异类文档中的表检测

会议文件·2010年1月

DOI：10.1145/1815330.1815361 · 来源：DBLP

2位作者，其中包括：

穆萨尔·沙菲特

西澳大学

178出版物3,820个引用

查看个人资料

异类文档中的表格检测

Faisal Shafait *
德国人工智能研究中心（DFKI GmbH），德国凯泽斯劳滕faisal.shafait@dfki.de

雷·史密斯Google Inc.Mountain View，CA，美国，theraysmith @ gmail.com

摘要在文档图像中检测表格很重要，因为表格不仅包含重要信息，而且大多数布局分析方法都在文档图像中存在表格时失败。现有的表检测方法主要集中在检测文本的单列中的表，并且在具有可变布局的文档上不能可靠地工作。本文提出了一种适用于表格检测的实用算法，该算法在具有不同布局的文档（公司报告，报纸文章，杂志页面等）上具有很高的准确性。 Tesseract OCR引擎的一部分提供了算法的开放源代码实现。对来自公开可用的UNLV数据集的文档图像进行算法评估表明，与商用OCR系统的表格检测模块相比，该算法具有竞争优势。

类别和主题描述符I.7.5
[文档和文本处理]：文档捕获—文档分析

关键字页面分割，表检测，文档分析

1.简介自动将纸质文档转换为可编辑的电子表示形式，取决于光学字符识别（OCR）技术。典型的OCR系统包括三个主要步骤。首先，执行布局分析以在文档图像中定位文本行并识别其阅读顺序。然后，字符识别引擎处理文本行图像，并通过识别文本行图像中的各个字符来生成文本字符串。最后，语言建模模块使用字典或语言模型对文本字符串进行更正。

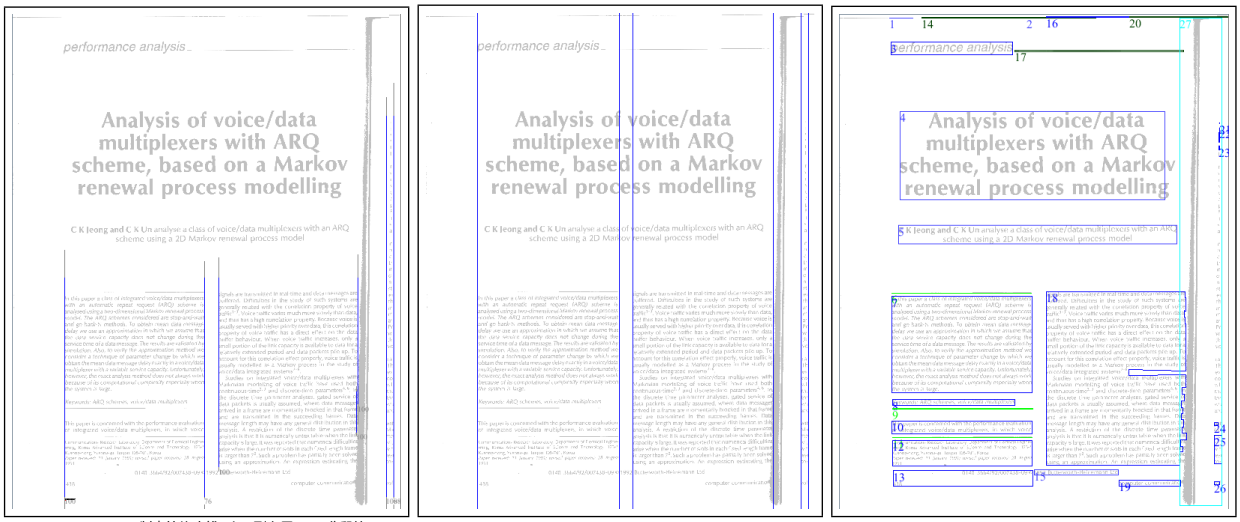
□作者非常感谢GoogleInc的资助。为了支持这项工作

由于布局分析是该过程的第一步，因此所有后续阶段都依赖于布局分析才能正常工作。布局分析面临的主要挑战之一是检测表区域。表格检测是一个难题，因为表格的布局有很大的差异。现有的开放源代码OCR系统缺乏表检测功能，其布局分析模块由于存在表区域而崩溃。在此阶段，应该在表检测和表识别之间进行区分[8]。表检测处理在页面图像中查找表的边界问题。另一方面，表识别则专注于通过发现行和列来分析检测到的表，并尝试提取表的结构。我们在本文中的重点是表检测问题。

Kieninger等人完成了表格检测和识别方面的开创性工作之一。 [11、10、12]。他们开发了一个名为T-Recs的表格识别和结构提取系统。系统依赖于单词边界框作为输入。这些单词框通过自下而上的方法通过构建“分段图”而聚类到区域中。如果这些区域满足特定标准，则将它们指定为候选表区域。该方法的主要局限性在于，仅基于单词框，无法非常准确地处理多列布局。因此，它仅适用于单列页面。

Wang等。 [20]采用统计学习的方法来解决表格检测问题。给定一组候选文本行，可以根据连续单词之间的间隙来识别候选表行。然后，将具有大间隙的垂直相邻的行和水平相邻的单词组合在一起，以构成表实体候选。最后，基于统计的学习算法用于优化候选表并减少误报。他们假设最大列数为2，并设计了三个页面布局模板（单列，双列，混合列）。他们应用列样式分类算法来找出页面的列布局，并将此信息用作发现表格区域的先验知识。这种方法只能处理经过训练的那些布局。此外，训练算法需要大量的标记数据。

Hu等。 [6]提出了一种用于从扫描的页面图像或纯文本文档中进行表格检测的系统。他们的系统假定一个单列输入页面可以很容易



(a) 制表符停止线 (b) 列布局 (c) 分段块

图1：在文档图像上，Tesseract的布局分析模块的不同步骤的输出。

ily分割成单独的文本行（例如，通过水平投影）。然后将表检测问题作为一个优化问题，其中通过优化某种质量函数来识别属于一个表的开始和结束文本行。像以前的方法一样，该技术不能应用于多列文档。

在[7]中，Hu等人。通过使用地面真实区域信息（确定每个地面真实区域是否为表），在UW-III数据集上评估了他们的表检测算法[5]。这种评估是不切实际的，因为将一个表分割为一个区域实际上是一个表检测系统的硬部分。这更多地涉及文档区域分类的方向[21，9]，目标是将每个分割的文档区域分配给一组预定义的类（文本，数学，表格，半音等）。

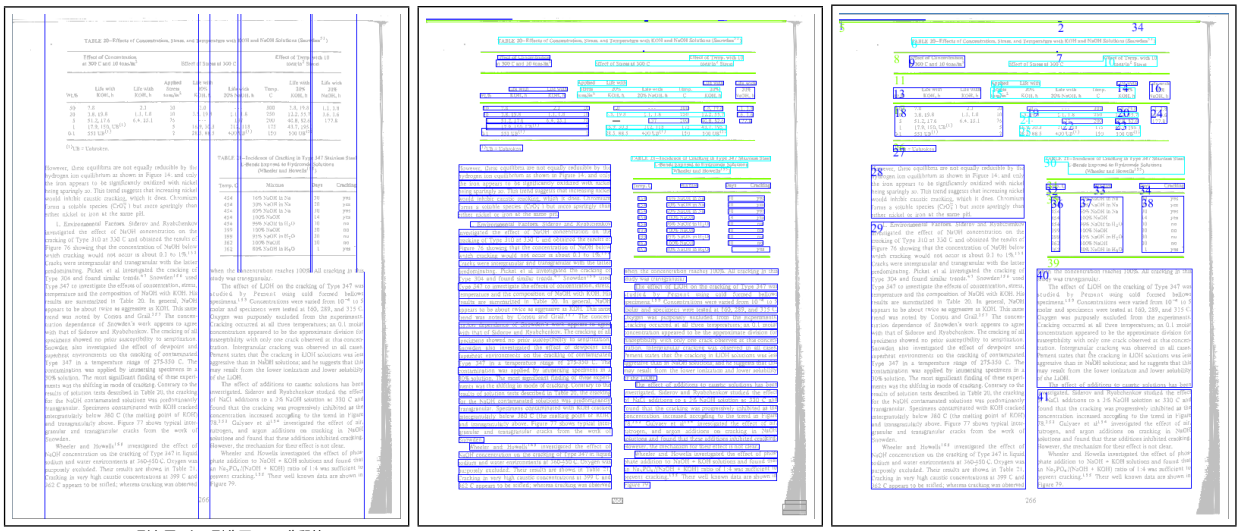
Cesarini等。 [2]提出了一种通过检测平行线来定位桌子区域的系统。然后通过在平行线之间的区域中放置垂直线或空白来验证以此方式形成的表假设。但是，仅依靠水平或垂直线进行表检测会限制系统范围，因为并非所有表都具有此类线。
Gatos等人报道了表检测方面的最新工作。 [4]和Costa e Silva [3]。 Gatos等。 [4]着重于查找同时具有水平和垂直标尺并找到其交点的桌子。然后，通过绘制连接所有线交叉点对的相应水平线和垂直线来实现表重构。该系统可以很好地用于其目标文档，但是当表格的行/列没有用分隔线隔开时，则无法使用。Costa e Silva [3]的工作着重于使用隐马尔可夫模型（HMM）从PDF文档中提取表格区域。他们使用pdftotext Linux实用程序从PDF中提取文本。提取的文本中的空格用于计算特征向量。显然，此方法不适用于文档图像。

总结表检测的最新技术，我们可以看到现有方法的明显局限性。方法在多列文档图像上不能很好地工作。这可能是由于以下事实：大多数现有方法都专注于表识别以提取表的结构（行，列，单元格），因此在表检测部分进行了一些简化假设。当必须处理布局简单的某些特定类别的文档图像时，此方法效果很好。但是，在处理文档的异构集合时，需要更强大的表检测算法。在本文中，我们试图弥合这一差距。我们的目标是在复杂的异类文档（公司报告，期刊文章，报纸，杂志等）中准确发现可疑区域。一旦发现了餐桌区域，可以使用一种现有的餐桌识别技术（例如[10]）来提取餐桌的结构。

本文的其余部分安排如下。首先，我们在第2节中描述了Tesseract [18，19]的布局分析模块，该模块将用作表检测算法的基础。然后，在第3节中说明了我们的表检测算法。在第4节中介绍了用于评估我们的系统的各种性能指标。实验结果在第5节中进行了讨论，然后在第6节中进行了总结。

2.通过制表符检测进行布局分析Tesseract的布局分析是开源OCR系统的新增功能[19]。它基于检测文档图像中的制表位的想法。在设置文档类型时，制表位是文本对齐的位置（左，右，居中，十进制，...）。因此，制表位可以用作文本块开始或结束位置的可靠指示。通过制表符停止检测来查找页面的布局，操作如下（请参见图1）：

•首先，执行文档图像预处理步骤，以识别水平和垂直标线或分隔符并确定半色调或图像区域



(a) 列布局 (b) 列分区 (c) 分段块

图2：在存在表格区域的情况下，Tesseract的布局分析不同步骤的结果。请注意，两个表中的列布局不一致。同样，列分区有时也会合并不同表列中的文本，有时会使它们分开。这导致在表区域中页面图像的严重过度分割。

在文档中。然后，执行连接成分分析以基于候选文本成分的大小和笔划宽度来识别它们。

•过滤后的文本组件被评估为处于制表符停止位置的候选对象。这些候选者被分组为垂直线，以找到垂直对齐的制表位。作为最后一步，调整成对的连接的制表符线，使它们在相同的y坐标处结束（请参见图1（a））。在此阶段，垂直制表符线标记文本区域的开始和结束。

•根据选项卡行，推断页面的列布局，并将连接的组件分组到“列分区”中。列分区是一系列连接的组件的序列，这些组件不跨越任何制表行，并且具有相同的类型（文本，图像等）。 Textcolumn分区可以视为文本行的初始候选对象（请参见图1（b））。

•最后一步创建列分区流，以便将相同类型的相邻列分区分组到同一块中（图1（c））。具有不同字体大小和行距的文本列分区被分组为不同的块。然后，确定这些块的读取顺序。块的边界表示为等角多边形（具有与轴平行的所有边的多边形）。

3.表格检测我们的表格检测算法是基于布局分析模块的两个组件构建的：

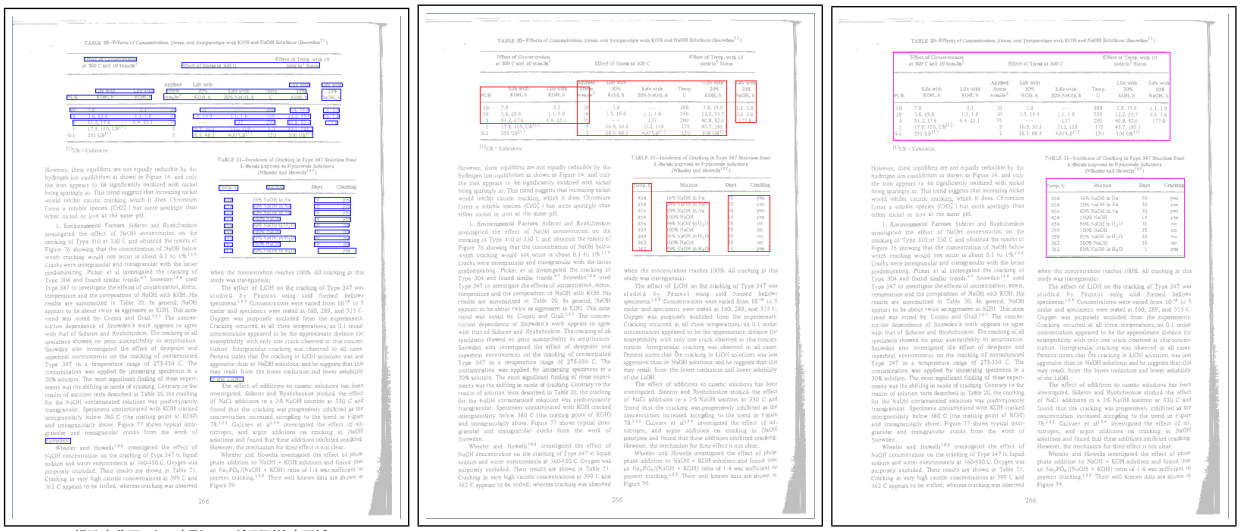
1。列分区

2.列布局

列分区使我们可以将连接的组件按其类型分组为不跨越制表符行的分区。因此，文本列分区近似于文档中的文本行。半色调区域和黑色水平线（标线）报告为“图像”和“水平线”类型的列分区。除了列分区之外，columnlayout还为我们提供了特定列分区是完全位于一列内还是跨越多列的信息。如图2所示，在存在表区域的情况下，列分区和列布局都可能给出错误的结果。

在存在表格区域的情况下对布局分析结果的进一步分析显示了两种主要情况。在第一种情况下，表列被报告为页面列，从而破坏了页面的列结构。当表格单元格非常对齐时，这种情况尤其会发生。对齐导致要检测大量的制表位，因此制表行足够强壮以报告列的存在。因此，表中的每个单元格都报告为一个单列分区。在第二种情况下，由于单元格排列不正确，系统将忽略表列。因此，页面的列结构已正确识别。在这种情况下，列分区跨越表的不同列。这两种情况都在图2的示例图像中进行了说明。基于此分析，我们的表检测算法设计如下。

3.1识别表分区我们算法的第一步是识别可能属于表区域的文本列分区，称为表分区。根据上一段落中提到的观察，将三种类型的分区标记为不稳定分区：（1）在其连接的组件之间至少有一个大间隙的分区；（2）仅由一个单词组成的分区（两个分区之间没有明显的间隙） com-



（a）候选表分区（b）表列（c）检测到的表区域

图3：我们的表格检测算法在样本图像上不同步骤的结果。

（3）沿y轴与同一列内的其他分区重叠的分区。第一种情况识别表分区，该表分区是由于将表的不同列中的单元格合并到一个分区中而产生的。第二种情况是检测由单个数据单元组成的表分区。第三种情况标识位于一个列中但由于存在强大的制表行而未连接在一起的表分区。

此阶段尝试非常积极地查找候选表分区。这样做的好处是，即使是表存在的少量证据也不会丢失，因为在此阶段丢失的任何表在以后的阶段都将无法恢复。侵略性方法的缺点是可能会产生多个错误警报，例如，来自单个单词部分的标题，页面页眉和页脚，编号的等式，边缘噪声中的一小部分文本单词以及线条绘制区域。应用平滑过滤器来检测隔离的表分区，该分区在其上方或下方没有其他表分区邻居。这些分区将从候选表分区列表中删除。图3（a）显示了我们示例图像的候选表分区。

3.2检测页面列拆分下一步是检测由于存在表而导致页面列布局中的拆分。当表格的单元格非常对齐时，就会发生这种分裂。为了检测这种情况，我们将页面分为几列，然后在每一列中找到表分区的比率。错误地报告为页面列的表列很容易被检测到，因为与普通文本分区相比，它们具有较高的表分区比率。但是，由于错误的决定会导致合并两个文本列，导致分页布局分析本身出现大量错误，因此在此阶段需要特别注意取消列拆分（即合并两个列）。

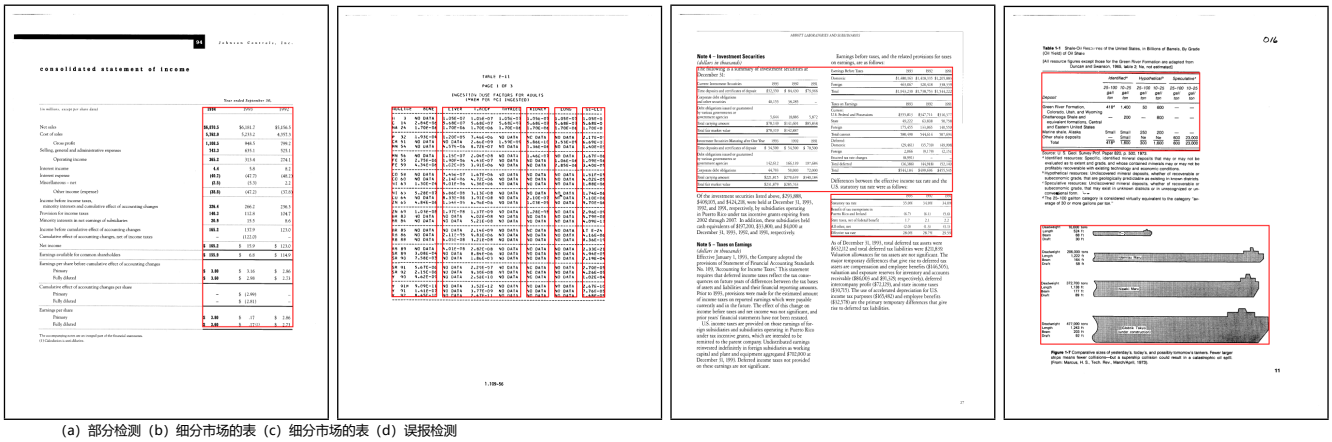
因此，只有在足够的情况下，我们才会撤消页面列拆分

存在跨越两列的文本分区的数量，并且列中的拆分以表分区开始。当页面中没有流文本时，这种额外的注意可防止合并表列与全页表。由于在布局分析错误方面，错误决定的代价非常高，因此我们选择防御性地执行此步骤。

3.3查找表列此步骤的目标是将表分区分组为表列。为此，将垂直相邻的表分区的运行分配给单个表列。如果遇到“水平裁定”类型的列分区，则运行继续。找到任何其他类型的分区后，最终确定的表列将最终确定。如果一个表列仅包含一个表分区，则将其作为错误警报删除。在图3（b）中显示了exampleimage的已识别表列。

3.4标记表区域在前面的步骤中获得的表列为该区域中的表的存在提供了明显的提示。我们在这里做一个简单的假设：在单个页面列中，流动文本不会沿y轴与表格共享空间。在实践中，大多数情况下这种布局都适用于计数器之间的布局，因为如果表格与流动垂直共享空间文本，很难看到文本是否属于表格。基于此假设，我们将表列的边界水平扩展到包含它们的页面列。因此，我们为每个页面列获取了内置列表区域。

在此阶段，正确识别列在一列中的表。但是，跨多个页面列的表被过度细分。尽管如果相邻页列中的两个表区域的开始位置和结束位置对齐，则可以合并它们，但这可能会错误地合并两个列中的不同表。因此，仅当至少一个任何类型的列分区（文本，表格，



（a）部分检测（b）细分市场的表（c）细分市场的表（d）误报检测

图4：此工作中使用的不同绩效指标的说明。每个图都显示了一种类型的细分误差，该细分误差可以通过相应的度量来量化。

可以找到与两个表都重叠的水平分隔线。相邻表中还包括未包含在任何表中，并且在沿x轴有较大重叠的可使用区域的正上方或正下方的表分区和水平分隔分区。这样获得的示例图像的表格区域如图3（c）所示。

3.5删除错误警报尽管在正常情况下，大多数来自普通文本区域的错误警报已被删除，但其他错误警报源（如边际噪声[17]和数字）仍然存在。因此，已识别的表区域通过简单性有效性测试通过：有效表应至少有两个列。通过分析其在x轴上的投影，可以消除由单列组成的错误警报。在x轴上显示有效表格的投影应至少比该页面的全局x高度中值大一个零谷。因此，删除在垂直投影中没有零谷的候选表。

4.性能指标文献中已经报告了用于评估表格检测算法的不同性能指标。这些范围从简单的基于精确度和召回率的度量[6,13]到用于基准化完整表结构提取算法[8]的更复杂的度量。在本文中，由于我们仅专注于表格点画，因此我们将标准措施用于关注表格区域的文档图像分割。因此，根据[13、14、16、20]，我们使用了几种方法来定量评估我们的表格发现算法的不同方面。

真实表和我们的算法检测到的表均由其边界框表示。让Gi代表第i个真值表的边界框，Dj代表文档图像中第j个检测到的表的边界框。两者之间的重叠量定义为：

$$A(Gi, Dj) = 2 \times |Gi \cap Dj| \div (|Gi| + |Dj|) \quad (1)$$

| Gi∩Dj |代表两个区域，以及| Gi |，| Dj |代表地面真相和检测到的表格的各个区域。面积重叠量A将在零与一之间变化，这取决于真实情况表Gi与可检测的表Dj之间的重叠量。如果两个表完全不重叠，则A = 0，并且两个表完全匹配，即|Gi∩Dj|。 = | Gi | = | Dj |，则A = 1。

•正确的检测：这是与其中一个检测到的表有很大重叠（A≥0.9）的地表的数量。

•部分检测：这些是与检测到的表一一对应的地面真相表的数量，但是重叠的量不足以将其分类为正确的检测值（0.1 <A <0.9）（请参见图4（a））。

•过度细分的表：这些表是具有大量重叠（0.1 <A <0.9）且有多个检测到的表的地面真相表的数量。这表明已将地面真相表的不同部分检测为单独的表（请参见图4（b））。

•分段不足的表：这些是与一个检测到的表具有主要重叠（0.1 <A <0.9）的地面真相表的数量，但是对应的检测到的表也与其他地面真相表具有很大的重叠。这表明检测算法合并了一个以上的表（可能是相邻的表），并被报告为一个表（参见图4（c））。

•丢失的表：这些是与任何检测到的表（A≤0.1）没有实质性重叠的地面真相表的数量。这些表被检测算法视为丢失。

•错误肯定检测：这些是与任何事实表（A≤0.1）没有重大重叠的被检测表的数量。这些表被视为误报检测，因为系统将某些非表区域误认为是表（请参见图4（d））。

•区域精度：虽然上面定义的措施有助于理解表检测算法犯了哪些类型的错误，但此措施的目的是通过测量实际属于表检测区域的百分比来总结算法的性能真实图像中的表格区域。当保守地做出关于表格区域的存在的决策时，可以实现高精度。

•区域召回：此度量评估被算法标记为属于表的地面真值表区域的百分比。精确和召回措施的概念类似于它们在信息检索社区中的使用[13]。

5实验和结果为了评估我们的表检测算法的性能，我们选择了UNLV数据集[1]。 UNLV数据集包含各种文档，从技术报告，商务信函到报纸和杂志。该数据集专门用于分析领先的商业OCR系统在UNLV年度OCR准确性测试中的性能[15]。它包含10,000多个不同分辨率的扫描页面和1000个传真文档。扫描的页面被分为双吨级灰度文档。黑白文档再次分为不同的扫描分辨率（200、300和400dpi）。对于每一页，都提供了手动键入的地面真实文本以及手动确定的区域信息。这些区域还根据其内容（文本，表格，半色调，...）进行标记。我们在实验中选择了300 dpi级别的黑白文档，因为这是扫描文档的最常用设置。在这些图像中，选择了427个包含表格区域的页面，并将这些页面图像进一步分为213个图像的训练集和214个图像的测试集。在算法的开发中使用了训练图像，并在这些图像上广泛评估了算法的不同步骤。最后使用测试图像来评估整个系统。

我们的表检测算法对来自UNLV数据集的一些样本图像的结果如图5所示。表1和图6给出了对该算法的详细评估以及与最新的商用OCR系统的比较。请注意，UNLV数据集提供的地面真相表区域还包括该区域内部的表标题。由于表格标题不是表格结构，因此所有OCR系统都将其排除在表格之外，因此，我们通过手动标记所有文档中的表格标题区域来编辑基本信息，然后将该区域排除在外随数据集提供的真值表区域。这是通过缩小地面真值表区域以紧密包围不属于表格标题的所有前景像素而实现的。实验结果表明，我们的系统能够在测试数据上以86％的精度定位表格区域。召回率也很高（79％），显示出精确度和召回率之间的妥协。另一方面，商用OCR系统的召回率较低（37％），但精度较高（96％）。

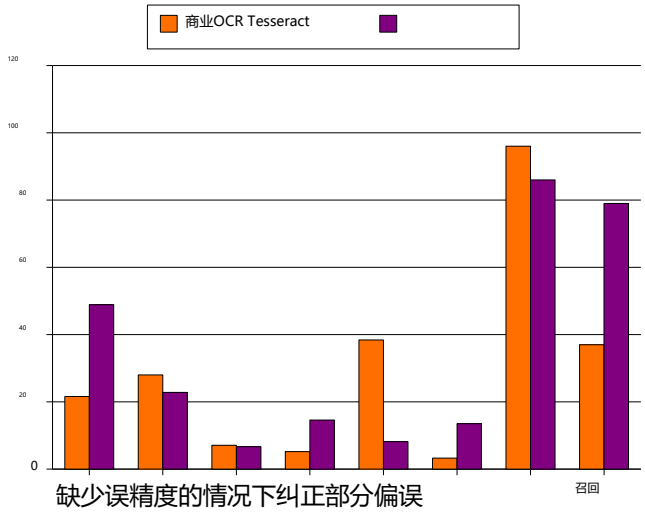


图6：在UNLV测试仪上建议的表格检测系统与商用OCR的准确性的条形图（包含268张表格的214页）。

图4显示了我们的算法产生的一些错误。对结果的分析表明，错误的主要来源是整页表。在这些情况下，列查找算法报告几列文本。由于newspa-pers也具有多个文本列，而没有使用关于文档类型（报告，报纸等）的先验知识，因此很难检测到大量列归因于整页表格。一个典型的示例是包含“目录”的页面。此类页面在UNLV数据集提供的真实信息中标记为稳定区域。但是，我们的算法将它们视为常规文本页面，因此完全丢失了这些“表”或部分检测到了它们。

还分析了我们算法进行的误报检测。我们注意到了我们算法的一个有趣的副作用。由于许多图形区域内部具有间隔开的文本，因此这些区域也被标记为表。尽管此类情况被报告为虚假警报，但在某些情况下，也可能还需要额外发现图形区域。错误警报的其他情况也源于列表方程式。纯文本区域的错误警报非常少见。

6.结论本文提出了一种表格检测算法，作为Tesseract开源OCR系统的一部分。呈现的算法使用Tesseract的布局分析模块的组件来定位具有多种布局的文档中的表。来自UNLV数据集的不同类别文档（公司报告，期刊文章，报纸文章，杂志页面）的实验结果表明，我们的表检测算法与具有更高召回率和较低精度的商业OCR系统相竞争。我们计划将来在表结构提取的方向上扩展这项工作

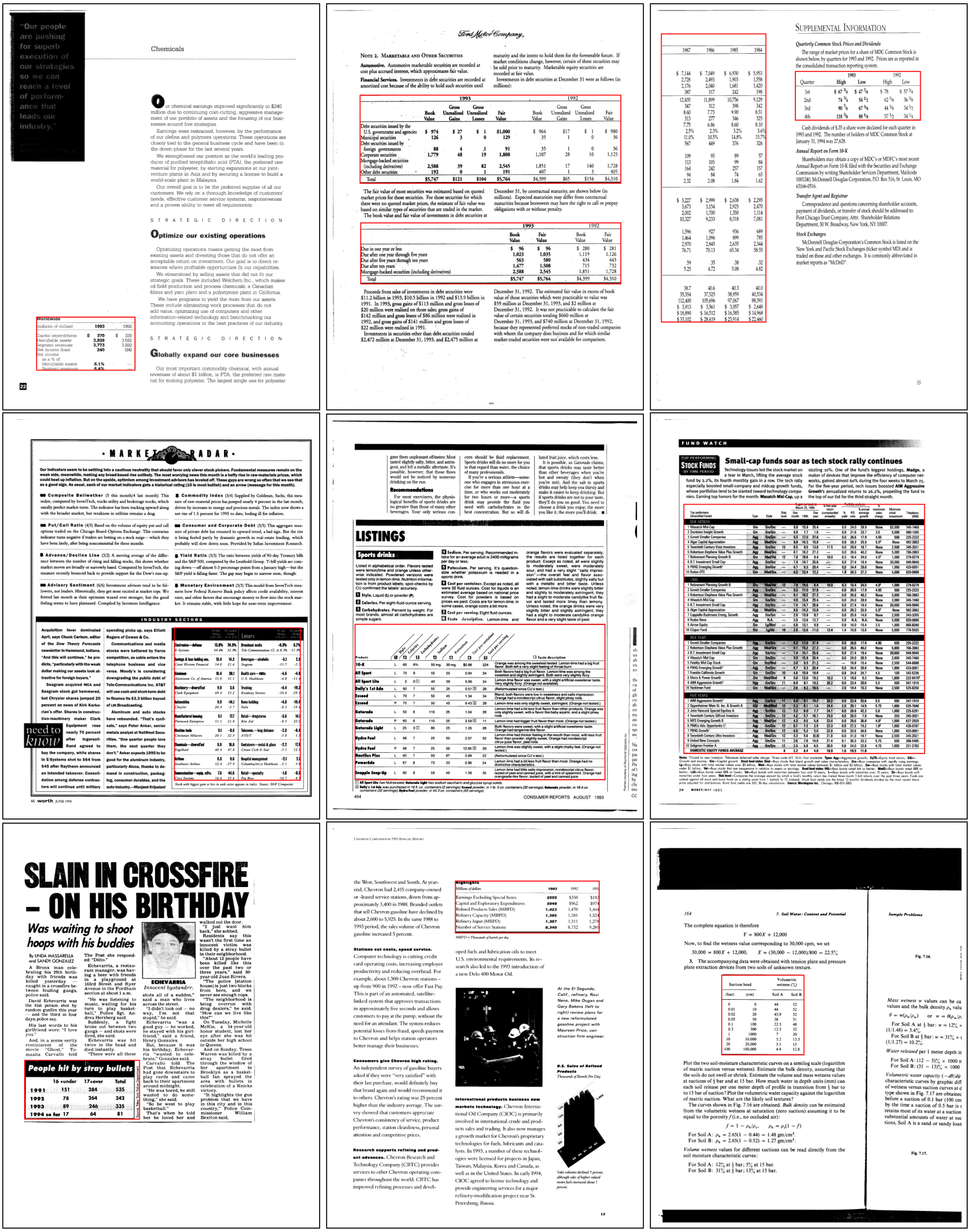


图5：来自UNLV数据集的一些示例图像显示了我们算法的表格发现结果

表1：在包含表格区域的427个二进制300 dpi扫描的UNLV数据集页面上评估商用OCR系统和建议的表格检测算法的结果。

	培训图像（302个表格）测试图像（268个表格）			
	商业Tesseract商业Tesseract 系统系统			
正确检测79130 58131				
部分检测66 65 75 61				
细分桌25 30 19 18				
细分市场表17 55 14 39				
错过的桌子120 31103 22				
误报检测6 17 7 29				
面积精度97.4%90%96.3%86%				
区域召回40.7%78%36.7%79%				

7.参考资料[1] http://www.isri.unlv.edu/ISRI/OCRtk.[2] F.Cesarini，S.Marinai，L.Sarti和G.Soda。文档图像中可培训的表格位置。在Proc.Int。 Conf。关于模式识别，第236-240页，加拿大魁北克，2002年8月。[3] A. C. e Silva。在文档分析中学习丰富的隐藏markov模型：表位置。在过程中。诠释文件分析与识别联盟，第843–847页，西班牙巴塞罗那，2009年7月。[4] B. Gatos，D。Danatsas，I。Pratikakis和S. J. Perantonis。文档图像中的自动表格检测。在过程中。诠释Conf。关于模式识别的进展，第612-621页，英国路径，2005年8月。[5] I. Guyon，R.M。Haralick，J.J。Hull和I.T.Phillips。OCR和文档图像理解研究的数据集。在H. Bunke和P. Wang的编辑中，《字符识别和文档图像分析手册》，第779–799页。世界科学，新加坡，1997年。[6] J.Hu，R.Kashi，D.Lopresti和G.Wilfong，独立于媒体的表格检测。在过程中。SPIEDocument Recognition and Retrieval VII，第291-302页，美国加利福尼亚州圣何塞，2000年1月。[7] J.Hu，R.S。Kashi，D.Lopresti和G.Wilfong。表识别实验。在过程中。文件布局解释与应用国际研讨会，美国华盛顿，9月。2001. [8] J.Hu，R.S。Kashi，D.Lopresti和G.Wilfong，评估表处理算法的性能。诠释周杰伦关于文档分析与识别，4（3）：140–153，2002年。[9] D. Keysers，F。Shafait和T.M. Breuel。 Documentimage区域分类-一种简单的高性能方法。在第二国际Conf。关于计算机视觉理论和应用，第44-51页，西班牙巴塞罗那，2007年3月。[10] T. Kieninger和A. Dengel。纸到HTML表格转换系统。在过程中。文件分析系统，第356-365页，日本长野，1998年11月。[11] T. Kieninger和A. Dengel。使用固有布局功能进行表识别和标记。在过程中。国际会议关于模式识别的进展，英国普利茅斯，1998年11月。[12] T. Kieninger和A. Dengel。应用T-RECS表识别系统到商务信函域。诠释Conf。文件分析和识别，第518-522页，美国华盛顿州西雅图，2001年9月。[13] T. Kieninger和A. Dengel。基准测试表结构识别结果的方法。InProc。 8th Int。 Conf。文件分析和识别，第1232-1236页，韩国首尔，2005年8月。[14] S. Mandal，S。Chowdhury，A。Das和B. Chanda。来自文档图像的简单有效的表格检测系统。诠释周杰伦文件分析与识别，2006年，第8（2-3）：172–182页。[15] S. V. Rice，F。R. Jenkins和T. A. Nartker。第四次OCR准确性年度测试。技术报告，内华达大学信息科学研究所，拉斯维加斯，1995年。[16] F. Shafait，D。Keysers和T.M. Breuel。六个页面细分算法的性能评估和基准测试。 IEEE Trans。模式分析与机器智能研究，2008，30（6）：941-954。[17] F. Shafait，J.van Beusekom，D.Keysers和T.M.Breuel。使用页框检测清除文档周杰伦文件分析与识别研究，11（2）：81-96，2008年。[18]史密斯（R. Smith）。Tesseract OCR引擎概述。第九国际Conf。文件分析和识别，第629-633页，巴西库里提巴，2007年9月。[19]史密斯（R. Smith）。通过制表符停止检测进行混合页面布局分析。在过程中。诠释Conf。文件分析和识别，第241-245页，西班牙巴塞罗那，2009年7月。[20] Y. Wang，R。Haralick和I. T. Phillips。自动生成地面事实和基于背景分析的表结构提取方法。在过程中。诠释Conf。文件分析和识别，第528-532页，美国华盛顿州西雅图，9月。 2001. [21] Y. Wang，I。Phillips和R. Haralick。文档区域内容分类及其性能评估。模式识别，39（1）：57-73，2006年。