

# 基于几何变换模型的换脸软件设计

郑有为<sup>1)</sup>

<sup>1)</sup> 中山大学计算机学院 广州 510000

**摘 要** FACE SWAP 是一个基于 Python 编写, 使用到了 OpenCV, dlib 等图像处理库的智能换脸软件, 核心代码在 300 行内, 并相应地编写了图形界面。本篇论文阐述了换脸软件 FACE SWAP 的模型原理、设计过程, 并在从不同角度对软件的换脸效果进行测试, 该软件借鉴了参考开源代码, 对其核心算法部分进行了优化, 并设计一个简洁, 易于使用的用户界面。

目前的换脸软件背后的模型主要分为两类, 一类是基于数学变换的换脸软件, 另一类是基于深度学习网络、需要训练模型的换脸软件。FACE SWAP 考虑到硬件设备的限制和使用低门槛的初衷, 选择了前者作为基本的模型框架。FACE SWAP 模型是基于几何变换的, 由人脸检测阶段和面部替换阶段组成, 人脸检测阶段的步骤主要为: 人脸捕获、人脸选择和面部关键点检测, 面部替换阶段的步骤主要为面部仿射变换、掩膜生成、色彩校正(可选)和无缝接合。

论文中, 我们首先详细介绍了该软件的模型和实现(第三节内容), 然后从不同的侧面测试 FACE SAWP 换脸的效果(第四节内容), 除常规的换脸测试外, 包括侧面面部测试、肤色差异测试、遮挡物测试和不规则光影测试, 从多个角度衡量该软件换脸的实际效果。

**关键词** 图像处理; 仿射变换; 智能换脸; 人工智能; 计算机视觉

**中图法分类号** TP31

## Design of face changing software based on triangle affine transformation model

Youwei Zheng<sup>1)</sup>

<sup>1)</sup>(School of Computer Science, Sun Yat-sen University, GuangZhou 510000)

**Abstract** FACE SWAP is a Python written, using OpenCV, dlib and other image processing libraries intelligent face replacement software. The core code is in 300 lines, and it has a corresponding written user graphical interface. This paper describes the model principle and design process of FACE SWAP, and tests the effect of FACE SWAP from different angles. The software uses the open source code for reference, optimizes its part of core algorithm, and designs a simple and easy to use user interface.

The models behind the current face-swapping software are mainly divided into two categories: one is face-swapping software based on mathematical transformation, and the other is face-swapping software based on deep learning network that needs to train the model. Considering the limitations of hardware devices and the original intention of using low threshold, FACE SWAP chose the former as the basic model framework. FACE SWAP model is based on geometric transformation and consists of face detection stage and face replacement stage. The steps of face detection stage mainly include face capture, face selection and face key point detection. The steps of face replacement stage mainly include face affine transformation, mask generation, color correction (optional) and seamless bonding.

In the paper, we first introduced the model and the implementation of the software (third section), then test from different sides FACE SAWP in face of effect (section 4 content), in addition to the regular in face of the test, including side face, skin color difference testing, mask and irregular light testing, the measure of this software from multiple angles in face of actual effect.

**Key words** Image processing; Affine transformation; Intelligent face change; Artificial intelligence; Computer vision

## 1 选题背景及意义

换脸指将一个人的脸与另一个人、动物或无生命物体的脸进行交换的行为。在狭义上,换脸指的是媒体中的某人脸被替换成另一个人脸。而在广义上人脸可以被换入任何物体、任何部位,即便不合常理。

智能换脸,就是通过 AI 人工智能技术进行换脸,而且保证效果脸部表情自然,效果逼真。

传统上是采用基于图形学的 3D 模型重建追踪技术。较新的研究则是采用深度学习来达到换脸效果,为了解决深度学习的训练难度和生成质量,又进一步融合了生成对抗网络技术。

智能换脸往往与表情伪造结合在一起,表情伪造是将其他人脸图像的表情替换到目标人脸上,从而达到目标人物做指定表情的目的。此外,换脸伪造和表情伪造还常常结合语音伪造技术。通过文本到语音合成和语音转换技术来制作虚假语音。[1]

图形和深度学习换脸各有优势,前者易于实现,无需大量数据集,对硬件设备要求低;后者能提供理论上更好的生成图片质量,在表情、不同层面等细节方面处理的更合理,但训练模型较为耗时,且对计算设备要求高,普通群众难以很好地使用。

### 1.1 换脸技术的历史

换脸的做法始于 21 世纪初的 Photoshop。2012 年,当一个美国的新闻聚合网站 BuzzFeed 开始发布一些引人注目的例子时,它开始引起了社会对换脸的关注。[2]

自动换脸应用程序的出现让换脸变得更加简单,使用者无需对人脸图片进行拼接,转而由计算机自动处理。早在 2014 年,安卓系统推出了一款名为“Face Swap”的应用程序,用户可以轻松地用手机拍摄的照片换脸。Face Swap Live 在这项技术上进行了创新,使视频和照片的即时交换成为可能。2016 年,Snapchat 向其超过 1.5 亿的日活跃用户发布了一款换脸滤镜,显著推动了这一趋势。这些应用程序有时会失败,导致幽默的错误,为更多的表情包提供支持。[2]

例如,许多病毒式的人脸交换会把婴儿的脸和祖父母的脸交换,或者把狗的脸和主人的脸交换,创造出搞笑或令人不安的图像。还有一些更有创意的换脸镜头,比如篮球明星迈克尔·乔丹(Michael Jordan)在自己的脸上扣篮。这些应用程序的流行使得“换脸”成为这种现象最常见的

术语。

今天,我们可以在开源网站 Github 找到许多换脸开源项目,包括著名的 Deepfakes[3]、Deepfacelab[4]、ALAE[5],国内也有成熟的换脸应用 ZAO,但功能比较局限。

### 1.2 AI 融入生活,智能换脸火热

今天,人工智能不断改变着我们的生活,从人脸识别门锁到智能机器人、无人驾驶,人工智能在将许多“不可能”变为可能。人工智能该我们带来的不仅是生活方式的转型,更还有思考方式和娱乐模式的转变。

随着整个社会生活条件的改善,人们对娱乐消遣有了新的追求:参与感、新奇感、多样性。并且,人们的娱乐方式在从“现实娱乐”向“虚拟娱乐”转变,即人们通过手机、电脑来玩游戏、看视频以消遣。数字媒体正在成为人们特别是年轻人娱乐消遣的核心。

智能换脸在国内掀起一阵热潮,近期视频网站哔哩哔哩某视频发布者发布的合成视频,引起病毒式传播。

### 1.3 智能换脸选题意义

智能换脸选题有以下两点意义:

1. 市场上,现有的开源换脸软件的效果好,但门槛高,使用时需要使用者有一定的编程语言基础和对应的程序运行环境。同时,这些换脸程序往往需要通过训练模型来生成换脸图片和视频,训练模型需要大量的图像或视频,训练的设备要求也较为严格,在普通电脑上训练的训练时间难以忍受,普通用户难以使用以及进行创造。

因而开发一款简单实用,对设备要求不高的,在效果与速度之间平衡的换脸软件是十分必要的。

2. 换脸应用已经在影视方面有应用,例如在电影《速度与激情 8》中,换脸就被用在一名已故演员上,换脸应用能为影视作品、视频娱乐、自媒体提供新的思路,并且具有广大的市场。

### 1.3 换脸技术的争议

深度学习在计算机视觉领域取得了重大成功,超越了众多传统的方法,然而换脸技术被用在不当的地方。这种深度伪造技术通过篡改或替换原始视频的人脸信息,并合成虚假的语音来制作色情电影、虚假新闻、政治谣言等。[6]

为了消除此类伪造技术带来的负面影响,众多学者对假视频的鉴别进行了深入的研究,并提出一系列的检测方法去识别此类伪造图像和视频,Facebook 每年会联合微软

举办全球 Deepfakes 检测竞赛。

## 2 现有换脸方法回顾

目前，换脸技术主要分为两类：一类是基于图形学的换脸，另一类是基于学习的换脸。

### 2.1 图形学换脸技术

Zollhofer 等人阐述了比较主流的 3D 模型重建追踪等技术。[7]

开源应用 FaceSwap 基于图形学的换脸方法,其主要分为两个阶段：面部特征提取（包括检测、对准、掩膜生成和对其文件四个子阶段）和模型训练。

FaceSwap 换脸大致步骤如下：首先获取人脸关键点,然后通过 3D 模型对获取到的人脸关键点位置进行渲染,不断缩小目标形状和关键点定位间的差异,最后将渲染模型的图像进行混合,并利用色彩校正技术获取最终的图像。[6]

Nirkin 等人用分割的思路促进换脸,通过网络分割出来的人脸估计 3D 人脸形状,最后融合源和目标这两个对齐的 3D 人脸形状。[8]

### 2.2 基于学习和训练的换脸技术

随着深度学习技术的飞速发展,研究者们开始关注深度学习在人脸篡改上的应用。

Deepfakes 是网络上较早开源的基于深度学习的换脸项目,Deepfakes 的核心结构是编码解码结构,其工作原理大致如下：。

通过训练两个自动编码器,两个编码器共享权重参数,使得两个解码器学会重建人脸的能力。训练结束后,在换脸阶段,交换两个解码器,从而使得换脸效果达成。[6]

GAN 生成对抗网络技术也被用于换脸技术。Faceswap-GAN 就是增加了 GAN 技术的 Deepfakes,引入判别器的对抗损失函数,在生成的时候判别生成图像和原图的相似度,使得生成的图像质量有大幅度提高,另外引入了感知损失函数增加眼珠的转动效果。GAN 生成对抗网络的引入让换脸效果更加逼真,更加自然。[6]

从 GAN 网络延伸出的 CycleGAN 生成对抗网络模型,Face2Face 模型、Pix2pPix 模型、Vid2Vid 模型作为图像或者视频的生成模型,也能用在换脸模型的训练上。

### 2.3 最近提出的换脸模型

#### 2.3.1 Face Shifter 模型

论文[9]提出了一个基于二阶段的 GAN 生成对抗网络

方式的换脸方法 FaceShifter。整体网络结构由两层自网络：AEI-Net, HEAR-Net 组成。FaceShifter 通过获取输入的两张图片中第一张图片的 ID 和第二张图片的属性,通过 GAN 生成对抗网络生成最终的合成图片。合成的图片具有第一个图片的 ID,第二个图片的属性,从而实现了换脸。

对 ID 和属性进行了区分,Face Shifter 找到了一个处理面部遮挡的方式,通过寻找重建后的面部图像与原图像的差异,来提供关于遮挡位置的线索。

#### 2.3.2 ALAE 模型

ALAE (Adversarial Latent Autoencoder)[5]是一种新型自编码器,也是一种无监督方法。ALAE 采用的是自编码器(AE)而不是生成对抗网络(GAN)。

ALAE 功能强大,图像中人脸的性别、年龄、笑容、丰满程度、嘴唇和鼻子的大小等都是可以调节的。

ALAE 有两个自编码器：一个是基于 MLP 编码器,另一个基于 StyleGAN 对抗神经网络的生成模型,也可以称之为 StyleALAE。

其中,自编码器指的是一种神经网络,经过训练后能尝试将输入复制到输出,该网络可以看作由两部分组成：一个由函数  $h = f(x)$  表示的编码器和一个生成重构的解码器  $r = g(h)$

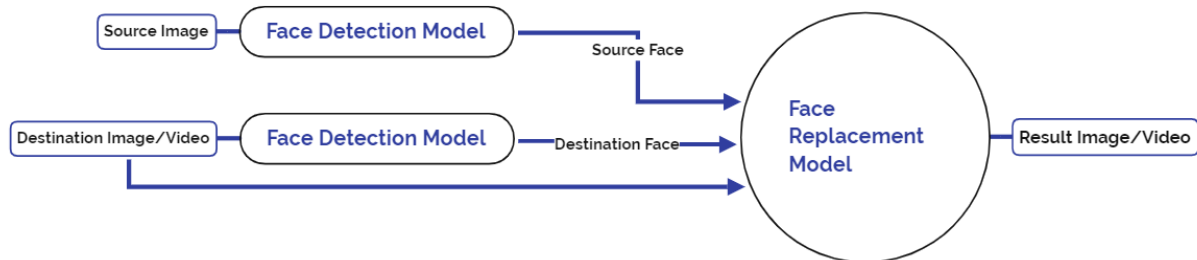
## 3 换脸模型介绍

换脸软件 FACE SWAP 是基于几何变换来实现的,整个换脸算法分为大体分为两个阶段：第一个阶段是人脸检测阶段,第二个阶段是面部替换阶段,整体的换脸模型如图 1 所示。

### 3.1 人脸检测模型

人脸检测模型输入为一张图片,输出为 68 个人脸关键点、人脸在图像中的位置和人脸子图。人脸检测模型基于开源的机器学习算法库 dlib 和人脸关键点数据集 shape\_predictor\_68\_face\_landmarks.dat [10] (已附加在软件压缩包中),人脸检测的三个主要过程是：人脸捕获、人脸选择和面部关键点检测,其中人脸捕获和面部关键点识别均基于 dlib 和数据集实现。

## FACE SWAP



模型将人脸捕获的捕获器命名为 Face Detector，它的作用是从将输入的一张图片中检测出所有人脸；并将人脸选择的选择器命名为 Face Selector，作用是选择一张目的人脸，它可以是源图的人脸，也可以是将目的图的人脸；将 68 个面部关键点（也称为面部特征点）检测的检测器称为 Face Predictor，命名为 Predictor 的原因是关键点识别本质上是通过数据集和图像通过 dlib 库的形状预测器预测得

图 1 换脸软件 FACE SWAP 模型图 1 换脸软件 FACE SWAP 模型到的。

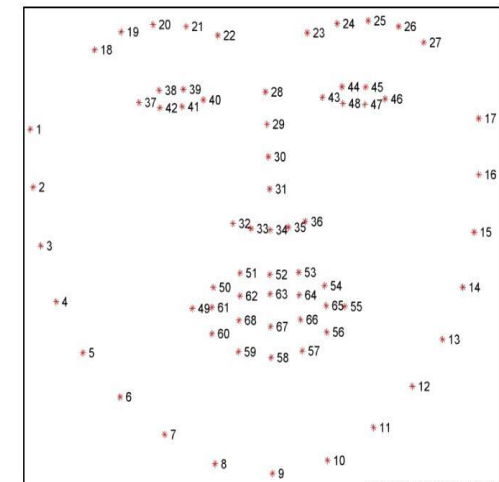


图 2 人脸 68 个关键点分布

在 68 个面部关键点中，第 1-17 个关键点连接成了人脸的面部下轮廓，第 18-22 个关键点和第 23-27 个关键点分别概括了人的左眉毛和右眉毛的位置和形状特征，第 37-42 个关键点和第 43-48 个关键点分别围绕出了人的左眼和右眼的位置和形状特征，第 28-36 个关键点连接成了人整个鼻梁的形状，剩余的点描述的是人的嘴唇的形状，并对上嘴唇和下嘴唇进行了划分，位置信息反映了嘴的大小和开合程度。

人脸捕获模型如下图所示：

## FACE SWAP Face Detection

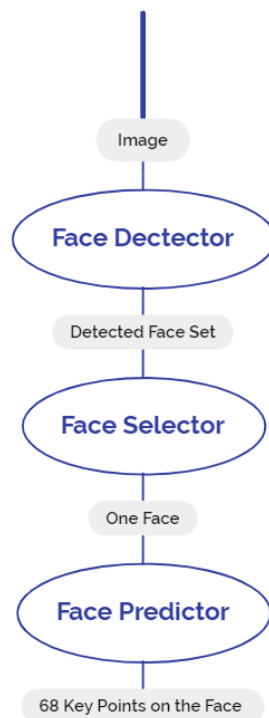


图 3 人脸检测模型

### 3.1.1 基于 dlib 的人脸捕获

我们使用 dlib 提供的库函数 `get_frontal_face_detector()` 以创建一个捕获器，以图片和上采样数作为参数捕获人脸并返回人脸矩阵的集合。

其中，上采样数（`upsample times`）代表将原始图像放大的倍数，1 表示放大 1 倍再检查，提高较小人脸的检测效果。

人脸捕获在 `detect.py` 文件的 `keypoints_detection(image, face:dlib.rectangle)` 函数中实现。

### 3.1.2 人脸选择

完成人脸捕获后，我们获得图片中所有人脸的集合，我们默认选取在原图中像素面积最大的人脸，每个人脸用包含在一个小的面部矩形中，我们对人脸像素面积采取近似计算，人脸像素的面积用面部矩形的面积近似，即矩形的长乘宽。

人脸选择实现与 detect.py 文件的 face\_selection 函数，在实现中，我们还提供了用户选取人脸的功能，即弹出图片让使用者通过鼠标点击人脸已选中指定人脸，但这项功能未被加入到最后的 FACE SWAP 用户接口中。

### 3.1.3 基于 dlib 的面部关键点检测

面部关键点的检测基于 dlib 官网提供的数据集 shape\_predictor\_68\_face\_landmarks.dat，通过调用关键点预测器函数 dlib.shape\_predictor() 以获取给定面部的 68 个关键点的位置，最后转化为二维矩阵输出，每列元素即为一个关键点在源图中的坐标。

该功能于 detect.py 文件中的 keypoints\_detection(image, face:dlib.rectangle) 函数中实现。

以上是人脸检测模型的三个模块，即捕捉器、选择器和关键点检测器，主要基于 dlib 库和 dlib 提供的面部关键点数据集实现，一并实现在 detect.py 文件中。

## 3.2 面部替换模型

以下是面部替换模型，分为四个阶段：面部仿射变换、掩膜生成、色彩校正（可选）和无缝接合。其中面部仿射变换提供 2D 变换和 3D 变换，其中 3D 变换能够将替换的五官进行变形，以更好地匹配原面部的五官位置和形态，面部替换模型如图 4 所示。

### 3.2.1 2D 仿射变换

仿射变换，指在几何中一个向量空间进行一次非奇异的线性变换并接上一个平移，变换为另一个向量空间。

[11]

在三维视觉技术中大量应用到这些仿射变换以实现图形的平移，缩放，旋转，翻转和错切。程序中可利用 cvWarpAffine 解决稠密仿射变换，或用 cvTransform 解决稀疏仿射变换。仿射变换可以将矩形转换成平行四边形，它可以将矩形的边压扁但必须保持边是平行的，也可以将矩形旋转或者按比例变化。

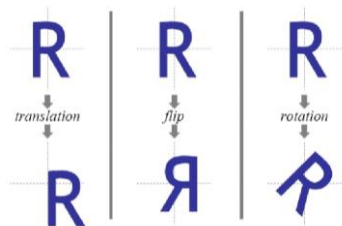


图 3-1 简单仿射变换示例

## FACE SWAP Face Replacement



图 4 面部替换模型

在有限维下，仿射变换可以由一个矩阵  $A$  和一个向量  $b$  给出，一个仿射变换对应于一个矩阵和一个向量的乘法，而仿射变换的复合对应于普通的矩阵乘法。

在实现上，我们先通过计算出仿射变换矩阵  $M$ ， $M$  的计算公式如下：

$$\begin{aligned}
 &\text{令两个点列为 } P_1, P_2 \\
 &C_1 = \sum P_1/n, C_2 = \sum P_2 \\
 &S_1 = \sqrt{\frac{\sum (P_1 - C_1)^2}{n}}, S_2 = \sqrt{\frac{\sum (P_2 - C_2)^2}{n}} \\
 &P'_1 = \frac{P_1 - C_1}{S_1}, P'_2 = \frac{P_2 - C_2}{S_2} \\
 &A = P'^T_1 \cdot P'_2 \\
 &A = U \Sigma V^T \text{ (奇异值分解)} \\
 &R^T = U \cdot V \\
 &A_1 = \frac{S_2}{S_1} R \\
 &A_2 = C_2^T - A_1 \cdot C_1^T \\
 &M = [[A_1, A_2]^T, [0, 0, 1]^T]^T
 \end{aligned}$$

在计算出仿射变换矩阵  $M$  之后，调用 OpenCV 的 cvWarpAffine 计算出仿射后的面部，cvWarpAffine 的函数

原型为：`cv2.warpAffine(src, M, dsize[, dst[, flags[, borderMode[, borderValue]]]])` → `dst`，其中 `src` 为输入图像、`M` 为变换矩阵、`dsize` 是输出图像的大小、`flags` 选定插值方法，`borderMode` 和 `borderValue` 分别指定边界像素模式和边界填充值。在现实，我们将边界像素设置为透明 `cv2.BORDER_TRANSPARENT`，将边界填充设置为 `cv2.WARP_INVERSE_MAP` 已指定输出图像到输入图像的反变换。

`convert.py` 文件的 `points_transformation(points1, points2)` 函数就是用于计算两个点列的仿射变换矩阵，两个点列分别是源图面部关键点和目标图面部关键点。2D 仿射变换是基于整个面部的变换，不改变关键点的相对位置。

### 3.2.2 3D 仿射变换

3D 变换不同于 2D 变换，它会调整面部关键点的相对位置，以使得两个面部更加匹配，让生成面部同时具有源面部和目的面部的特征。

在实现上，不同于 2D 变换，我们使用的是三角变换仿射。我们首先引入面部矩形区域的概念，并引入双线性插值来处理每一个三角形。

我们首先通过 `scipy.spatial.Delaunay` 在面部关键点中建立三角形网络，获得三角形网络根据这些三角形关系构建三角仿射变换矩阵（该部分实现于 `convert.py` 文件中的 `triangular_affine_matrices(vertices, src_points, dst_points)` 函数）。

计算三角仿射变换矩阵的步骤包括：遍历所有三角网络中的单元，找到于源面部关键点和目的面部关键点的位置关系，最后计算源三角矩阵和目的三角的逆矩阵的内积。

然后，获得目的面部的面部矩形区域，即涵盖整个面部的矩形像素区域（该部分实现于 `convert.py` 文件中的 `grid_coordination(points)` 函数）。再通过 `scipy` 库提供的 `find_simplex` 函数找到面部矩形区域中面部关键点三角的位置，对于每个三角，通过双线性插值进行大小的调整。

双线性插值是有两个变量的插值函数的线性插值扩展，其核心思想是在两个方向分别进行一次线性插值，相较于一次插值，能更好地处理缩放图像过程中的颜色过渡，以下是双线性插值的推导过程：

$$\begin{aligned} & \text{令点列为 } P, \text{ 像素点 } (x, y) \text{ 的像素值为 } Img(x, y) \\ & X = \lfloor P_x \rfloor, Y = \lfloor P_y \rfloor \\ & dX = P_x - X, dY = P_y - Y \\ & P_{top} = (1 - dX)Img(Y, X)^T + (dX)Img(Y, X + 1) \\ & P_{bottom} = (1 - dX)Img(Y + 1, X)^T + (dX)Img(Y + 1, X + 1) \\ & P_{interpolation}^T = (1 - dY)P_{top} + (dY)P_{bottom} \end{aligned}$$

### 3.2.3 掩膜生成模块

掩膜本身是由 0 和 1 组成的一个二进制图像。当在某一功能中应用掩膜时，1 值区域被处理，0 值区域被屏蔽。通过指定的数据值、数据范围、有限或无限值、感兴趣区和注释文件来定义图像掩膜，也可以应用上述选项的任意组合作为输入来建立掩膜。使用掩膜，能将目的图像面部部分过滤掉，以便于换入面部。

在实现上，我们通过 `OpenCV` 提供的 `cv2.convexHull` 函数来根据目标面部关键点来生成面部掩膜区域，并借助 `cv2.fillConvexPoly` 函数来应用掩膜，将该部分过滤掉。

对于掩膜边缘，可以进行腐蚀处理。侵蚀如同土壤侵蚀，侵蚀了前景物体的边界(始终尝试使前景保持白色)，以此优化掩膜的边界，以使结果图像更加自然。我们通过调用 `cv2.erode` 来对掩膜进行腐蚀。

### 3.2.4 色彩校正模块

我们使用高斯滤波对肤色进行调和和处理，并引入模糊参数 `blur factor`，默认值为 0.75。模糊参数在设置为 0 是不进行颜色校正，设置为 1 时校正程度最高。

我们分别对源图片和目的图片进行高斯滤波处理。高斯滤波是一种线性平滑滤波，适用于消除高斯噪声，应用于图像处理的减噪过程，高斯滤波就对整幅图像进行加权平均，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。

在实现上，我们先计算两眼的平均距离和范数。该范数（偶数，非偶数则加一）作为高斯滤波矩阵的行数和列数。借助 `OpenCV` 的 `GaussianBlur` 函数实现高斯滤波，获得源图片的模糊量（`src_blur`）和目标图片的模糊量（`dst_blur`），然后对目的图片进行一下数学处理：

$$Img_{result} = Img_{dst} \cdot \frac{Blur_{src}}{Blur_{dst}}$$

### 3.2.5 无缝接合模块

我们使用 `OpenCV` 内置函数 `seamlessClone` 将变换后的脸、目标图像和掩膜进行无缝结合，从而生成换脸图像，自此图像换脸完成。

## 3.4 视频换脸处理

对于视频换脸，我们将视频的每一帧取出来，将其视为图像进行换脸，换脸后重新拼接成视频，视频格式默认为 `avi`。

视频读取和换脸在 `gui.py` 文件中的 `VideoHandler` 视频句柄类中实现。



## 4 实验结果和分析

以下是 FACE SWAP 软件界面：

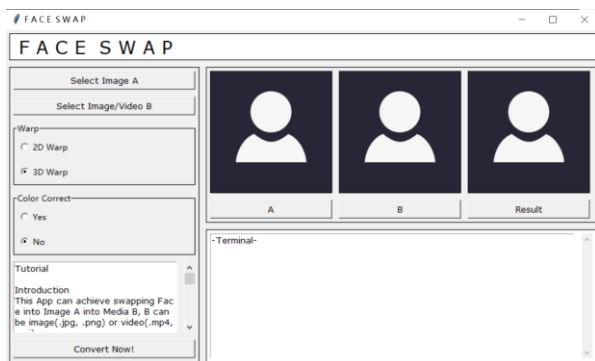


图 5 FACE SWAP 软件界面

软件提供以下功能：这是一个使用 Python 编写的换脸软件，所提供的换脸功能包含以下两个场景：将图像 B 上的人脸换成图像 A 上的人脸，生成换脸图片；将视频 B 上的人脸换成图像 A 上的人脸，生成换脸视频。

在换脸效果上，软件提供可选选项：包括图像变换的方法（2D 变换和 3D 变换）和是否进行色彩调和。

在图形界面上（实现文件：`gui.py`），提供生成的图片、视频预览功能，生成的图像视频会自动保存到`out`文件夹。

### 4.1 正脸换脸效果测试

相对而言，3D Warp 面部变化的效果由于 2D Warp，2D 在边缘区域处理得不是很好；使用肤色调和（Color Correction）的效果也往往优于不使用肤色调和。

选取以下两张图片进行图片换脸：

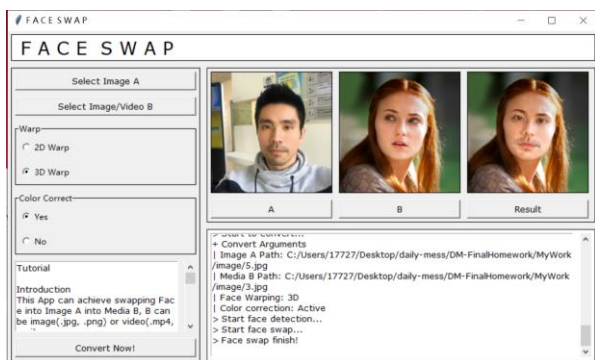


图 6 3D Warp + Color Correction 测试

不同模式下换脸效果如下：

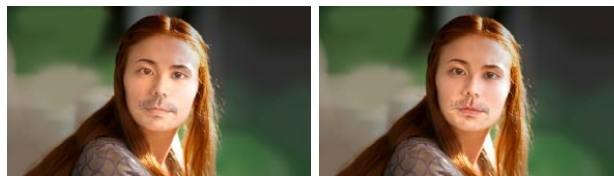
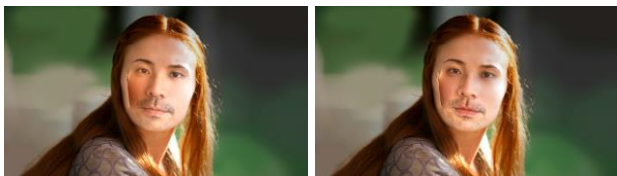


图 7 不同模式下换脸效果

左上角为 2D 变换+无颜色校正，右上角为 2D 变换+颜色校正，左下角为 3D 变换+无颜色校正，右下角为 3D 变换+颜色校正。可以看到 3D 变换效果更好，因此我们在软件介绍中也极力推荐 3D 变换换脸，2D 换脸在边缘处做的不是很好。

### 4.2 侧脸换脸效果测试

以如下两张图片面部偏侧面的进行换脸测试，比较不同选项的效果：



图 11 3D Warp + Color Correction 测试

通过侧面换脸测试，可以看到 2D 变换的效果不如 3D，特别是在源图与目标图五官不对准时；而使用颜色校正能够使换脸后的图像在光线明亮上更加自然。

下图为生成效果比较：2D 变换在左眼和右眼的变换上比较失败，而 3D 仿射变换很好地解决了这个问题。



图 12 不同模式下换脸效果

左上角为 2D 变换+无颜色校正，右上角为 2D 变换+

颜色校正, 左下角为 3D 变换+无颜色校正, 右下角为 3D 变换+颜色校正。

交换原图像和目的图像, 可以看到 2D 变换的另一个不足, 即在边缘的处理上不够自然, 3D 在面部调整上不一定总能达到最自然的效果, 但对边缘处理的很好。

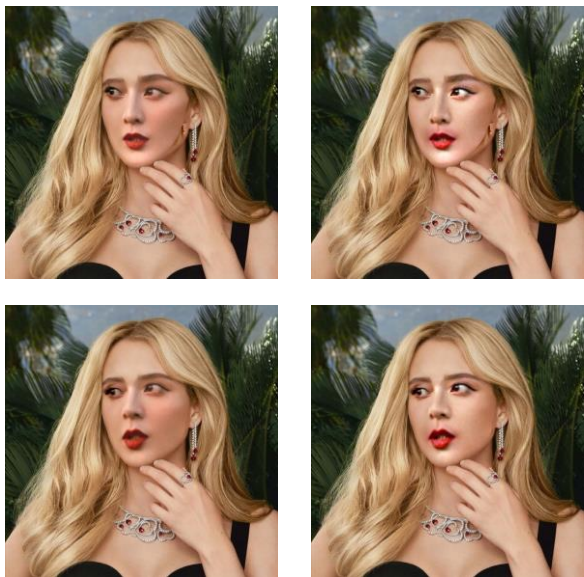


图 13 不同模式下换脸效果

左上角为 2D 变换+无颜色校正, 右上角为 2D 变换+颜色校正, 左下角为 3D 变换+无颜色校正, 右下角为 3D 变换+颜色校正。

以下, 总结出 2D 变换和 3D 变换的优缺点:

表 1 2D 变换和 3D 变换特点比较

	2D 变换	3D 变换
面部边缘处理	差	优
五官贴合图片	中等	优
五官自然	优	中等

#### 4.3 肤色差异换脸效果测试

选取不同肤色的人进行换脸测试, 换脸结果很好地保留了目的图片的肤色特征:



图 14 不同肤色换脸测试样例

下图为换脸结果:

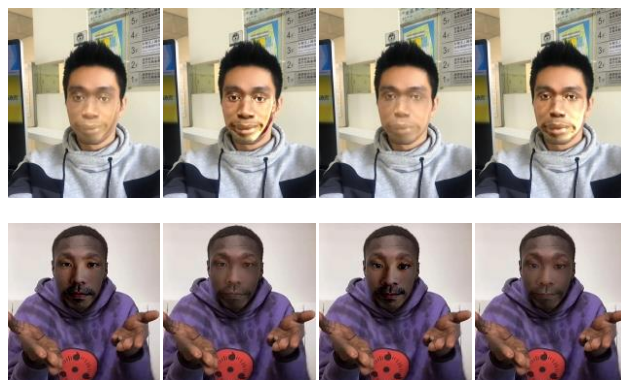


图 15 不同肤色换脸测试效果

我们发现从深色换到浅色皮肤的情况下, 不适用颜色校正的效果优于使用颜色校正, 使用颜色校正会使得换脸部分亮度偏高。

#### 4.4 遮挡物换脸效果测试

在模型中没有对遮挡物做鉴别和处理, 但人脸检测部分仍然可以捕获并识别到受眼镜、甚至口罩遮挡的面部, 在遮挡物处理这一方面还有大的改进空间。

两组测试遮挡物 (眼镜): 第一行为两组原图片和目的图片, 第二行和第三行的每一列分别为 2D 变换+无颜色校正、2D 变换+颜色校正、3D 变换+无颜色校正、3D 变换+颜色校正选项下的换脸结果。

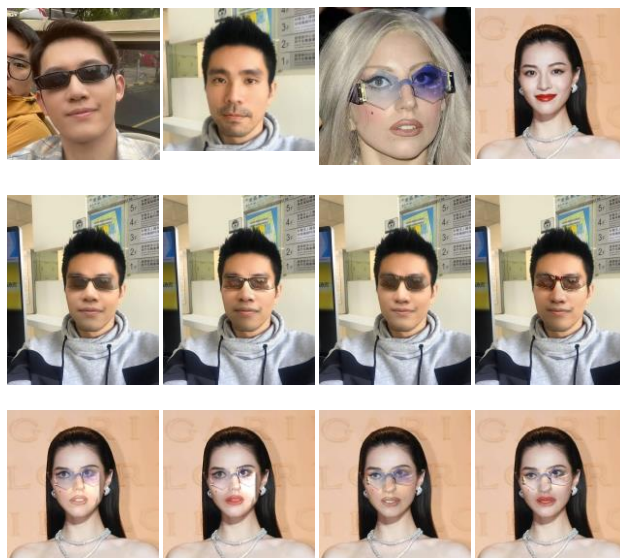


图 16 遮挡物换脸测试

除遮挡物遮挡外, 我们对不规则阴影的背景图进行测试:





图 17 面部阴影换脸样例

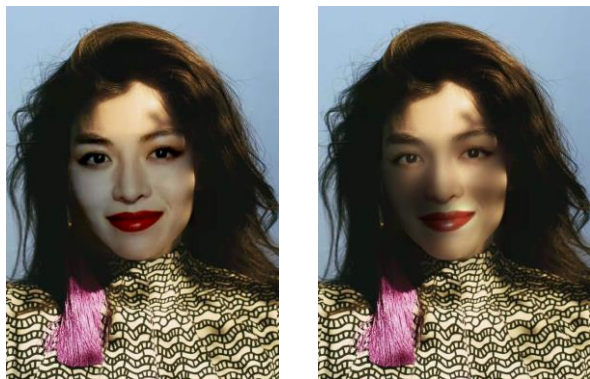


图 18 面部阴影换脸结果

在开启颜色校正后，保留了目的图片自身的光影，效果较好。

#### 4.5 视频换脸效果测试

视频换脸效果与图片换脸无异，但较为分帧处理较为耗时，下图为视频处理结果：

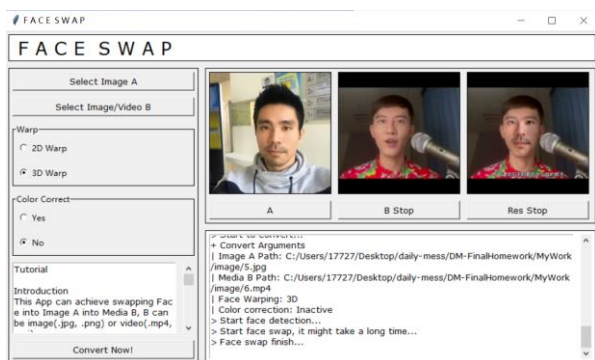


图 18 视频换脸结果

## 5 总结

本篇论文阐述了换脸软件 FACE SWAP 的模型原理、设计过程，并在从不同角度对软件的换脸效果进行测试，该软件借鉴了参考开源代码[12]，对其核心算法部分进行

了优化，并设计一个简洁，易于使用的用户界面。在模型构建之处，考虑到设备支持的限制和方便使用，快速换脸的初衷，在 AI 学习深度换脸和传统数学变换换脸中选择了后者。在测试部分中，也验证了它的效果。

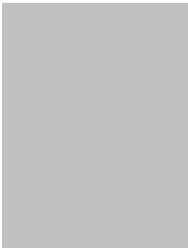
FACE SWAP 模型由人脸检测阶段和面部替换阶段组成，人脸检测阶段的步骤主要为：人脸捕获、人脸选择和面部关键点检测，面部替换阶段的步骤主要为面部仿射变换、掩膜生成、色彩校正（可选）和无缝接合。

换脸技术是图像处理，计算机视觉的一项成功的技术成果，尽管在受限于技术门槛和道德争议，但也发挥着起作用，催生出一系列识别伪造图像的技术。今天，换脸技术并未停止脚步，而是不断革新，FaceShifter、ALAE 等技术就是这项技术不断进步的代表。

最后，该软件仅用于学习，所有测试图片源于网络，换脸测试仅用于呈现软件效果，无恶意。

## 参 考 文 献

- [1] 维基百科: Deepfake <https://zh.wikipedia.org/wiki/Deepfake>
- [2] 字典解释: Face-swap <https://www.dictionary.com/e/memes/face-swap>
- [3] 开源项目 Deepfakes <https://github.com/deepfakes/faceswap>
- [4] 开源项目 DeepFaceLab <https://github.com/iperov/DeepFaceLab>
- [5] 开源项目 ALAE <https://github.com/podgorskiy/ALAE>
- [6] 李旭嵘; 纪守领, 吴春明, 刘振广, 邓水光, 程鹏, 杨琨, 孔祥维. 深度伪造与检测技术综述. 软件学报. 2021, 32 (2): 496 - 518 [2021-03-02]. doi:10.13328/j.cnki.jos.006140
- [7] Zollhöfer M, Thies J, Garrido P, Bradley D, Beeler T, Perez P, Stamminger M, Niessner M, Theobalt C. State of the art on monocular 3D face reconstruction, tracking, and applications. Computer Graphics Forum, 2018, 37(2): 523-550. [doi:10.1111/cgf.13382]
- [8] Nirkin Y, Masi I, Tuan AT, Hassner T, Medioni G. On face segmentation, face swapping, and face perception. In: Proc. of the 13th IEEE Int'l Conf. on Automatic Face and Gesture Recognition (FG 2018). IEEE, 2018. 98-105.
- [9] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, Fang Wen, FaceShifter: Towards High Fidelity And Occlusion Aware Face Swapping. <https://arxiv.org/pdf/1912.13457.pdf>
- [10] shape\_predictor\_68\_face\_landmarks.dat <http://dlib.net/files>
- [11] Stearns C C, Kannappan K. Method for 2-D affine transformation of images: U.S. Patent 5,475,803[P]. 1995-12-12.
- [12] 开源项目 Face Swap <https://github.com/wuhuikai/FaceSwap>



**Youwei Zheng** born in 2001, undergraduate, He has no main research interests.

### Background

The content of this paper is the intelligent face changing technology in computer vision. It can't compare with the current popular Deepfakes and ALAE, but it has the advantages of simple use and simplified model.

The software designed in this paper is written based on Python, using OpenCV, dlib and other image processing libraries. The core code is within 300 lines, and the corresponding graphical interface is written to minimize the threshold for users to use.

Implementing an intelligent face change is meaningful by the following two points:

1. On the market, the existing open source face-changing software has a good effect, but the threshold is high, and users need to have a certain programming language foundation and

the corresponding program running environment when using it. At the same time, these face-swapping programs often need to generate face-swapping pictures and videos through training models. The training models require a large number of images or videos, and the training equipment requirements are also relatively strict. The training time on ordinary computers is intolerable, and it is difficult for ordinary users to use and create. And the market for another type of face Tujfty. Therefore, it is very necessary to develop a simple and practical, not high requirements for equipment, balance between the effect and speed of face change software.

2. Face-swapping apps have already been applied in film and television. For example, in the movie Fast and Furious 8, face-swapping apps were used on a deceased actor.