



基于CTPN和CRNN模型的 中文检测和识别项目

汇报人:徐鹏博

2021.06.29

目录

01

项目简介

02

检测模块

03

识别模块

04

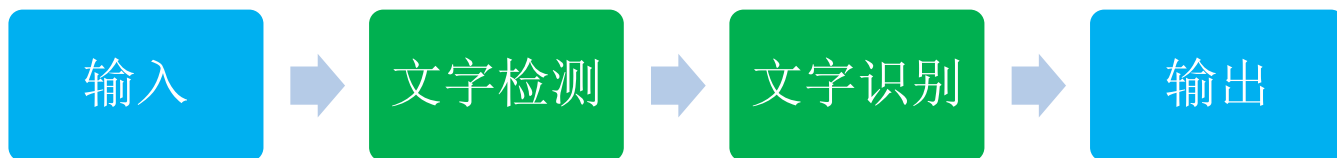
实验过程

05

总结展望

项目简介

图像的文字识别有着广泛的应用场景,它主要分为两个具体步骤:文字检测和文字识别。两者缺一不可,检测是识别的前提,识别是检测的目标。



近年来出现了各种基于深度学习的技术解决方案。对于检测功能,出现了CTPN、PSE、PAN等一系列模型,而对于识别部分,有Dense Net、NRTR、CRNN、RARE等多种模型。

本项目基于CTPN模型实现检测功能(detection),基于CRNN模型实现识别功能(recognition)。

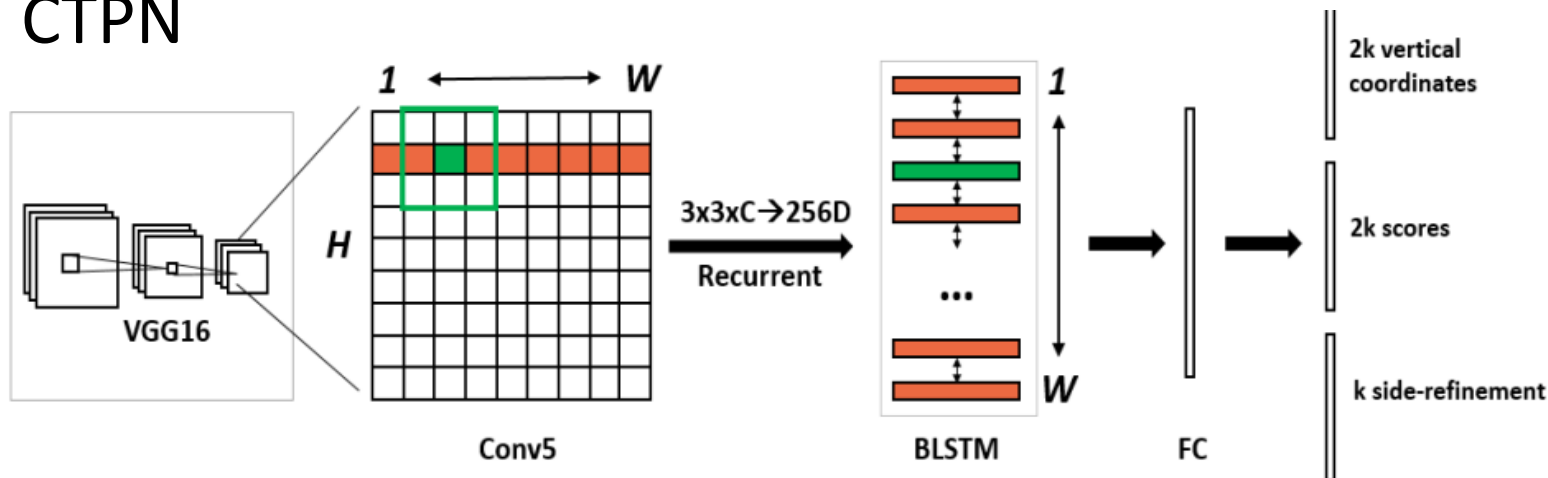
CTPN

CTPN是目前流传最广、影响最大的开源文本检测模型,它来自2016年的一篇论文:《Detecting Text in Natural Image with Connectionist Text Proposal Network》,直到今天这个网络框架一直是OCR系统中做文本检测的一个常用网络。

CTPN可以检测水平或微斜的文本行。文本行可以被看成一个字符sequence,而不是一般物体检测中单个独立的目标。同一文本行上各个字符图像间可以互为上下文,在训练阶段让检测模型学习图像中蕴含的这种上下文统计规律,可以使得预测阶段有效提升文本块预测准确率。

检测模块

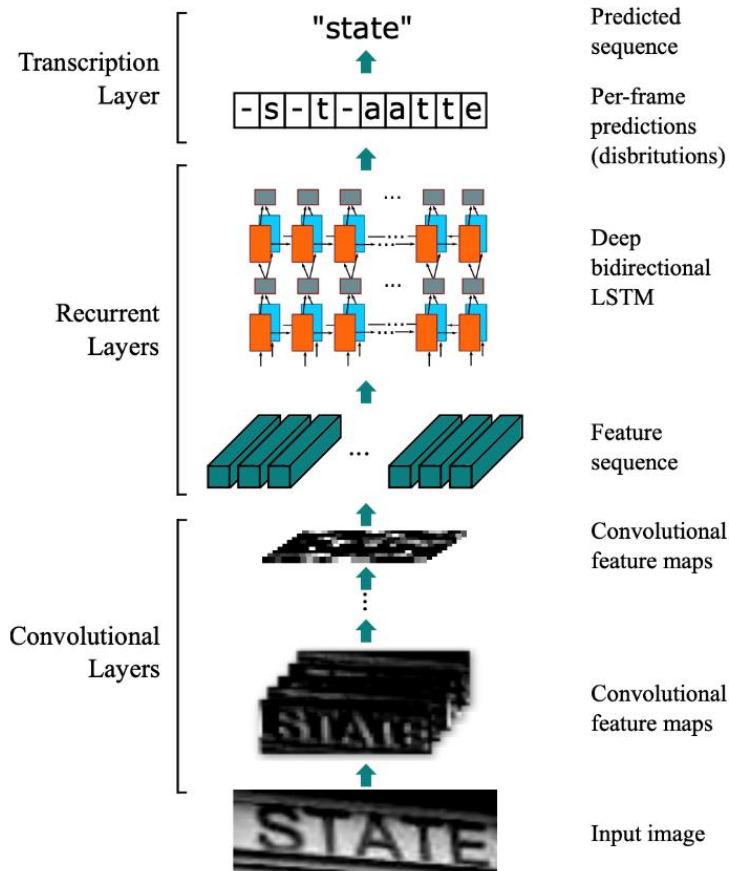
CTPN



- 前端使用VGG16网络来提取各字符局部图像特征, 通过CNN学习到空间特征;
- 中间使用BLSTM层提取字符序列上下文特征, 学习到序列特征;
- 然后通过FC全连接层, 末端经过预测分支输出文字块坐标值和分类结果概率。
- 在数据后处理阶段, 将合并相邻的小文字块为文本行。

识别模块

CRNN



现今基于深度学习的端到端OCR技术有两大主流技术：**CRNN OCR**和**attention OCR**。其实这两大方法主要区别在于最后的输出层(翻译层),即如何将网络学到的序列特征转化为识别结果。

特征学习阶段他们都采用了**CNN+RNN**的网络结构,但在最后的对齐阶段,**CRNN**采取的方式是**CTC**算法,而**attention**采取的方式则是**attention**机制。

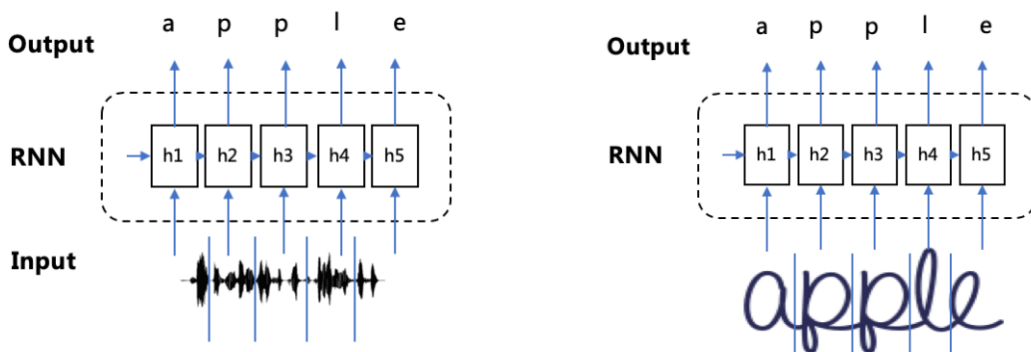
本项目采用的识别算法是更为广泛的**CRNN**。

- **CNN(卷积层):** 使用深度CNN提取图像特征得到特征图。
- **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。

识别模块

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,但是在语音和文本的识别任务中,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,导致RNN无法直接进行端到端的训练和预测。



而CTC的提出解决了对齐问题,所以CTC也被广泛应用于文本识别和语音识别中。

- 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的Forward-Backward算法思路,利用动态规划算法计算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。

实验过程

训练阶段

1. 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连接成文本检测框

2. 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

测试阶段

- 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。
- 对文本行按原图中位置顺序重新组织排列。

实验过程

改进工作

- 在CRNN中构造dataset对象时,将label文本按长度重新排序,保证同一batch相对等长。
- 针对微斜情况进行处理,将检测的倾斜字框旋转分割传入识别模块。
- 尝试换用VGG19和Resnet作为基础网络进行训练,但提升作用不明显,放弃

部分训练日志

CTPN

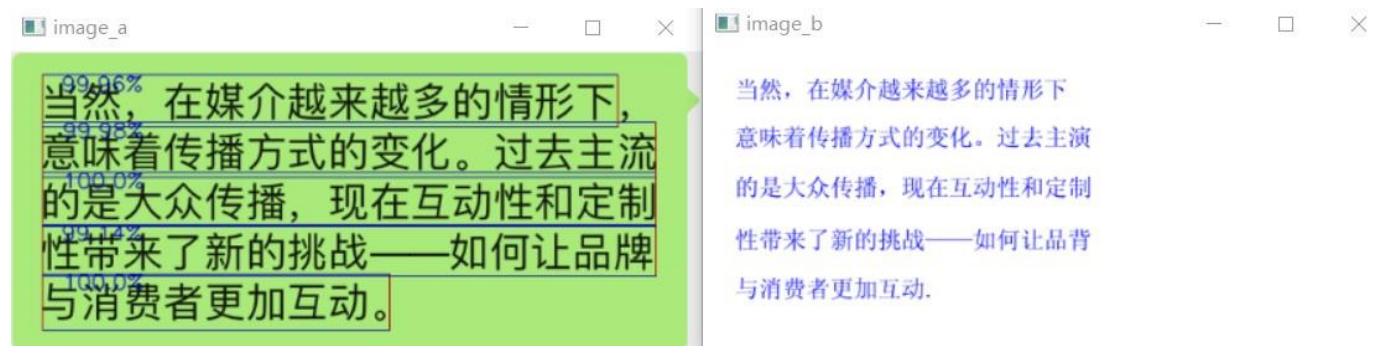
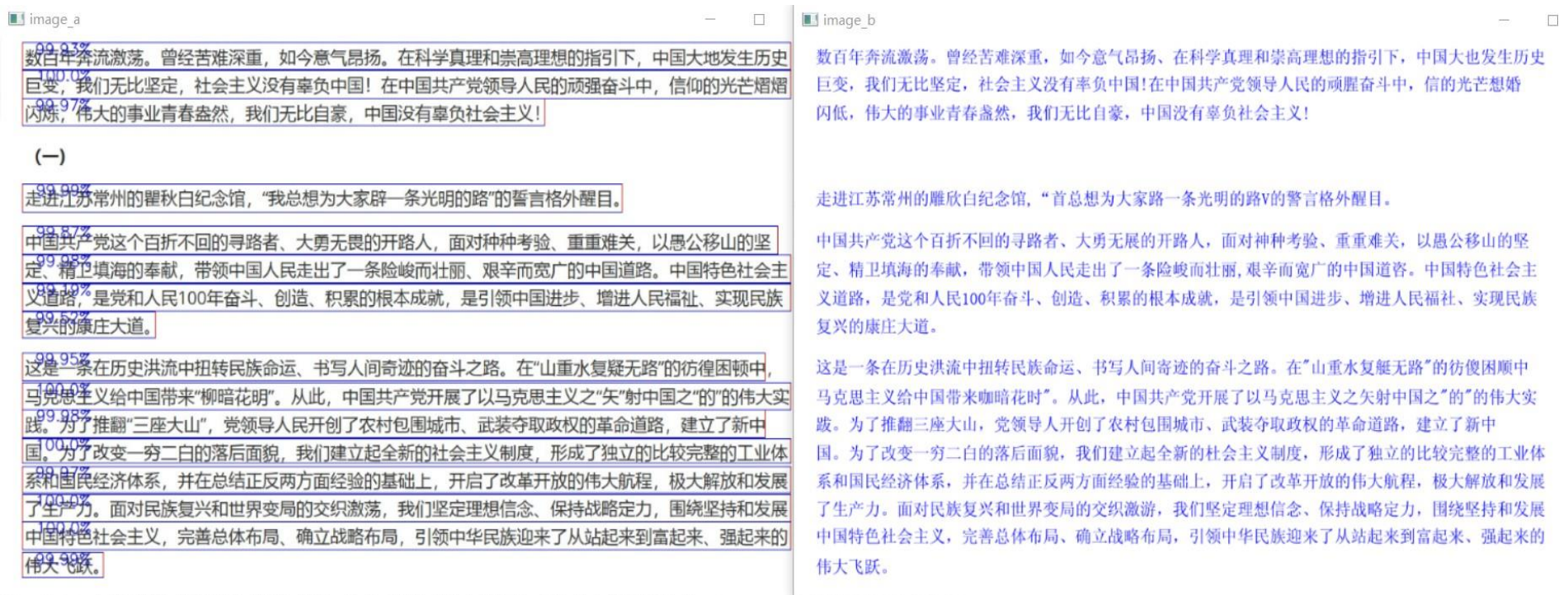
ctpn_ep01_0.1561_0.1756_0.3317.pth
ctpn_ep02_0.0963_0.1407_0.2371.pth
ctpn_ep03_0.0741_0.1220_0.1961.pth
ctpn_ep04_0.0626_0.1111_0.1736.pth
ctpn_ep05_0.0551_0.1020_0.1572.pth
ctpn_ep06_0.0473_0.0943_0.1417.pth
ctpn_ep07_0.0426_0.0856_0.1282.pth
ctpn_ep08_0.0385_0.0827_0.1211.pth

CRNN

checkpoint_0_acc_0.1052.pth	checkpoint_0_acc_0.0199.pth
checkpoint_1_acc_0.4003.pth	checkpoint_1_acc_0.2780.pth
checkpoint_2_acc_0.5042.pth	checkpoint_2_acc_0.4168.pth
checkpoint_3_acc_0.5502.pth	checkpoint_3_acc_0.4826.pth
checkpoint_4_acc_0.5832.pth	checkpoint_4_acc_0.5104.pth
checkpoint_5_acc_0.6002.pth	checkpoint_5_acc_0.5455.pth
checkpoint_6_acc_0.6185.pth	checkpoint_6_acc_0.5569.pth
checkpoint_7_acc_0.6218.pth	checkpoint_7_acc_0.5607.pth
checkpoint_8_acc_0.6293.pth	checkpoint_8_acc_0.5708.pth
checkpoint_9_acc_0.6308.pth	checkpoint_9_acc_0.5737.pth
checkpoint_10_acc_0.6457.pth	checkpoint_10_acc_0.5841.pth
checkpoint_11_acc_0.6520.pth	checkpoint_11_acc_0.5873.pth
checkpoint_13_acc_0.6553.pth	checkpoint_12_acc_0.6018.pth
checkpoint_16_acc_0.6598.pth	checkpoint_14_acc_0.6040.pth
checkpoint_17_acc_0.6663.pth	checkpoint_17_acc_0.6082.pth
checkpoint_19_acc_0.6695.pth	checkpoint_18_acc_0.6126.pth
checkpoint_21_acc_0.6697.pth	checkpoint_23_acc_0.6180.pth

实验过程

实验效果



总结展望

CTPN 从Faster R-CNN改进而来,能有效的检测出复杂场景的横向分布的文字,比如用来进行标准格式印刷体的检测时效果很好,但是对于倾斜角度较大的场景不太适应。

目前已经出现一些对于任意角度的文字检测,如 EAST 模型在目标框回归预测时,如果加上回归框的角度信息,就可以用来检测旋转文本; SegLink模型既融入CTPN小尺度候选框的思路,又加入了SSD算法的思路。

不过现在也出现了一些端到端模型,直接从图片中定位和识别出所有文本内容来。

FOTS是图像文本检测与识别同步训练、端到端可学习的网络模型。检测和识别任务共享卷积特征层,既节省了计算时间,也比两阶段训练方式学习到更多图像特征。另外它引入了旋转感兴趣区域,可以从卷积特征图产生定向文本区域,从而支持倾斜文本的识别。

STN-OCR集成了图文检测和识别功能,可进行端到端的学习。它的检测部分嵌入了一个空间变换网络来对原始输入图像进行仿射变换。利用这个空间变换网络,可以对检测到的多个文本块分别执行旋转、缩放和倾斜等图形矫正动作,而在后续文本识别阶段得到更好的识别精度。STN-OCR属于半监督学习,只需要提供文本内容标注,而不要求文本定位信息。

文本检测和识别任务可以利用上述模型实现更好的效果。



Thanks
