# 人像分割与背景替换设计文档

清华大学

## 1 项目开发目的和意义

人像分割及背景替换技术一向是学术界以及工业界中研究的热点话题，在娱乐平台的视频直播、在线教育、远程视频会议方面具有广泛的应用。除此之外，人像分割技术还在VR领域有诸多应用，在便利数据采集、提升用户体验，多样化商业模式等方面能够发挥巨大的作用。

本项目属于人像分割与背景替换应用，旨在华为Atlas200DK上实现实时对摄像头拍摄的视频进行人像分割与背景替换的系统。该系统能够使用摄像头捕获视频/图像，实时检测人像区域并对背景进行替换。该系统与用户交互的部分包括从摄像头图像采集、模型推理到输出结果的完整流程。

目前工业界通用的人像分割主要采用绿屏技术，需要专门的绿屏设备及环境，不利于普通用户的广泛使用。传统的人像分割使用背景减除法，严重依赖人像和背景的色差。在颜色相近的区域会产生不稳定的分割结果。并且具有对环境的高度依赖性。而基于简单神经网络的人像分割方法具有较低的精度，会在分割时产生人像关键部位的缺失，不完整。复杂的神经网络方法由于其较高的复杂度通常难以满足实时高效的需求。

本项目拟在华为Atlas200DK上实现借助摄像头对人像视频进行实时检测和分割，并对其背景进行替换，华为Atlas200DK设备速度快、精度高，能够满足在实际场景下用摄像头进行人像视频的拍摄、实时分割和背景替换的需求。

## 2 相关工作

PortraitNet网络的设计主要与语义分割和轻量级卷积神经网络的研究相关。在这一小节，将会简要介绍经典的语义分割模型与最新的轻量级卷积神经网络的设计工作。

语义分割是计算机视觉领域的基础研究方向之一，是很多视觉任务的基础。近年来随着深度学习的快速发展，相比传统方法，基于卷积神经网络的语义分割模型在准确率上有了极大的提高。全卷积网络（fully convolution networks， FCNs）[1]是语义分割领域一篇具有里程碑意义的研究工作，它提出了一种端到端的逐像素分割模型，并设计了“跳跃连接”的结构来提高网络的分割效果。SegNet[2]提出了一种经典的encoder-decoder网络结构来解决语义分割问题，与之相似的工作还有UNet[3]，两者的主要区别是SegNet保存了encoder部分的pool indices，并通过上采样操作连接到decoder部分，而UNet是将encoder部分与decoder部分相同尺寸的特征图连接到decoder部分。以Deeplab为基础的系列工作是目前语义分割模型准确率最高的模型，Deeplabv1[4]提出了使用空洞卷积在获取较大感受野的同时保持特征图的尺寸不变，避免下采样层导致特征图过小影响分割精度，同时它使用条件随机场（CRFs）作为后处理来优化分割结果。Deeplabv2[5]设计并提出了空间金字塔池化模块（atrous spatical pyramid pooling,ASPP）通过特征融合提高了网络的预测精度。Deeplabv3[6]去掉了CRFs后处理模块，并改进了Deeplabv2中的空间金字塔池化模块进一步调高了分割的准确率。上述这些工作虽然具有较高的检测精度，但由于模型设计复杂，导致运行速率比较低，难以应用于移动设备。

与具有较高复杂度的大模型相比，也有部分研究工作更关注语言分割的速率问题。ENet[7]设计并提出了一个“深而窄”的网络结构，网络的运行速率非常快但分割准确率下降也比较明显。ICNet[8]通过融合不同分辨率的特征提高了模型的精度，但对于移动设备来说，该模型还是复杂度比较高。

随着移动端应用需求的日益增长，设计更适合移动设备的轻量级卷积神经网络结构越来越吸引研究者们的关注。MobileNet-v1[9]使用depthwise separable卷积来近似传统卷积操作，将VGG网络结构中的传统卷积层替换为depthwise卷积层，在ImageNet数据集上可到达到与VGG近乎相同的准确率，模型速率提高了接近10倍，但模型中的1×1卷积层是整个网络的计算瓶颈。为了解决这个问题，ShuffleNet-v1[11]利用分组卷积和通道重排（channel shuffle）来减少模型的计算量。MobileNet-v2[10]提出了倒置残差模块（inverted residual block）来减少模型的复杂度。ShuffleNet-v2[12]认为评价模型的速率应该使用更直接的方式，即直接用模型运行速率来评价而非模型的理论计算量，并提出了设计轻量级卷积神经网络的四个设计原则。

这些针对移动设备，研究轻量级卷积神经网络设计的相关工作，为我们设计高效率人像分割模型奠定了基础，也启发了我们设计兼顾准确率与速率的网络模型。

## 3 算法设计

### 3.1 模型结构

基于深度学习的语义分割模型是目前语义分割领域准确率最高的方法，我们考虑将语义分割与针对移动设备的轻量级卷积神经网络研究工作相结合，设计一个适用于移动设备的轻量级语义分割模型，使得该模型既可以取得较高的准确率（IOU）又可以有较快的运行速度，满足在移动设备上运行的需求。在本章节，主要介绍PortraitNet的模型结构与设计思路。

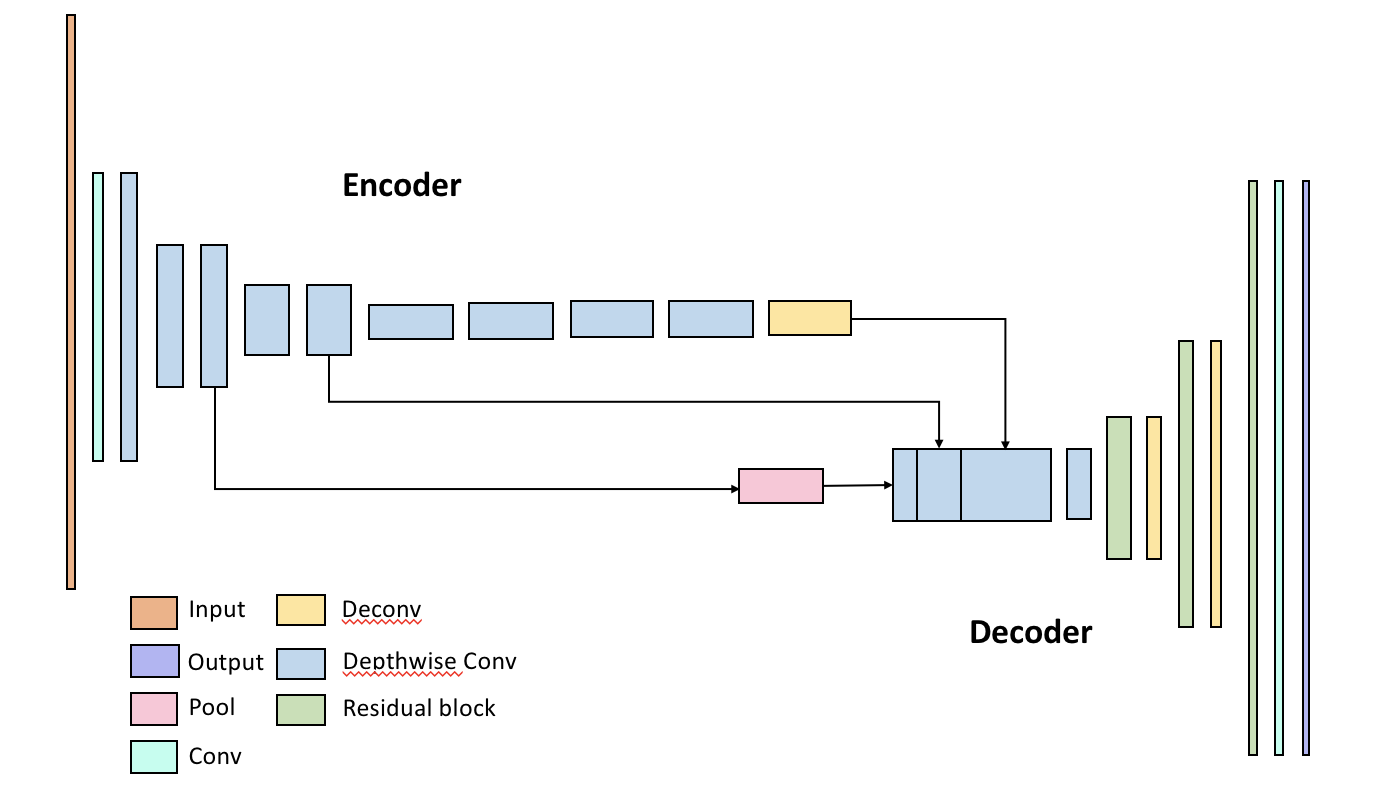


图1 PortraitNet网络模型结构图

PortraitNet主要分为两个模块：encoder模型与decoder模块，网络结构如图1所示。网络的输入为三通道的图像，输出为与输入相同大小的逐像素分割结果，对每个像素点分别进行预测为背景还是人像。受经典语义分割研究工作的启发，网络结构采用了经典的encoder-decoder网络结构，encoder部分对输入的三通道图像进行特征提取，decoder部分对特征图进行上采样得到与输入图像相同大小的分割结果。在encoder部分，主要利用深度可分离卷积（depthwise separable convolution）来降低网络的参数量，提高网络的运行速度。为了减少特征提取部分的运行时间，我们只使用了10层卷积层来进行特征提取，最大下采样倍数为16倍，encoder部分的网络结构参数如表1所示。为了充分利用模型的特征提取能力，在使用较少卷积层的情况下提高分割准确率，考虑将不同分辨率的特征图在channel维度进行拼接进行后续的decoder操作。由于在encoder部分分别进行了2倍、4倍、8倍、16倍的下采样，以下采样8倍的特征图的尺寸为基准，将下采样4倍、8倍、16倍的特征进行channel维度的拼接，由于不同下采样倍数的特征图分辨率大小不同，在channel维度上拼接时需要对下采样4倍的特征图进行核为2×2的max pool操作，对下采样16倍的特征图进行2倍的上采样操作，用反卷积层实现，这样可以将特征图的尺寸对齐为下采样8倍的特征图的尺寸，因为可以在channel维度上进行特征图拼接，拼接后的特征图channel数为896（896=128+256+512），在该融合后的特征图后使用1×1的卷积将通道数从896降低为128，降低模型的复杂度。

表1 PortraitNet encoder网络结构

|  |  |  |
| --- | --- | --- |
| Type/Stride | Filter Shape | Input Size |
| Conv/s2 | 3×3×3×32 | 224×224×3 |
| Conv dw/s1 | 3×3×32 dw | 112×112×32 |
| Conv/s1 | 1×1×32×64 | 112×112×32 |
| Conv dw/s2 | 3×3×64 dw | 112×112×64 |
| Conv/s1 | 1×1×64×128 | 56×56×64 |
| Conv dw/s1 | 3×3×128 dw | 56×56×128 |
| Conv/s1 | 1×1×128×128 | 56×56×128 |
| Conv dw/s2 | 3×3×128 dw | 56×56×128 |
| Conv/s1 | 1×1×128×256 | 28×28×128 |
| Conv dw/s1 | 3×3×256 dw | 28×28×256 |
| Conv/s1 | 1×1×256×256 | 28×28×256 |
| Conv dw/s2 | 3×3×256 dw | 28×28×256 |
| Conv/s1 | 1×1×256×512 | 14×14×256 |
| Conv dw/s1 | 3×3×512 dw | 14×14×512 |
| Conv/s1 | 1×1×512×512 | 14×14×512 |
| Conv dw/s1 | 3×3×512 dw | 14×14×512 |
| Conv/s1 | 1×1×512×512 | 14×14×512 |
| Conv dw/s1 | 3×3×512 dw | 14×14×512 |
| Conv/s1 | 1×1×512×512 | 14×14×512 |

PortraitNet网络结构的decoder部分相对较为简单，主要涉及两个操作，一个是对特征图进行上采样的反卷积层，每次扩大倍数为2倍，由于encoder部分下采样倍数为8倍，故共需三层反卷积层将特征图上采样为与输入图像相同的尺寸。在反卷积层之间，添加Residual block来进行过渡，采用Residual block的出发点是希望能更好的保存图像的边缘信息。Decoder的网络结构如表2所示。其中Residual Block的结构如图2所示。

表2 PortraitNet decoder网络结构

|  |  |  |
| --- | --- | --- |
| Type/Stride | Filter Shape | Input Size |
| Deconv/s2 | 4×4×128×128 | 28×28×128 |
| Residual Block | - | 56×56×128 |
| Deconv/s2 | 4×4×64×64 | 56×56×64 |
| Residual Block | - | 112×112×64 |
| Deconv/s2 | 4×4×16×16 | 112×112×16 |
| Conv/s1 | 3×3×16×2 | 224×224×16 |

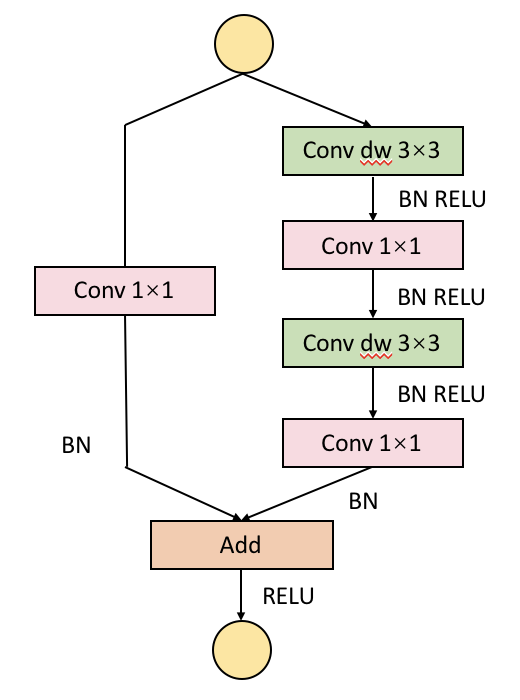


图2 Residual Block结构图

由于任务目标是处理以半身肖像为主的图像数据，训练数据的输入输出如图3所示，输入图像为原始三通道彩色图像，groundtruth标注为逐像素的mask标注，每个像素点分别为0或1，0代表该像素点为背景类，1代表该像素点为人像类。损失函数使用交叉熵损失函数（CrossEntropy loss）。



图3 原始图像与groundtruth标注

### 3.2 辅助loss

为了提高模型的运行速率，PortraitNet使用了较少的卷积层来进行特征提取与上采样，与经典大模型相比，在模型准确率上会有所下降，为了提高模型的准确率同时不增加额外的时间消耗，充分发挥模型的特征提取能力，除正常的分割loss外，我们考虑增加辅助loss来帮助模型的训练，提高模型的准确率。在本章节，将主要介绍添加的两种辅助loss，并在EG1800公开人像数据集上进行验证。

#### 3.2.1 边缘约束loss

由于人物肖像分割问题的特殊性，相比常规的语义分割任务，人物肖像分割在背景替换或背景需要的情况下对边缘分割的准确率尤其敏感。在传统的语义分割模型设计中，往往没有针对边缘部分进行特殊的优化，因此，我们提出了在保持原有分割loss的情况下，再增加一个边缘约束性loss，加强网络对边缘的敏感度，提高模型的分割效果。

训练数据集包括原始图像以及每张图像对应的逐像素点mask标注，根据mask标注生成对应的边缘groundtruth 。生成方法为首先根据mask图像得到人像的边缘，然后对得到的边缘带进行高斯滤波得到边缘的高斯带状groundtruth，即中间置信度高，离中心越远置信度越低，如图4所示。边缘约束loss使用L2 loss监督学习。



图4 原始图像与groundtruth标注

#### 3.2.2 一致性约束loss

在语义分割模型的训练过程中，为了扩充训练数据，通常会采用数据增强的方式来使得训练样本更丰富。比较常使用的数据增强方式有图像旋转、翻转、缩放、裁切，改变图像的亮度、对比度、锐化、高斯滤波，添加随机噪点等，在进行数据增强时会对groundtruth标注同步进行图像旋转、翻转、缩放、裁切操作，使得输入图像与标注保持一致。Groundtruth标注为0/1的 hard label，使用交叉熵损失函数进行约束。

针对原始图像，在经过图像旋转、翻转、缩放、裁切后的图像为A，对图像A改变图像的亮度、对比度、锐化、高斯滤波，添加随机噪点等可以得到对应的图像A’,分别预测图像A和A’的分割结果如图5所示，会发现图像A的分割结果会比图像A’更准确，因为改变图像的亮度或对比度或增加高斯滤波会使得图像更难以辨认，人像与背景的差别更小，导致预测结果会弱于直接用图像A进行预测。

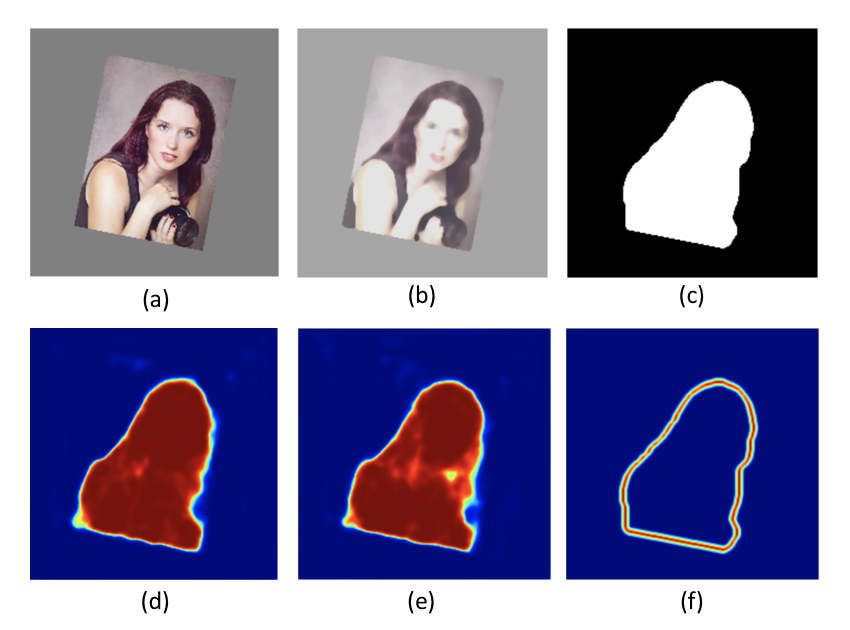


图5 预测效果对比：(a)为图像A；(b)为对图像A进行数据增强得到的图像A’；(c)为groundtruth标注；(d)为图像A的网络预测结果；(e)为图像A’的网络预测结果；(f)为根据mask的groundtruth生成的边缘groundtruth。

为了提高模型的准确率，同时受知识蒸馏相关工作的启发，soft label相比hard label包含更丰富的信息，可以使得模型的准确率进一步提高。为简化训练过程，充分发挥soft label的优势，我们按照如下思路设计并添加了一致性约束loss。针对模型M，令输入图像A的网络分割预测结果为B，令输入图像A’的网络分割预测结果为B’。除去用groundtruth作为hard label分别监督B和B’外，再添加新的loss，将B作为soft label去监督B’的分割结果。由于分割结果B与B’是相同维度的置信度为0~1的两个分布，可以用L2 loss约束或用KL散度loss约束，具体可见下一小节的实验验证。

#### 3.2.3 实验验证

为快速验证辅助loss是否对训练有帮助，选择公开数据集EG1800进行试验，从EG1800中随机选择300张图像作为测试集，剩下的图像作为训练集。分别设置以下四组对比试验：

* Base模型
* Base模型+边缘约束loss
* Base模型+一致性约束loss
* Base模型+边缘约束loss+一致性约束loss

所有实验使用相同的数据增强方式，数据增强方式包括图像旋转、翻转、缩放、裁切，改变图像的亮度、对比度、锐化、高斯滤波，添加随机噪点等，输入图像的大小为224×224，对于原始长宽不相等的图像，首先会对较短的边进行padding，填充为像素值128。所有实验均使用单张NVIDIA-TIANX显卡训练，训练框架使用Pytorch。实验结果如表3所示。

表3 对比实验结果

|  |  |
| --- | --- |
| EG1800(300 imgs for test) | IOU |
| 实验1 | 93.40% |
| 实验2 | 94.23% |
| 实验3 | 94.39% |
| 实验4 | 94.90% |

从实验结果来看，边缘约束loss与一致性约束loss均可以提高模型的预测准确率，帮助模型的训练。

### 3.3 模型设计

为进一步适用于服务器端或移动端，在PortraitNet模型的基础上，通过使用空洞卷积层进一步增大模型的感受野，并通过修改模型卷积层的channel数分别设计了适用于服务器版的模型与适用于移动版的模型，下面分别进行介绍。

#### 3.3.1 服务器版模型特征提取部分设计

表4 服务器版模型网络encoder结构

|  |  |  |  |
| --- | --- | --- | --- |
| Layers | Type/Stride | Filter Shape | Input Size |
| 1 | Conv/s2 | 3×3×3×32 | 224×224×3 |
| 2 | Conv dw/s1 | 3×3×32 dw | 112×112×32 |
|  | Conv/s1 | 1×1×32×64 | 112×112×64 |
| 3 | Conv dw/s2 | 3×3×64 dw | 112×112×64 |
|  | Conv/s1 | 1×1×64×128 | 56×56×128 |
| 4 | Conv dw/s1 | 3×3×128 dw | 56×56×128 |
|  | Conv/s1 | 1×1×48×128 | 56×56×128 |
| 5 | Conv dw/s2 | 3×3×128 dw | 56×56×128 |
|  | Conv/s1 | 1×1×128×256 | 28×28×128 |
| 6 | Conv dw/s1 | 3×3×256 dw | 28×28×256 |
|  | Conv/s1 | 1×1×256×256 | 28×28×256 |
| 7 | Conv dw/s2 | 3×3×256 dw | 28×28×256 |
|  | Conv/s1 | 1×1×256×384 | 14×14×256 |
| 8 | Conv dw/s1 | 3×3×384 dw | 14×14×384 |
|  | Conv/s1 | 1×1×384×384 | 14×14×384 |
| 9 | Conv dw/s1 | 3×3×384 dw | 14×14×384 |
|  | Conv/s1 | 1×1×384×384 | 14×14×384 |
| 10 | Conv dw/s1 | 3×3×384 dw | 14×14×384 |
|  | Conv/s1 | 1×1×384×512 | 14×14×384 |
| 11 | Conv\_dw/s1 | 3×3×512 dw | 14×14×512 |
|  | Conv/s1 | 1×1×512×512 | 14×14×512 |
| 12 | Conv\_dw/s1 | 3×3×512 dw | 14×14×512 |
|  | Conv/s1 | 1×1×512×512 | 14×14×512 |

在特征融合层，将第4层、第6层、第9层和第12层的特征进行channel维度的拼接，再进行后续的decoder操作。其中不同尺寸的特征图在拼接的同时需要进行下采样或上采样对应到相同的尺度。

服务器版模型的decoder网络结构如表5所示，其中Residual Block如图2所示。

表5服务器版模型网络decoder结构

|  |  |  |
| --- | --- | --- |
| Type/Stride | Filter Shape | Input Size |
| Deconv/s2 | 4×4×128×128 | 28×28×128 |
| Residual Block | - | 56×56×128 |
| Deconv/s2 | 4×4×64×64 | 56×56×64 |
| Residual Block | - | 112×112×64 |
| Deconv/s2 | 4×4×16×16 | 112×112×16 |
| Conv/s1 | 3×3×16×2 | 224×224×16 |

#### 3.3.2 移动版模型特征提取部分设计

表6 移动版模型网络encoder结构

|  |  |  |  |
| --- | --- | --- | --- |
| Layers | Type/Stride | Filter Shape | Input Size |
| 1 | Conv/s2 | 3×3×3×8 | 224×224×3 |
| 2 | Conv dw/s1 | 3×3×8 dw | 112×112×8 |
|  | Conv/s1 | 1×1×8×24 | 112×112×24 |
| 3 | Conv dw/s2 | 3×3×24 dw | 112×112×24 |
|  | Conv/s1 | 1×1×24×48 | 56×56×48 |
| 4 | Conv dw/s1 | 3×3×48 dw | 56×56×48 |
|  | Conv/s1 | 1×1×48×48 | 56×56×48 |
| 5 | Conv dw/s2 | 3×3×48 dw | 56×56×48 |
|  | Conv/s1 | 1×1×48×96 | 28×28×48 |
| 6 | Conv dw/s1 | 3×3×96 dw | 28×28×96 |
|  | Conv/s1 | 1×1×96×96 | 28×28×96 |
| 7 | Conv dw/s2 | 3×3×96 dw | 28×28×96 |
|  | Conv/s1 | 1×1×96×144 | 14×14×96 |
| 8 | Conv dw/s1 | 3×3×144 dw | 14×14×144 |
|  | Conv/s1 | 1×1×144×144 | 14×14×144 |
| 9 | Conv dw/s1 | 3×3×144 dw | 14×14×144 |
|  | Conv/s1 | 1×1×144×144 | 14×14×144 |
| 10 | Conv dw/s1 | 3×3×144 dw | 14×14×144 |
|  | Conv/s1 | 1×1×144×196 | 14×14×144 |
| 11 | Conv\_dw/s1 | 3×3×196 dw | 14×14×196 |
|  | Conv/s1 | 1×1×196×196 | 14×14×196 |
| 12 | Conv\_dw/s1 | 3×3×196 dw | 14×14×196 |
|  | Conv/s1 | 1×1×196×196 | 14×14×196 |

在特征融合层，将第4层、第6层、第9层和第12层的特征进行channel维度的拼接，再进行后续的decoder操作。其中不同尺寸的特征图在拼接的同时需要进行下采样或上采样对应到相同的尺度。

移动版模型的decoder网络结构如表7所示，其中Residual Block如图2所示。

表7移动版模型网络decoder结构

|  |  |  |
| --- | --- | --- |
| Type/Stride | Filter Shape | Input Size |
| Deconv/s2 | 4×4×96×96 | 28×28×96 |
| Residual Block | - | 56×56×96 |
| Deconv/s2 | 4×4×32×32 | 56×56×32 |
| Residual Block | - | 112×112×32 |
| Deconv/s2 | 4×4×16×16 | 112×112×16 |
| Conv/s1 | 3×3×16×2 | 224×224×16 |

#### 3.3.3 decoder部分上采样层

在decoder部分使用双线性插值进行上采样，上采样通过反卷积实现，固定卷积核，并设置学习率为0，效果与使用Upsample层相同，使用反卷积层是为了方便后续的模型转换。

## 4 软件运行结果

### 4.1 模型训练方法

#### 4.1.1 训练数据集

由于使用目标场景为肖像分割，因此训练数据集采用网上公开数据集加自己爬图标注的方式进行。公开数据集使用EG1800、Supervise.ly以及ATR数据集作为正样本图像，从MSCOCO数据集中抽取了5000张不包含人的图像作为背景图像，即负样本。自己标注图像为从Flickr与直播网站上爬取的主播图像组成，共计约11500张。

为方便训练与比较，本文档中对比实验仅使用EG1800作为训练集，其中训练数据约1500张，测试数据为剩余的300张。实际交付模型使用全部数据集训练，因此实际准确率高于仅使用EG1800数据训练的模型。

#### 4.1.2 训练环境

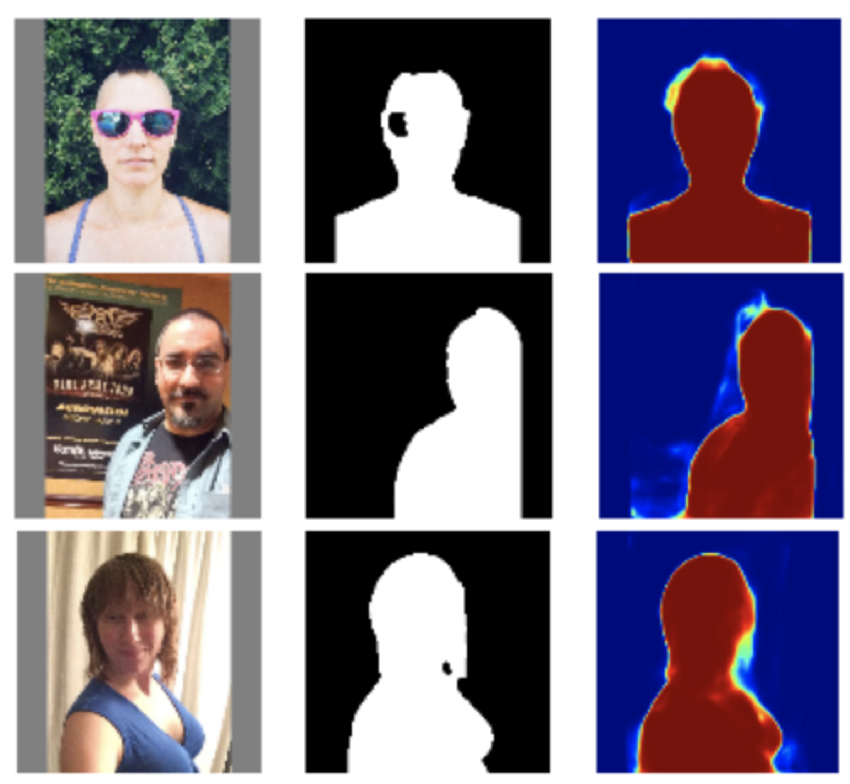
模型训练框架为tensorflow，所有模型均使用单张NVIDIA-TIANX显卡训练完成,得到tensorflow模型后可以转换为om模型，用于在Atlas200DK上运行。

#### 4.1.3 训练步骤

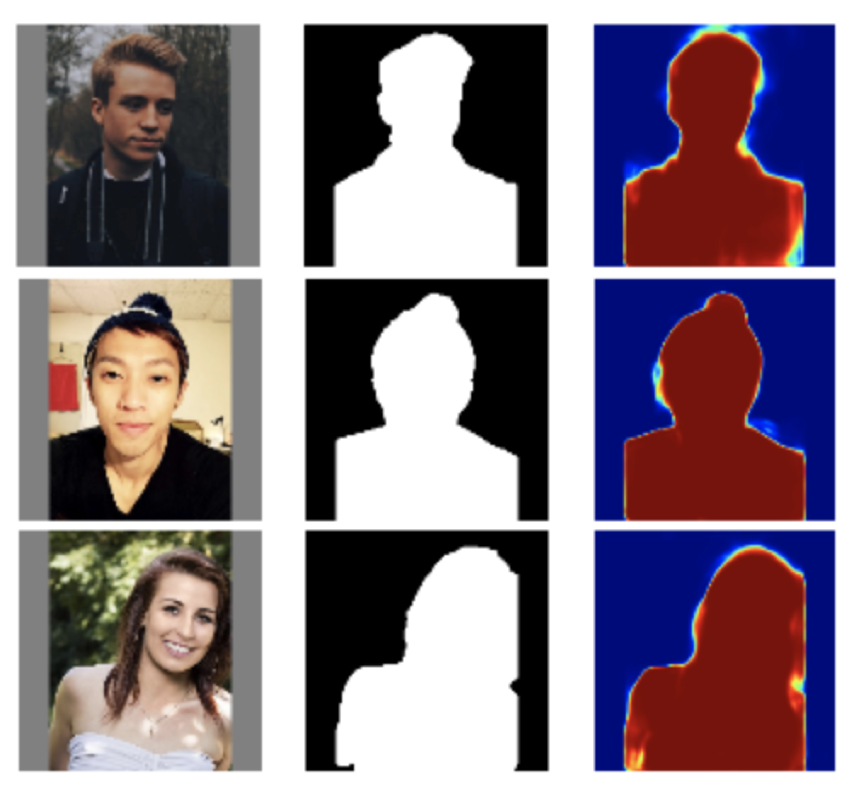
所有的模型均使用tensorflow框架训练，初始学习率为0.001，使用Adam训练2000个epoch，每20个epoch学习率降低为原来的0.95。

### 4.2 实验结果展示

* 不同类型的背景



* 不同光照和天气条件



* 不同肤色和人种



* 不同衣服颜色式样



* 具有可以接受的人像边缘轮廓



### 4.3 实验结果统计(只用EG1800数据集训练)

|  |  |  |
| --- | --- | --- |
| EG1800数据集(300张test) | 服务器版模型 | 移动版模型 |
| IOU | 94.95% | 93.11% |

## 5 项目部署

项目参考了人脸检查样例，主要分为数据库——摄像头输入的图像或视频，执行引擎、后处理引擎。

数据库由摄像头输入的图像或视频构成，设置摄像头的帧率、图像分辨率、图像格式等相关参数，从摄像头中获取视频数据，每一帧传给执行引擎进行计算。

执行引擎接收摄像头数据，对每帧图像进行处理：将其转为RGB格式的图像并进行resize，接下来将预处理后的视频帧通过模型进行推理，得到人像分割结果。由于原项目由pytorch实现，而mind studio还暂时不支持pytorch模型的转换，固先将pytorch模型转换成tensorflow模型，再通过mind studio将其转换为davinci模型。然后将原视频每帧图像集合和每帧的人像分割结果集合作为输入传给后处理引擎模块。

后处理引擎接收上一个引擎的推理结果与摄像头原图像，将分割出来人像和准备好的背景结合，得到背景替换后的结果。

硬件搭设情况如图6所示。



图6 硬件搭设情况

## 6 参考文献

[1] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[2] Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." arXiv preprint arXiv:1511.00561 (2015).

[3] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

[4] Chen, L., et al. "Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. CoRR abs/1606.00915." (2016).

[5] Chen, Liang-Chieh, et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." IEEE transactions on pattern analysis and machine intelligence 40.4 (2018): 834-848.

[6] Chen, Liang-Chieh, et al. "Rethinking atrous convolution for semantic image segmentation." arXiv preprint arXiv:1706.05587(2017).

[7] Paszke, Adam, et al. "Enet: A deep neural network architecture for real-time semantic segmentation." arXiv preprint arXiv:1606.02147 (2016).

[8] Zhao, Hengshuang, et al. "Icnet for real-time semantic segmentation on high-resolution images." arXiv preprint arXiv:1704.08545 (2017).

[9] Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).

[10] Sandler, Mark, et al. "MobileNetV2: Inverted Residuals and Linear Bottlenecks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[11] Zhang, X., et al. "Shufflenet: an extremely efficient convolutional neural network for mobile devices (2017). arXiv preprint." arXiv preprint arXiv:1707.01083.

[12] Ma, Ningning, et al. "Shufflenet v2: Practical guidelines for efficient cnn architecture design." arXiv preprint arXiv:1807.11164 (2018).