# PINGAN-VCGROUP’S SOLUTION FOR ICDAR 2021 COMPETITION ON SCIENTIFIC LITERATURE PARSING TASK B: TABLE RECOGNITION TO HTML

Jiaquan Ye, Xianbiao Qi, Yelin He, Yihao Chen, Dengyi Gu, Peng Gao, and Rong Xiao
Visual Computing Group, Ping An Property  Casualty Insurance Company
Ping An Technology Company

**摘要**

本文介绍了对 ICDAR 2021 科学文献解析竞赛任务 B：将表格识别为 HTML 代码的第二名解决方案的复现与尝试改进工作。在复现论文的解决方法中，作者将表格内容识别任务分为四个子任务：表格结构识别、文本行检测、文本行识别和框分配。本文复现的实验结果在 9,115 个验证样本上取得了 96.49% 的 TEDS 分数。随后进行了三次改进尝试实验，分别对表格识别模型 TableMASTER 增加一层 transformer decoder、增加残差连接、替换为第一名的表格识别模型 LGPMA，表格结构识别实验结果分别取得 77.84%，77.53% 的 accuracy，使用 LGPMA 模型在 9,115 个验证样本上取得了 91.25% 的 TEDS 分数。

**关键词：**Table Structure Recognition；TableMASTER；LGPMA

## 1　引言

随着文档尤其是通过扫描、拍照等方式生成的文档的快速增长，自动地进行文档识别并从中提取有用的数据成为了一个备受关注的研究问题。这其中，表格作为一种高效的数据组织和展现的方式，是文档页面中最重要的数据对象之一。表格识别包括表格检测与结构识别，作为文档识别一个重要的子任务，一直是该领域研究者关注的研究问题。近年来，国内外专家、学者针对这一问题进行了大量研究，引入深度学习方法和模型来进行自动化识别表格信息。表格识别也逐渐演变成了多个分支研究领域，包括：专门的数据集构建、表格检测、表格结构识别、表格检测与结构识别等。表格自动化识别的意义在于人工处理表格的方式存在很多问题。一是，人工处理方法经常会出现表格处理错误、不一致等问题。二是，手工提取表格信息往往是一个繁琐而耗时的过程。三是，在金融业和许多其他领域，表格往往是以非结构化的数字文件公开的，这些文件难以直接进行人工提取和处理。因此，高效地从文档中找到表格，同时有效提取表格中的数据与结构信息即表格识别，成为了一个亟待解决的问题。这篇复现论文中的 ICDAR 2021 竞赛中的科学文献解析任务 B 是将表格图像重构为 HTML 代码。在本次比赛中，PubTabNet 数据集 (v2.0.0)[1]作为官方评估数据提供，并使用 Tree-Edit Distance-based similarity (TEDS) 度量进行评估。

## 2 相关工作

### 2.1 传统方法

思想：基本都是基于规则和图像处理方法，腐蚀、膨胀，找连通区域，检测线段、直线，求交点，合并猜测框，按大小过滤。如 pdfplumber 表格抽取。

pdfplumber 抽取表格主要包含以下几步：因为表格及单元格都是存在边界的（由可见或不可见的线表示），所以第一步，pdfplumber 是找到可见的或猜测出不可见的候选表格线。因为表格以及单元格基本上都是定义在一块矩形区域内，所以第二步，pdfplumber 是根据候选的表格线确定它们的交点。根据得到的交点，找到它们围成的最小的单元格。把连通的单元格整合到一起，生成一个检测出的表格对象。

### 2.2 深度学习方法

TableNet[2]

思想：一种端到端的、多任务的、基于编解码器的图像语义分割模型，整体架构类似于 U-Net, 该解决方案准确检测图像中的表格区域，然后检测和提取检测到表的行和列中的信息。

SPLERGE[3]

思想：一种先自顶向下、再自底向上的两阶段表格结构识别方法 SPLERGE，分为 Split 和 Merge 两个部分。Split 部分先把整个表格区域分割成表格所具有的网格状结构，模型预测每一行或列像素是否属于单元格间的分隔符区域。而 Merge 部分则是对 Split 的结果中的每对邻接网格对进行预测，判断它们是否应该合并。

DeepTabStr[4]

思想：可变形卷积替换传统卷积，将表格结构检测视为一个对象检测问题，将表格的行和列当做是要检测的对象。变形卷积网络加入了各个像素的偏移向量 Offset 来训练卷积窗口的形状。传统的 ROI-pooling 层将 ROI 转换为 k*k 的固定大小，可变形的 ROI-pooling 层也引入了额外的偏移量，使得 ROI-pooling 层也具有了变形的属性，以适应不同区域的对象检测。

## 3 本文方法

### 3.1 本文方法概述

此部分对本文将要复现的工作进行概述。本文作者将表格内容识别任务分为四个子任务：表格结构识别、文本行检测、文本行识别和框分配。识别表格转换为 HTML 代码流程如图 1所示，首先对输入的表格图片进行文本检测，使用 PSENet[5]模型用于检测表格图像中的每个文本行，可以得到对应文本框的坐标信息，通过该坐标信息可以对表格图片进行裁剪，将原表格裁剪为单行的文本图片，对单行的文本图片使用文本识别模型 MASTER[6]得到每个裁剪图片的文本信息。同时通过本文提出的 TableMASTER[7]对输入的表格图片进行表格结构识别，可以得到表格结构的 HTML 预测代码以及表格 cell 回归框坐标。将文本框坐标和表格 cell 回归框坐标进行匹配得到对应文本在表格中的位置信息，通过整合该 cell 文本信息和位置信息即可还原表格 HTML 代码。

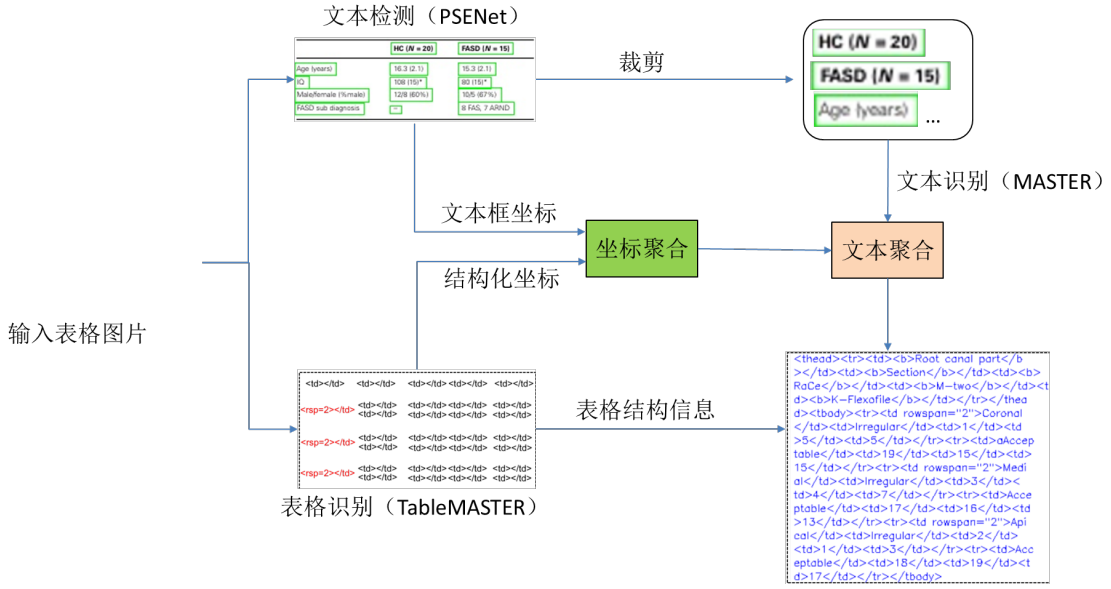


图 1: 表格转换为 HTML 代码流程图

### 3.2 TableMASTER 模型

本文作者提出的 TableMASTER 模型，图 2(b)，是通过 vanilla MASTER，图 2(a)，改进而来，不同于 vanilla MASTER，TableMASTER 采用两个分支，其中一个分支用于预测 HTML 代码序列，另一个分支进行表格 cell 的框回归。需要注意的是，本文作者在 vanilla MASTER 第一层 Transformer 解码层之后便将模型分为两个分支，而不是在最后一层 Transformer 解码层进行分支。实验结果显示，这样设计网络结构得到的表格识别效果更好。

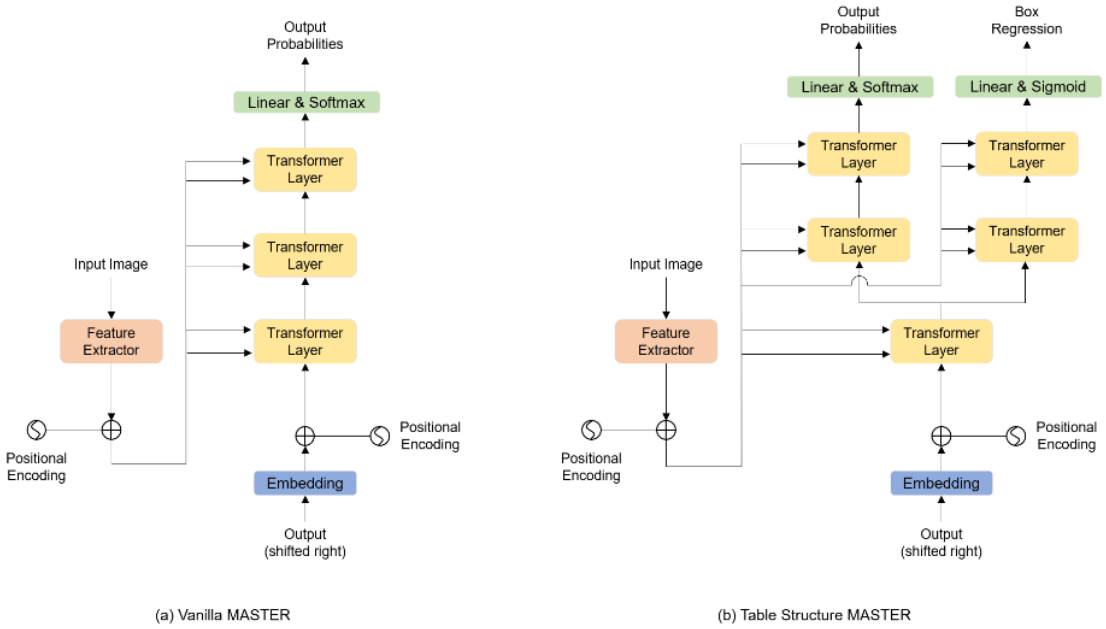


图 2: (a)vanilla MASTER 结构;(b)TableMASTER 结构

## 4 复现细节

### 4.1 与已有开源代码对比

本次复现工作通过修改作者提供的开源代码来进行改进尝试，开源代码基于 mmdetection 和 mmocr 开源项目框架的基础上进行开发。

### 4.1.1 复现论文实验

1. 首先运行 data_preprocess.py 脚本对数据集进行预处理，在运行脚本之前修改脚本中对应的数据集路径和保存路径，并将 split 标签设置为 train，表示生成训练数据。预处理完成后，将 split 标签设置为 val，表示生成验证数据。

2. 下载作者提供的表格文本行检测数据集，通过运行 table_text_line_detection_dist_train.sh 脚本来训练文本行检测模型，运行脚本之前修改 psenet_r50_fpnf_600e_pubtabnet.py 配置文件中的文本行检测数据集路径。

3. 训练文本行识别模型和表格识别模型之前需要将预处理好的数据集转换为 lmdb 文件，分别将脚本文件 lmdb_maker.py 中 parse_master_args 和 parse_tablemaster_args 的数据集路径修改为预处理后的数据集路径，运行脚本，分别得到文本识别和表格识别的训练和验证数据集。

4. 通过运行 table_text_line_recognition_dist_train.sh 脚本来训练文本行识别模型，运行脚本之前修改 master_lmdb_ResnetExtra_tableRec_dataset_dynamic_mmfp16.py 配置文件中的字母表 textline_recognition_alphabet.txt 路径和文本识别训练集和验证集 lmdb 文件路径。

5. 通过运行 table_recognition_dist_train.sh 脚本来训练表格结构识别模型，运行脚本之前修改 table_master_lmdb_ResnetExtract_Ranger_0930.py 配置文件中的字母表 structure_alphabet.txt 路径和表格结构识别训练集和验证集 lmdb 文件路径。

6. 通过运行 run_table_inference.py 脚本来对训练好的文本检测，文本识别，表格结构识别模型在验证集中进行推理，运行脚本之前修改 table_inference.py 脚本中文本检测，文本识别，表格结构识别的配置文件和模型参数路径，分别保存文本行检测和识别端到端推理结果和表格结构识别推理结果。

7. 通过运行 match.py 脚本将文本行检测和识别端到端推理结果与表格结构识别推理结果进行合并，运行脚本之前修改相应的路径，得到最终的推理结果。

8. 通过运行 get_val_gt.py 脚本得到验证集真实标签。

9. 通过运行 mmocr_teds_acc_mp.py 得到最后的 TEDs 分数，运行脚本之前修改推理结果和真实结果的路径。

### 4.1.2 改进尝试 1 代码与已有开源代码对比

通过修改开源代码中的表格结构识别配置文件 table_master_lmdb_ResnetExtract_Ranger_0930.py，将表格识别模型 TableMASTER 原来的三层 transformer decoder 改为四层，其他源文件不变。

### 4.1.3 改进尝试 2 代码与已有开源代码对比

通过在模型文件 master_decoder.py 中增加一个新模型 TableMasterDecoderRes，实现在原表格识别模型 TableMASTER 模型前两层 transformer decoder 加上残差边，并在配置文件 table_master_lmdb_ResnetExtract_Ranger_0930.py 中引用该模型进行训练，其他源文件保持不变。

#### 4.1.4 改进尝试 3 代码与已有开源代码对比

将该比赛第一名的表格识别模型 LGPMA[8]替换本论文模型 TableMASTER。通过自己写的数据预处理脚本 data_preprocess.py 将原数据集标签中的验证集标签保存为独立的文件 PubTabNet_2.0.0_val.jsonl。修改 LGPMA 配置文件 lgpma_pub.py 中验证集的标签路径为 PubTabNet_2.0.0_val.jsonl 所在路径。通过修改脚本 test_pub_with_ocr.py 中的 obtain_ocr_results 函数，将复现论文中的文本行检测和识别结果返回到 LGPMA 模型的推理参数中，完成表格识别模型的替换。

### 4.2 创新点

根据表 2实验结果推断，作者提出的表格识别模型 TableMASTER 识别精度不高，故尝试改动作者提出的 TableMASTER 模型来进行改进。

#### 4.2.1 改进尝试 1

根据作者提出的表格识别模型 TableMASTER，如图 2(b)，考虑到网络越深，模型越强大的思路在原有的模型基础上在两个输出分支各加一层 Transformer 解码层。改进尝试后的表格识别模型如图 3所示。

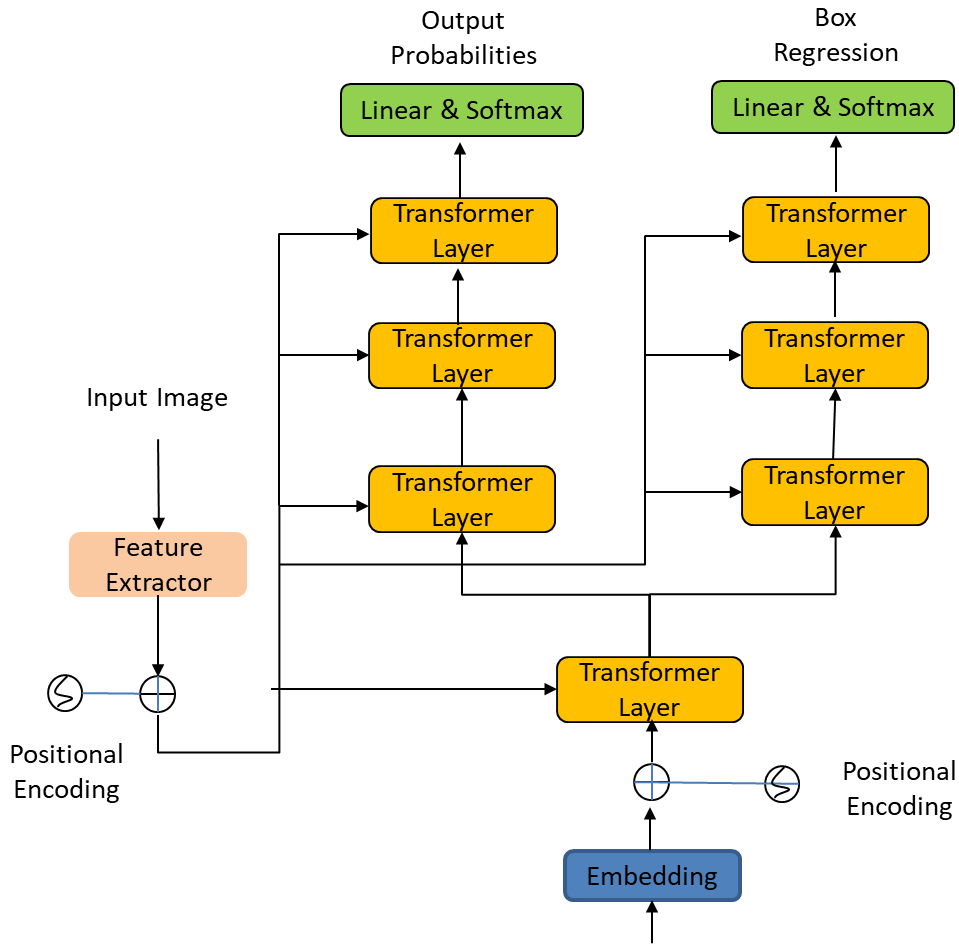


图 3: 改进尝试后的表格识别模型 TableMASTER1

#### 4.2.2 改进尝试 2

后面尝试残差网络的思想，在两个分支的前两层 Transformer 解码层加上残差边。改进尝试后的表格识别模型如图 4所示。

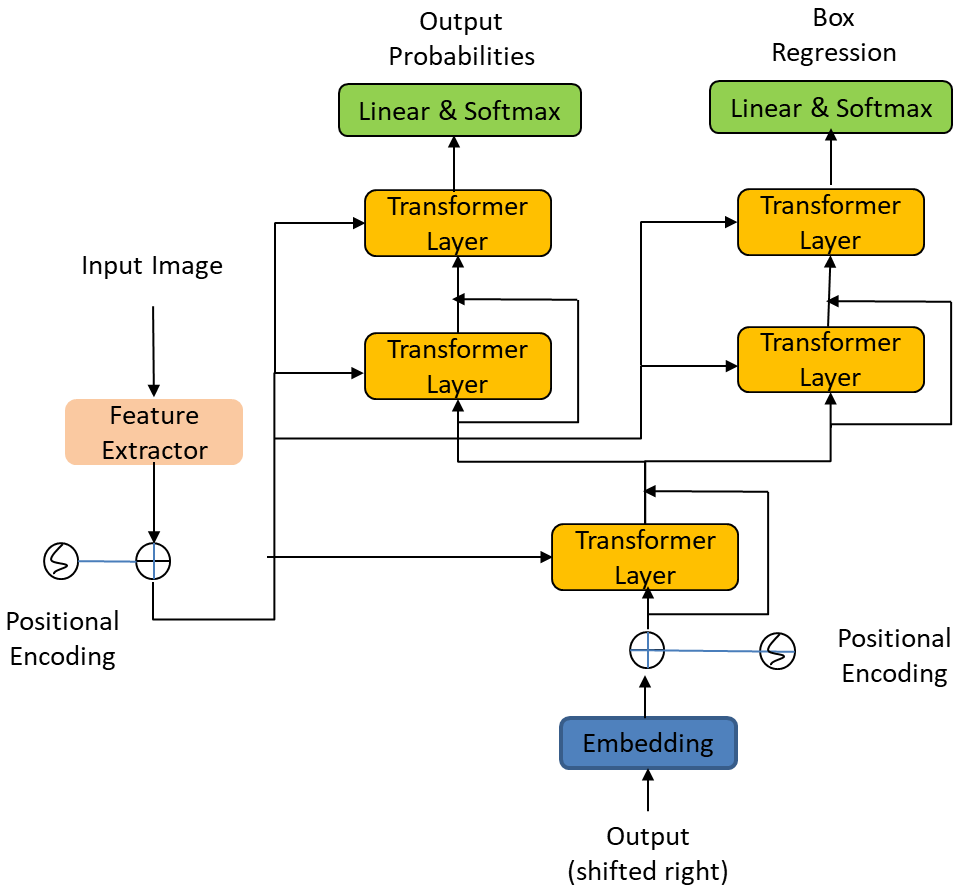


图 4: 改进尝试后的表格识别模型 TableMASTER2

### 4.2.3 改进尝试 3

尝试采用本次竞赛第一名的表格识别模型 LGPMA 替换本复现论文提出的 TableMASTER 模型。LGPMA 模型如图 5所示。

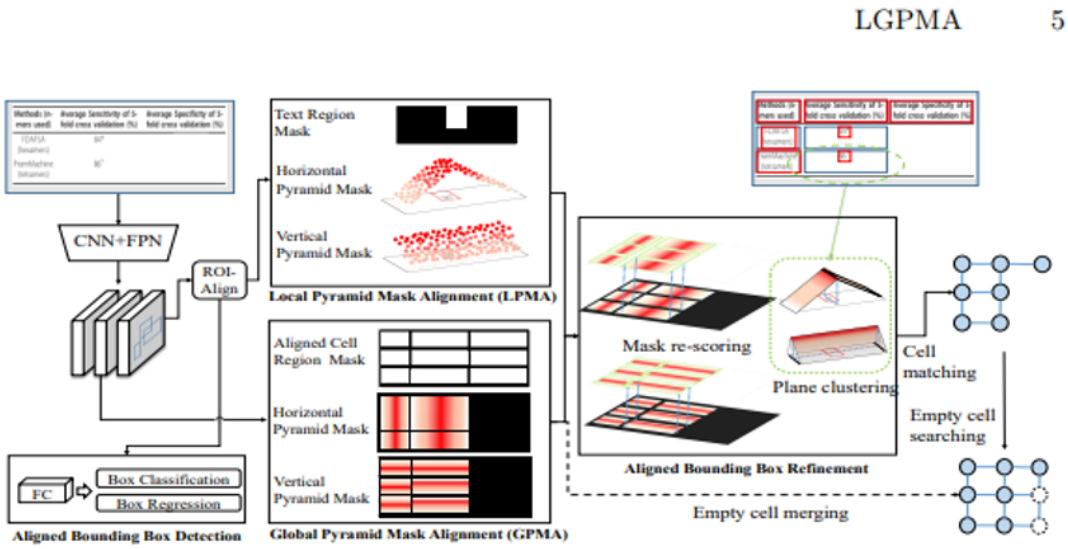


图 5: LGPMA 模型

# 5 实验结果分析

## 5.1 论文复现实验结果

本文采用 PSENet 和 MASTER 方法实现文本检测和文本识别已经取得非常好的效果，实验结果如表 1所示。

表 1: PSENet 和 MASTER 方法实现文本检测和文本识别实验结果

| **Models** | **Accuracy** |
|---|---|
| PSENet | 0.9968 |
| MASTER | 0.98 |

本文采用提出的 TableMASTER 实现表格结构识别，其实验结果如表 2所示。

表 2: TableMASTER 实现表格结构识别实验结果

| **Models** | **Accuracy** |
|---|---|
| TableMASTER | 0.7773 |

综合实验结果与作者提供的实验结果对比如表 3所示。

表 3: 综合实验结果与作者提供的实验结果对比

| **Models** | **Average TEDS scores** |
|---|---|
| PSENet + MASTER +TableMASTER | 0.9649 |
| PSENet + MASTER +TableMASTER(author) | 0.9658 |

实验结果表明，复现的实验结果与作者提供的实验结果已经非常接近。如表 2所展现的 TableMASTER 模型对表格结构识别效果欠佳，原因在于该模型将原本存在两个单元格的表格结构预测成只有一个单元格的结构，如图 6所示。

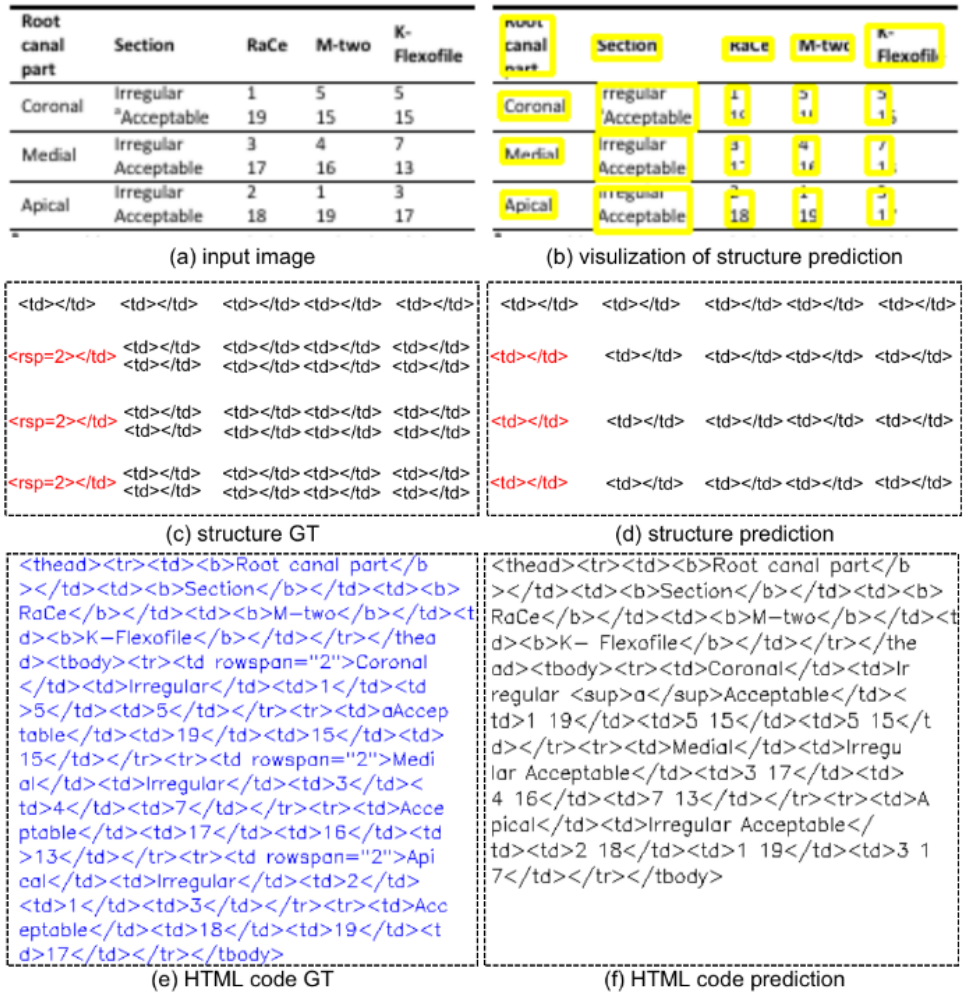


图 6: TableMASTER 模型预测错误例子

### 5.2 改进尝试 1 实验结果

仅表格识别部分，改进尝试 1 实验结果与复现实验结果对比如表 4所示。

表 4: 尝试改进 1 实验结果与复现结果对比（仅表格识别部分）

| Models | Accuracy |
|---|---|
| TableMASTER | 0.7773 |
| TableMASTER1 | 0.7784 |

实验结果表明，表格识别精度相比未加改动的复现实验结果只有极小的提升，但是意义不大。

### 5.3 改进尝试 2 实验结果

仅表格识别部分，改进尝试 2 实验结果与复现实验结果、改进尝试 1 实验结果对比如表 5所示。

表 5: 尝试改进 2 实验结果与复现结果、改进尝试 1 实验结果对比（仅表格识别部分）

| Models | Accuracy |
|---|---|
| TableMASTER | 0.7773 |
| TableMASTER1 | 0.7784 |
| TableMASTER2 | 0.7753 |

实验结果表明，表格识别精度比之前的模型都要低，改进失败。

### 5.4 改进尝试 3 实验结果

改进尝试 3 尝试采用竞赛第一名的表格识别模型 LGPMA 替换本复现论文提出的 TableMASTER 模型，综合实验结果如表 6所示。

表 6: 使用不同表格识别模型综合实验结果对比

| Models | Average TEDS scores |
|---|---|
| PSENet + MASTER +TableMASTER | 0.9649 |
| PSENet + MASTER + LGPMA(first) | 0.8771 |
| PSENet + MASTER + LGPMA(second) | 0.9125 |

第一次实验结果 TEDS score 为 87.71% 左右，如表 6第二行所示，后面发现是文本裁剪和识别输出的差异问题，做出修改后第二次实验结果得到 91.25% 左右，如表 6第三行所示，比一开始复现实验结果低 5% 左右。其原因在于我是直接将复现论文的文本行检测和识别结果替换掉竞赛第一名解决方案中的文本行检测和识别模块，在两个模型对接的过程中虽然已经考虑到了图像裁剪大小等预处理和后处理的问题，但不可避免遗漏部分细节未能与原配模块高度契合，故导致只有 91.25% 左右的 TEDS score。

## 6 总结与展望

总结：完成了本论文的复现工作，并对文章提出的表格识别模型进行改进尝试并实验。

展望：针对作者提出的表格识别错误的例子（图 6）进一步挖掘，或许可以找到更佳的改进方法；针对作者提出的 box assignment 子任务可以通过 Graph Neural Network (GNN) 来实现，可以进一步实验进行验证。

## 参考文献


[1] ZHONG X, SHAFIEIBAVANI E, JIMENO YEPES A. Image-based table recognition: data, model, and evaluation[C]//European Conference on Computer Vision. 2020: 564-580.

[2] PALIWAL S, D V, RAHUL R, et al. TableNet: Deep Learning Model for End-to-end Table Detection and Tabular Data Extraction from Scanned Document Images[J]. international conference on document analysis and recognition, 2019.

[3] TENSMEYER C, MORARIU V I, PRICE B, et al. Deep Splitting and Merging for Table Structure Decomposition[J]. international conference on document analysis and recognition, 2019.

[4] SIDDIQUI S A, FATEH I A, RIZVI S T R, et al. DeepTabStR: Deep Learning based Table Structure Recognition[J]. international conference on document analysis and recognition, 2019.

[5] WANG W, XIE E, LI X, et al. Shape robust text detection with progressive scale expansion network[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 9336-9345.

[6] LU N, YU W, QI X, et al. Master: Multi-aspect non-local network for scene text recognition[J]. Pattern

Recognition, 2021, 117: 107980.

[7] YE J, QI X, HE Y, et al. PingAn-VCGroup's Solution for ICDAR 2021 Competition on Scientific Literature Parsing Task B: Table Recognition to HTML.[J]. arXiv: Computer Vision and Pattern Recognition, 2021.

[8] QIAO L, LI Z, CHENG Z, et al. LGPMA: Complicated Table Structure Recognition with Local and Global Pyramid Mask Alignment[C]//Document Analysis and Recognition-ICDAR 2021, 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part I. 2021: 99-114.