实 验 报 告

实验名称：基于 CTPN 和 CRNN 模型的中文检测和识别

课程名称：认知科学与类脑计算

实验地点：K2-206

实验日期：2021-05-29

班　　级：人工智能 18.1 班

姓　　名：徐鹏博

学　　号：201800130086

一 实验目的：

基于 CTPN 模型实现检测功能(detection),基于 CRNN 模型实现识别功能

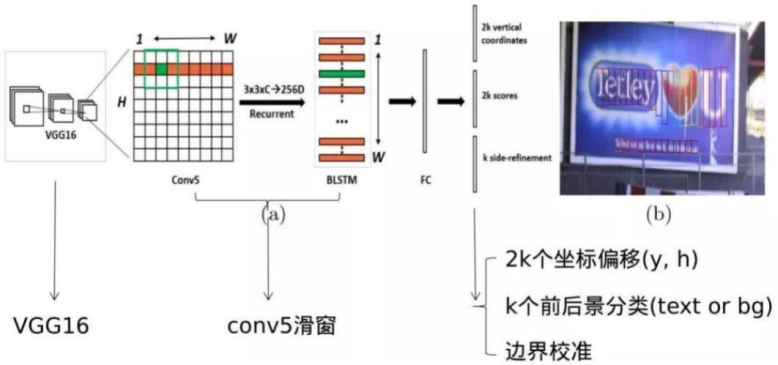
二 实验环境：

Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz
win10-x64
PyCharm
CUDA 11.1
python 3.8
torch==1.9.0+cu111

三 实验原理

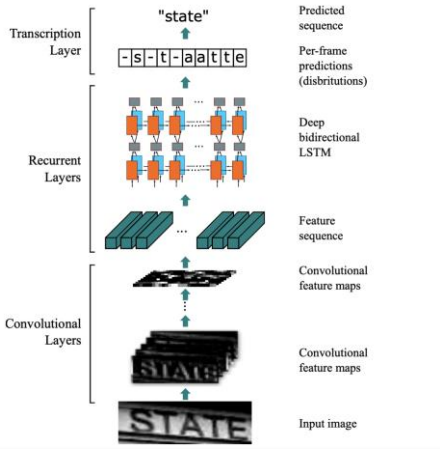
CTPN

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。



- 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。

CRNN



整个 CRNN 网络结构包含三部分，从下到上依次为：

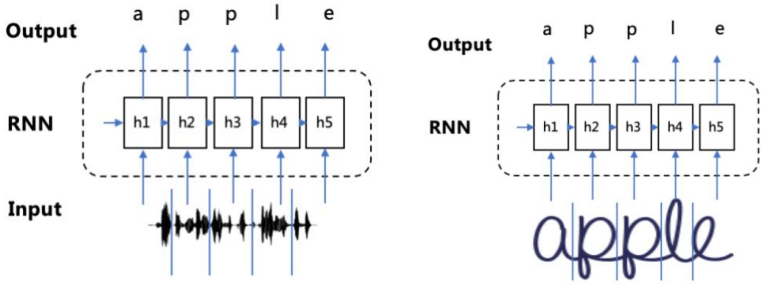
CNN(卷积层): 使用深度 CNN，对输入图像提取特征，得到特征图；

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，把从循环层获取的标签分布转换成最终的标签序列。

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.

四 实验步骤：

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。

五 实验结果

1. 实验日志(部分)

CTP 日志

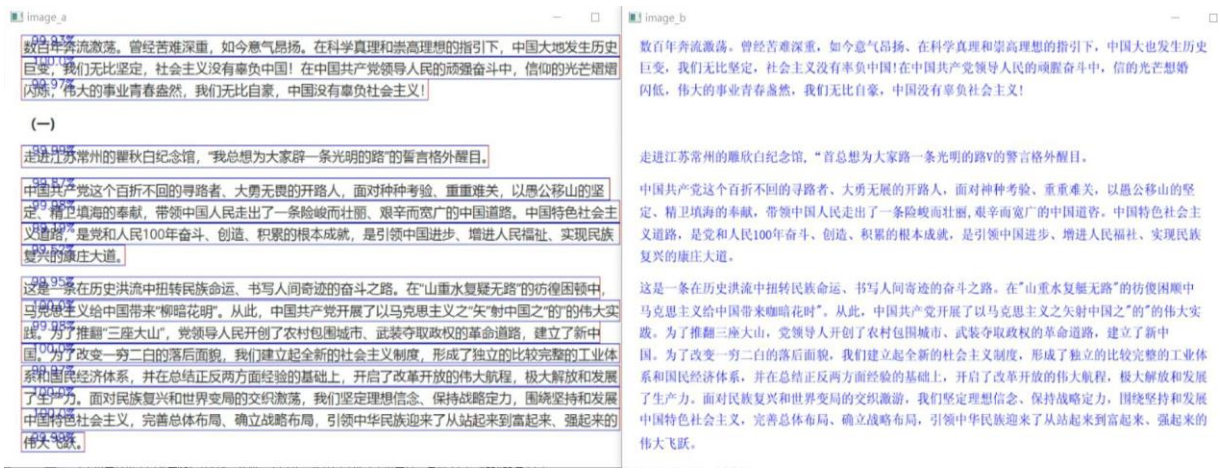
ctpn_ep01_0.1561_0.1756_0.3317.pth
ctpn_ep02_0.0963_0.1407_0.2371.pth
ctpn_ep03_0.0741_0.1220_0.1961.pth
ctpn_ep04_0.0626_0.1111_0.1736.pth
ctpn_ep05_0.0551_0.1020_0.1572.pth
ctpn_ep06_0.0473_0.0943_0.1417.pth
ctpn_ep07_0.0426_0.0856_0.1282.pth
ctpn_ep08_0.0385_0.0827_0.1211.pth

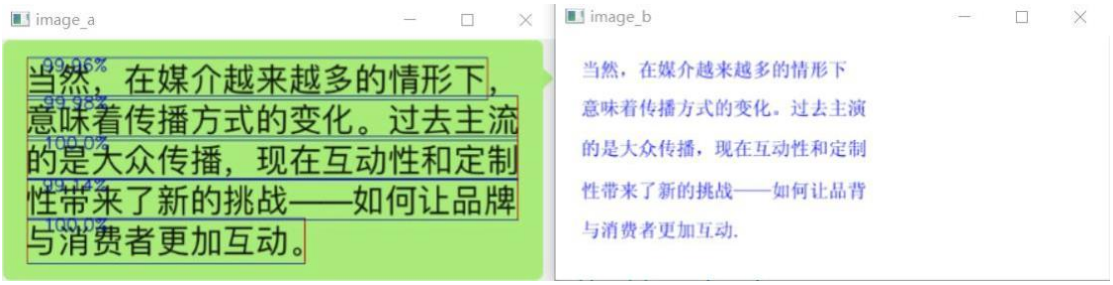
CRNN 日志

checkpoint_0_acc_0.1052.pth
checkpoint_1_acc_0.4003.pth
checkpoint_2_acc_0.5042.pth
checkpoint_3_acc_0.5502.pth
checkpoint_4_acc_0.5832.pth
checkpoint_5_acc_0.6002.pth
checkpoint_6_acc_0.6185.pth
checkpoint_7_acc_0.6218.pth
checkpoint_8_acc_0.6293.pth
checkpoint_9_acc_0.6308.pth
checkpoint_10_acc_0.6457.pth
checkpoint_11_acc_0.6520.pth
checkpoint_13_acc_0.6553.pth
checkpoint_16_acc_0.6598.pth
checkpoint_17_acc_0.6663.pth
checkpoint_19_acc_0.6695.pth
checkpoint_21_acc_0.6697.pth

checkpoint_0_acc_0.0199.pth
checkpoint_1_acc_0.2780.pth
checkpoint_2_acc_0.4168.pth
checkpoint_3_acc_0.4826.pth
checkpoint_4_acc_0.5104.pth
checkpoint_5_acc_0.5455.pth
checkpoint_6_acc_0.5569.pth
checkpoint_7_acc_0.5607.pth
checkpoint_8_acc_0.5708.pth
checkpoint_9_acc_0.5737.pth
checkpoint_10_acc_0.5841.pth
checkpoint_11_acc_0.5873.pth
checkpoint_12_acc_0.6018.pth
checkpoint_14_acc_0.6040.pth
checkpoint_17_acc_0.6082.pth
checkpoint_18_acc_0.6126.pth
checkpoint_23_acc_0.6180.pth

2. 检测和识别结果





六 分析改进

- 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 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,放弃替换。

七 实验小结：

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,从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。