计算机学院 软件工程 数字媒体技术基础



智能换脸软件

梁冠轩

中山大学计算机学院 软件工程 19335118

摘 要智能换脸技术多数用于娱乐领域，依靠AI学习和算法模型训练，对人脸进行提取与替换，可达到视频换脸的级别。本文主要从介绍智能换脸的意义，回顾智能换脸的现有方法，如何利用算法实现智能换脸技术，并且对使用的算法进行解析，对使用结果进行分析，最后总结大作业。

关键词 深度学习；智能换脸；

1. 引言

脸部是辨别一个人的身份的主要方式之一，如今还可以用脸部来进行加密，支付。换脸技术的出现，人们看到的一张照片甚至一段视频里面的人物的脸可能是另一个人脸，这使得人们分辨网络上照片的真实性更加困难。当然换脸技术还可以用于制作影视作品，该技术已成熟地被应用于好莱坞的工业电影。

早在上世纪90年代，好莱坞电影就使用过了换脸技术。1991年的《终结者2》就使用过换脸技术，使用数字擦除，将替身演员的脸擦掉，再通过特效师一帧一帧地替换。1993年地《侏罗纪公园》，先由特效演员进行拍摄，然后演员本身要拍摄大量素材，将这些素材数字化转描，最后将角色的脸遮盖到特效演员脸上[1]。这些技术仍未成熟，换脸后的视频可以看到有明显的换脸痕迹。

如今随着计算机行业的发展，计算能力已远超过以前。AI深度学习技术的出现，让视频换脸技术的门槛大大降低了。曾经只有好莱坞才能完成的换脸特效，现在普通人通过一台电脑，甚至一部手机就可以做到。2019 年市面上名为“ZAO”的人工智能“换脸”手机软件引发人们很高关注，用户上传清晰正脸照，并且在摄像头前完成眨眼、扭头等规定动作，由系统判断这张照片为本人之后就可以将视频中的人物替换为自己的脸[1]。

值得肯定的是，人工智能在影视产业中

的再创作功能有了长足的进步，能够促进影视产业的繁荣发展。人工智能在换脸方面的应用带来便利的同时也引发大量争议，很大程度上是因为 它的用户协议带来的个人隐私风险。譬如手机软件“ZAO”的影响持续发酵，即是如此。用户的数据信息被泄露和截取能够对个体生成数据画像，从而搭建出完整的个体形象，这不仅牵扯到隐私甚至是对个体进行全程全方位的监控。甚至有不法分子，利用换脸技术，将一个人的脸换到色情影片中进行传播，这严重侵犯了人们的权利。

所以如何利用现有的技术判别一张照片或一个视频中的人物是否被换脸，开发识别和破解虚假视频的技术，深度学习算法来识别和过滤伪造视频，去伪存真。是现在最重要的问题。美国国防部研究机构 DAPRA 已经研发出了首款“反变脸”的 AI 刑侦检测程序，有利于打击虚假视频以及减轻换脸技术带来的负面影响，这是值得我们借鉴的。

第二要强调技术研发者与应用者的自律。科学家与科技工作者的责任伦理能够有效遏制科技道德失范现象。责任伦理，是指一种以“尽己之责”作为基本道德规范的伦理。是否履行自己应尽的责任是道德主体善恶的判定关键所在。人工智能技术的研发与应用者应该承担起相应的社会责任，将 AI 用在诸如影视产业等对社会有益的领域。

本文主要回顾实现换脸的现有方法，阐述自己所使用的算法，对实验结果进行分析，最后总结这一次大作业。

1. 回顾智能换脸的现有方法

DeepFake是现在在github上最火热的智能换脸技术，“ZAO”的人工智能换脸技术也是从DeepFake算法演变出来的。顾名思义，Deep是深度学习，Fake是造假。一般地，DeepFake可划分为如下四类：重现（reenactment），替换(replace)，编辑(editing)和合成(synthesis)，而智能换脸技术，就是运用了replace。DeepFake技术由几种神经网络结合而来：编码-解码网络，卷积神经网络，生成对抗网络，图像转移网络和递归神经网络。

* 1. 编码-解码网络（ED）

编码-解码网络(ED)由编码网络encoder和解码网络decoder组成。

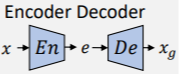


图1 编码-解码网络原理

连接编码和解码的中间层较窄，因此当训练：De ( En ( x ) ) = xg时，网络将被迫学习、汇总训练样本的高层语义概念。

给定x的分布X，En ( x ) = e，通常称e为编码或嵌入，而E = En ( x )被称为“潜在空间”。

DeepFake技术通常使用多个编码器或解码器，并操纵编码来影响输出xg。如果编码器和解码器是对称的，并且以目标

De ( En ( x ) ) = x训练网络，则该网络称为自动编码器，输出是x的重建。

ED的另一种特殊类型是变分自动编码器（VAE），其中编码器学习给定X的解码器后验分布。VAE比自动编码器在生成内容方面更好，因为潜在空间中的表征可以被更好地解耦。

* 1. 卷积神经网络（CNN）

与全连接网络相反，卷积神经网络擅长学习数据中结构模式并组合得到高层次表征，因此在处理图像方面效率更高。

卷积神经网络中的卷积层学习的是卷积核/滤波器参数，这些滤波器在输入图像上移动，提取抽象的特征图作为输出。

随着网络变得越来越深，池化层降低维数，上采样层提高维数；它们可以灵活地构建用于图像的ED CNN。

卷积网络在本质上是一种输入到输出的映射，它能够学习大量的输入与输出之间的映射关系，而不需要任何输入和输出之间的精确的数学表达式，只要用已知的模式对卷积网络加以训练，网络就具有输入输出对之间的映射能力。

* 1. 生成对抗网络（GAN）

生成对抗网络由互相作用的两个神经网络组成：生成器G和鉴别器D。

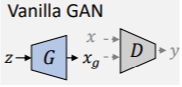
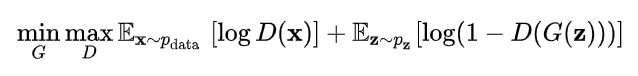


图2 生成对抗网络原理

生成网络G: 期望能够产生尽量真实的图片，进而骗过判别器。

判别网络D: 期望能够准确的区分真假图片。

G旨在欺骗D来创建伪样本xg，D学会区分真实样本( x∈X )和伪样本xg = G ( z )。具体来说，有一个分别用于训练D和G的对抗损失：



在对抗博弈下，G学习如何生成与原始分布无法区分的样本。训练后，将D丢弃，并使用G生成内容。当应用于图像时，此方法的优点是通常可以生成更高质量、逼真的图像样本。

* 1. 图像转换网络（Pix2Pix CycleGAN）

pix2pix可以完成从一个图像域到另一个图像域的转换。在pix2pix中，G以输入图像xc作为输入，在给定目标标签xt，希望G学习xc - > xt 的映射。即希望G所生成的图像xg和xt 无限接近，而D区分

( xt , xc ) 和 ( xg , xc )。

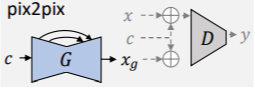


图3 Pix2Pix原理

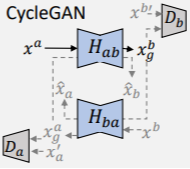
pix2pix是一种监督式、成对式的训练方式，对数据有严苛要求。提升版本的pix2pixHD可以用来生成具有更好保真度的高分辨率图像。 

图4 CycleGAN原理

CycleGAN可以通过不成对的训练样本来进行图像转换。该网络由两组GAN构成，并形成一个循环约束：将图像从一个域转换为另一个域，然后再次返回时，确保一致性。

* 1. 递归神经网络（RNN）

RNN是可以处理序列和可变长度数据的神经网络。

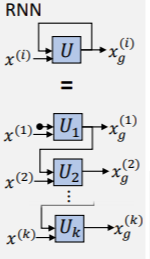


图5 递归神经网络原理

在Deepfake制作中，RNN通常用于处理音频、视频。RNN的更高级版本包括长期短期记忆（LSTM）和门递归单元（GRU）。

* 1. Deepfake的流程示例

要生成目标的xg，重现和人脸替换类型的网络一般遵循以下过程：

（1）检测并裁剪面部；

（2）提取中间特征表示；

（3）根据一些驱动信息（例如，另一张脸）生成新面部内容；

（4）将生成的脸融合到目标帧中。

通常，一些驱动图像生成的方法有：

（1）让网络直接在图像上执行映射学习，

（2）使用ED解耦身份，然后进行修改/交换编码特征；

（3）在将其传递给解码器之前添加其他编码；

（4）在生成之前将中间人脸/身体的特征表示转换为所需的身份；

（5）通过光流驱动生成

（6）使用3D渲染，或生成的内容进行组合，以创建初始内容（头发，场景等），然后将该合成（粗略结果）通过另一个网络（例如pix2pix）以改善真实感。

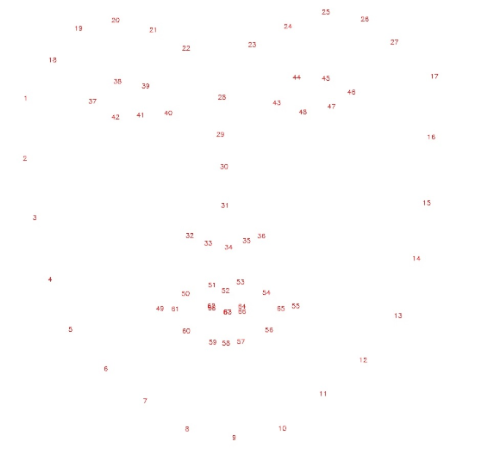


图6 人脸的68个特征点

1. 阐述自己使用的算法

本人的智能换脸软件，主要使用了python的paddlehub库和opencv库。

PaddleHub是飞桨预训练模型管理和迁移学习工具，通过PaddleHub开发者可以使用高质量的预训练模型结合Fine-tune API快速完成迁移学习到应用部署的全流程工作。其提供了飞桨生态下的高质量预训练模型，涵盖了图像分类、目标检测、词法分析、语义模型、情感分析、视频分类、图像生成、图像分割、文本审核、关键点检测等主流模型。

OpenCV是一个用于图像处理、分析、机器视觉方面的开源函数库，提供了Python、Ruby、MATLAB等语言的接口，并且实现了图像处理和计算机视觉方面的很多通用算法，可以给开发者调用。OpenCV提供了计算机视觉领域方向，人机互动，物体识别，图像分割，人脸识别

，动作识别，运动跟踪等功能。

* 1. 提取人脸的68个特征点

智能换脸的最关键部分，就是识别图

片中人像的脸部。利用PaddleHub的face\_ landmark\_localization模块，可以快速地从

图片中提取人像的68个特征点，人脸轮廓17个点，左右眉毛各5个点，左右眼睛各6个点，鼻子9个点，嘴巴20个点，还支持同一张图中的多个人脸检测。

face\_landmark = hub.Module(name = "face\_landmark\_localization")

利用face\_landmark\_localization的

keypoint\_detection函数，就可以获得人脸的68个特征点。

实现代码：

def face\_feature\_points(image):

    dets = face\_landmark.keypoint\_detection([image])

    num\_faces = len(dets[0]['data'][0])

    if num\_faces == 0:

        return None

    feature\_points = np.array([[p[0], p[1]] for p in dets[0]['data'][0]])

    return feature\_points

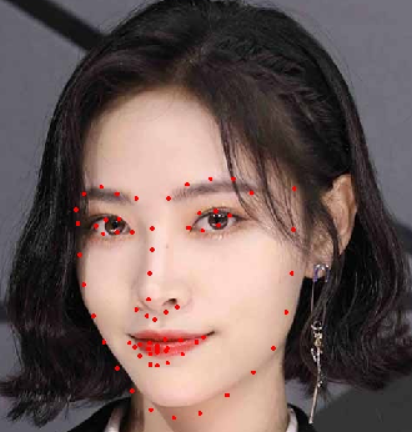


图7 人脸特征点提取实现

获取了图片1和图片2的人脸特征点后，还要根据这些特征点，获取图片1和图片2的人脸掩模。

* 1. 获取人脸掩模

获取了人脸特征点后，要根据特征点，获取人脸的掩模。获取人脸掩模要用到opencv库的fillPoly函数。

cv2.fillPoly()函数可以用来填充任意形状的图型，可以用来绘制多边形。也可以使用非常多个边来近似的画一条曲线。cv2.fillPoly()函数可以一次填充多个图型。函数原型：cv2.fillPoly( image , [ 多边形顶点array1, 多边形顶点array2, … ] , RGB color)。

所以，根据68个特征点，设置多边形的68个顶点，再指定RGB color为255后，可以获得人脸的掩模。

实现代码：

def face\_mask(image\_size, feature\_points):

    mask = np.zeros(image\_size, dtype = np.int32)

    points = np.concatenate([feature\_points[0:16], feature\_points[26:17:-1]])

    points = np.array(points, dtype = np.int32)

    cv2.fillPoly(img = mask, pts = [points], color = 255)

    return mask.astype(np.uint8)

得到了人脸掩模后，还要获取人脸掩模的中心点。

* 1. 获取人脸掩模的中心点

获取人脸中心点很简单，只需要把x轴的最大值加最小值再除2，y轴的最大值加最小值再除2即可。

实现代码：

def mask\_center\_point(mask):

    mask\_index = np.argwhere(mask > 0)

    miny, minx = np.min(mask\_index, axis = 0)

    maxy, maxx = np.max(mask\_index, axis = 0)

    center\_point = ((maxx + minx) // 2, (maxy + miny) // 2)

    return center\_point

* 1. 获取两个人脸掩模掩盖部分的并集

获取了图片1和图片2的人脸掩模后，要获取这两个人脸掩模掩盖部分的并集。

先利用np.min函数，获取两个掩模矩阵的每列的最小值。然后利用opencv库的cv2.blur ( img, ksize )函数，对这个掩模掩盖部分的并集进行均值滤波。它只取内核区域下所有像素的平均值并替换中心元素，对于椒盐噪声的滤除效果比较好。

实现代码：

def mask\_union(mask1, mask2):

    union = np.min([mask1, mask2], axis = 0)

    union = ((cv2.blur(union, (5, 5)) == 255) \* 255).astype(np.uint8)

    union = cv2.blur(union, (3, 3)).astype(np.uint8)

    return union

获取了两个人脸掩模掩盖部分的并集后，还要依靠图片1和图片2的68个特征点，将图片1按照图片2来进行仿射变换。

* 1. 获取仿射变换

仿射变换，又称仿射映射，是指在几何中，一个向量空间进行一次线性变换并接上一个平移，变换为另一个向量空间。仿射变换需要一个M矩阵，但是由于仿射变换比较复杂，一般直接找很难找到这个矩阵，opencv提供了根据变换前后三个点的对应关系来自动求解M的函数，这个函数就是：M=cv2.GetAffineTransform(src, dst)。src：原始图像中的三个点的坐标，dst：变换后的这三个点对应的坐标，M：根据三个对应点求出的仿射变换矩阵。

设置三个点来进行仿射变化，分别为68个特征点中的18，8，25。仿射变换前后都是依靠这个三个点。依靠这三个点获得仿射变换矩阵后，用这个矩阵来合成图片。这就要使用opencv的warpAffine函数。函数原型为：cv2.warpAffine(src, M, dsize)。

实现代码：

def affine\_image(image1, image2, feature\_points1, feature\_points2):

    three\_points\_index = [18, 8, 25]

    M = cv2.getAffineTransform(feature\_points1[three\_points\_index].astype(np.float32),feature\_points2[three\_points\_index].astype(np.float32))

    dsize = (image2.shape[1], image2.shape[0])

    affine\_image = cv2.warpAffine(image1, M, dsize)

    return affine\_image.astype(np.uint8)

* 1. 依据图片2对图片1进行人脸肤色调整

当不需要用到人脸掩模时，只需要对图像进行进行高斯滤波，使用opencv的GaussianBlur函数。高斯滤波是一种线性平滑滤波，适用于消除高斯噪声，广泛应用于图像处理的减噪过程。高斯滤波是对整幅图像进行加权平均的过程，每一个像素点的值都由其本身和邻域内的其他像素值经过加权平均后得到。

当用到人脸掩模时，先使用opencv的bitwise\_and函数，对图像和掩模进行按位与操作，利用掩膜进行“与”操作，即掩膜图像白色区域是对需要处理图像像素的保留，黑色区域是对需要处理图像像素的剔除，其余按位操作原理类似只是效果不同而已。然后利用np.mean函数，对图像的第0和1维共同展成的二维平面进行求均值。

最后利用np.clip函数，对图像颜色进修改。

实现代码：

def skin\_color\_adjustment(image1, image2, mask = None):

    if mask is None:

        image1\_ksize = 55

        image2\_ksize = 55

        image1\_factor = cv2.GaussianBlur(image1, (image1\_ksize, image1\_ksize), 0).astype(np.float)

        image2\_factor = cv2.GaussianBlur(image2, (image2\_ksize, image2\_ksize), 0).astype(np.float)

    else:

        image1\_face\_image = cv2.bitwise\_and(image1, image1, mask = mask)

        image2\_face\_image = cv2.bitwise\_and(image2, image2, mask = mask)

        image1\_factor = np.mean(image1\_face\_image, axis = (0, 1))

        image2\_factor = np.mean(image2\_face\_image, axis = (0, 1))

    image1 = np.clip((image1.astype(np.float) \* image2\_factor / np.clip(image1\_factor, 1e-6, None)), 0, 255).astype(np.uint8)

    return image1

* 1. 换脸的实现步骤

实现了所有要使用的函数后，利用这些函数，开始进行换脸。

1. 获取图片1和图片2的特征点，
2. 获取图片1和图片2的掩模
3. 图片1依据图片2进行仿射变换
4. 图片1的人脸掩模依据图片2进行仿射变换
5. 对图片2的人脸掩模和(4)得到的仿射变换掩模，进行掩模合并
6. 将(3)得到的仿射变换图片，依据图片2和(5)得到的合并掩模，进行人脸肤色调整
7. 获取(4)得到的掩模的中心点
8. 使用图片2，(5)得到的合并掩模，(3)得到的仿射变换图片，(7)得到的掩模中心点，这些元素为参数，进行泊松融合，获得换脸后的照片。

所以，最关键的步骤，就是最后利用泊松融合实现换脸。图像融合是图像处理的一个基本问题，目的是将源图像中一个物体或者一个区域嵌入到目标图像生成一个新的图像。在对图像进行合成的过程中，为了使合成后的图像更自然，合成边界应当保持无缝。但如果源图像和目标图像有着明显不同的纹理特征，则直接合成后的图像会存在明显的边界。

针对这个问题，有人提出了一种利用构造泊松方程求解像素最优值的方法，在保留了源图像梯度信息的同时，融合源图像与目标图像。该方法根据用户指定的边界条件求解一个泊松方程，实现了梯度域上的连续，从而达到边界处的无缝融合。

总的来说，将泊松方程引入图像融合后，接下来的操作很容易在梯度域中进行应用，并且可以通过局部的图像编辑得到全局融合的效果。我们都知道，图像梯度的最重要性质是梯度可以用来反映图像中亮度改变最明显的区域，也就是说可以用梯度来捕捉图像上的亮度变化，梯度的方向在图像灰度的最大变化率上，它恰好可以反映出图像边缘的灰度变化。

实现泊松融化，要使用到opencv库的seamlessClone函数。函数原型为：cv2.seamlessClone(obj, im, mask, center, cv2.NORMAL\_CLONE)。

obj代表的是子图，由cv2读进来的数组文件；im代表的是母图，也是由cv2都进来的数组文件；mask代表掩模，并不需要把子图所有的部分都贴进来，所以可以用mask划分出一个兴趣域。center表示坐标，打算在母图的哪个位置放子图；cv2.NORMAL\_CLONE代表融合的模式。

实现代码：

feature\_points1 = face\_feature\_points(image1)

image1\_mask = face\_mask(image1\_size, feature\_points1)

feature\_points2 = face\_feature\_points(image2)

image2\_mask = face\_mask(image2\_size, feature\_points2)

affine\_image1 = affine\_image(image1, image2, feature\_points1, feature\_points2)

affine\_image1\_mask = affine\_image(image1\_mask, image2, feature\_points1, feature\_points2)

union = mask\_union(image2\_mask, affine\_image1\_mask)

affine\_image1 = skin\_color\_adjustment(affine\_image1, image2, mask=union)

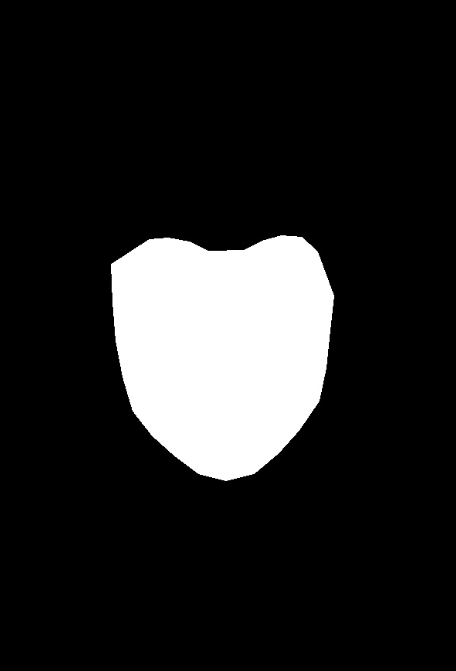
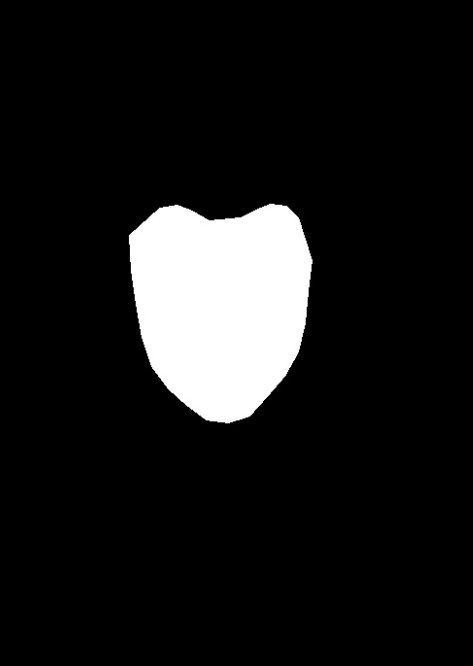
center\_point = mask\_center\_point(affine\_image1\_mask)

seamless\_image = cv2.seamlessClone(affine\_image1, image2, mask = union, p = center\_point, flags = cv2.NORMAL\_CLONE)

（a）原脸 （b）被换脸

图8 进行换脸的两张原图

（a）图8a的人脸掩模 （b）图8b的人脸掩模

图9 两张人脸掩模

1. 实验结果和分析

先获取图8a和图8b的人脸掩模，可以看到，人脸掩模按照68个特征点完美地匹配获取了。

然后，获取图8a依据图8b进行仿射变换得到图片10。

再然后，获取图9a和图8b进行仿射变换得到图片11。

再然后，获取图9b和图11进行掩模

合并得到图12。

再然后，获取图10依据图8b进行肤色调整得到图13。

最后，依据图10，图8b，图12，图11的中心点，进行泊松融合，获得实验结果，换脸后的图14。

可以看到，换脸是十分成功的，利用泊松融合，边缘的融合处理十分完美。

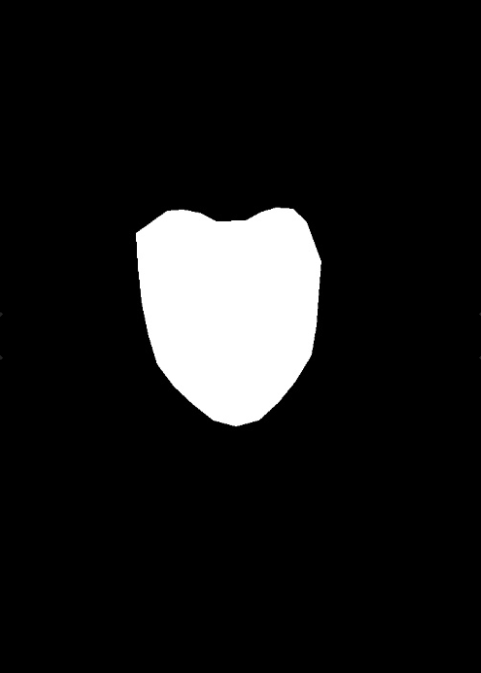
 

图10 图8a依据图8b进行仿射变换 图11 图9a和图8b进行仿射变换

图12 图9b和图11进行掩模合并 图13 图10依据图8b进行肤色调整



图14 泊松融合换脸后的成果

1. 总结

通过这次大作业，了解了大量关于智能换脸的知识。学会了如何使用paddlehub库和opencv库来实现简单初步的智能换脸技术。还了解了现在市场上最热门的智能换脸应用软件“ZAO”的技术原理。

换脸技术，从90年代开始已经出现，随着计算机行业的发展，换脸技术越来越成熟，换脸的技术门槛也逐渐下降，不再需要什么好莱坞特效大公司，只需要一台电脑或者手机就可以实现智能换脸。智能换脸技术的发展是越来越快，但是智能换脸的应用要被制约，不能被用于违法行为，这需要不断地努力研究。让智能换脸技术真正地服务大众，为大众提供遍历，而不是让智能换脸技术成为侵犯大众权益的帮手。

参 考 文 献

[1] 叶晓娴，王润彬，从“换脸”技术看人工智能背景下影视产业的发展