百度网盘 AI 大赛-美颜祛斑祛痘

1. 核心思路

我们首先观测赛题的输入和输出，发现均为图像，输入是含有斑痘的 1024*1024 高分辨率图像，而输出则是去除了色斑后的美颜图像。因此，我们直观的想法是构建一个输入和输出均为图像的模型，可供选择的是类似 autoencoder 这样的模型。我们选择了 paddlehub 中有在大规模人脸数据集上预训练权重的 u2net 网络。之后，我们又观测到色斑的本质是只在人脸上存在，因此考虑在 u2net 网络中进行多任务学习，即同时输出包含人脸的分割 mask。以上是我们对于模型的选择，除此之外还包括一些数据集的预处理和增强，如何选择损失函数以及多个损失函数间系数的分配权重问题，训练的时候是否冻结一些层等，这些将在下文进行详细解释和介绍。

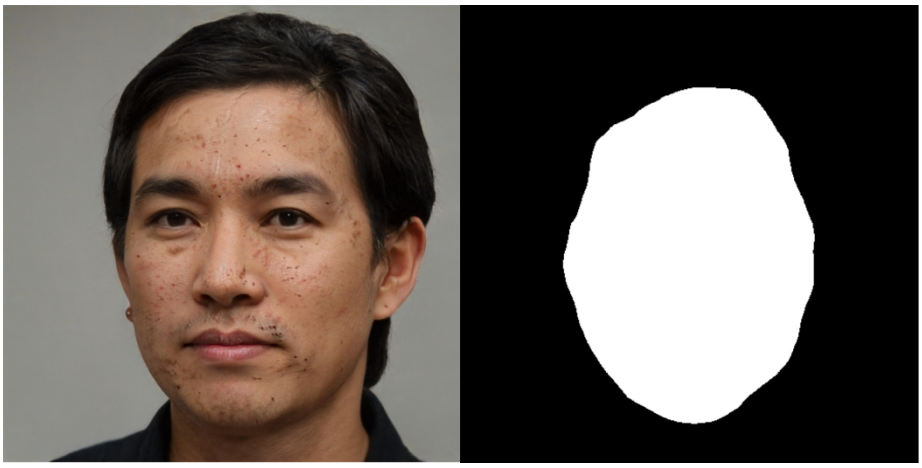
2. 数据预处理

针对官方发布的数据集，5000 张 image 和 5000 张 groundtruth，为了更充分的利用给到的数据进行训练，我们并没有分出额外的验证集和测试集，我们首先对每张图片做了水平镜像反转，这样数据集可以扩增一倍，在对这 1W 张图片随机选取 4000 张进行±10°的旋转，再选取 500 张加入轻微的椒盐噪声，共 14500 张图片，以此增加数据的丰富度。

针对人脸部分 mask 区域的 groundtruth 生成问题，我们寻求了一个捷径。我们注意到 paddlehub 中有一个 human parsing 的模型 “ace2p”。因此我们直接利用了该模型对于人体的解析，把其中标签为 13 的结果存储下来，作为人脸的 mask。如下代码：

```
chunk_size = 10
human_parser = hub.Module(name="ace2p")
items = os.listdir('train_datasets/image')
path   = [os.path.join('train_datasets/image', item) for item in items]
chunk_path = [path[i:i + chunk_size] for i in range(0, len(path), chunk_size)]
for pth in tqdm(chunk_path):
    parsing_gt = human_parser.segmentation(paths=pth, use_gpu=True, batch_size=chunk_size, output_dir='ace2p_output', visualization=True)
    for i in range(len(parsing_gt)):
        pa, data = parsing_gt[i]['path'],parsing_gt[i]['data']
        face = np.where(data==13,255,0)
        pa = os.path.join('train_datasets/face', pa.split('/')[-1])
        cv2.imwrite(pa, face)
```

Paddlehub 提供的该模型预训练权重还是比较准确的，见下图。



3. 模型搭建

我们注意到原始输入是 stylegan 生成的人脸图，而 stylegan 是在 celeb 上进行训练的。因此，我们很希望寻找一个同样在 celeb 或者人脸数据集上训练过的预训练网络，（如下图，输入是人脸，输出是素描图），之后在我们的需求下进行微调。很幸运，我们在 paddlehub 中找到了 u2net 网络，并且下载到了它的权重。因此，我们后续的实验都使用了 u2net。

U2net 是基于 unet 提出的一种新的网络结构，U2Net 名称的来源在于其网络结构由两层嵌套的 Unet 结构，可以在不需要预训练骨干网络的情况下从零开始训练，同样基于 encode-decode，作者参考 FPN，Unet，在此基础之上提出了一种新模块 RSU(ReSidual U-blocks)对于分割物体前背景取得了惊人的效果

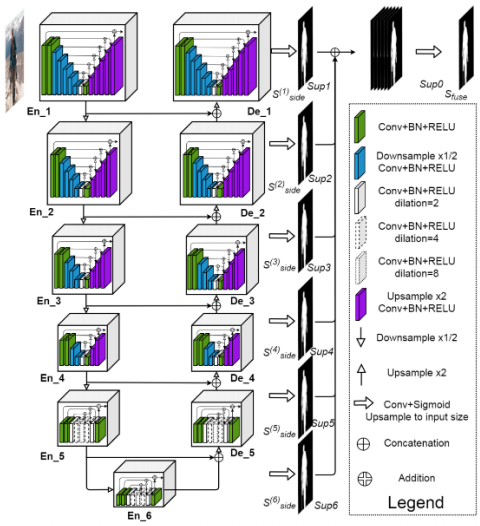
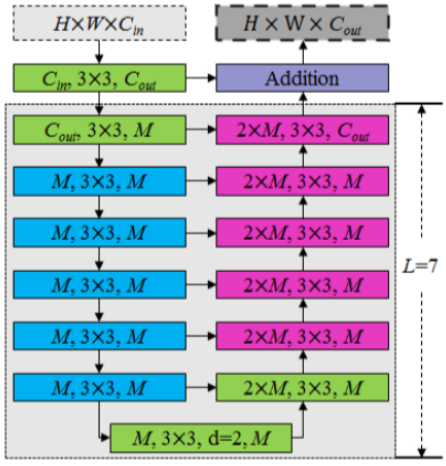


Figure 5. Illustration of our proposed U^2-Net architecture. The main architecture is a U-Net like Encoder-Decoder, where each stage consists of our newly proposed residual U-block (RSU). For example, **En_1** is based on our RSU block shown in Fig. 2(e). Detailed configuration of RSU block of each stage is given in the last two rows of Table 1.

下面是 RSU 结构:



从这个结构可以很容易的看出，所谓的 RSU 其实就是一个很简单的 Unet，通过类似 FPN 的结构，将多个 Unet 输出结果进行组合，最后取得了非常理想的效果。

我们输入是经过 resize 的图片，3 通道彩色图，模型输出为 4 通道，我们把 1，2，3 通道拿出来与原图做 loss，最后一个通道和生产的 mask 做 loss

4. 损失函数

我们使用了四个损失函数联合训练。首先两个是题目中包含有的 PSNR 和 MSSSIM，这很直接，我们直接按照比赛的打分标准给每个按照 0. 5 的权重进行模型的训练。另外，我们需要进行多任务学习，因此还需要分割的两个损失函数，分别是每个像素点之间的 celoss

和判别重合区域的 diceloss。这两个也是分割中比较常见的损失函数。我们同样给予每个损失函数 0.5 的权重。

celoss：

$$L = -\sum_{i=0}^{N} y_i ln(\sigma(x_i))$$

Diceloss：

$$L_{dice} = 1 - \frac{2I+\varepsilon}{U+\varepsilon}$$

5. 训练细节

我们冻结 encoder 层的后 3 层进行训练，我们认为这样可以保留预训练模型的高层语义信息，加快训练速度，学习率为 0.001，batch_size 为 12，对输入图片做 resize 为 512*512

6. 可视化