骗判别器。

当模型都是多层感知器时，对抗建模框架最容易应用。要了解发生器在数据 x 上的分布 p_g ，我们在输入噪声变量 z 上定义一个先验概率 $p_z(z)$ ，然后将到数据空间的映射表示为 $G(z; \theta_g)$ ，其中 G 是一个微分函数，由具有参数 θ_g 的多层感知器表示。我们还定义了第二个多层感知器 $D(x; \theta_d)$ ，它输出一个标量。 $D(x)$ 表示 x 来自数据而非 p_g 的概率。我们训练 D 来最大化为 G 训练样本和样本分配正确标签的可能性。我们同时训练 G 来最小化。换句话说， D 和 G 玩以下具有价值函数^[1] $V(G, D)$ 的两人最大最小博弈，价值函数如下：

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

2. cartoonGAN

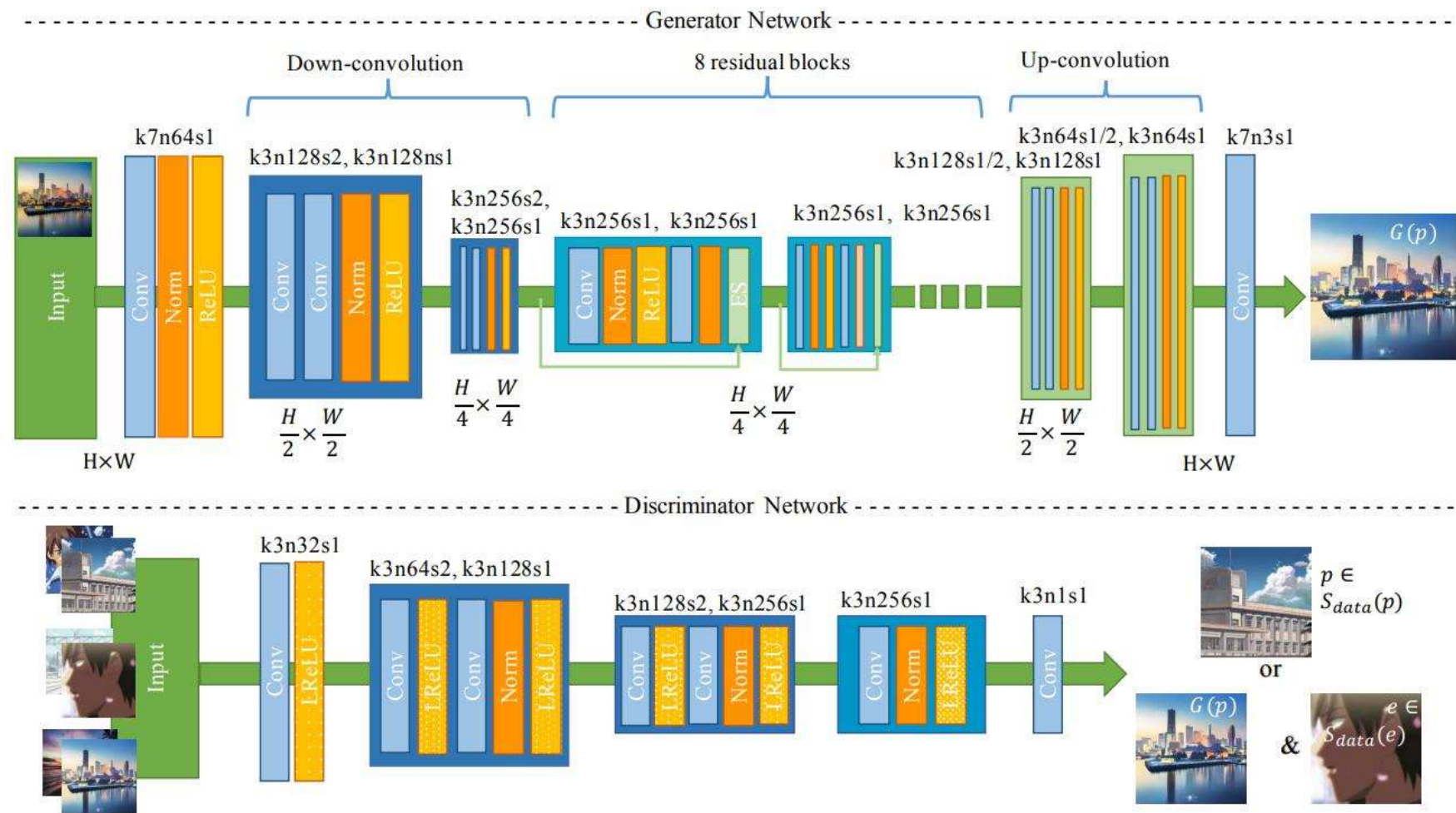
CartoonGAN是一种基于生成对抗网络（GAN）的图像风格转换模型，旨在将真实照片转换为卡通风格的图像。它是由李飞飞等人于2018年提出的。

CartoonGAN的网络结构由生成器和判别器组成，类似于传统的GAN模型。生成器接收真实照片作为输入，并输出对应的卡通风格图像。判别器则负责判断生成器输出的图像是真实照片还是卡通风格图像。

CartoonGAN的训练过程是通过对抗训练来实现的。在每一轮训练中，生成器生成卡通风格的图像，判别器根据真实照片和生成图像进行判别。生成器和判别器通过对抗学习的方式相互调整参数，使得生成的卡通风格图像更加逼真，判别器更加准确。CartoonGAN的优点是能够将真实照片转换为卡通风格的图像，具有广泛的应用前景。

与生成器网络互补的是，判别器网络D用于判断输入图像是否为真实卡通图像。由于判断图像是否为卡通是一项要求较低的任务，而不是常规的全图像判别器，因此我们在D中使用带有较少参数的简单patch级别的判别器。与对象分类不同，卡通风格判别依赖于图像的局部特征。因此，网络D被设计为较浅。在具有平坦层的阶段之后，该网络使用了两个跨步卷积块以降低分辨率并编码分类的基本局部特征。^[2]

下图是CartoonGAN结构图：



3. animeGAN

AnimeGAN 是一个轻量级的生成对抗模型，具有较少的网络参数，并引入了Gram矩阵来生成更生动的风格图像。

为了生成更好的视觉效果图像，提出了三个损失函数：灰度样式损失(grayscale style loss)、颜色重建损失(color reconstruction loss)和灰度对抗损失(grayscale adversarial loss)。在生成网络中，灰度风格损失和颜色重建损失使生成的图像具有更明显的动漫风格，并保留了照片的颜色。识别网络中的灰度对抗损失使生成的图像具有鲜明的色彩。在鉴别器网络中，我们也使用了 CartoonGAN 提出的促进边缘的对抗损失（edge-promoting adversarial loss）来保留清晰的边缘。^[3]

为了生成高质量图片，并且稳定训练，我们采用了LSGAN的最小二乘损失函数(least squares loss)作为对抗损失 $L_{adv}(G, D)$ 。

content loss类似于CartoonGAN:

$$L_{con}(G, D) = E_{p_i \sim S_{data}(p)} [\|VGG_l(p_i) - VGG_l(G(p_i))\|_1]$$

grayscale style loss促使网络生成的图像拥有动漫风格的纹理和线条，分别计算生成图像和训练之前得到动漫灰度图片的Gram matrix，然后计算二者的L1 loss：

$$L_{gra}(G, D) = E_{p_i \sim S_{data}(p), E_{x_i \sim S_{data}(x)}} [\|Gram(VGG_l(G(p_i))) - Gram(VGG_l(x_i))\|_1]$$

- 下采样模块
- 生成器网络结构

而判别器只需要对生成器生成的模型进行判别，因此使用传统的CNN网络即可

以上使用的生成器和判别器网络构建位于./net文件夹的generator.py 和 discriminator.py 文件中

2.模型搭建

- 模型初始化
- 模型创建

以上代码位于model.py中

3.模型训练

提出的 AnimeGAN 可以很容易与未配对的训练数据进行端到端的训练。由于 GAN 模型是高度非线性的，在随机初始化的情况下，优化很容易被困在局部最优解。CartoonGAN 建议，对生成器的预训练有助于加速 GAN 的收敛。因此，只使用内容损失函数Lcon(G, D)对生成网络 G 进行预训练。其中对一个 epoch 进行初始化训练，学习率设置为2e-4。在 AnimeGAN 的训练阶段，生成器和鉴别器的学习率分别为2e-5和4e-5。AnimeGAN的 epochs 是100，batch size 设置为12。Adam 优化器用来最小化总损失。

- 模型训练

以上训练代码位于model.py中，训练超参数和执行训练代码位于train.py中

Picture stylized

首先加载已经训练好的模型权重，然后导入带风格化的图片，最后对图片进行动漫风格化。

- 图片风格化

成果展示：





Video stylized

实现了图片风格化，就可以实现视频风格化，只需要逐帧对图形进行动漫风格化，然后适当进行平滑处理即可

► 视频风格化

成果展示：

原视频：



处理后：

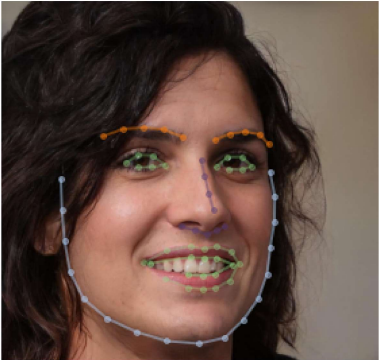




Face stylized

在上述视频中出现了人，可以看到该模型对人脸的处理效果不佳，比较模糊。因此为了改进模型，我们采用dlib库的人脸特征检测器对人脸特征进行提取，然后对人脸特征进行风格化处理，这样就能解决人脸模糊化的问题。

首先是加载人脸特征数据库，然后生成特征提取器，同时利用matplotlib将特征点可视化如下：



对导入的图片进行对齐、裁剪、缩放、填充等处理，然后利用提取到的特侦点进行人脸动漫风格化

成果展示：



四、Conclusion and Distribution

1.总结

项目前期，确定了选题后，学习了基本的生成对抗网络GAN。然后学习了cartoonGAN，一种动漫风格的生成对抗网络，但是效果以及训练量上都有不足，因此采用了一种新型轻量级的animeGAN。在阅读并学习该论文中的特点和提出的三种损失函数后，经过编写代码实现了图片动漫风格化，视频动漫风格化以及人脸动漫风格化。

其中在项目中遇到的一些问题主要是库版本不兼容问题，例如在人脸检测用到的dlib需要适配python3.9版本，整体框架使用的tensorflow是基于1.x版本进行开发的，但是安装的tensorflow是2.x不向前兼容，而如果对tensorflow进行版本降低，又不适配python3.9以及其他库，因此尝试了很多方法兼容tensorflow 1.x版本。因此下次在环境配置时应该创建一个整体适配深度学习框架如tensorflow，硬件cuda的python环境。

2.成员分工

组长：华佳彬(60%)

- 1. (40%)负责项目整体模型构建，代码实现编写，调试与后期测试
- 2. (20%)负责项目报告的撰写

组员：鄢凯瑞(40%)

- 1. (20%)负责项目选题报告的撰写和项目报告的撰写
- 2. (20%)负责项目介绍视频的准备和录制

五、References

[1]Goodfellow, I. J., "Generative Adversarial Networks", *arXiv e-prints*, 2014. doi:10.48550/arXiv.1406.2661.

[2]Y. Chen, Y. -K. Lai and Y. -J. Liu, "CartoonGAN: Generative Adversarial Networks for Photo Cartoonization," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 9465-9474, doi: 10.1109/CVPR.2018.00986.

[3]Chen, J., Liu, G., Chen, X. (2020). AnimeGAN: A Novel Lightweight GAN for Photo Animation. In: Li, K., Li, W., Wang, H., Liu, Y. (eds) Artificial Intelligence Algorithms and Applications. ISICA 2019. Communications in Computer and Information Science, vol 1205. Springer, Singapore.

- 1. [1](#) ↩
- 2. [2](#) ↩
- 3. [3](#) ↩