

人民币 编码识别

团队：wei

目录

1



编码检测

CTPN

2



编码识别

CRNN+OCR_DENSENET+ASTER

3



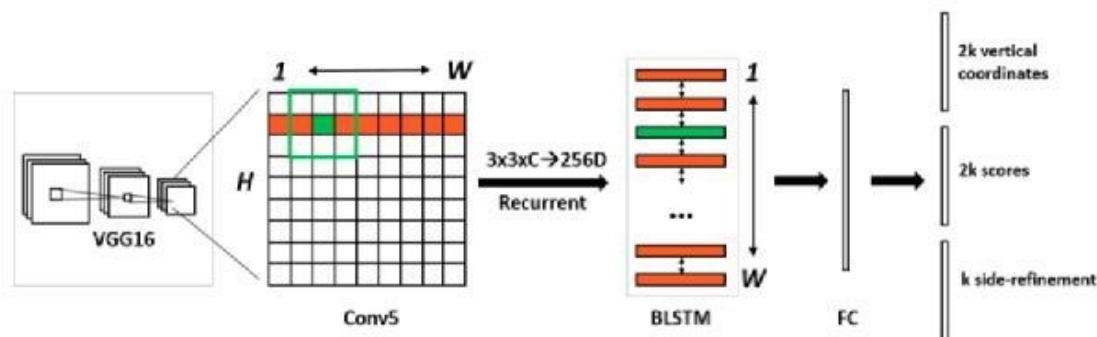
细节处理

4



模型融合

CTPN

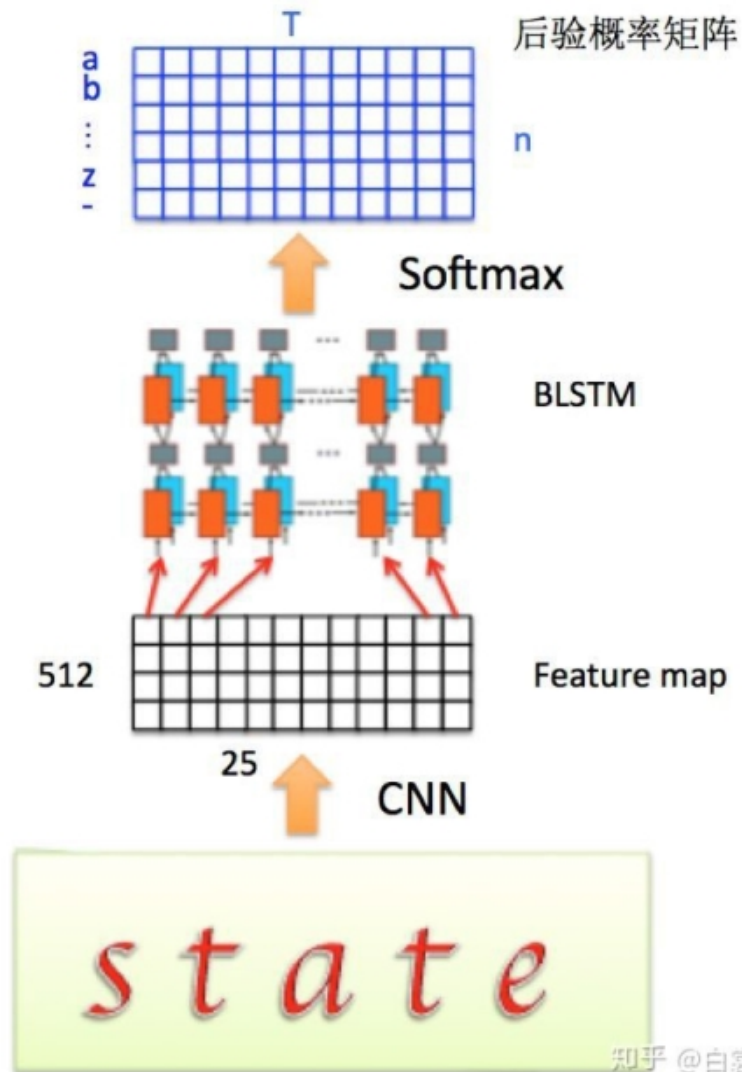


- 1.首先，用VGG16的前5个Conv stage得到feature map，大小为 $W \times H \times C$
- 2.用 3×3 的滑动窗口在前一步得到的feature map上提取特征，利用这些特征来对多个anchor进行预测,这里anchor定义与之前faster-rcnn中的定义相同，也就是帮我们去界定出目标待选区域。
- 3.将上一步得到的特征输入到一个双向的LSTM中，输出 $W \times 256$ 的结果，再将这个结果输入到一个512维的全连接层（FC）。
- 4.最后通过分类或回归得到的输出主要分为三部分，根据上图从上到下依次为 $2k$ vertical coordinates:表示选择框的高度和中心的y轴的坐标； $2k$ scores:表示的是 k 个anchor的类别信息，说明其是否为字符； k side-refinement表示的是选择框的水平偏移量。本文实验中anchor的水平宽度都是16个像素不变，也就是说我们微分的最小选择框的单位是“16像素”。
- 5.用文本构造的算法，将我们得到的细长的矩形（如下图），然后将其合并成文本的序列框。



CRNN+CTC

1. 首先CNN提取图像卷积特征
2. 然后LSTM进一步提取图像卷积特征中的序列特征
3. 最后引入CTC解决训练时字符无法对齐的问题



OCR_DENSENET

模型：

采用densenet结构，模型输入为 (64×512) 的图片，输出为 $(8 \times 64 \times 35)$ 的概率。
将图片划分为多个 (8×8) 的方格，在每个方格预测35个字符的概率。

Loss：

将 $(8 \times 64 \times 35)$ 的概率沿着长宽方向取最大值，得到 (35) 的概率，表示这张图片里有对应字符的概率。

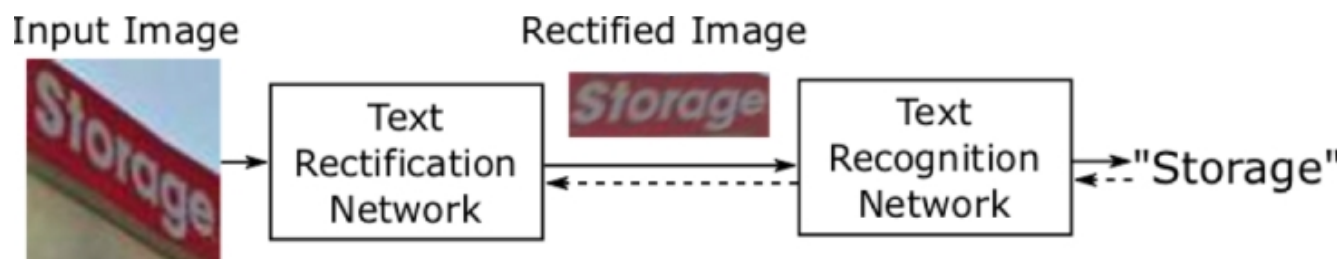
balance：

对正例和负例分别计算loss，使得正例loss权重之和与负例loss权重之和相等，解决数据不平衡的问题。

文字检测：

将 $(8 \times 64 \times 35)$ 的概率沿着宽方向取最大值，得到 (64×35) 的概率。沿着长方向一个个方格预测文字，然后连起来可得到一句完整的语句。

ASTER



ASTER是一种端到端神经网络模型，它由校正网络和识别网络组成。校正网络自适应地将输入图像转换为新的图像，对其中的文本进行校正。它由一个灵活的Thin-Plate Spline transformation作为核心，该转换处理各种文本不规则性，并在没有人工注释的情况下进行训练。识别网络是一个注意序列到序列的模型，它直接从校正后的图像预测字符序列。整个模型从头到尾进行训练，只需要图像和它们的groundtruth文本。通过大量的实验，验证了校正的有效性，并证明了ASTER的最新识别性能。此外，我们还证明了ASTER在端到端识别系统中是一个强大的组件，因为它能够增强检测器。

细节处理

标记、检测阶段:

1.每种面值标记15-20张

2.考虑到尽量减少背景对识别的影响, 标签框与编码区域的IOU尽可能的大

3.0.1、0.2、0.5面值的图片很多比较模糊,进行去雾、锐化处理(图像识别/目标检测都适用)



细节处理

训练阶段:

- 1、错误标签处理:在训练过程中，校验验证集发现提供的编码文件含有少量标记错误的样本，用模型对训练集进行一次识别，找出明显错误的标签，我的处理比较简单粗暴，直接删除了对应的图片
- 2、误检样本重采样:对误检样本进行分析，发现大部分为0.1、0.2、0.5面值中的3，8，9造成的，人眼基本都难以分辨，生成训练样本时，增加这部分的比重
- 3、样本尺寸选则:为了方便训练，我们一般会将图片缩放到统一的尺寸，有两种处理方法，1)缩放到特定宽度，保持长宽比的同时，对剩余部分填充0. 2)直接找出中间比例，进行**resize**，我们的样本都为定长字符，长宽比差异很小，我这里选择了2，直接将图片**resize**到（32,192）

细节处理

CTC/DENSENET连续字符识别不住的解决方案

模型预测结果preds : [1,64,35]

[13.810794904825986, -14.016372033970892, -13.247148407280104, -12.5448224270312476,
-8.709649026730835, -3.024013329570577,

.....

-8.651944672590915]

获取概率最大的下标，通过dict转成我们的预测结果，背景填充*

示例1:

*****HH*888*YYY**666*44422222222000*444*000*****

原始识别结果: H8Y642040

校验结果: H8Y6422040

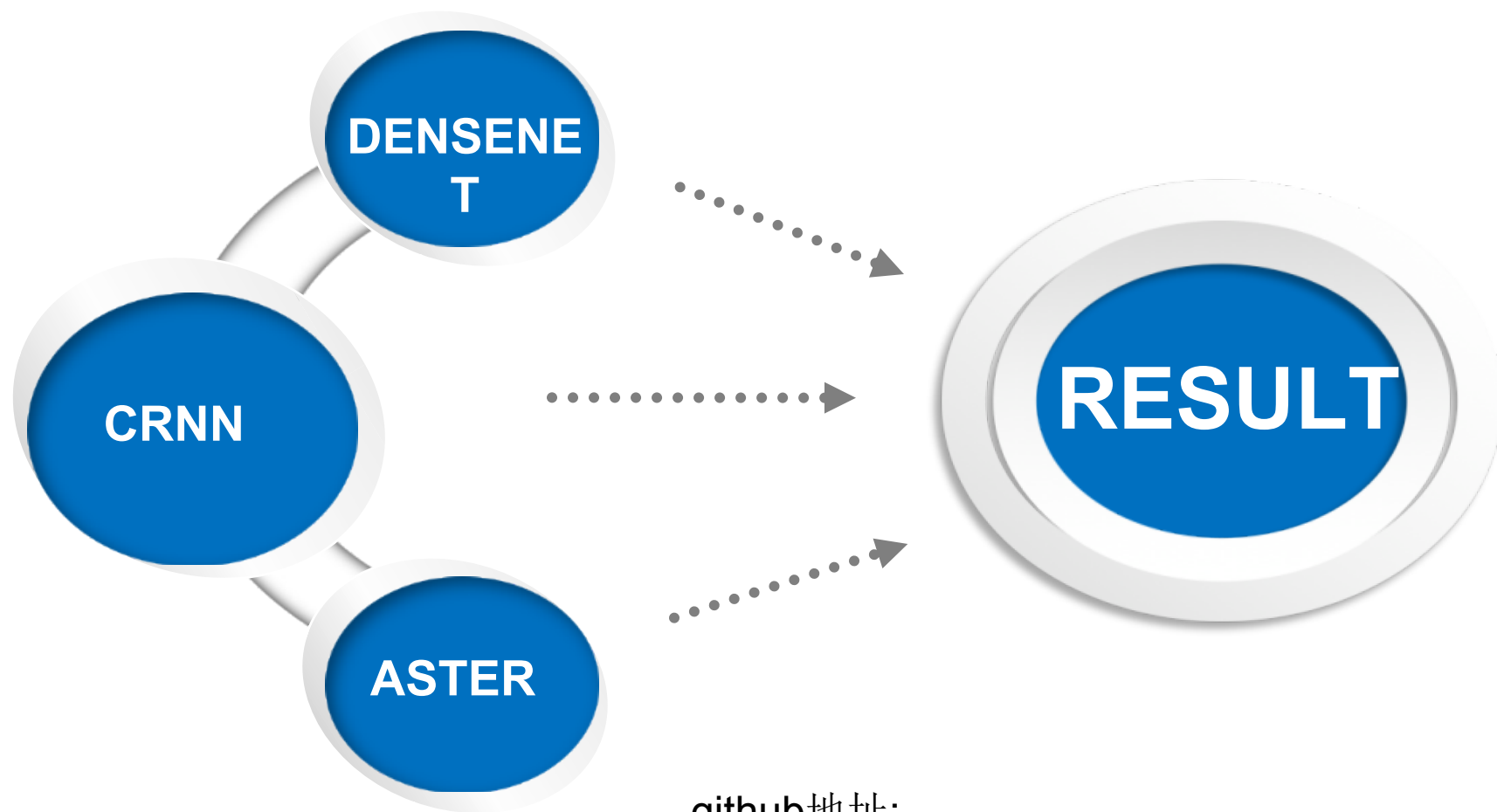
示例2:

*****SDD***PHY*111*555*444*0003333*777*6663333*****

原始识别结果: SDPHY15403763

校验结果: DH15403763

模型融合



github地址:

<https://github.com/hwwwu/ctpn-crnnc>

https://github.com/yinchangchang/ocr_densenet

<https://github.com/bgshih/aster>

THANKS