

文章编号: 2095-2163(2020)12-0024-05

中图分类号: TP3-05

文献标志码: A

# 基于 Faster-RCNN 网络的表格检测算法研究

马志远<sup>1</sup>, 余粟<sup>2</sup>

(1 上海工程技术大学 机械与汽车工程学院, 上海 201620; 2 上海工程技术大学 工程实训中心, 上海 201620)

**摘要:** 为了快速、准确地识别出文档图片中存在的表格, 为表格信息提取提供表格图像数据, 为表格内容的语义分割打下基础。本文首先使用 OpenCV 图像处理工具对包含表格的文档图片进行预处理, 再采用 Labelme 标注工具对图片中的表格位置进行标注; 其次, 把图片数据集按照 4:1 比例分为表格识别模型训练数据集和模型验证数据集; 最后, 借助 Tensorflow 深度学习工具, 采用 Faster-RCNN 目标检测框架对表格识别模型进行训练, 并用验证数据集对训练好的模型进行验证实验。实验结果表明, 基于 Faster-RCNN 算法的表格检测模型系统平均每张图片的处理时间为 1.31s, 识别准确率达到 92.4%。说明 Faster-RCNN 目标检测算法能准确且快速地检测出文档图像中存在的表格。

**关键词:** 图像处理; 表格检测; 深度学习; Faster-RCNN; Tensorflow

## Table detection algorithm based on Faster-RCNN

MA Zhiyuan<sup>1</sup>, YU Su<sup>2</sup>

(1 School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China;

2 Engineering Training Center, Shanghai University of Engineering Science, Shanghai 201620, China)

**[Abstract]** In order to identify the table in the document image more quickly and accurately, make a foundation for the table information extraction. This paper using the OpenCV image processing tools to preprocess the document images containing tables, then uses the Labelme labeling tool to mark the position of the table in the picture, and the image data set is divided into training data and verification data set according to the ratio of 4:1, finally, the Tensorflow deep learning tool is used to train the table recognition model and the target detection framework of Faster-RCNN is adopted. The trained model was validated with the validation data set. The experimental results show that the processing time of the table detection algorithm system based on Faster-RCNN is 1.31s, and the recognition accuracy reaches 92.4%. It means that Faster-RCNN can accurately and quickly detect the existing table in the document.

**[Key words]** Image processing; Table detection; Deep learning; Faster-RCNN; TensorFlow

## 0 引言

表格因其信息丰富、简明扼要的特点, 被大量地运用到了现代的商业活动中, 许多人每天的工作就是不断地处理大量的文档和表格<sup>[1]</sup>。由于表格的应用范围广泛, 且拥有信息量大的特点, 在处理表格的过程中就要耗费掉大量的时间、精力。近些年来, 光学字符识别 OCR ((Optical Character Recognition) 技术取得了蓬勃的发展, 人们开始尝试利用一些先进的图像识别算法在表格文档中提取表格图像, 并通过语义分割等技术来提取表格中的数据信息, 并加以整合。这种方法在一定程度上减少了在表格信息获取工作中花费的时间和精力<sup>[2]</sup>。OCR 引擎运用中, 首先要获得完整的表格图像, 进而进行语义分割等后续操作。但目前的表格检测算法并不十分成熟, 因此用准确、快速的表格识别算法来检测文档图

像中存在的表格就显得尤为重要。

近年来, 通用的表格处理系统的研究取得了一些的进展, 为了实现表格数据的提取, 研究出很多表格识别算法, 例如: 最小二乘法拟合表格线算法<sup>[3]</sup>, 基于直线交点交角的区域定位算法<sup>[4]</sup>, 形态学操作提取表格框线并用 Hough 变换检测单元格内斜线的算法<sup>[5]</sup>。

最小二乘法表格线拟合算法: 由于构成表格的框线大部分都是水平直线和垂直直线, 该方法首先采用数学形态学提取水平和垂直的表格框线, 通过选取适当的结构元素, 过滤掉由于文字和噪声等因素产生的干扰信号, 对提取出的表格框线细化处理, 最后采用最小二乘法拟合表格框线中断掉的部分, 使表格线连贯。但这种方法没有考虑针对表格中斜线等情况的具体解决方案, 对倾斜表格的适用性较

**作者简介:** 马志远(1995-), 男, 硕士研究生, 主要研究方向: 机器视觉; 余粟(1962-), 女, 硕士, 教授, 硕士生导师, 主要研究方向: 计算机科

学。

**通讯作者:** 余粟 Email: suyush@ hotmail.com

**收稿日期:** 2020-09-04

差。

基于直线交点交角的区域定位算法: 由于表格中存在大量互相垂直的直线, 互相垂直的两条直线存在一个交点, 通过表格中表格线的交点位置, 定位出单元格的位置, 再根据每个单元格的位置对表格进行切分, 最后在切分的表格中提取信息。因为该方法依赖框线交点定位, 所以在单纯水平线组成的表格即无交点表格中无法应用。

形态学操作提取表格框线并用霍夫变换检测单元格内斜线的算法: 表格线的提取采用了基于形态学膨胀、腐蚀的算法, 提取出矩形表格框线, 并定位出表格框区域位置。采用霍夫变换检测单元格内的斜线, 根据提取出的表格框线位置以及表格斜线进行表格重建。但是当图像中表格存在干扰时候, 以及表格线比较模糊的时候, 不能很好的确定交点的位置, 甚至会丢失原有交点位置, 导致表格提取效果不理想。

近些年来, 由于深度学习技术尤其是卷积神经网络的发展, 基于深度学习的目标检测算法也取得了很大的进步, 得益于高性能 GPU (Graphics Processing Unit | 图形处理单元) 的强大计算力基础, 深度卷积神经网络的特征提取能力大大增强<sup>[6]</sup>。表格图像作为一种特殊的数据形式, 有着极为典型的特征, 为基于深度神经网络的表格检测任务提供了可能<sup>[7]</sup>。因此, 本文提出一种基于 Faster-RCNN 目标检测框架的表格检测算法, 充分利用表格的典型特征, 以及神经网络高准确性、鲁棒性强的特点, 以完成对不同类型文档图像中的表格的检测识别。

## 1 数据预处理

图像数据的预处理是表格识别任务的第一步, 采集到的表格图像, 如图 1 所示。经过预处理后的图像数据能更好的进行特征的提取, 图像预处理的好坏也很大程度上影响到识别的结果, 表格图像识别的预处理过程, 主要经过以下几个步骤: (1) 图像灰度化; (2) 图像二值化; (3) 距离变换。

### 1.1 图像灰度化

由于采集到的包含表格的图像数据的每个像素点都是由红绿蓝 (RGB) 三种色彩按照不同的比例混合构成的, 为了方便研究, 首先把三通道的 RGB 图像转换成单通道的灰度图像, 灰度图像使用亮度作为度量依据, 亮度指标是由人眼对 RGB 图像的三个色彩的不同敏感程度叠加得到的<sup>[8]</sup>。采用 RGB 三通道图像中 30% 的红色 (R)、59% 的绿色 (G)、

11% 的蓝色 (B) 为基础, 将其叠加后可以得到亮度值  $K$ , 计算表达式 (1), 也就代表了灰度化后的图像的每个像素值。

$$K = 0.3 \times R + 0.59 \times G + 0.11 \times B. \quad (1)$$



图 1 包含表格的图像

Fig. 1 Image contains table

### 1.2 图像二值化

由于表格文档经过扫描仪产生的图片存在着墨迹不清晰、深浅以及清晰度不一致等问题, 需要把灰度化的表格图像二值化处理, 把像素值分布不均的图像转变为只有黑色 (像素值为 0) 与白色 (像素值为 255) 的二值图像<sup>[9]</sup>。借助开源计算机视觉库 OpenCV (Open Source Computer Vision Library) 中提供的二值化工具, 采用自适应阈值的方法, 将灰度图片分为 15 个 block 块, 分别计算 15 个 block 中灰度值的平均值, 在每个 block 中采用正向二值化的方法设置其二值化的后的像素值。其中正向二值化是指如果当前的像素值  $src$  大于设置的阈值 ( $thresh$ ), 则将该点二值化后的像素值  $dst$  设置为  $maxval(255)$ ; 否则, 将该点的像素值设置为 0; 正向二值化表达式 (2):

$$dst = \begin{cases} maxval & \text{if } src > thresh \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

### 1.3 距离变换

图像的距离变换广泛应用在图像分析、计算机视觉、模式识别领域, 常用它来实现目标的骨架提取、前后景区分<sup>[10]</sup>。在表格检测之前, 需要对包含表格的图像进行前景背景的区分, 用距离变换方法输入的是一幅二值图像, 可以认为仅仅包含目标和背景二种像素, 距离变换的结果不是一幅二值图像, 而是一幅灰度图像, 也称之为距离图像。距离图像中每个像素的灰度值  $Distance$  为该像素在其 block 区域内与其最近的背景像素间的距离。距离即灰度值的计算, 选用 OpenCV 中的  $DIST\_L1$  类型, 计算方法为式 (3):

$$Distance = |x_1 - x_2| + |y_1 - y_2| \quad (3)$$

其中,  $x_1$  代表该像素  $x$  轴的坐标值;  $y_1$  代表该像素  $y$  的坐标值; 而  $x_2, y_2$  代表与其最近的背景像素的坐标值。block 设置为 5, 代表  $5 \times 5$  大小的区块。经过距离变换输出的图像如图 2 所示。



图 2 距离变换后的图像

Fig. 2 Image after distance transform

## 2 标注

运用 python3.5 中的 Labelme 标注工具标注包含表格的文档图片, 用矩形的标注框标注图片中表格的位置。每张图片经过标注后能得到包含其表格位置信息 (两个点: 矩形框的左上点和右下点) 以及相关文件, 将 train 数据集中所有图片所对应的文件合并, 即可得到 train 数据集中所有的标注信息, 以 csv 的文件格式保存这些信息。图 3 为 Labelme 图像标注工具使用截图, 其中红色区域即为标注的表格位置。



图 3 Labelme 标注工具截图

Fig. 3 Screenshot of Labelme annotation tool

## 3 Faster-RCNN 表格检测

### 3.1 Faster-RCNN 目标检测算法流程

Faster-RCNN 是目标检测领域一个经典的算法, 以其精度高、速度快的特点著称。Faster-RCNN 在本文中的运用就是为了在含有表格图片 (经过距离变换后) 的图像中准确地定位出表格的位置。Faster-RCNN 最大的特点就是集成 RPN 网络 (region proposal network | 区域建议网络) 和 Fast-RCNN 网络, 通过 RPN 网络生成目标候选区域, 把这些区域建议输入 Fast-RCNN, 再进行目标类别的区分以及边框的回归, 最后得出图像中目标的种类信息以及位置信息<sup>[11]</sup>。Faster-RCNN 目标检测网络的流程如图 4 所示。

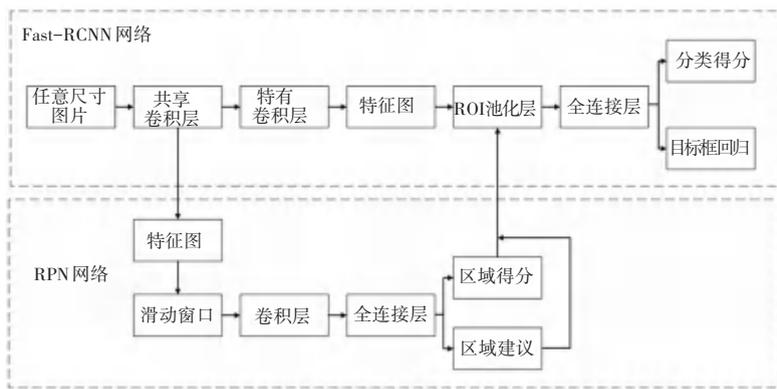


图 4 Faster-RCNN 流程

Fig. 4 Faster-RCNN flow

### 3.2 RPN 网络

RPN 网络的主要目的是为 Faster-RCNN 网络提供区域建议, 就像人类的眼睛一样, 能捕获和感知可能存在物体的空间位置<sup>[12]</sup>。RPN 网络的输入为特征图, 这些特征图由任意大小图片经过特征提取网络, 如: VGG-Net, ZF-Net, ResNet 提取, 特征图经过 RPN 网络会输出一些局部区域的位置, 这些区域

很大程度地包含了所要识别的物体。输入特征图上  $n \times n$  大小的区域, 称为滑动窗口, 滑动窗口是 RPN 网络的计算单位。本文仅对“table”类进行预测, 称滑动窗口上与“table”类目标在原始图片上候选区的关联区域为“锚点”, 锚点所对应的区域根据图像尺度和长宽比确定为 9 种大小规模。也就是说每种目标类型在锚点上都对 9 个区域位置, 特征图的

每个点都是“锚点”。输入的每个滑动窗口经过多次卷积转变为一个向量,这个向量(锚点数据信息)会被分别用来进行前景、背景分类任务和边框回归任务,前景背景分类任务会得到该锚点属于前景的概率值也就是区域得分,而边框回归任务会得到该锚点属于前景区域的位置坐标也就是区域位置建议。

### 3.3 Fast-RCNN 网络

Fast-RCNN 在得到 PRN 网络提供的区域位置建议和区域前景得分概率时,也得到了由特有卷积层提供的特征图,该特征图是由特征提取网络经过特有池化层进一步提取特征生成的。Fast-RCNN 通过 ROI 池化层对其得到的建议筛选,并将筛选后的区域位置映射到得到的特征图上,有针对性地对这些特征图上的区域提取,使用全连接层对候选区域进行类别预测任务,得到目标类别以及对边框位置执行回归任务得到位置坐标。

### 3.4 Faster-RCNN 网络损失函数

RPN 网络和 Fast-RCNN 网络使用的都是随机梯度下降和反向传播方法优化网络中的参数,而这两个网络的损失函数也都由矩形框的回归损失和目标类别的分类损失组成。对于 RPN 网络的损失函数,式(4):

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i + p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*). \quad (4)$$

其中,分类损失  $L_{cls}$  为前一部分; $i$  表示第  $i$  个锚点; $p_i$  表示锚点被预测为目标概率; $p_i^*$  是标签。 $p_i^* = 0$  代表负样本; $p_i^* = 1$  代表正样本; $N_{cls}$  设置为训练的  $batch\_size = 256$ ;  $batch\_size$  表示一次训练所选取的样本数。分类损失函数  $L_{cls}$  是一个二分类交叉熵函数为式(5):

$$L_{cls}(p_i, p_i^*) = -\log[p_i p_i^* + (1 - p_i)(1 - p_i^*)]. \quad (5)$$

回归损失  $L_{reg}$  为第二部分,  $\lambda$  取 10,  $N_{reg}$  特征图的大小为 2400,  $t_i$  表示 RPN 预测的位置偏移量,  $t_i^*$  为实际位置偏移量,回归损失函数  $L_{reg}$  为式(6):

$$L_{reg}(t_i, t_i^*) = R(t_i - t_i^*). \quad (6)$$

其中,  $R$  函数为式(7):

$$smooth_{L1}(x) = \begin{cases} 0.5 x^2, & |x| < 1 \\ |x| - 0.5, & otherwise \end{cases} \quad (7)$$

Fast-RCNN 的联合损失函数为式(8):

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda L_{loc}(t^u, v) \quad (8)$$

同样,前部分  $L_{cls}$  表示分类损失,后一部分  $L_{loc}$  表示回归损失。

其中,  $p$  为 Fast-RCNN 输出层对目标种类预测的概率值,而  $u$  为真实类别的标签。因为本文要识别的目标只有一个“table”类,所以分类损失和 RPN 网络的损失一样为二分类的交叉熵损失,分类任务是为了区分目标是否为“table”类。 $t^u$  表示网络预测目标框位置(包含目标框中心点的坐标以及目标框的长度和宽度),  $v$  代表实际标签中的目标框位置信息。 $L_{loc}$  函数和 RPN 网络选择一样的  $R$  函数,  $\lambda$  取 10。

## 4 实验结果与分析

将预处理过的包含表格的 406 幅图片按照 5 : 1 的比例分为二组,分别是 train 组 338 张图片和 val 组 68 张图片,其中 train 组用来对表格识别模型进行训练,并设置训练轮次为 10 次,即将 train 组的 338 张图片循环训练 10 次。模型训练平台为: Windows 10 系统 AMD Ryzen 5 2600@ 3.4GHz CPU, 1660TI@ GPU, 16G 内存计算机,选用 Tensorflow 1.14.0 搭建 Faster-RCNN 表格检测模型。

训练完成后,在 Tensorboard(Tensorflow 中模型训练损失查看工具)中对 Faster-RCNN 表格检测模型中的 4 个损失截图,如图 5 所示。横轴代表输入图片的数量(单位为:张),纵轴代表 loss 损失值。

综合上述训练损失图像可知,随着训练图片摄入量增大,4 个损失都呈下降趋势,且在训练结束时模型的各个损失趋近于稳定。保存训练模型,对 val 组数据输入模型检测得到 92.4% 的正确率,且平均处理时间为 1.31s,说明 Faster-RCNN 表格检测模型有准确率高、响应速度快的特点。

## 5 结束语

本文通过对表格图像的预处理、标注,以及对 Faster-RCNN 模型针对性地训练、验证,最终得到了一个能快速高效且准确检测出图像中表格位置的表格检测模型。该模型可以用来检测日常见到的多种表格,有着广泛的适用性。所提取出的表格图像不包含多余的信息,为表格信息的语义分割奠定了数据基础,可以提供优质的表格图像数据。但是本文并没有对干扰性强的表格图像的提取做具体的实验,该模型的抗干扰能力有待验证。表格信息的分割、提取以及提升表格检测模型的抗干扰能力等问题仍然需要进一步的研究。

(下转第 31 页)