人工智能应用实践

结题报告

——人脸检测任务

学号：201720121206

班级：17 级人工智能班

姓名：吕程

目录

1 DSFD....1
1.1 概括....1
1.2 网络结构....1
1.3 FEM：Feature Enhance Module....2
1.4 PAL: Progressive Anchor Loss....2
1.5 IAM：Improved Anchor Matching....3
2 Face-swap....4
2.1 概括....4
2.2 网络结构....4
2.3 转换网络....5
2.4 损失函数....6
3 实现图片及视频换脸....7
3.1 子目标....7
3.2 人脸识别任务....7
3.3 大体流程....7
3.4 实现步骤....8
3.4.1 使用 FFmpeg 工具将视频转为图片....8
3.4.2 人脸检测和校准....8
3.4.3 人脸 Encoder/Decoder 训练....9
3.4.4 人脸转换....11
3.4.5 图片转视频....11
3.5 改进之处....11
3.5.1 问题....11
3.5.2 整体架构....12
3.5.3 成因分析....12
3.5.4 改进....12
3.5.5 数据质量对最终效果的影响....13

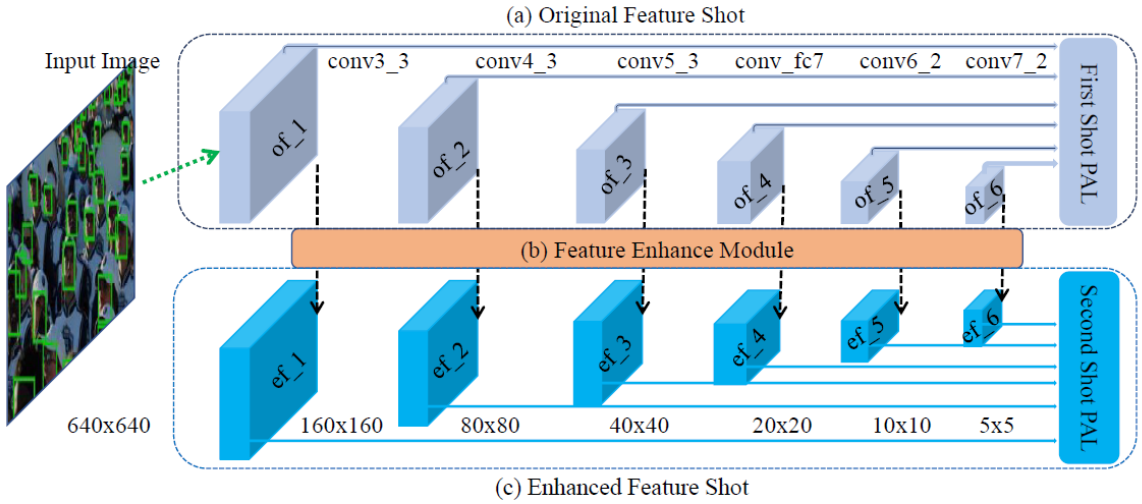
4 成果展示.....14
4.1 图片上的换脸效果.....14
4.2 视频上的换脸效果.....16
5 理解与体会.....17

1 DSFD

1.1 概括

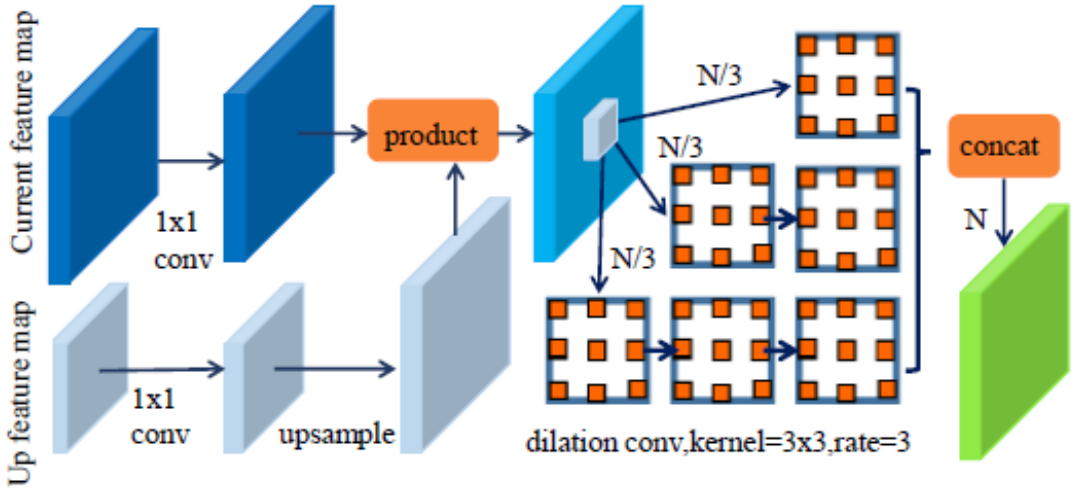
人脸检测任务目前存在的问题：尺度、姿势、遮挡、表情、外观、照明等具有高度可变性，论文三个主要创新点：一是继承 SSD 的检测框架，引入 feature enhance module（FEM），用于传输原始特征图来将单发检测器扩展到双射检测器；二是采用通过两组 anchor 计算的 progressive anchor loss（PAL）来有效促进特征；三是通过在 DSFD 中集成创新的数据增强技术和锚设计策略来提出一种改进的锚点匹配方法（IAM），能给回归器提供更好的初始化。

1.2 网络结构



网络结构采用 VGG16 作为骨干网络，在分类层之间进行截断并增加一些辅助结构，选择 conv3_3，conv4_3，conv5_3，conv_fc7，conv6_2 和 conv7_2 作为第一个检测层生成六个原始的特征图，然后通过 FEM（特征增强模块）把这些原始的特征图转换成六个增强的特征图，这些增强的特征图和对应的原始特征图具有同样的尺寸，然后把他们送入第二个检测层

1.3 FEM：Feature Enhance Module



FEM 可以使原始特征图更具辨识度和鲁棒性，FEM 利用不同维度的信息，包括上一层的神经元和当前层非局部神经元，增强的神经元可以用下面的数学公式表示：

$$ec_{(i,j,l)} = f_{concat}(f_{dilation}(nc_{(i,j,l)}))$$
$$nc_{i,j,l} = f_{prod}(oc_{(i,j,l)}, f_{up}(oc_{(i,j,l+1)}))$$

FEM 先使用 1*1 卷积归一化输入的特征图，然后对上一层的输入进行上采样和当前层输入执行元素乘积，最后将 N 个特征图分成三份分别作为包含不同扩张卷积层数的子网络，最后连接操作合并子网结果，得到增强特征图

1.4 PAL：Progressive Anchor Loss

采用多任务损失，不仅对框架的不同层次，而且对不同 shot 设计渐进式锚点大小，低级特征更适合于小脸，在第一张照片中分配较小的锚点尺寸，在第二张照片中使用较大的尺寸，提出基于锚点的 second shot 多任务损失函数

$$\mathcal{L}_{SSL}(p_i, p_i^*, t_i, g_i, a_i) = \frac{1}{N_{conf}}(\Sigma_i L_{conf}(p_i, p_i^*) + \frac{\beta}{N_{loc}}\Sigma_i p_i^* L_{loc}(t_i, g_i, a_i)),$$

同一层上的原始特征图比增强特征图具有更多用于检测高分辨率定位信息，原始特征图可以用于检测和分类较小的人脸，由此基于一组较小的锚点提出 first shot 多任务损失函数

$$\mathcal{L}_{FSL}(p_i, p_i^*, t_i, g_i, sa_i) = \frac{1}{N_{conf}} \Sigma_i L_{conf}(p_i, p_i^*) + \frac{\beta}{N_{loc}} \Sigma_i p_i^* L_{loc}(t_i, g_i, sa_i)$$

两个层的损失进行加权求和得到整个渐进式锚点损失

$$\mathcal{L}_{PAL} = \mathcal{L}_{FSL}(sa) + \lambda \mathcal{L}_{SSL}(a)$$

1.5 IAM：Improved Anchor Matching

在训练期间需要计算正负锚点确定哪一个锚点对应人脸框，当前锚点匹配策略是在锚点和 ground truth 之间双向进行的，在增强阶段为了使得锚点和人脸匹配，锚点设计和人脸采样是相互协作的

Feature	Stride	Size	Scale	Ratio	Number
ef_1 (of_1)	4	160 × 160	16 (8)	1.5 : 1	25600
ef_2 (of_2)	8	80 × 80	32 (16)	1.5 : 1	6400
ef_3 (of_3)	16	40 × 40	64 (32)	1.5 : 1	1600
ef_4 (of_4)	32	20 × 20	128 (64)	1.5 : 1	400
ef_5 (of_5)	64	10 × 10	256 (128)	1.5 : 1	100
ef_6 (of_6)	128	5 × 5	512 (256)	1.5 : 1	25

根据人脸比例统计，设置 anchor 比率 1.5：1，其中原始特征的 anchor size 为增强特征的一半；在数据增强阶段，有 2/5 的概率使用基于 anchor 的采样策略，在图像中随机选择人脸，然后裁剪包含人脸的子图，并设定五种不同的子图和选择人脸之间大小比率，剩下的 3/5 采用 SSD 的数据增强方法

2 FACE-SWAP

2.1 概括

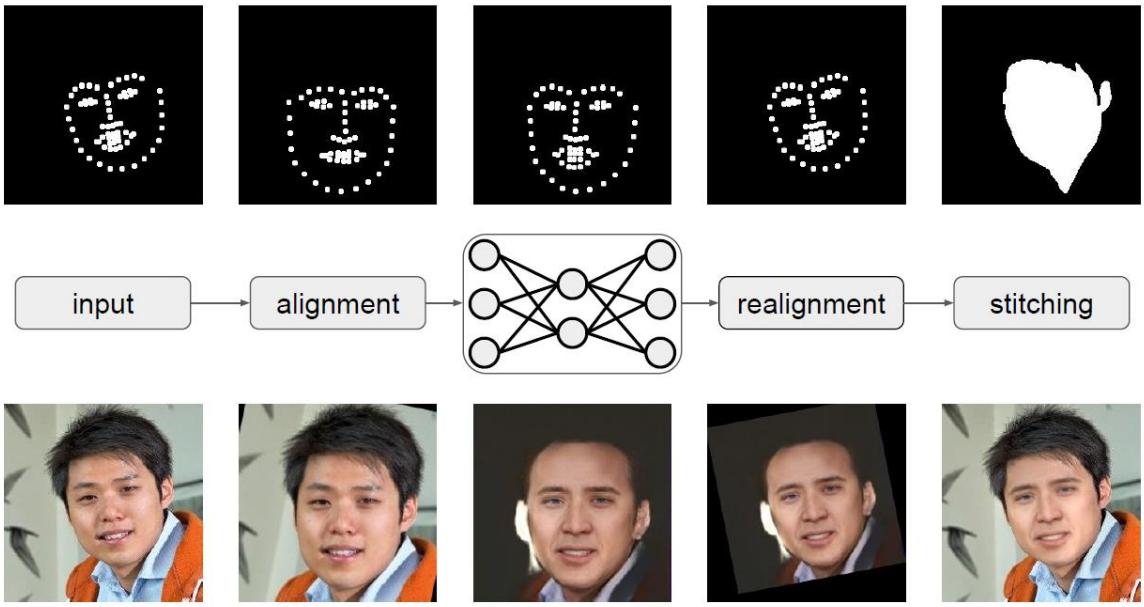
考虑在图像中的换脸问题，在保留姿势、面部表情以及光照的情况下将输入身份换成目标身份，为了执行这种映射，使用训练过的卷积神经网络来从非结构化的目标身份的图像中捕捉到他的外观信息，这种方法通过将换脸问题转化为风格迁移来实现，目标是以另一个人的脸来渲染图片，设计了一种新的损失函数来使网络生成高度逼真的结果，通过利用简单的预训练和后处理步骤结合神经网络，目的是在没有用户输入的情况下实时进行人脸交换。

2.2 网络结构

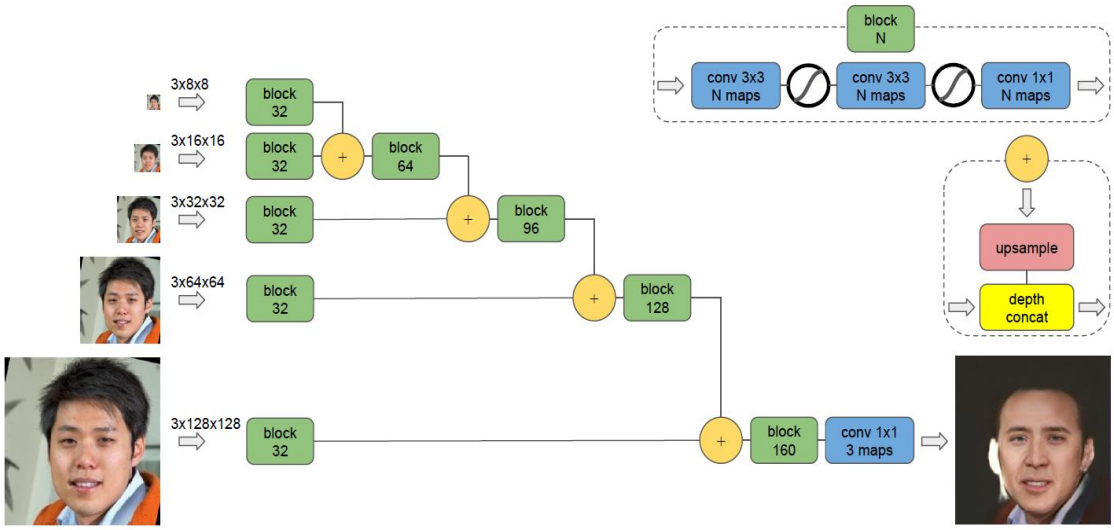
任务介绍：有一张 A 的图像，能够将其换成 B 的脸，同时保持姿势和表情以及光照情况，就风格迁移而言，考虑到输入 A 的姿势和表情作为内容，输入图像 B 的身份作为风格，光照作为一个单独的方法提到。

使用一个权重 W 参数化的卷积神经网络来转换内容图像 x，如输入图像 A，输出图像则为$\bar{\mathrm{x}} = f_W(\mathrm{x})$，不像先前的工作，假定被给定不止一张风格图像，而是大量的风格图像，定义为$\mathrm{Y} = y_{1,\ldots,n}$，这些图像描述了想要匹配的人的身份，并且只在训练的时候使用。

系统有另外两个组成成分执行人脸对齐和背景/头发/皮肤分割，假定所有的图像都与一个正面视图参考图对齐，这使用一个仿射变换来实现，对给定图像和 68 个面部参考关键点，面部关键点由 dlib 提取，分割被用来对恢复输入图像 x 的背景和头发，目前不由转换网络保留。使用 opencv 中的泊松克隆来缝合背景和结果换脸图像，存在又快且相对准确的分割方法，包含一些基于神经网络的方法，给定一个分割掩膜会更加简单，并且只需要注重于剩下的问题。整个系统给定在下图中：



2.3 转换网络



这是一个有不同下采样版本的输入图像 x 的带分支操作的多尺度结构。每个分支都有一个 0 填充卷积层伴随 ReLU 的块，分支通过由两个因子组合的最近邻上采样组合并且在通道维数上进行串联，最后一个分支是网络的结尾，有一个 1x1 卷积带有 3 个颜色通道。

网络设计为 128x128 的输入，有 1M 参数。对于更大的输入，256x256 或者 512x512，直接加到其他分支上是很直接的，网络的输入只由最高的分辨率的分支

决定，首先在 128x128 的输入上训练非常方便，然后使用其作为起始来训练更大的图像。这样，能够不需要重新训练整个模型来实现更高分辨率。

2.4 损失函数

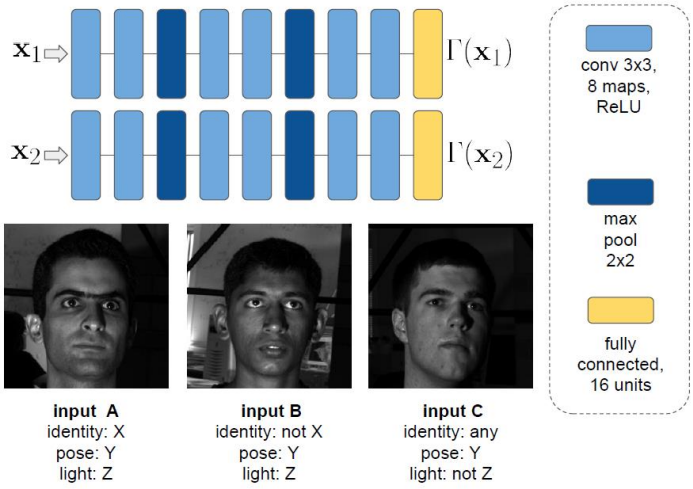
对于每一个输入 x，目的是生成一个$\bar{x}$，其联合了最小化 content 和 style 的损失。这些损失在 19 层的 VGG 网络的特征空间中定义。假定 x 和每个风格图像 y 都与参考脸对齐了，所有图像都有 3xHxW 的维度。

content loss：能够在网络的所有层上被计算

style loss：损失函数受到了基于 patch-based 损失的启发，对每个点提取一个 kxk 的方形邻域，这个过程拥有 M=(Hl−k+1)(Wl−k+1)个神经 patches，假定为通过欧式距离来在面部的标记点和输入图像 x 的标记点来进行分类，这样，每个训练图像都有一系列有相似姿势和表情的风格图像。

light loss：内容图像 x 的光照条件在生成图像$\bar{x}$中没有被保留，当只使用上面提到的在 VGG 特征空间中定义的损失的话。针对这个问题对目标通过引入一个额外的项来惩罚光照的改变，为了定义光照惩罚项，运用风格特征以及内容特征在预先训练的 VGG 中提取特征相同的方式来使用特征空间，这样如果特征空间能够代表光照条件的不同，那么就可以运用成功。VGG 网络对于这项任务不是非常合适，其被训练用来对目标进行分类，然而光照信息与其不是特别的相关。

为了得到光照灵敏度的可取属性，构建一个小型的孪生卷积神经网络，其被训练来在要么相同光照要么不同光照条件的一对图像，成对的图像总是有相同的姿态，网络结构如下图所示：



3 实现图片及视频换脸

3.1 子目标

人脸检测（Detection）、人脸校准（Alignment）、人脸 Decoder/Encoder（将一张脸训练成另一张脸）、视频与图像的互相转换

3.2 人脸识别任务

人脸识别任务主要分为四大板块：

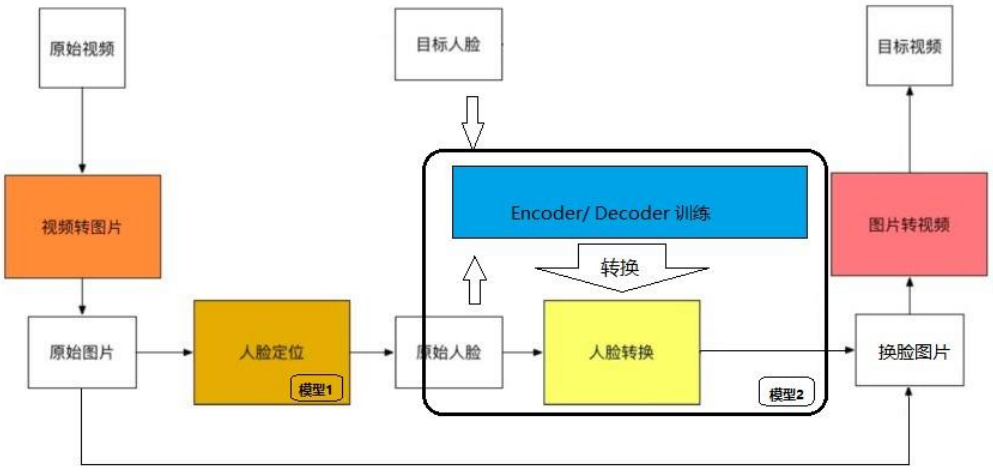
- Detection-识别出人脸位置
- Alignment-人脸上的特征点定位
- Verification-人脸校验
- Identification（Recognization）-人脸识别

后两者的区别在于，人脸校验是要给你两张脸问你是不是同一个人，人脸识别是给你一张脸和一个库问你这张脸是库里的谁。

此次只用到前两个板块 Detection 和 Alignment：

- Detection - 用于找到视频中被换脸人的脸的位置
- Alignment - 用于解决 B 脸和 A 脸的表情同步，判断正脸侧脸等问题

3.3 大体流程



五个步骤：视频转图片、人脸检测和校准、Encoder/Decoder 训练、人脸转换、图片转视频

两个模型：人脸定位模型+人脸转换模型（训练+转换）

3.4 实现步骤

3.4.1 使用 FFmpeg 工具将视频转为图片

FFmpeg 介绍：

- 支持几乎所有音频、视频格式合并、剪切、格式转换、音频提取、视频转图片、图片转视频
- 播放器、直播等音视频行业核心

3.4.2 人脸检测和校准

人脸检测有两种方案:

- 基于传统的 HOG 模型

调用 dlib 人脸识别函数库中的 HOG 模型做人脸检测，Dlib 这是一个很有名的库，有 c++、Python 的接口。使用 dlib 可以大大简化开发，比如人脸检测，特征点检测之类的工作都可以很轻松实现。同时也有很多基于 dlib 开发的应用和开源库，dlib 官方文档有如下描述，表明该模型经过大量训练在人脸检测领域确实比较成熟：This face detector is made using the classic Histogram of Oriented Gradients (HOG) feature combined with a linear classifier, an image pyramid, and sliding window detection scheme. The pose estimator was created by using dlib's implementation of the paper: One Millisecond Face Alignment with an Ensemble of Regression Trees by Vahid Kazemi and Josephine Sullivan, CVPR 2014 and was trained on the iBUG 300-W face landmark dataset. HOG 模型多用于行人检测，行人检测很多基于经典论文 HOG+SVM 模型。

- 基于 CNN 模型

将第一步得到的连续帧序列图片作为输入，通过对输入的处理将每一帧人脸位置的信息输出到 json 文件中，并且截取人脸位置的照片。

人脸特写照片：用于第三步训练人脸转换 GAN 模型。

Json 文件：用于第四步人脸转换，只对标记区域进行转换，即只换人脸。

人脸校准

目的：保证转换后的脸和原脸保持同一表情，同一朝向，要让每一帧的图像都符合原图，彻底让图像动起来

方法：提取人脸特征点，对转换后的脸按照原脸提取到的特征点排布进行变换

特征点提取分为三种方式：基于 ASM 和 AAM 的传统方法；基于级联形状回归的方法；基于深度学习的方法。直接调用 dlib 库实现，找到 68 个人脸特征点

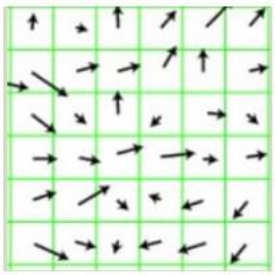
对提取到的特征点，通过以下两种方案对特征点进行处理来做到校准人脸：

- 普氏分析（PA）

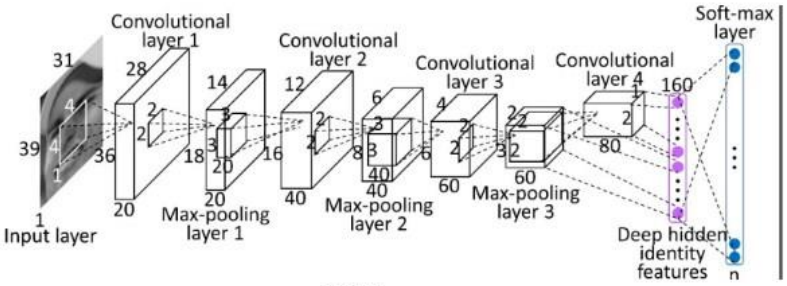
PA 包含了常见的矩阵变换和 SVD 的分解过程，最终返回变换矩阵，调用变换矩阵，将原图和所求得矩阵进行仿射变换即可获得新图片，这个矩阵能够让 a 脸和 b 脸具有相同的表情和朝向

- 点云匹配（PCL）

将源点云(source cloud)变换到目标点云(target cloud)相同的坐标系下，包含了常见的矩阵变换和 SVD 的分解过程，最终返回变换矩阵，计算过程与普氏分析类似。



HOG



CNN

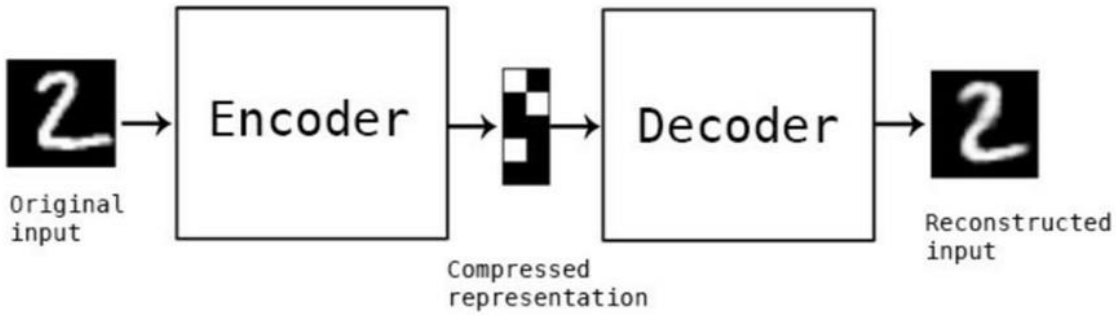
3.4.3 人脸 Encoder/Decoder 训练

使用的模型：Autoencoder

核心思想：GAN

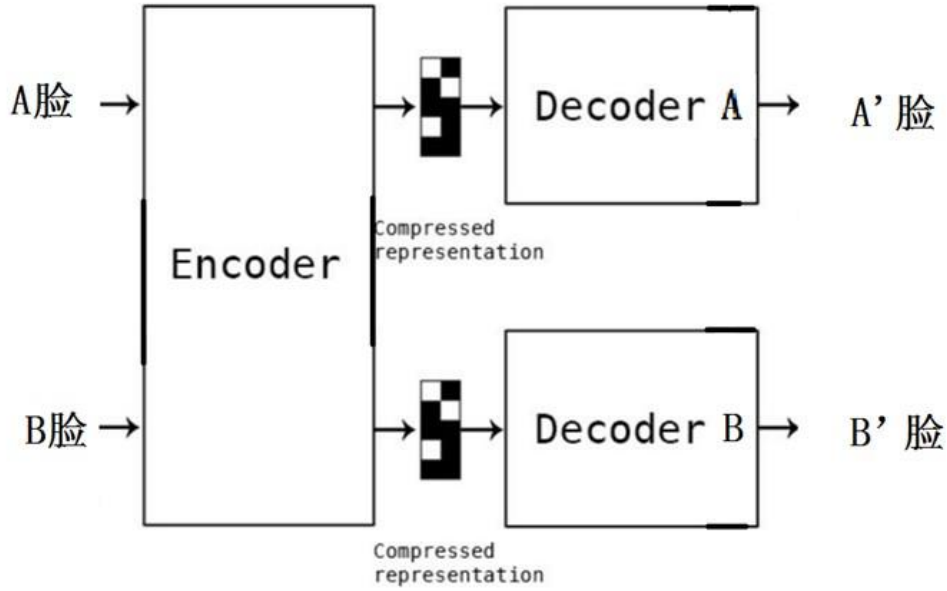
✓ 这个模型所做的是基于原始的图片再次生成原始的图片。

✓ Autoencoder 的编码器（Encoder）把图片进行压缩，解码器（Decoder）将图片进行还原



问题：即使输入的是另外一个人脸，也会被 Autoencoder 编码成为一个类似原来的脸。

解决方案：把人脸共性相关的属性和人脸特性相关的属性进行学习，对所有的脸都用一个统一的编码器 Encoder，这个编码器的目的是学习人脸共性的地方， 然后对每个脸有一个单独的解码器 Decoder，这个解码器的目的是学习人脸特性的地方。 当用 A 的脸通过编码器，再使用 B 的解码器的话，会得到一个与 A 的表情一致 B 的脸。

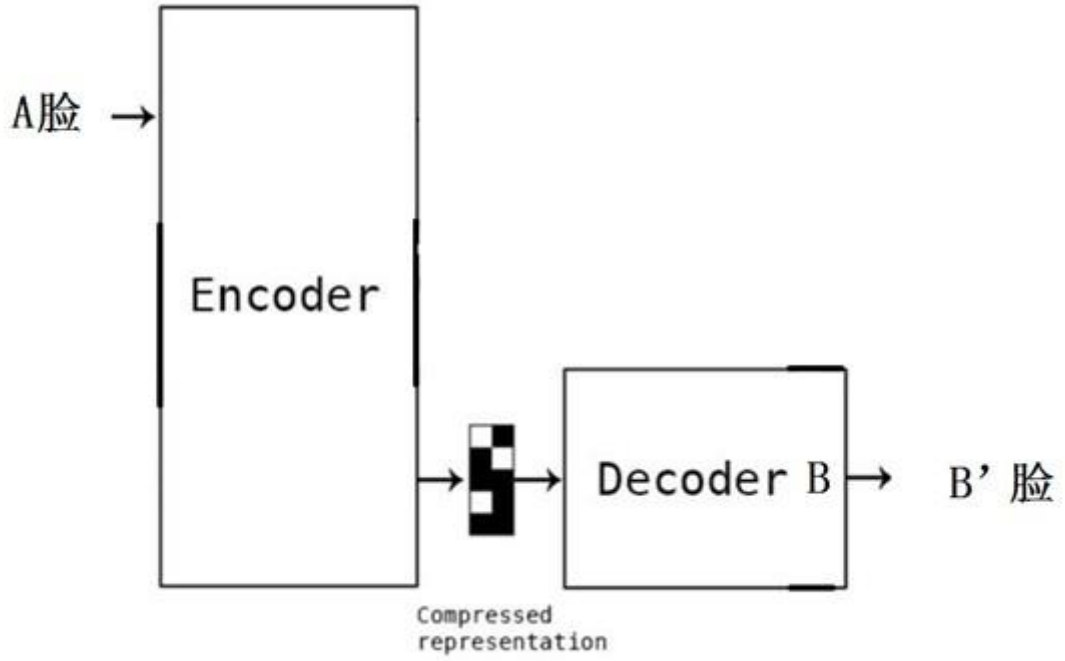


Encoder 是 4 层卷积+2 层全连接+1 层 upscale，Decorder 是 3 层 upscale+1 层卷积， Upscale 嵌套在 Encoder 和 Decoder 中，Upscale 的核心是 PixelShuffler() ，该函数能够把图像进行一定的扭曲，而这个扭曲增加了学习的难度，反而能让模型实现最终的效果

3.4.4 人脸转换

目标：将带有 landmark（ A 脸）的帧图片转换成新的图片（只换 landmark 区域，A 脸变 B 脸）

- ✓ 生成每一帧图片中 A 脸区域对应的 B 脸区域图片
  过程：Decoder_B (Encoder(A)) （输入 A ，输出 B）
- ✓ 将生成的 B 脸图片替换 A 脸区域，根据之前人脸检测生成的 json 文件找到 A 脸的 landmark，逐帧替换，生成一系列新图。



3.4.5 图片转视频

将经过第四步转换后的图片通过 FFmpeg 组合成视频，然后将原视频的音频加进去。

3.5 改进之处

3.5.1 问题

对于 successful example，生成图像清晰，转换是全局的转换(Deepfakes 是局部的转换)，而对于 failed example 则问题比较多：

（1）表情不能一一对应

（2）出现混杂无意义的像素。

3.5.2 整体架构

首先，对于任意人脸 A，经过仿射变换后得到 WA ，其中 WA 等价于任意一张脸，即任意一张脸都是 A 扭曲而来，如果网络能学会从 WA 去噪修复至 A，那么网络就能从任意一张脸转换成 A

而在网络结构的设计上，在编码器中采用全链接层破坏了之前卷积层提取的特征内的空间关系，让每一个像素之间都充分运算，而在解码器中采用 PixelShuffler 结构亦是如此考虑，整体网络共用一个编码器，分用两个解码器的目的是让编码器学习到更丰富的脸部特征，把不同的人脸都编码都同一个隐空间上，再通过不同的解码器用不同的方式“重构”回来。

3.5.3 成因分析

对于 CycleGAN 的结果，分析其 loss，CycleGAN 是利用对抗损失衡量 A 转换到 B 的，无论是 KL 散度，JS 散度……都是衡量两个分布之间的距离。由基本的概率论知识可知，当 X,Y 同分布的时候，指 X,Y 在概率上具有相同性质，如 EX=EY，但不能得到 X=Y，同时，当数据量不够时，GAN 的训练是有问题的(因为数据不能很好的代表分布，容易发生崩塌到生成某个特例上)，而这些问题，在使用 MSE 和 MAE 这种逐像素误差作为优化目标时，得到缓解。

效果也并非完美，其一是因为使用 MAE 作为 loss 具有均值性的，会导致图片模糊。其二，WA=任意一张脸，这个假设是有局限的。

第一点，将男脸 A 换成女脸 B，比女脸 A 换成女脸 B 先天要困难。

第二点，这个假设局限了只能五官周围一部分进行转换，且人脸对齐与否，仿射变换的参数都会对结果造成影响，因为只能在局部进行变换，且原模型的生成图片大小固定在 64x64，因此生成后几乎必然还要做一次 resize，原本模糊的人脸会进一步模糊。

第三点，这一点则完全是对数据的理解能力了。在美图横行的时代，若只考虑五官数据，部分女星的人脸数据五官部分区分度极低，造成效果令人不满。

3.5.4 改进

在分析问题后，首要也是最为容易解决的问题就是清晰度问题。

对于生成图片的清晰度问题是因为 MAE 的均值性，这一点通过引入 GAN 来进行解决，引入两个 Discriminator，分别对应于 AB 图片，如同 cycleGAN，除了判别真假，还能判别是否配对。

在我的实验中，引入一个 Discriminator，用于补充生成图片的细节，而不考虑 A 与 B 的差异，如同其他 SR 和 denoise 的任务一样，这样得出的结果在视觉上是相仿的，一来可以省点参数，二来会使训练更稳定。其中特别指出的，GAN 虽然会使得图像变清晰，但并不能保证细节和原图一致，这就涉及两种应用场景。更注重视觉效果还是更注重细节恢复。

相比五官，人是否有胡子，眼镜，发型也同样重要，其实五官单拿出来，对人来说辨识率其实不高。如果是全身照中，身体的体态也同样重要，那么如果一个常年以有胡子的形象示人的人，你看到他没胡子的形象，反而感觉违和了。所以 GAN 从数据中抓取主要特征，导致多了胡子，其实不是什么主要问题，更重要的是解决视频转换中，胡子这种特征会在某一帧中突然丢失，产生的违和感要更重一些。

为了解决这点问题，采用了 Mask 机制，Mask 网络和生成网络共享编码器和解码器除最后一层外的其他所有层，最后一层生成一个数值范围 0~1 的 Mask，假设生成的图片记为 G，原图为 S，那么最后采用的图片为

$$Mask \odot G + (1 - Mask) \odot S$$

通过这个方法，让模型自己学习图像需要转换的部分。

之后我又加入了 self-attention 机制，即 SAGAN，且 self-attention 的层数比 SAGAN 要多，带来额外的计算量，最终的效果并没有很大的提升

3.5.5 数据质量对最终效果的影响

采用了 denoiseGAN 的结构，使用 probrandomcolor_match 交换两张不同人脸颜色的均值和方差来制作噪音，以及 motion blurs，运动模糊来损坏原脸，使用这两种处理方法进行数据强化。

为了让模型更好的训练，我又使用了 eye-aware，edge loss 以及 Mask 机制，改用 MTCNN 来做人脸的定位，和关键点定位，以及人脸对齐等操作，获得更好，更灵活的数据，因为 Dlib 在侧脸和有遮挡的情况下表现比 MTCNN 要差。

另外对于数据集的来源，直接在百度上找的图片效果并不是太好，杂音很多，我是用的 google_image_download 工具下载的图片，这样得到的数据更加清晰，噪音更少，并且相比在百度上找的图片姿势和表情更多，数据质量对模型最终的效果也是影响很大的。

4 成果展示

4.1 图片上的换脸效果

选择了关晓彤和江疏影的图片

换脸后的效果

选择了孙杨和庞博的照片

换脸后的效果

选择了徐峥和吴彦祖的照片

换脸后的效果

4.2 视频上的换脸效果

原始视频

换脸后的视频效果

5 理解与体会

这一学期人工智能应用实践课程的开设，我对神经网络有了更深的认识，并且对人脸检测领域进一步的了解，之前对换脸的具体网络结构以及实现方法理解的还不是很透彻，通过这一学期的实践，将问题具体化，另外之前也没有怎么接触过论文，通过一学期的努力，在相关论文方面读了很多，学到了很多东西，这门课程的开设为自己以后的研究打下了基础，对深度学习领域有了一个大概的了解，为自己以后的学习充满了信心。