**小票识别系统报告**

1700012903朱立人

1700012908何颂华

一、简介

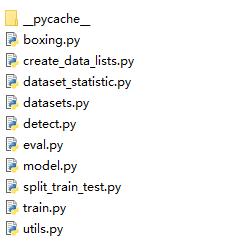
本组网络实习的project是实现一个简单的在线小票识别系统，主要目的为输入小票图片，在网页端将关键信息进行提取。主体的算法参考的是https://github.com/zzzDavid/ICDAR-2019-SROIE

代码总共分为四个部分，扫描收据文本本地化，扫描票据文本识别，关键信息提取，系统网页化。所有的代码已在https://github.com/PKU-ZLR/Computer-Networks-Project开源，以下会简要介绍各个部分的算法。目前开源的代码由于大小限制没有包含已经得到的训练模型的checkpoint和一些数据库文件。另外，一些文件路径也是根据本地实际情况设置，但是我们已经在演示中展示了系统运行的情况。

二、文本本地化

这一部分的主要目标为，定位发票图片的文本，输出4个顶点的坐标，就是所需要从给出的图片中确定文本框的bounding box. 参考算法中给出了CTPN 和SSD300 两种模型的代码，由于本次lab 硬件条件的限制，我们实现的是SSD300，之后将重点介绍该模型。

Part1代码主要结构如下：



split\_train\_test.py: 将原始图像和txt文件分割为训练集和测试集

create\_data\_lists.py: 创建json文件存储图像，对象和标签的ID

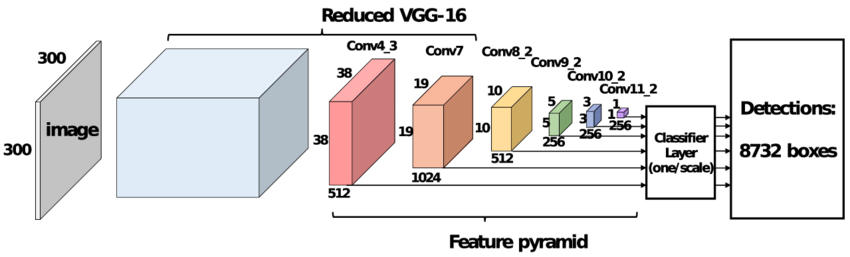
train.py : 定义训练的主要过程

model.py: 定义SSD模型

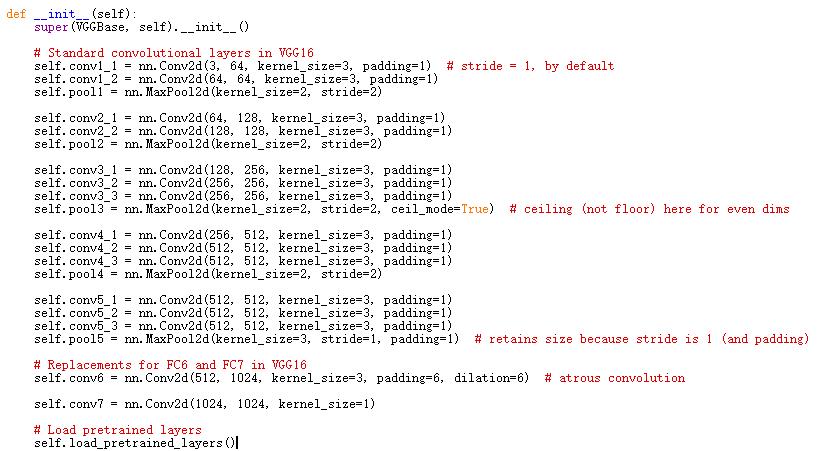
detect.py: 定义检测的主要程序，包括数据预处理和注释。

eval.py: 定义评估的主要过程

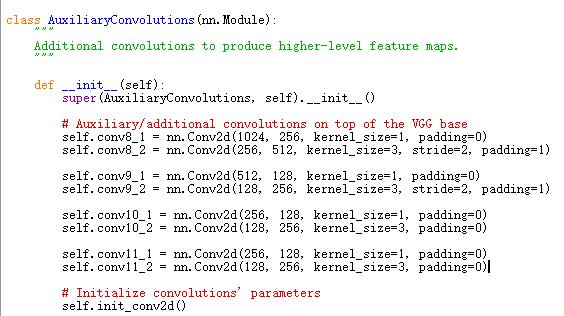
SSD300模型如下图所示：



SSD300修改了传统的VGG16网络将VGG16的FC6和FC7层转化为卷积层，可以从模型的初始化代码中看出图片所示的布局



在VGG16的基础上，SSD300多了4层的卷积，提供更高层的feature map，因此可以准确检测到不同尺度的物体



输入图片在SSD300中，经历了多个卷积层和池化层，我们可以在不同的卷积层会输出不同大小的feature map而且不同的feature map中含有不同的特征，而不同的特征可能对检测有不同的作用

SSD算法中使用到了conv4\_3，conv\_7，conv8\_2，conv7\_2，conv8\_2，conv9\_2，conv10\_2，conv11\_2这些大小不同的feature maps

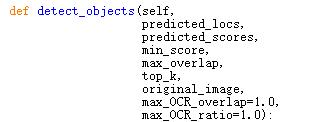
对于NMS算法，使用传统的NMS，有两个参数 min score，max overlap可以提高性能。如果算法min score设置得很更高，则很多正确的box会被消去。如果算法将min score设置为较低，那么将会出现很多empty box。 至于max overlap，我们倾向于降低以便清晰地分割。但是，它忽略了我们要保留更长的盒子而不是较短的盒子的目标。

因此，算法将根据统计信息再添加两个NMS：1、像素值总和，对于特定的单词组，带有更多单词的较长框具有较高的像素值总和；2、框的像素值平均值，这是与具有内容的框相比没有内容的框的像素值平均值更高。

以下是参考算法给出的图片，在这里引用以补充说明：



由此可以看到detect的对应参数如下：



这对应着我们的NMS的各个指标

SSD会用一种匹配策略给每个检测器分配一个真实标签(背景为0，前景为对应的物体类别，在这里我们只有0和1)。指定的标签为前景的预测称为正样本。标签为背景的预测是负样本，并不是所有的负样本都用来计算loss，因为负样本的数量远远多于正样本。此时按照预设正负样本比例1:3，挑选出一定数量的负样本。

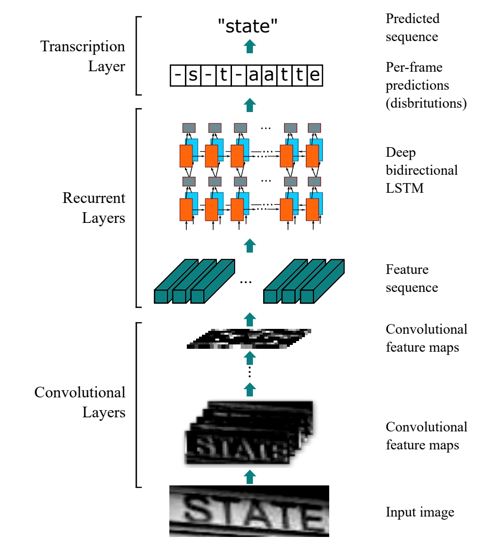
损失函数中N正负样本的数量之和，位置损失是预测框和真实标签值框参数之间的smooth-l1损失。置信损失是softmax对多类别的损失。为置信损失和位置损失的权重。

对应论文中的Loss函数公式为：

三、文本识别

这一步的目的主要是根据已经得到的bounding box进行字符识别，实际上已经有很多已有的图像识别的模型。参考算法中也根据https://github.com/meijieru/crnn.pytorch实现的，也就是比较常用的CRNN+CTC。

这里简要介绍一下CRNN，模型图示如下：



CRNN网络可以分为三个部分，Convolutional Layers，Recurrent Layers，Transcription Layers。

Convolutional Layers就是一个普通的CNN网络，用于提取输入图像的Convolutional feature maps，即将大小为(32,100,3)的图像转换为(1,25,512) 大小的卷积特征矩阵

Recurrent Layers是一个深层双向LSTM网络，在卷积特征的基础上继续提取文字序列特征。由于CNN输出的Feature map是(1,25,512)大小，所以对于RNN最大时间长度T=25

对于Recurrent Layers，如果使用常见的Softmax cross-entropy loss，则每一列输出都需要对应一个字符元素。那么训练时候每张样本图片都需要标记出每个字符在图片中的位置，再通过CNN感受野对齐到Feature map的每一列获取该列输出对应的Label才能进行训练。在这里无法标记这种对齐样本，另外，由于每张样本的字符数量不同，字体样式不同，字体大小不同，导致每列输出并不一定能与每个字符一一对应，所以，CTC是一种对不需要对齐的Loss计算方法。

CTC算法并不要求输入输出是严格对齐的。但是为了方便训练模型我们需要一个将输入输出对齐的映射关系，知道对齐方式才能更好的理解之后损失函数的计算方法和测试使用的计算方法。

由于我们在实现小票系统时基本上是引用的上述代码，本人和小组成员对这方面涉猎不深，这里就不详细解释CTC具体的实现了。我们在part2的主要工作是将原先的识别代码移植到我们自己的机器上，然后进行训练。但是可以肯定的是CRNN+CTC这种实现方式对于大部分的OCR问题在确定bounding box后是有效的

我们的代码结构如下包括create\_dataset.py,dataset.py创建训练数据，train.py进行训练，main.py用于detect，由于这方面已有前人的工作所以这方面的效果较好。

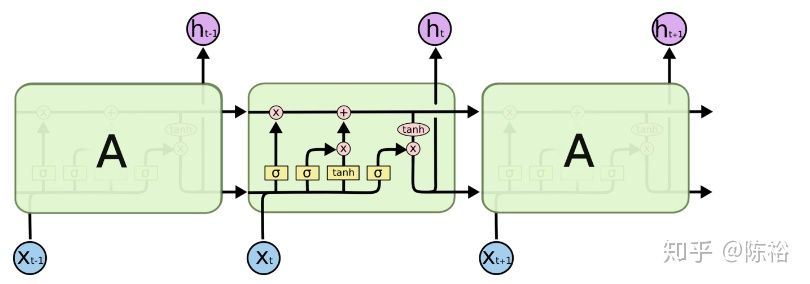
四、关键信息提取

这一步骤的目的是将步骤二中的文本提取关键信息，这里我们提取的关键信息有 "company"，"date"，"address"，"total"，当然只要有其他的训练和标注数据，也可以提取出其他的关键信息。对于票据我们主要关心的是上面四种文本。

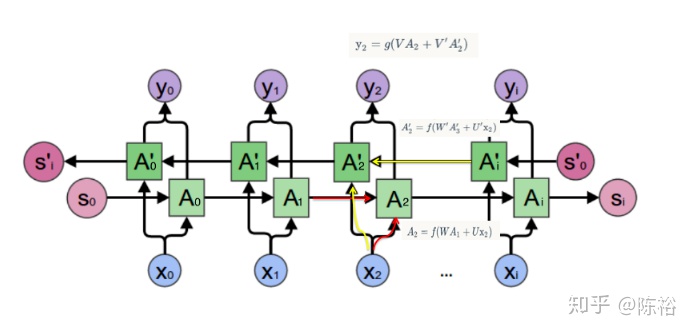
这一步本质上对应这一个文本分类问题。首先，将图像中的文本格式化为单个序列。然后我们对每个字符生成5种类别的分类标签，也就是4个关键信息类别和一个“其他”类别。 这里我们主要利用的PyTorch中实现两层双向LSTM，即可充分根据文本的上下文并输出高度准确的结果。

LSTM是对于RNN的一种变形，主要是为了解决长序列训练过程中的梯度消失和梯度爆炸问题。相比普通的RNN，LSTM在更长的序列中有更好的表现。

LSTM在RNN的基础上增加了细胞状态，以及用于控制细胞状态的忘记门、输入门和输出门，分别用于决定是否遗忘上一次的细胞状态、是否用当前输入更新细胞状态、是否输出当前细胞状态到隐藏层。LSTM隐藏层结构如下图所示：



而由于LSTM是单向的，为了能同时捕获上下文信息，又改进为Bi-LSTM，也即双向LSTM。大致思路是分别训练相反方向的LSTM，而输出时同时输出正向和反向的LSTM隐藏层输出。如下图所示：



代码是调用了pytorch.nn中现成模块实现的，num\_layers为2。其中my\_models模块用于创建LSTM模型，train.py和test.py分别用于训练和测试。代码的输入是上一部分输出的字符文本的pth格式文件，以及hidden\_size等参数，输出则是上述四个类型的文本信息。

最终我们会把分类的字符输出到对应的json文件之中，而我们的小票识别系统最终的输出也是基于这个json文件的。

五、系统网页化

这一部分主要是介绍是如何链接前三部分的代码，以及如何使用网页化。这里我们使用的轻量级的Python的网页化包Flask来实现的。

首先，先介绍如何将三个part结合起来。主要是通过system函数来执行代码，分别执行task1中的python task1 SSD\\_Method src detect.py --path --name，task2的python task2 main.py --name 和task3的python task3 src test.py --name，每一段代码会生成下一个代码的task所需要的输入到制定的文件路径之中。

上传的文件相当于向服务器post了一个新文件，然后我们会保存在static之下，而最终结果会在网页端展示，同时本地上的result文件也会保留结果的json文件。而Templates中的upload.html和upload\\_ok.html分别表示未上传文件时的网页模板和已上传后的网页模板。由于没有云端服务器，所以代码就在localhost上执行。

使用运行run.py之后，就可以在127.0.0.1:5000上看到网页

六、总结与反思

首先，这次的project是我们第一次接触有关OCR的问题，自身也没有很多的经验，许多代码也是参考了已经实现的算法。也有很多模型的深层数学原理没有完全理解，但是基本的步骤的理论还是有所理解的。这次实习课的前半段读的论文也让我对于我没有涉及过的方向有了新的认识。虽然，花费了很多时间进行学习，但是还是有很多的收获的。

由于没有并行设备，我们的代码只能在自己电脑的cpu上跑，所以训练出的模型识别效果不能说特别好，但是已经能够基本提取出关键的信息，如果有更好的设备应该可以有更好的效果

另外，没有网页端服务器，我们这次也是实现了一个简单的网页化，但是我们的代码还是可以在有服务器的情况下跑的，只需要修改其中运行的IP 即可。

如果有更多的标注数据，我们还可以提取更多不同的关键信息。但是由于我们在task1中的bounding box训练主要针对小票所以目前实现的是一个简单的小票系统，如果训练集是其他数据可以针对其他的进行识别。

最后，非常感谢助教和老师提供的这次学习的机会，也非常感谢助教和老师对于这次project的帮助，在线上学期这么艰难的情况下还是能够定期与我们讨论。

七、小组分工

主要代码实现工作及相应报告由朱立人完成（由于本人选修了网络实习所以大部分任务由本人完成），何颂华阅读了相关代码并撰写了第三部分关键信息提取的报告