**实验三 语义分割**

1. **实验要求**

使用基于深度学习的方法全卷积网络（Fully Convolution Network，FCN）在 Pascal VOC训练集上进行训练，然后将训练后的模型在 Pascal VOC 验证集上进行测试。为了减小所需的计算资源，训练时统一使用 480x480 的图片尺寸。

此外，使用的深度模型包括：

1. 一个使用 ResNet-18 作为骨干网络的 Output Stride=16 的 FCN，即将 ResNet 最后一个 stage 的下采样层去掉、卷积层的 dilation 设置为 2；
2. （与第三点二选一）基于上述的 FCN，结合至少一种课上提到的、对上下文语境建模（Context Modeling）的方法，例如 ASPP、PPM 等；
3. （与第二点二选一）基于上述的 FCN，结合 Transformer Block 的方法。
4. **实验原理**

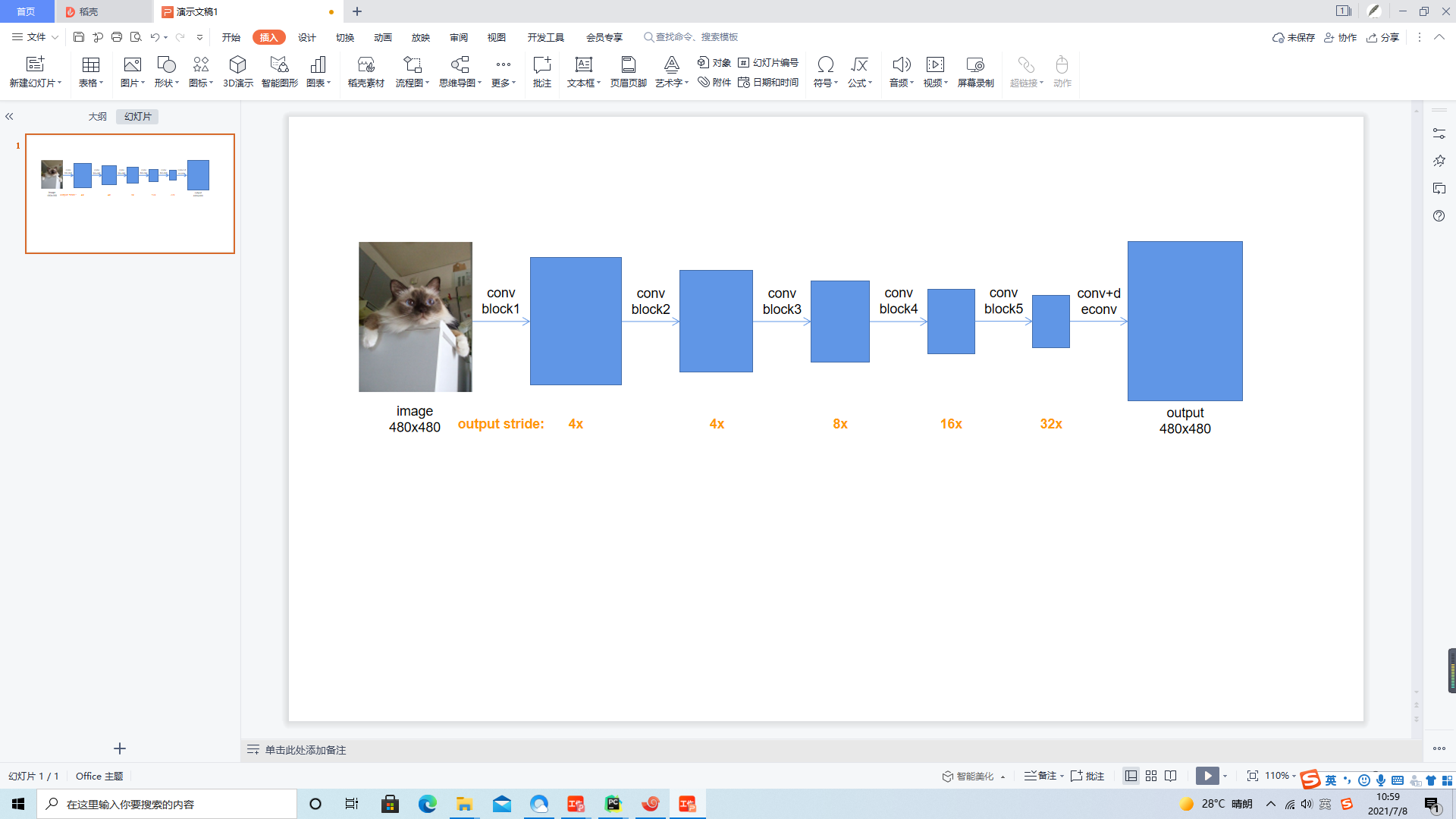
语义分割，即给定一张输入图像，利用模型输出该图像上每一个像素对应的类别预测或者属于某一类别的概率。由于该任务需要对每个像素都进行预测，语义分割又被称作稠密预测任务（Dense Prediction）。不同于图像分类只需对整张图片输出一个概率分布向量，语义分割中需要对图片中的所有像素都生成一个与之对应的概率分布向量，因此要求模型最终输出的特征图与输入图像具有相同的分辨率，并且具有与像素所属类别数同样多的通道数。

本次实验采用两个模型完成语义分割任务（即“实验要求”中的“1)”和“2)”），分别是下采样率为32的全卷积ResNet18模型（记作ResNet18-FCN），和结合了ASPP的ResNet18（记作ResNet18-ASPP）。

1. 全卷积ResNet18

全卷积ResNet18（ResNet18-FCN）的基本设计与原版FCN1相同，只是1中使用VGG作为卷积网络，而此处我们将其替换为ResNet18。

相比于标准ResNet18，ResNet18-FCN去掉了其最高层的全局池化和全连接层，用一个1x1卷积和一个32倍上采样层取而代之。其中1x1卷积用于将512通道的特征图转换为N通道（N为像素的类别数），而上采样层用于将特征图的分辨率恢复到与原输入图像相同，以进行稠密预测。

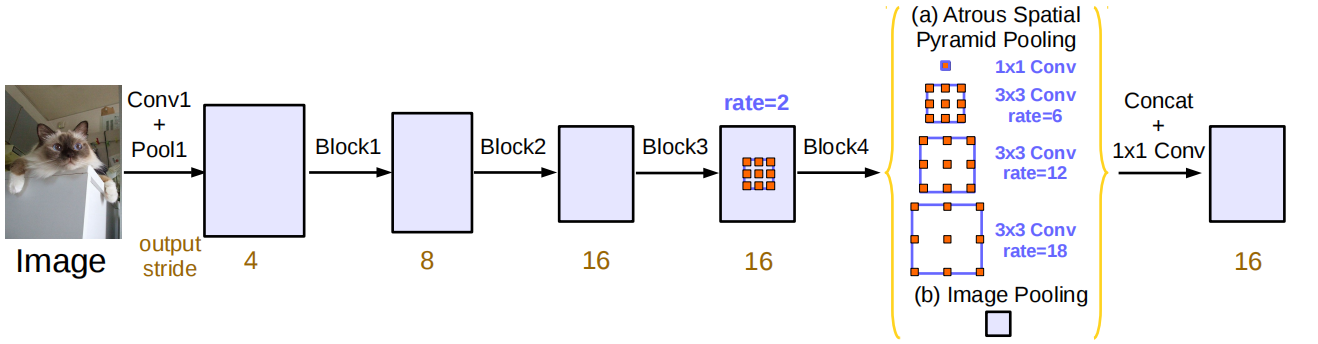


**图1.1** ResNet18-FCN的构建细节

图1.1给出了模型的结构细节。

1. 结合了ASPP的ResNet18

ASPP2是一种融合多尺度信息的构建图像分割模型的策略。2中依旧以去掉池化层和全连接层的ResNet18作为整个模型的特征提取器，其实现思路如图1.2所示：



**图1.2** ASPP的实现思路

其核心思想就是把ResNet18中的“平均池化层+全连接层”换成了“并行空洞卷积块”，这个“并行空洞卷积块”由两部分组成：

1. 空洞空间金字塔池化：四个卷积层：1x1卷积\*1+3x3卷积\*3（空洞率也即dilation分别为6,12,18，当output\_stride=16时，output\_stride定义为输入图像的尺寸与输出图像尺寸的比值），其中每一个卷积层的通道数都是256；
2. 图像池化层：平均池化层\*1+1x1卷积\*1（256通道）+双线性上采样层\*1；

具体运算过程如下：

1. 将ResNet输出的最后一个特征图（也即最后输出）输入“b”得到一个图像级的全局特征；
2. 将ResNet输出的最后一个特征图（也即最后输出）分别输入“a”的四个卷积层中得到四个特征图；
3. 由于上述得到的五个特征图都是256通道的，于是把它们在通道维上进行连接，然后再做一个256通道的1x1卷积操作，得到16倍下采样的特征图；

需要注意的是：上述并行空间金字塔池化模块并非ResNet的输出层，而只是一个融合上下文信息的组件，过程结束后，得到的不是整个模型的输出，而只是一个16倍下采样的特征图，该特征图需要经过1x1卷积将通道数变换为类别数s，然后再经上采样（反卷积），才得到最终的logits。

此外，在ASPP模块中所使用的卷积不是普通的卷积，而是空洞卷积（atrous convolution），这种卷积方法将卷积核中相邻两个元素间置以一定的距离（或用0填充），在不降低特征图分辨率的同时增大了卷积核的感受野，在图像分割中取得了非常好的效果。

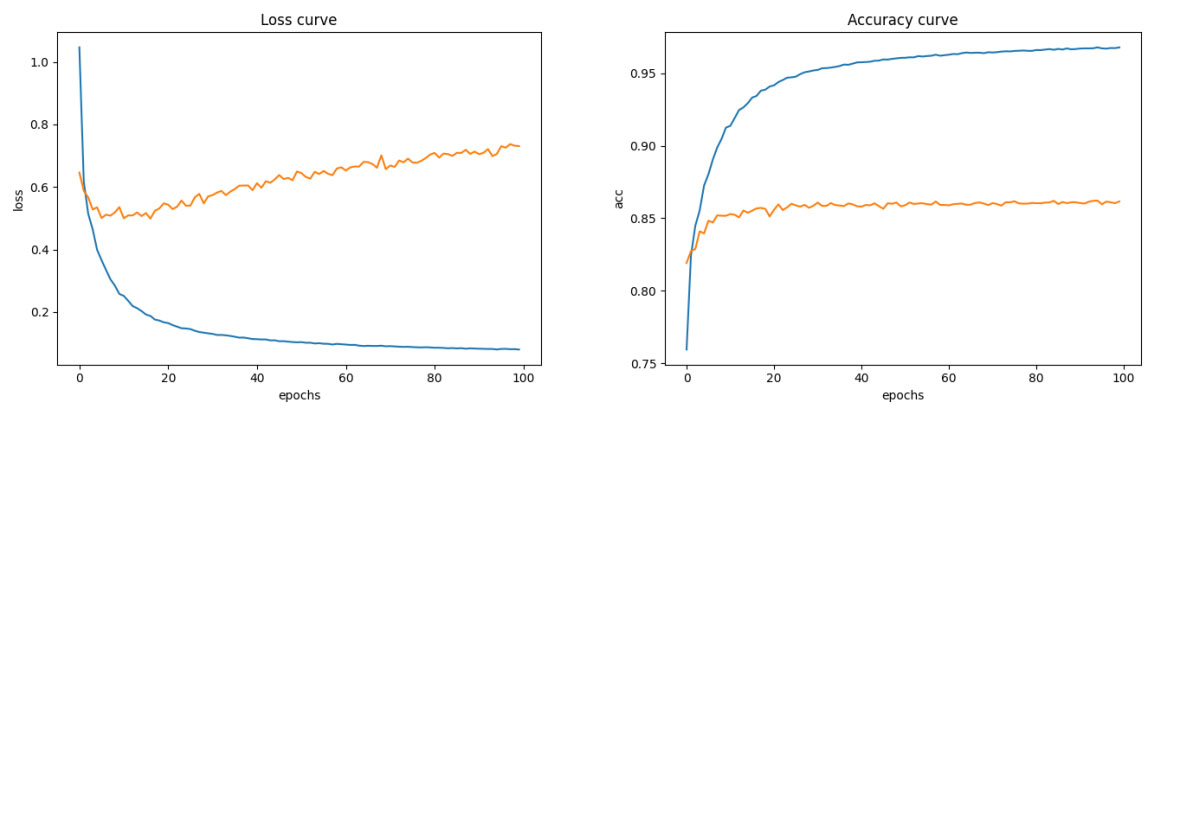
1. **实验结果**

本次实验主要用到的模型是ResNet18，以及其他一些简单的卷积（包括空洞卷积）、反卷积模块。为实现方便起见，实现过程中直接调用了在ImageNet上经过预训练的ResNet18，并去除了它的最后两层，作为模型的backbone。Pytorch中预训练的ResNet18共有五个卷积模块，分别作了4倍、1倍、2倍、2倍和2倍下采样，可得到32倍下采样的特征图。因此，我们采用卷积核尺寸为64、步长为32、padding为16的反卷积（ConvTranspose）来实现32倍的上采样（设输入图像尺寸为N，卷积核大小为K，步长为S，padding为P，则反卷积输出特征图的大小为N+(N-1)(S-1)+K-1-2P），采样规则为“双线性”上采样。

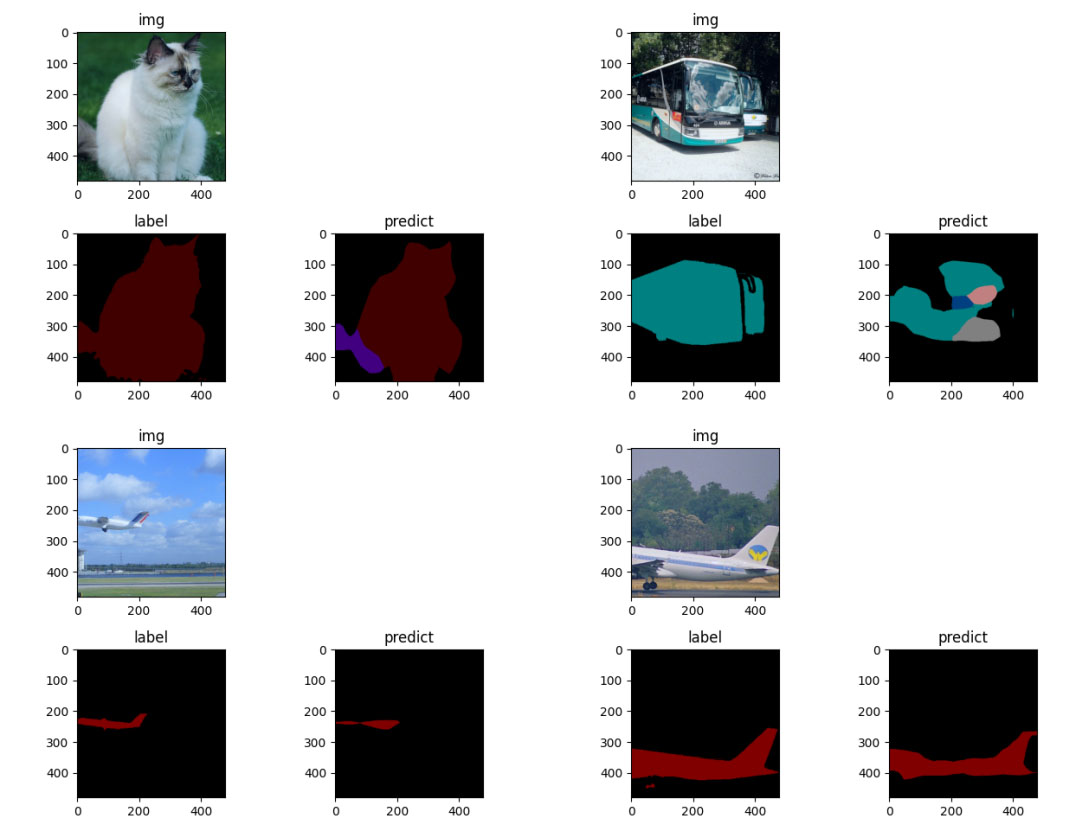
训练方法采用SGD，学习率为0.01，冲量为0.7，权重衰减率为1e-5，最大训练次数为100轮，批大小为32。表1给出了两个模型在验证集上的像素准确率和mIoU情况，图2.1，2.2分别给出了ResNet18-FCN训练过程中的准确率、损失曲线和在四组验证数据上的分割效果，图2.3，2.4分别给出了ResNet18-ASPP训练过程中的准确率、损失曲线和在四组验证数据上的分割效果（蓝色线条代表训练集上的指标，黄色代表验证集上的指标）。

**表1** 两个模型的评价结果

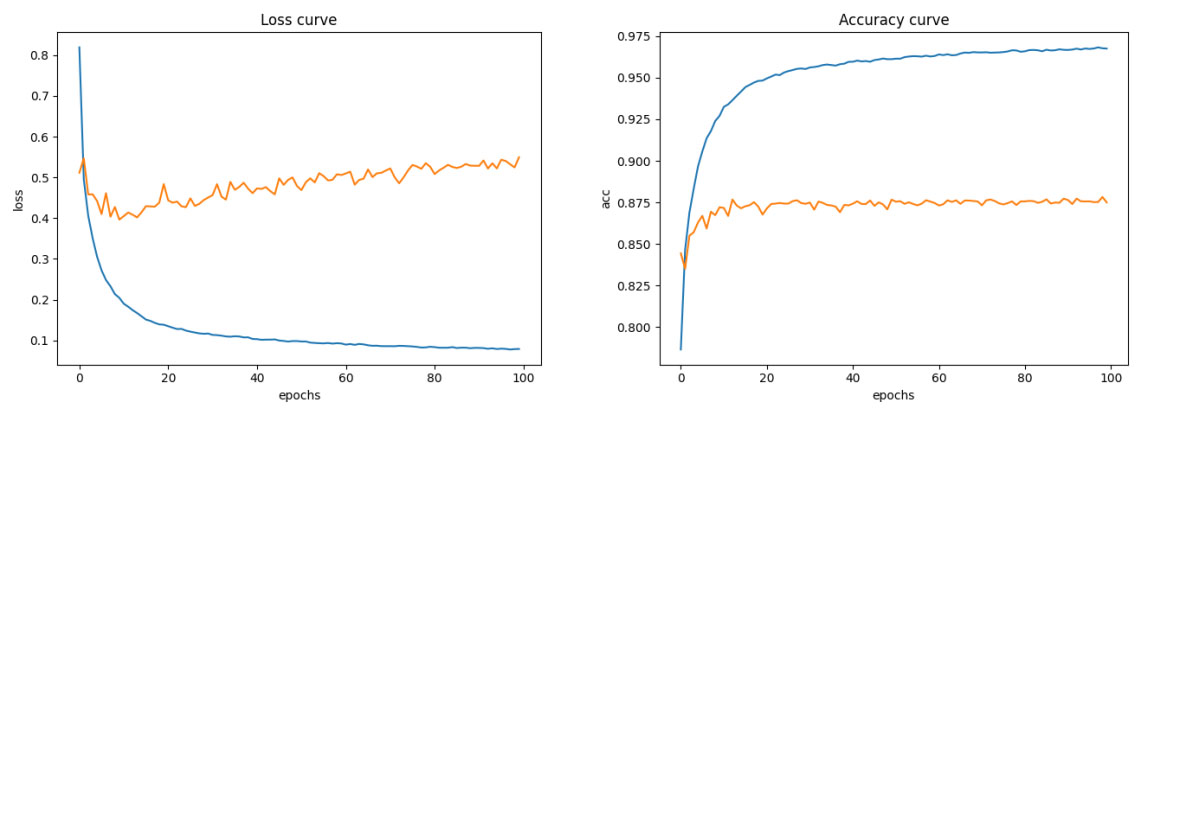
|  |  |  |
| --- | --- | --- |
| 模型 | 像素准确率 | mIoU |
| ResNet18-FCN | 86.05% | 44.71% |
| ResNet18-ASPP | 87.50% | 48.95% |



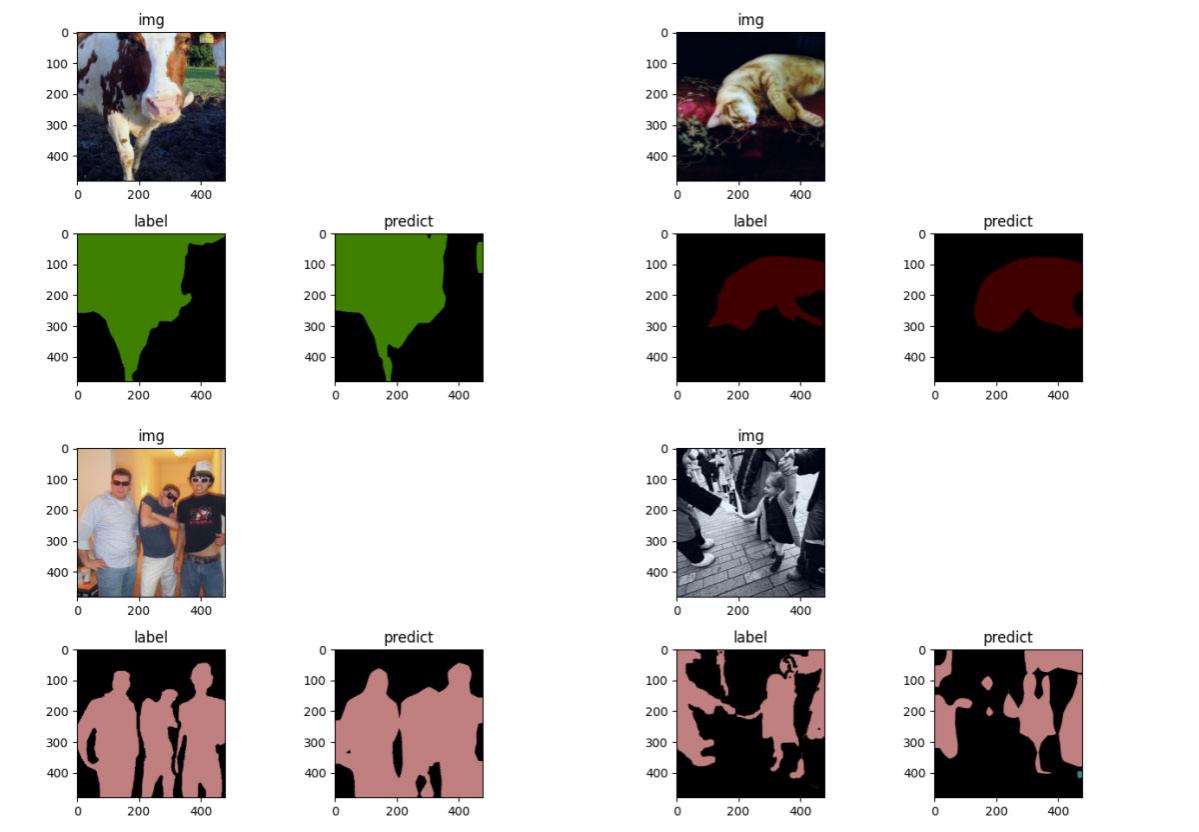
**图2.1** ResNet18-FCN训练情况



**图2.2** ResNet18-FCN测试效果



**图2.3** ResNet18-ASPP训练情况



**图2.4** ResNet18-ASPP测试效果

如果单纯从像素准确率的角度或者实验结果来看，无论是单纯的使用全卷积还是在此基础上再融入多尺度信息，模型的分割效果总体说来还是不错的，并且融入多尺度信息后，模型性能的上限得以提高，证明了多尺度信息在图像分割中的重要性。但是考虑到mIoU的话，实验发现只有当像素准确率提升到90%以上，mIoU才可能达到80%及以上，然而实验中模型虽然可以在训练集上通过训练持续获得更高的性能，在验证集上却早早进入了收敛期，导致最后的mIoU相当之低。关于这个问题，可能与数据的分布、模型的过拟合处理以及backbone模型的深度和下采样率的设置（2原文中使用的是ResNet50、output\_stride=16，而本实验中使用的只是ResNet18、output\_stride=32）等都有关系。鉴于一次训练花费的时间过长（>12小时），截止实验报告提交的最后期限，我还没有找到解决这一问题的有效办法。是以“怎样获得更好的多尺度信息、以怎样的机制融入到卷积模型中”、“怎样根据数据集的分布情况（尤其是验证集与训练集不同分布）设计模型”等都是值得在未来继续进一步深入研究的问题。

**参考文献**

1. Fully Convolutional Networks for Semantic Segmentation
2. Rethinking Atrous Convolution for Semantic Image Segmentation