运行参数如下：

```
python demos\demo_transfer.py --image_path TestSamples\examples\xxx.jpg --exp_path TestSamples\exp\7.jpg
--savefolder TestSamples\examples\results --device cpu --rasterizer_type pytorch3d
```

其中 image_path 是输入图片地址，exp_path 是要迁移的表情图片，savefolder 保存结果路径。（运行结果如下）

（4）姿态和表情动画(Resposing and Animatio)

运行参数如下：

```
python demos\demo_teaser.py --inputpath TestSamples\examples\xxx.jpg --exp_path TestSamples\exp
--savefolder TestSamples\teaser\results --device cpu --rasterizer_type pytorch3d
```

其中 inputpath 是输入图片地址，exp_path 是要迁移的表情文件夹，savefolder 保存结果路径。（运行结果如下）

二．项目改进

（1）将原来的 SGD 优化方法改进为 Adam 方法，提高训练速度

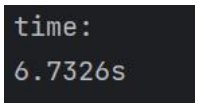
目前 DECA 模型在人脸重建方面取得了比较好的成果,但是对于光线影响较大的室外人脸图进行训练时,效率不高且总体性能一般,针对此问题,提出了一种基于 DECA 的改进算法.首先将原来的 SGD 优化方法改进为 Adam 优化器;其次加入正则化损失,提高重建后的效果,并以此来达到避免过度拟合等情况,提升算法的泛化能力.实验结果表明,改进后的 DECA 相比于原基础上效率平均提升了 5%,并且重建效果得到了很大的改善,在一定程度上改善了性能与效率不足的问题

Adam 代码（截取了部分，具体查看到 Lib\site-packages\torch\optim\sgd.py)

```
class Adam(object):
    def __init__(self, _func, _grad, _seed):
        '''
        _func: 待优化目标函数
        _grad: 待优化目标函数之梯度
        _seed: 迭代起始点
        '''
        self.__func = _func
        self.__grad = _grad
        self.__seed = _seed

        self.__xPath = list()
        self.__JPath = list()

    def get_solu(self, alpha=0.001, beta1=0.9, beta2=0.999, epsilon=1.e-8, zeta=1.e-6, maxIter=3000000):
        '''
        获取数值解,
        alpha: 步长参数
        beta1: 一阶矩指数衰减率
        beta2: 二阶矩指数衰减率
        epsilon: 足够小正数
        zeta: 收敛判据
        maxIter: 最大迭代次数
        '''
        self.__init_path()

        x = self.__init_x()
        JVal = self.__calc_JVal(x)
        self.__add_path(x, JVal)
        grad = self.__calc_grad(x)
        m, v = numpy.zeros(x.shape), numpy.zeros(x.shape)
        for k in range(1, maxIter + 1):
```

实验结果：分别使用 SGD 损失函数和 Adam 损失函数训练模型，训练时间如图所示，可见效率提升将近 5%。

使用 SGD 优化器



使用 Adam 优化器

（2）人脸对齐和数据增强，提高训练数据的质量

数据增强的作用：

1.避免过拟合。当数据集具有某种明显的特征，例如数据集中图片基本在同一个场景中拍摄，使用 Cutout 方法和风格迁移变化等相关方法可避免模型学到跟目标无关的信息。

2.提升模型鲁棒性，降低模型对图像的敏感度。当训练数据都属于比较理想的状态，碰到一些特殊情况，如遮挡，亮度，模糊等情况容易识别错误，对训练数据加上噪声，掩码等方法可提升模型鲁棒性。

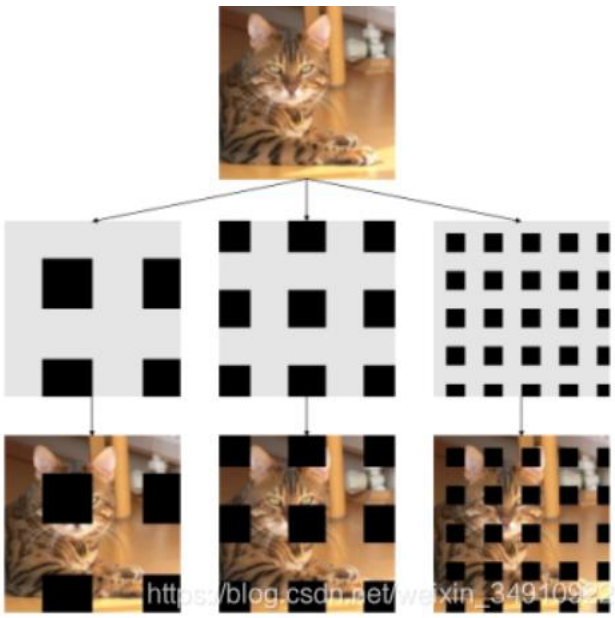
3.增加训练数据，提高模型泛化能力。

4.避免样本不均衡。在工业缺陷检测方面，医疗疾病识别方面，容易出现正负样本极度不平衡的情况，通过对少样本进行一些数据增强方法，降低样本不均衡比例。

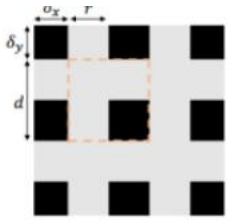
为提升训练数据的质量，首先对人脸进行对齐，对于检测任务，本使用到的数据增强方法为 GridMask

GridMask 主要思想是对 Mixup, Cutmix、Cutout 等类似方法的改进，由于掩码区域的选择都是随机的，因此容易出现对重要部位全掩盖的

情况。而 GridMask 则最多出现部分掩盖，且几乎一定会出现部分掩盖。使用的方式是排列的正方形区域来进行掩码,如下图所示。



具体实现是通过设定每个小正方形的边长，两个掩码之间的距离 d 来确定掩码，从而控制掩码细粒度。GridMask 对应 4 个参数，为 (x,y,r,d) ，四个参数的设置如下图所示：



从图中可以看出，r 代表了保留原图像信息的比例，有一个计算方法，具体可以阅读论文。d 决定了一个 dropped square 的大小， 参数 x 和 y 的取值有一定随机性。

具体代码如图所示：

```python
img_path = 'D:\DECA2\DECA\DECA\TestSamples\examples\alfw.png'  # 路径
img = cv2.imread(img_path, 1)
img_h, img_w = img[:, :, 0].shape
# 参数
x = 50
y = 50
w = 100
h = 100
l = 80

temp = x
while y < img_h:
    x = temp
    while x < img_w:
        if x + l >= img_w or y + l > img_h: break
        for j in range(x, x + l):
            for i in range(y, y + l):
                for k in range(3):
                    img[i, j, k] = 0
        x = x + l + w
    y = y + l + h

cv2.imshow('inshow', img)
cv2.waitKey(0)
```

（3）采用 GAN（生成对抗网络）技术：引入判别器，使生成的人脸更具真实感。

在人脸重建案例中，引入 GAN（生成对抗网络）技术的作用主要体现在以下三个方面：

1. 提高重建质量：GAN 技术能够有效地提高人脸重建的质量和真实感。传统的人脸重建方法通常依赖于深度学习模型，如卷积神经网络（CNN）等。虽然这些方法在某些程度上可以实现对人脸的重建，但它们往往存在一定的局限性，如重建结果的细节丢失、真实感不

足等问题。而 GAN 技术通过生成器和判别器的对抗过程，能够学习到更丰富的人脸细节，从而提高重建质量。

2. 生成多样化的人脸：GAN 技术具有生成多样化人脸的能力，这在传统的深度学习方法中是难以实现的。生成器在对抗过程中不断调整自己的人脸生成策略，使得生成的人脸具有更高的多样性和创新性。这一特点使得 GAN 技术在人脸重建领域具有更高的应用价值，不仅可以实现对人脸的重建，还可以用于创作全新的人脸形象。

3. 实时交互与反馈：GAN 技术中的生成器和判别器之间具有实时交互和反馈的特点。在生成过程中，生成器根据判别器的反馈不断调整自己的人脸生成策略，从而使得生成的人脸越来越接近真实人脸。这种实时交互和反馈机制使得 GAN 技术具有更高的自适应性和灵活性，有助于实现更高质量的人脸重建。

GAN 技术原理：

在 GAN 模型中，生成器（Generator）和判别器（Discriminator）是两个关键组件，它们分别扮演着不同的角色和功能

生成器（Generator）的作用是接收一个潜在向量作为输入，并生成与训练数据相似的样本。生成器的目标是生成逼真的样本，以至于判别器无法准确区分生成的样本和真实样本。生成器可以看作是一个生成模型，通过学习训练数据的分布特征，生成与之相似的新样本。

判别器的作用是接收样本作为输入，并预测样本的真实性。判别器的目标是对样本进行分类，判断样本是真实的还是生成的。判别器可以

看作是一个判别模型，它学习如何区分真实样本和生成样本，并提供对生成样本的反馈信号给生成器。

生成器和判别器通过对抗训练的方式相互竞争和协作。生成器的目标是欺骗判别器，使生成的样本越来越接近真实样本，以至于判别器无法准确区分。判别器的目标是尽可能准确地分类样本，使得真实样本和生成样本之间的差异更加明显。通过迭代的对抗训练过程，生成器和判别器不断调整自己的参数，以达到一个平衡点，最终生成器能够生成逼真的样本，而判别器无法准确区分真实和生成样本。

生成器和判别器的对抗性训练机制使得 GAN 能够学习到真实数据的分布，并生成具有多样性和创造性的样本。生成器和判别器之间的博弈过程推动了模型的学习和提高，使得生成的样本越来越逼真。

实验代码（截取部分，具体查看到 GAN.PY 文件)

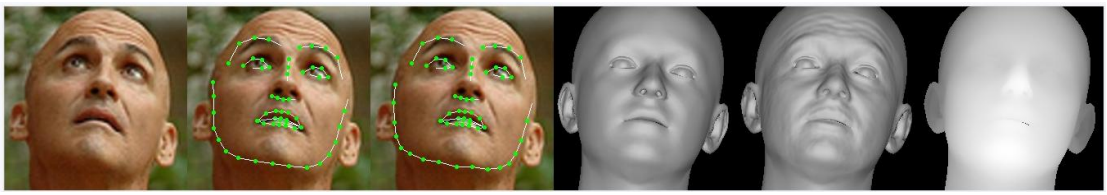
```
import os
import numpy as np

# 创建文件夹
os.makedirs( name: "./images/gan/", exist_ok=True)  ## 记录训练过程的图片效果
os.makedirs( name: "./save/gan/", exist_ok=True)  ## 训练完成时模型保存的位置
os.makedirs( name: "./datasets/mnist", exist_ok=True)  ## 下载数据集存放的位置

# 超参数配置
parser = argparse.ArgumentParser()
parser.add_argument( *name_or_flags: "--n_epochs",                                    epochs of training")
parser.add_argument( *name_or_flags: "--batch_size", type=int, default=64, help="size of the batches")
parser.add_argument( *name_or_flags: "--lr", type=float, default=0.0002, help="adam: learning rate")
parser.add_argument( *name_or_flags: "--b1", type=float, default=0.5, help="adam: decay of first order momentum of gradient")
parser.add_argument( *name_or_flags: "--b2", type=float, default=0.999, help="adam: decay of first order momentum of gradient")
parser.add_argument( *name_or_flags: "--n_cpu", type=int, default=2, help="number of cpu threads to use during batch generation")
parser.add_argument( *name_or_flags: "--latent_dim", type=int, default=100, help="dimensionality of the latent space")
parser.add_argument( *name_or_flags: "--img_size", type=int, default=28, help="size of each image dimension")
parser.add_argument( *name_or_flags: "--channels", type=int, default=1, help="number of image channels")
parser.add_argument( *name_or_flags: "--sample_interval", type=int, default=500, help="interval between image samples")
opt = parser.parse_args()
# opt = parser.parse_args(args=[])
# 在colab中运行时，换为此行
print(opt)

# 图像的尺寸:(1, 28, 28), 和图像的像素面积:(784)
img_shape = (opt.channels, opt.img_size, opt.img_size)
img_area = np.prod(img_shape)
# 设置cuda:(cuda:0)
```

True
The true value of the bool type.
Assignments to True are illegal and raise a SyntaxError.

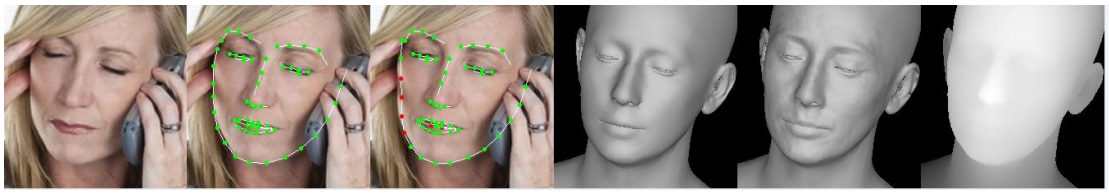
实验结果：

对于人脸重建，我们可以看到引入生成对抗网络技术生成的人脸更加

逼真（如下图所示）。



未引入生成对抗网络



引入生成对抗网络

（4）腾讯 AI Lab:使用基于 ASM（自适应骨骼-蒙皮模型）的高精度低成本 3D 人脸重建方案

腾讯 AI Lab 在 ICCV 2023 论文中提出，ASM 方法利用人脸先验，以高斯混合模型来表达人脸蒙皮权重，极大降低参数量使其可自动求解。该方法在不需要训练的前提下，仅使用少量的参数，即可显著提升人脸的表达能力及多视角人脸重建精度。这一创新性方法为低成本条件下的高保真 3D 人脸重建提供了新思路。

论文题目：ASM: Adaptive Skinning Model for High-Quality 3D Face Modeling

论文链接：https://arxiv.org/

与人脸重建领域常用的 DECA（Deep Encoder-Decoder）方法相比，ASM 在以下方面进行了优化：

1. 表达能力：ASM 方法提出了一种改进的自适应骨骼-蒙皮模型，利用人脸先验和高斯混合模型来表达人脸蒙皮权重，从而提高了人脸

的表达能力。相较于 DECA 方法，ASM 在人脸细节表达方面更具优势。

2. 参数化模型：ASM 方法在模型构建中降低了参数量，使其在无需训练的前提下，仅使用少量参数即可实现较高精度的多视角人脸重建。这使得 ASM 方法在低成本条件下取得了更好的重建效果，相较于 DECA 方法具有更高的实用价值。

3. 成像质量：ASM 方法通过利用人脸先验和高斯混合模型，使得重建结果在成像质量上具有更好的可控性。相较于 DECA 方法，ASM 方法在成像质量上有所提升，更接近真实人脸。