基于CTPN和CRNN模型的中文检测和识别项目

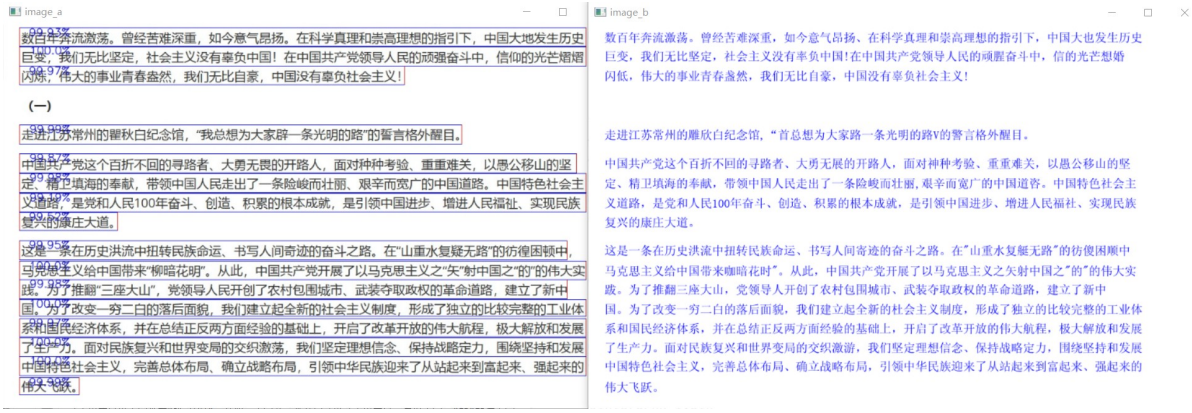
```
-detection:检测模块
-recognition:识别模块
-simsunttc:识别结果显示字体
-OCR.py 测试效果
```

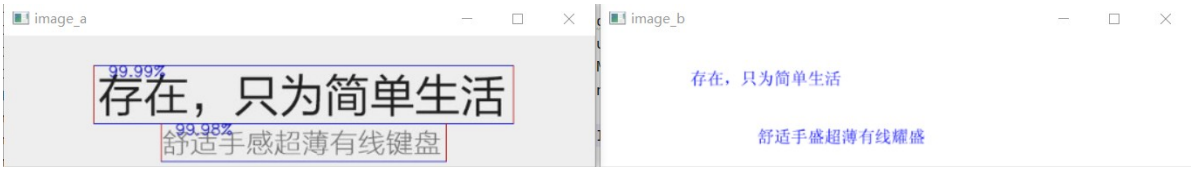
实验环境

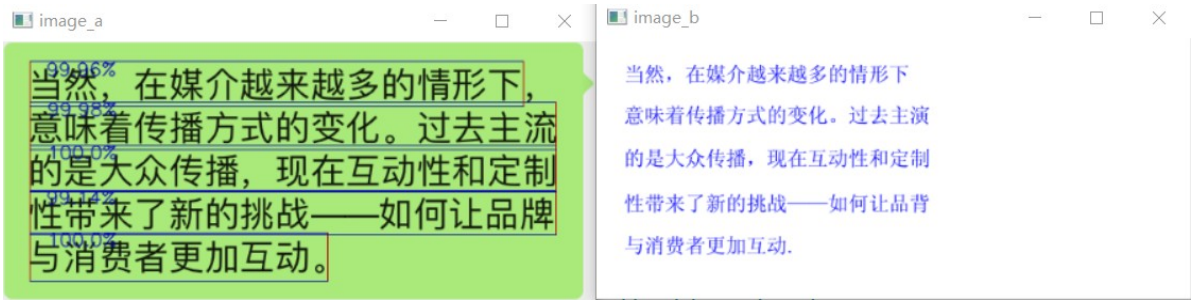
- win10-x64
- PyCharm
- CUDA 11.1
- python 3.8
- torch==1.9.0+cu111

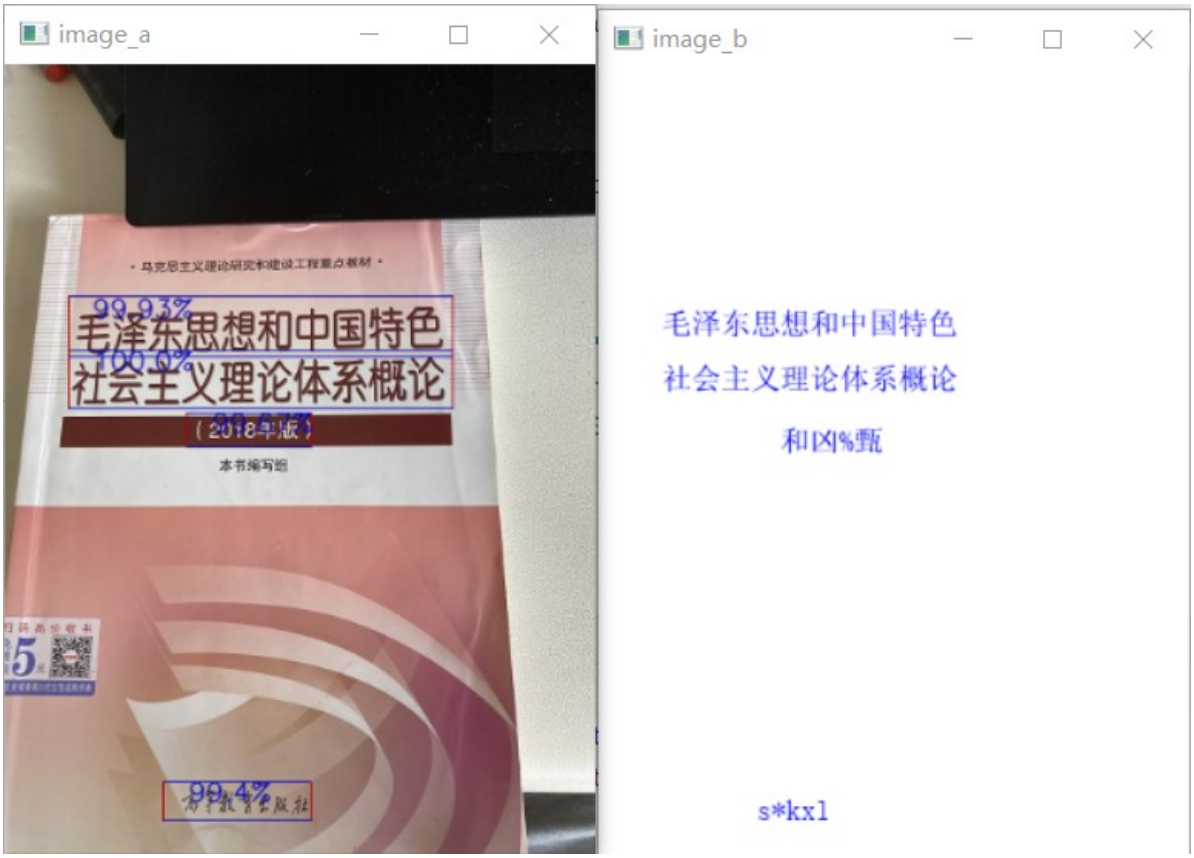
测试效果

对于水平文字(如印刷文字,消息截屏等)识别效果较好，对变形较大或者角度倾斜明显的文字难以识别准确。









代码解读

主要流程

训练

- 进行基于CTPN的文本检测训练
  - 通过VGG16提取特征
  - 通过CNN学习空间特征，Bi-LSTM学习序列特征
  - 经过RPN网络获得文本候选区域
  - 通过文本线构造方法将候选区域连接成文本检测框
- 进行基于CRNN+CTC的文本识别训练
  - 首先CNN提取图像卷积特征
  - 然后LSTM进一步提取图像卷积特征中的序列特征
  - 最后引入CTC解决训练时字符无法对齐的问题

测试:

- 给定图片，进行文本检测得到文本检测框
- 将检测框输入到文本识别模块得到预测结果。
- 将检测和识别结果分别保存。

基本架构

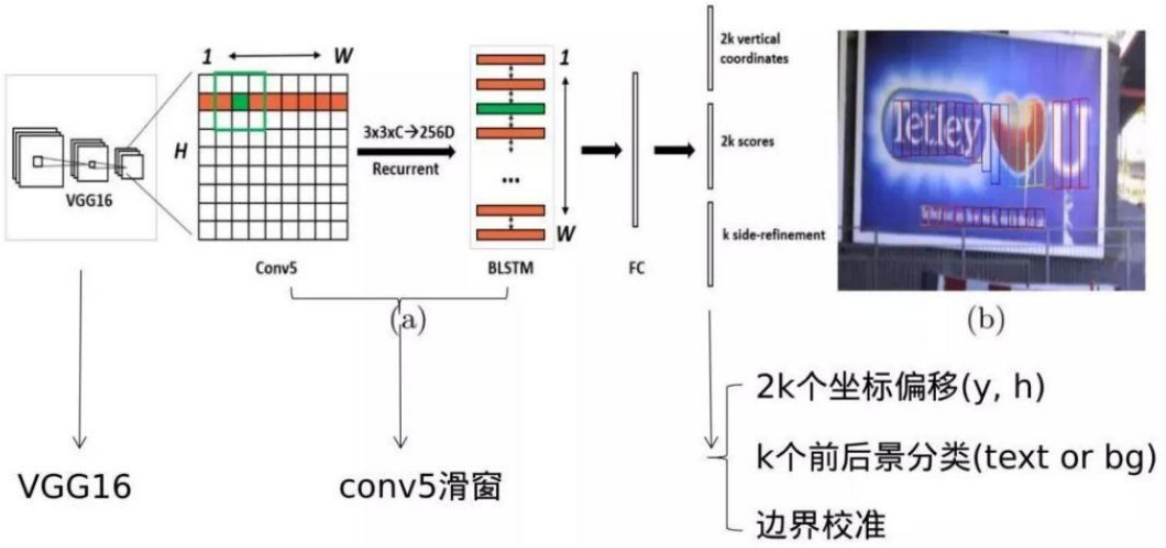
检测功能(detection): `CTPN`

识别功能(recognition): `CRNN`

模型介绍

CTPN: Connectionist Text Proposal Network

论文中给出的CTPN结构:



CTPN具体结构解读

➢ 通过 VGG16网络进行特征的提取，其Conv5层输出 N×C × H × W的特征图，由于VGG16 的卷积网络中经过 4个池化层累计的stride 为 16。也就是Conv5层输出的Feature Map中一个像素对应原图的16个像素。
➢ 在Conv5上做 3×3的滑动窗口，即每个点都结合周围 3×3区域特征获取一个长度为3×3 ×C 的特征向量。输出为 N×9C × H × W的 Feature Map ，该特征依然是由CNN 学习到的空间特征。
➢ 继续对上一步输出的Feature map进行Reshape操作：

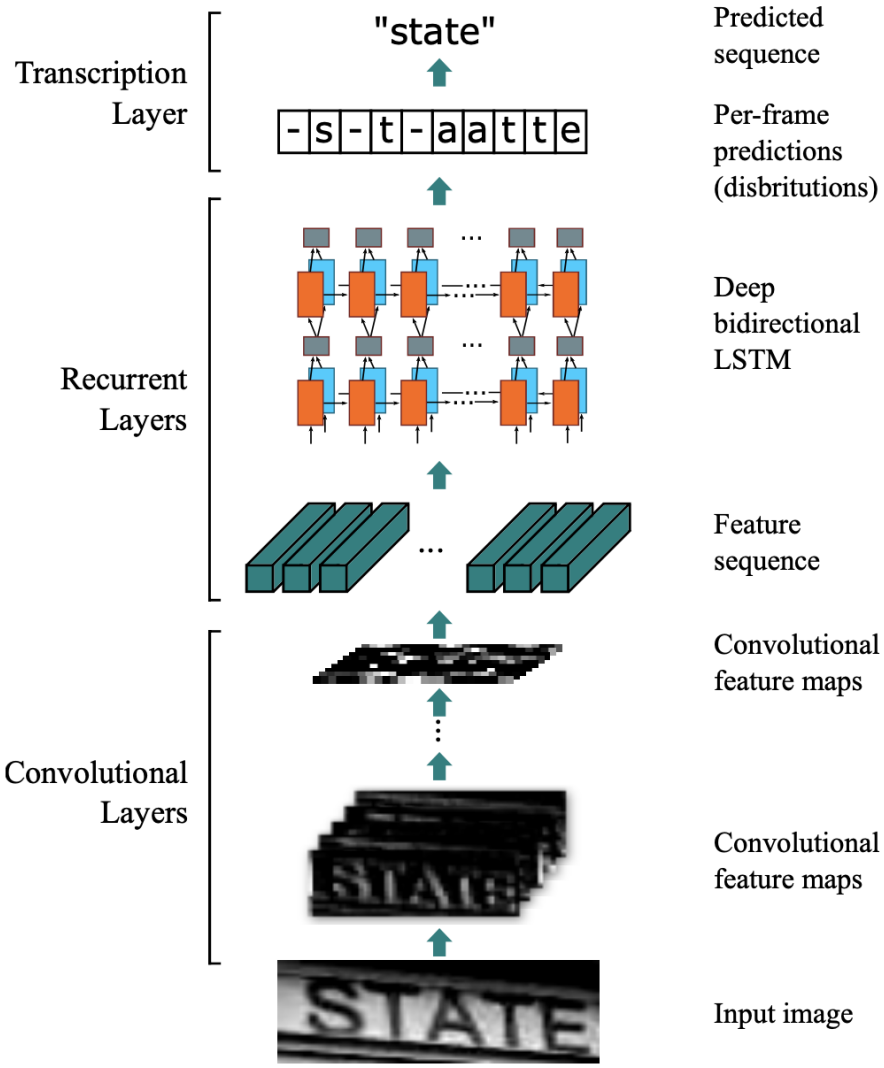
$$Reshape: N \times 9C \times H \times W -> (NH) \times W \times 9C$$

➢ 以Batch = NH且最大时间长度Tmax=W的数据流输入Bi-LSTM，学习每一行的序列特征。Bi-LSTM输出为(NH)×W×256，再经Reshape恢复形状。

$$Reshape: (NH) \times W \times 256 -> N \times 256 \times H \times W$$

➢ 此时学习到的特征既包含了空间特征，也包含了Bi-LSTM学习到的序列特征。
➢ 经过FC层，变为N×512×H×W的特征。
➢ 经过类似Faster-RCNN的RPN网络，获得Text Proposals,使用非极大值抑制算法来滤除多余的box。
➢ 用简单的文本线构造算法，把分类得到的文字的proposal合并成文本线。

CRNN 基本结构



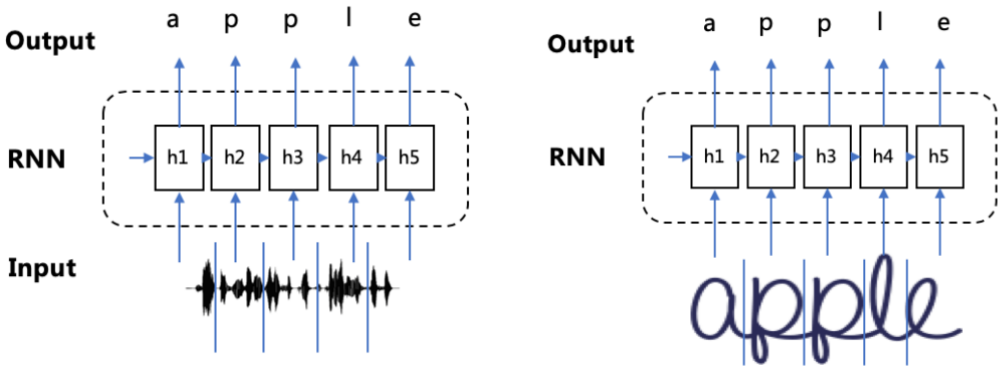
整个CRNN网络结构包含三部分，从下到上依次为：

- CNN（卷积层）：使用深度CNN，对输入图像提取特征，得到特征图；
- RNN（循环层）：使用双向RNN（BLSTM）对特征序列进行预测，对序列中的每个特征向量进行学习，并输出预测标签（真实值）分布；
- CTC loss（转录层）：使用CTC损失，把从循环层获取的一系列标签分布转换成最终的标签序列。

CNN和RNN这里不再赘述。

CTC:

一般来说RNN模型需要输入和输出序列是标注好的映射关系，但是在语音识别和文本识别中，采集的信号数据本身很难获取到大规模的具有良好映射关系的训练样本序列，此时RNN无法直接进行端到端的训练和预测。



CTC(Connectionist Temporal Classification)是Alex Graves等人在ICML 2006上提出的一种端到端的RNN训练方法，是一种不需要对齐的Loss计算方法，它解决了对齐问题。所以CTC也被广泛应用于文本行识别和语音识别中。

CTC的核心思路：

- 它扩展了RNN的输出层，在输出序列和最终标签之间增加了多对一的空间映射，并在此基础上定义了CTC Loss函数
- 它借鉴了HMM（Hidden Markov Model）的Forward-Backward算法思路，利用动态规划算法有效地计算CTC Loss函数及其导数，从而解决了RNN端到端训练的问题
- 最后，结合CTC Decoding算法RNN可以有效地对序列数据进行端到端的预测