FCN(全卷积神经网络)是一种用以对图像进行语义分割的深度学习模型。本文件夹内对FCN的实现来源于github上的项目https://github.com/aurora95/Keras-FCN， 使用的深度学习框架是Keras。本文件夹内后缀是.ipynb的文件需要用jupyter notebook打开。

本项目的使用主要分为三大步骤。

首先需要对数据进行预处理，即将高分辨率的材料图像切割成图像块，制作数据集。这一步使用的是dataPreprocess.ipynb。切割图像块的示意图如下。切割时按蓝色矩形框切割，拼接时按黄色框拼接。切块大小需按实际图像大小进行设定，保证分块能覆盖整张图片。

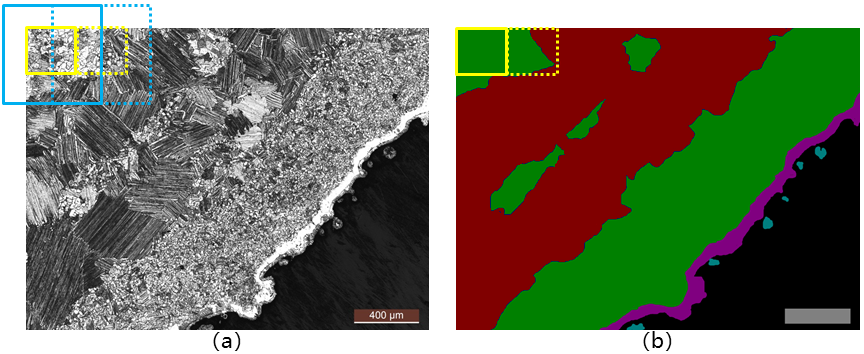


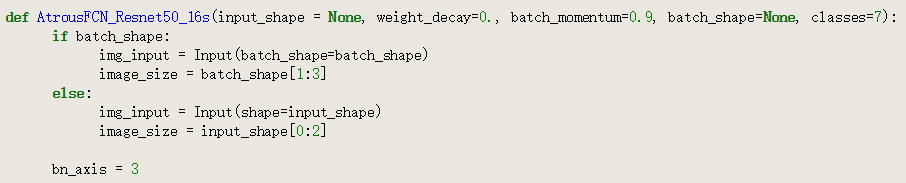
图1 图像分块的示意图。

第二步是模型的训练。用到的主要是train.py和model.py这两个文件。Train.py中需要根据实际需求修改一些参数：



其中batch\_size是批大小；epochs是迭代的轮数；lr\_base和lr\_power和学习率有关，可以不用管；target\_size是输入的尺寸，需按实际情况修改；train\_file\_path是训练集目录，需按实际情况修改；其他目录也要根据实际情况做相应修改；data\_suffix和label\_suffix分别是训练样本和标签数据的后缀，需根据实际情况修改；classes是类别数目，需按照实际需求修改。

由于model\_name选择的是AtrousFCN\_Resnet50\_16s，需要在model.py文件中找到相关模型的实现：



然后将函数定义中的参数”classes=7”根据实际情况进行修改。

都修改完成后，执行”python train.py”即可开始模型的训练。训练的模型会保存在文件夹Models/目录下。

第三步是模型的测试。这一步用到的是predict.ipynb和predict\_all.ipynb。predict.ipynb对指定的一张图片进行测试，predict\_all.ipynb对指定的多张图片进行测试。测试时也是先将高分辨率材料图像分块后，输入深度学习模型得到图像块的语义分割结果，然后将图像块的分割结果拼接恢复为原图大小。图像块的切割和拼接在图1中有说明。