计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(12):262−267 [doi: 10.15888/j.cnki.csa.008227]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于并行子空间优化的企业命名实体识别①

乔诗展, 陈逸伦

(西北工业大学 航天学院, 西安 710072)
通讯作者: 乔诗展, E-mail: qiaoshizhan@mail.nwpu.edu.cn

摘　要: 针对企业命名实体的识别任务的过程复杂、学科交叉、实时性差等难点, 提出了一种基于并行子空间优化的方法. 首先, 建立系统的目标-约束方程完成系统级优化; 其次, 再通过构建文字检测、文字识别两级模型, 并考虑现存不同模型的优缺点进行模型选择的方法对涉及学科进行并行优化; 随后, 再使用图像阈值、灰度化、霍夫变换等算法构建两级模型的衔接; 最后, 通过仿真实验, 验证了本文方法相比其他两级文字检测识别模型的识别准确率提高了 9%, 推理速度提升约 20%.

关键词: 命名实体识别; 文字检测; 文字识别; 并行子空间优化

引用格式: 乔诗展,陈逸伦.基于并行子空间优化的企业命名实体识别.计算机系统应用,2021,30(12):262–267. http://www.c-s-a.org.cn/1003-3254/8227.html

Enterprise Named Entity Recognition Based on Concurrent Subspace Optimization

QIAO Shi-Zhan, CHEN Yi-Lun

(School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: Concerning the complicated process, interdisciplinarity, and poor real-time performance of enterprise named entity recognition, a method based on concurrent subspace optimization is proposed. First, a target-constrained equation of the system is established to complete system-level optimization; secondly, a two-level model of text detection and recognition is constructed, and the model is selected, considering the advantages and disadvantages of different existing models, to optimize the discipline in parallel; then, the connection of the two-level model is constructed with the image threshold, grayscale and Hoff transform; finally, simulation experiments verify that the recognition accuracy of this method is 9% higher than that of other two-level text detection and recognition models, and the speed increases by about 20%.

Key words: named entity recognition; text detection; text recognition; concurrent subspace optimization

命名实体识别, 指在一张图像或一段文字中提取出特定的名词, 如人名、地名、企业名称等. 而随着相关智能手机软件, 如美团、大众点评的发展, 用户直接使用手机拍照并实时获得企业和店铺相关信息或将成为未来一段时间智能手机软件的发展趋势. 此外, 企业命名实体识别也为工商备案网站查找违建、非法营业企业等提供了便利.

识别企业命名实体首先需要检测出图片中包含的命名实体, 目前现有的目标检测框架主要有 YOLOv5[1] 端对端目标检测架构, 该算法将候选区域生成及物体分类集成至单网络中, 并使用 Efficient Net[2] 的策略将网络缩放至 5 个不同大小, 以获得不同精度的区域重合度 IOU (Intersection Over Union) 准确率和分类准确率, 其最小的模型推理一张图片仅需 0.007 s, 且在 COCO 数据集上的 mAP 达到了 45%. 此外, 如 Faster-RCNN[3]、Mask-RCNN[4] 等两步目标检测/分割网络也

① 收稿时间: 2021-03-10; 修改时间: 2021-04-07; 采用时间: 2021-04-13

得到了广泛应用, 其基本思路是使用全卷积神经网络作为候选区域的提取网络, 并使用 VGG16[5]、ResNet18[6]等物体分类网络进行分类, 从而分步获取物体边界矩形框及物体的类别. 而除了针对广泛目标检测的网络, 专门应用于文字检测的网络如 EAST[7], 采用并行式端对端的架构, 通过特征提取全卷积神经网络、非极大值抑制等方法, 直接预测文本行位置.

对于检测后的文本, 则还需要进行识别, 基于传统算法的 Tesseract 框架[8] 使用直方图阈值分割文字及背景, 并使用霍夫变换的方法将倾斜文本进行旋转, 随后通过模板匹配的方法进行识别, 对英文字体达到了 97.3% 的识别准确率, 而对中文字体则支持较差. 而目前基于神经网络的主流的文本识别框架如 Attention-OCR[9]、CRNN[10] 等, 均采用了 LSTM (Long Short-Term Memory networks) 结合 CTC (Connectionist Temporal Classification) 或 Attention 机制的方法, 进行图像大小更改、特征提取等步骤, 使文本识别准确率提高, 但其存在泛化性能较差、需要至少 100 万数据量进行训练、难以迁移学习等特点.

由于企业命名实体识别任务具有多学科交叉、处理过程繁琐等特点, 因此本文提出将机械设计中常用的并行子空间优化方法 (Concurrent SubSpace Optimization, CSSO) [11] 应用于该任务中. 该方法与单级的多学科优化 (Multidisciplinary Design Optimization, MDO) 方法[12] 步骤相类似, 二者均通过数学模型进行优化求解, 结合了并行化设计的思想, 但 CSSO 方法首先通过对系统级进行优化, 又再次对系统中的多学科进行并行优化, 比单级的多学科优化更能适应系统的复杂性.

因此, 本文考虑到企业命名实体识别任务的复杂性, 首先通过建立系统的目标-约束方程的方法对系统进行约束层面上的优化. 在满足系统约束的条件下, 对其中涉及的学科通过实验比对、算法优化等方法进行学科级并行优化, 并综合考虑不同目标检测框架的优缺点, 最后通过实验结果判别系统的性能优劣. 此外, 本文还通过实拍及标注原创中文企业命名实体数据集 (Naming-649)[13], 作为本文的实验数据集.

1 系统架构

1.1 系统目标-约束方程

企业命名实体识别任务的目标-约束方程如式 (1) 所示.

$$\begin{aligned} &\max P, \min D \\ &\text{s.t.}\begin{cases} P \le 1.00 \\ 0.007 \le D \le 0.1 \\ D = d_{\text{info}} + \sum d_i \\ P = f(p_i) \\ f(p_i) \le \sum p_i \\ IOU_i = kp_i \\ IOU_i \le 1.00 \end{cases} \end{aligned} \tag{1}$$

其中, 该任务首要保证的是识别系统的准确率 P, 其次, 系统的延迟 D 应小于 0.1 s 以减少用户等待的时间. 而在不引入新模型的情况下, 系统的约束为不同模型的延时d_i、区域重合度IOU_i、识别准确率p_i, 此外还需考虑多级模型间的信息交互延时d_{info}. 而在目前可选模型 (包括目标检测模型和文字识别模型) 中, 延时最小的为 YOLOv5-S 模型, FPS (Frame Per-Second) 达到了 140, 因此系统总延迟 D 应大于 YOLOv5-S 模型的延迟 0.007 s[1]. 区域重合度及网络最高准确率不会超过准确率上限 100%, 此外, 区域重合度IOU_i与网络识别准确率p_i成正相关, 即当区域重合度过低时, 网络就无法识别出正确的文字. 并且, 若采用并行化设计, 则由于样本和模型不一定独立, 因此系统的准确率并不是不同子系统的准确率p_i之和, 而是呈某种函数关系.

对于目标/文字检测模块, 选取主流检测框架在 COCO 数据集[14] 上进行多次测试取平均值的结果如图 1 所示.

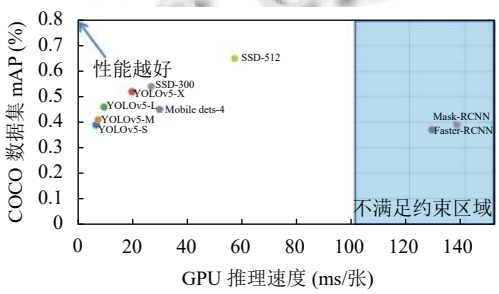

图 1 主流目标检测框架对比

图 1 中 YOLOv5 模型均小于系统最大许可延迟, 在测试图中使用目标点 (0,1) 计算各模型的欧氏距离, 从而得到计算模型的性能, 如式 (2) 所示.

$$E_i = \frac{1}{(d_i - 0)^2 + (IOU_i - 1)^2} \tag{2}$$

其中, E_i为计算模型的性能, d_i为目标点到各模型的欧式距离. 随后, 将各模型的性能进行排序, 最终本文选取了 YOLOv5-X 模型为目标检测子系统, 以进行命名

实体的检测.

对于文字识别模块, 由于各模型性能较为平均, 为了进一步提高模型的精度及降低模型延时, 最终采用了 CSSO 中并行化设计的方法, 采用多个模型, 将模型预先部署于服务器中.

由于各服务器均可以进行独立计算, 因此识别算法仅需考虑平均延时最长的子识别模块, 其余模块不受影响. 此外进行训练和测试时仅有图像大小的信息分发, 大大节省了时间, 该部分延时参见式 (3).

$$d_{\text{reco}} = d_{\text{info}} + \max(d_i),\ i \in \text{文字识别模块} \tag{3}$$

1.2 系统架构

系统主要架构如图 2 所示, 主要划分为两级. 其中第一级为文字检测模块, 主要任务是将图像中的命名实体所在的边界矩形框提取出来. 本文中, 文字检测模块使用 YOLOv5-X 架构, 并使用 Naming-649 数据集进行迁移学习.

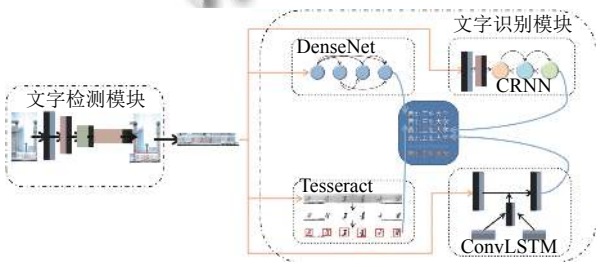

图 2 系统架构示意

随后, 一、二级衔接部分主要包括图像的裁剪、灰度化、直方图、图像二值化、霍夫变换等方法, 首先通过裁剪方法, 对命名实体所在的区域进行提取, 并对图像进行灰度化以便清除无用信息. 随后, 为了扩大图像的样本数和识别准确率, 采用了动态直方图阈值, 对于含有文字的图像而言, 企业店铺的文字一般为同一颜色, 背景为其他颜色, 因此在文字出现的位置及背景出现的位置在直方图上会呈现出两个波峰, 而文字和背景分离的关键就是取两个波峰中的波谷位置即最优阈值, 此外, 由于文字所对应的位置不确定, 因此阈值可能有偏差, 所以要对阈值信息进行反相、增加、减少等操作, 才可生成一系列经过不同阈值处理后的多张二值图像. 最后, 通过霍夫变换, 将文字进行旋转, 以便让文字识别模块可以更容易地识别.

文字识别模块主要包括几种目前主流的文字识别框架, 包括基于多分类的 DenseNet[15], 基于 CNN 特征提取、RNN 语义信息提取, 并使用 CTC 损失的 CRNN 架构, 基于模板匹配的 Tesseract, 以及可同时提取图像特征及语义特征并引入注意力机制的 ConvLSTM[16]. 其采用并行式设计, 在一二级衔接的文字处理模块完成后, 对应图片会拷贝 4 份, 分别通过 TCP/IP 协议传输给预先部署在不同服务器上的模型. 当模型计算完成后, 对应输出将通过 TCP/IP 协议传输给原服务器. 当所有服务器计算完成后, 使用竖向对齐算法, 对不同模型识别出的文字进行进一步对齐和处理, 清除文字中的标点符号和特殊字符, 最终输出识别完成的文字.

1.3 竖向对齐算法

由于文字识别模块的每个模型的独立性不好判别, 因此采用了一种竖向对齐方法来输出文字. 如图 3 所示.

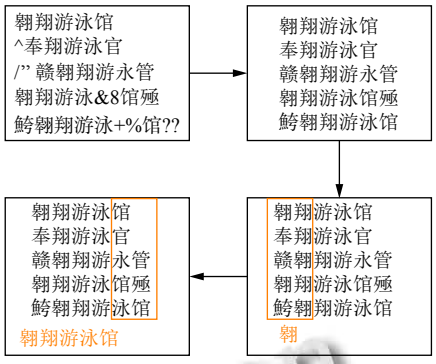

图 3 竖向对齐算法示意图

由于企业命名实体一般不包含标点符号和生僻字, 因此可对每个文字识别模型进行去标点及生僻字处理. 而由于每个模型的性能不相同, 因此对每行输出进行加权操作, 如式 (4) 所示.

$$\begin{cases} w_i = p_i a \\ a = \begin{cases} 2, & length(i) = Mo(i) \\ 1, & length(i) \neq Mo(i) \end{cases} \end{cases} \tag{4}$$

其中, 对于在先验数据集上测试准确率为 p_i 的模型 i, 输出长度为 $length(i)$ 的文字, 每一行长度的众数记为 $Mo(i)$, 为修正因子. 可以看出, 该公式保证了准确率 p_i 越高, 输出长度越接近平均值的模型输出的文本行所占的权重越大.

随后, 使用 s=2 的滑动窗口分别选中 s 列, 考虑到模型性能不齐, 因此直接计算滑动窗口中某文字的众数会产生误差, 因此需要按照式 (5) 计算每个文字的加权数:

$$word_i = \sum_{i \in line} word_i w_i \quad (5)$$

其中, $word_i$表示每行滑动窗口中第 i 个文字. 其表明, 计算每个文字的加权数只需将每行中滑动窗口中的每个文字乘上该行对应行的权重. 最后, 对滑动窗口中文字进行排序操作, 取加权数最大的文字, 作为该文字的输出, 即保证了准确率p_i越大的模型输出的文字权重也越大, 可以提高文字的识别准确率.

2 实验和结果分析

2.1 数据集

由于网络上的场景文字识别数据集中不仅包括企业命名企业实体, 还包括商家的描述等任务无关信息, 因此通过实地拍摄方式, 原创了 Naming-649 数据集, 包含 649 张训练集和 50 张测试集, 其标注格式为标准 COCO 格式, 即采用 (class, centerX, centerY, width, height) 的格式标注, 由于只有企业命名实体一个类别, 因此 class=0, (centerX, centerY) 表示边界矩形框中心点对应的归一化坐标, (width, height) 表示边界矩形框对应的归一化宽度及高度, 数据集中部分图片如图 4 所示.

图 4 Naming-649 数据集样本

2.2 训练策略

由于本系统具有多个子系统, 且训练集样本较少, 从头训练较为困难, 因此采用了迁移学习的方法, 使用 COCO 数据集上训练的 YOLOV5-X 模型在 Naming-649 上进行微调训练 10 Epoch, 使用在 Synthetic Chinese String Dataset 上训练 10 Epoch 的各文字识别模块, 使用经过裁剪、二值化等操作后的 Naming-649 数据集上进行微调. 本文使用搭载 2 块 RTX-2070Ti 的服务器进行 GPU 训练, 并使用 CUDNN 加速.

训练结束后, 将模型分别部署于搭载 1 块 RTX-1050Ti、2 块 RTX-2070Ti、1 块 RTX-1060Ti、2 块 RTX-2070Ti 共 4 台服务器上, 使用其中 1 台搭载 2 块 RTX-2070Ti 的服务器作为目标检测和数据发送/接收的主服务器, 其余 3 台服务器共 4 块 GPU 用于文字识别模块的分布式计算.

考虑到测试集样本数有限, 因此为了扩张样本数, 将测试集进行反色、裁剪等处理可得到约 400 张测试集图片.

2.3 系统性能对比分析

对系统每个模块和其他同类系统使用 GitHub 上的已训练模型, 随后通过 Synthetic Chinese String Dataset 中文识别数据集获取中文标签和特征, 最后在 Naming-649 数据集上采用学习率为10^{-4}以进行微调, 所用的训练批次 batchsize=16, 统一训练 10 轮. 在测试集上进行测试的结果如表 1 所示.

表 1 不同模型性能对比

模型结构	准确率 (%)	总时间 (ms)
FOTS[17]	54.8	58.46
EAST+CRNN	49.7	100.53
TextBoxes++[18]+CRNN	57.4	98.56
CTPN+CRNN	59.1	101.8
YOLOv5-X+并行识别(本系统)	64.3	73.4

从表 1 中可知, 相比端对端的 FOTS 模型, 本系统所用时间相对较长, 但是准确率提升了约 9%, 而对比其他传统两步方法, 即文字检测+文字识别方法, 所用时间较短, 识别准确率也是最高的. 这是由于 YOLO 等目标检测框架运行速度较快, 而对于本系统而言, 可将 YOLOv5-X 替换为其他的 YOLOv5 系列模型, 其性能指标变化如表 2 所示.

从表 2 中可见, 相比于 FOTS 的权重大小 (417 MB), YOLOv5 的权重相对较小, 推断速度较快, 因此可为识别模块的并行计算提供时间上的预留, 同时也为满足系统的约束提供了支持. 此外, 不同 YOLOv5 模型之间的准确率变化波动较小, 性能相近, 因此可根据实际系统的需求选择不同 YOLOv5 模型作为文字的检测模块. 对于文字识别模块, 系统采用了并行识别的方式, 而对于其中的单个文字识别的方法, 对系统性能的影

响参见表 3.

可以发现, 4 种模型所使得系统达到的准确率各不相同, 但由于独立性, 最终使用的 4 种模型并行计算所能达到的准确率较高, 这也表明实际训练中可以采用多个低级分类器聚合为一个高级分类器的随机森林思想.

表 2 YOLO 系列文字检测模块对比

模型结构	准确率 (%)	文字检测时间 (ms)	检测模型大小 (MB)
YOLOv5-S+并行识别	49.8	44.2	14.5
YOLOv5-M+并行识别	53.7	51.7	41.9
YOLOv5-L+并行识别	56.8	63.3	91.6
YOLOv5-X+并行识别	64.3	73.4	170.1

表 3 文字识别模块对比

模型结构	准确率 (%)
YOLOv5-X+并行识别	64.3
YOLOv5-X+DenseNet	53.3
YOLOv5-X+Tesseract	48.7
YOLOv5-X+CRNN	57.8
YOLOv5-X+ConvLSTM	56.2

2.4 系统优化分析

经实验可知, 本系统的自身具体性能和约束分析如表 4 所示.

表 4 系统优化指标列表

指标名称	指标数值
准确率P	64.3%
总延时D	73.4 ms
平均IOU	78.3%
平均p_i	54.0 ms
平均d_{info}	15.3 ms
平均d_{reco}	35.3 ms

从表 4 中可知, 对于 2.1 节提出的系统约束, 系统均满足, 并且留有部分余量. 可行性方面, 由于分布式/并行计算上早有如 SETI@home[19] 等项目的先例, 因此无需购买多台服务器, 只需构建网站, 并且运营网站邀请用户参与项目, 因此成本上消耗实际较低, 成本构成包括数据传输成本、内存/显存消耗成本、运营成本等. 扩展性方面, 由于系统所用相关库如 PyTorch、Tesseract 框架等均有 C/C++版本, YOLOv5 权重大小较低, 因此可部署于嵌入式设备如 51 单片机、智能手机中, 扩展性较强.

3 结论

本文针对企业命名实体识别的任务, 结合并行子空间优化的思想, 构建了两步模型, 通过建立系统目标和约束进行模型的选取, 最终选择 YOLOv5-X 和并行识别计算的方法, 获得了准确率为 64.3%, 总延时为 73.4 ms 的优化模型, 满足系统目标及约束的基本要求, 可在实际检测与识别中使用.

参考文献

1 Ultralytics. YOLOv5 in PyTorch. https://github.com/ultralytics/yolov5. [2021-03-07].

2 Tan MX, Le QV. EfficientNet: Rethinking model scaling for convolutional neural networks. Proceedings of the 36th International Conference on Machine Learning. Long Beach: PMLR, 2019. 6105–6114.

3 Ren SQ, He KM, Girshick R, *et al*. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137–1149. [doi: 10.1109/TPAMI.2016.2577031]

4 He KM, Gkioxari G, Dollár P, *et al*. Mask R-CNN. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 386–397. [doi: 10.1109/TPAMI.2018.2844175]

5 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv: 1409.1556, 2014.

6 He KM, Zhang XY, Ren SQ, *et al*. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas: IEEE, 2016. 770–778.

7 Zhou XY, Yao C, Wen H, *et al*. EAST: An efficient and accurate scene text detector. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu: IEEE, 2017. 2642–2651.

8 Yan H, Maltz DA, Ng TSE, *et al*. Tesseract: A 4D network control plane. 4th Symposium on Networked Systems Design & Implementation. Cambridge: USENIX Association, 2007.

9 Brzeski A, Grinholc K, Nowodworski K, *et al*. Evaluating performance and accuracy improvements for attention-OCR. 18th International Conference on Computer Information Systems and Industrial Management. Belgrade: Springer, 2019. 3–11.

10 Shi BG, Bai X, Yao C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Transactions on

Pattern Analysis and Machine Intelligence, 2017, 39(11): 2298–2304. [doi: 10.1109/TPAMI.2016.2646371]

11 Paul D, Saha S, Mathew J. Improved subspace clustering algorithm using multi-objective framework and subspace optimization. Expert Systems with Applications, 2020, 158: 113487. [doi: 10.1016/j.eswa.2020.113487]

12 Kim KJ, Yu KH. Multidisciplinary design optimization for a solar-powered exploration rover considering the restricted power requirement. Energies, 2020, 13(24): 6652. [doi: 10.3390/en13246652]

13 QiaoSZ.Naming-649Dataset.https://github.com/ShizhanQiao/Naming-649Dataset. [2021-03-07].

14 Lin TY, Maire M, Belongie S, *et al*. Microsoft COCO: Common objects in context. 13th European Conference on Computer Vision. Zurich: Springer, 2014. 740–755.

15 Huang G, Liu Z, Van Der Maaten L, *et al*. Densely connected convolutional networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu: IEEE, 2017. 2261–2269.

16 Shi XJ, Chen ZR, Wang H, *et al*. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Proceedings of the 28th International Conference on Neural Information Processing Systems. Montreal: ACM, 2015. 802–810.

17 Li H, Wang P, Shen CH. Towards end-to-end text spotting with convolutional recurrent neural networks. 2017 IEEE International Conference on Computer Vision (ICCV). Venice: IEEE, 2017. 5248–5256.

18 Sheng FF, Chen ZN, Mei T, *et al*. A single-shot oriented scene text detector with learnable anchors. 2019 IEEE International Conference on Multimedia and Expo (ICME). Shanghai: IEEE, 2019. 1516–1521.

19 Anderson DP, Cobb J, Korpela E, *et al*. SETI@home: An experiment in public-resource computing. Communications of the ACM, 2002, 45(11): 56–61. [doi: 10.1145/581571.581573]