

Table Recognition and Accessibility in RAVI with Layout Parser and MTL-TabNet

K Laxman 2018CS50408

COD893 Major Presentation

Advisors: Prof.M.Balakrishnan, Prof.Volker Sorge, Abhishek Baghel

Supervisor: Prof. Vireshwar Kumar

July 10, 2023



About RAVI

- 1 RAVI (Reading Assistant for Visually Impaired) aims to make digital content in PDFs accessible to readers with blindness and low vision.
- 2 **The overall objective** of this project is to develop a web application that automatically converts PDF files to editable HTML.
- 3 The output HTML can further be edited on the platform.

Overview of MTP1 work

- ① The generated HTML was not containing image descriptions. The challenges of **lacking image descriptions** was addressed.
- ② The aim Enhance text extraction and improve **NLP-based descriptions** in the output HTML.
- ③ Crowdsourcing feature, **allowing users to add Alt text** to images in the HTML.
- ④ Improve code by integrating detection modules, using **Azure Computer Vision API**, and optimizing for generalization.

MTP2: Table recognition and Integration in RAVI

- ❶ The main objective of this project is to **make tables accessible** in a document like PDF.
- ❷ Analysis of table recognition, content extraction, and **placing them in output HTML**.
- ❸ In previous work, integration of the **Bordered Table** Algorithm was **completed**

Previous work on Table Analysis

- This work was done by Amar Agnihotri (MTP -2022)
- Two types of tables considered : Bordered tables and Borderless tables
- Amar worked on the bordered tables
- In his work he proposed to use CascadeTabNet for table recognition for borderless tables.
- However, Content extraction and **integration of borderless** tables into **RAVI were not done.**

Examples:

Product ID	Product Name	Product Quality	Product Quantity
1	Wheat	Good	200 Bags
2	Rice	Good	250 Bags
3	Sugar	Good	200 Bags

Figure: Example of Bordered table

Topics	Days for Class	Timing
Design For Ux	Monday	8:30 – 10:30 am
Design Thinking		12:30 – 2:30 pm
Empathy Map	Tuesday	9:30 – 11:30 pm
Emotional Intelligence	Wednesday	
Usability	Thursday	8:30 – 10:30 am
Utility	Friday	8:30 – 10:30 am
Accessibility	Friday	8:30 – 10:30 am

Figure: Example of Borderless table

Bordered table recognition and content extraction by Amar Agnihotri.

- 1 Amar Agnihotri's work **focused** on detecting tables and extracting their data as tabular cell structures.
- 2 Various techniques and algorithms were discussed, including the Bordered Table Algorithm and Borderless Table Algorithm.

Bordered Table Algorithm

- Detect vertical lines in the table.
- Detect horizontal lines in the table.
- Identify the start and end points of each vertical and horizontal line.
- Map the cell data based on the intersection points of the vertical and horizontal lines.

3.2 Bordered Table Algorithm Explanation with an Example

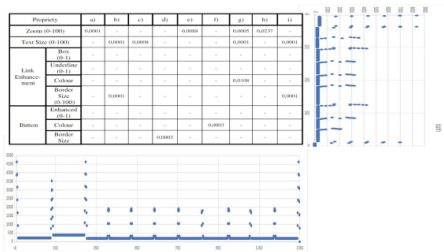


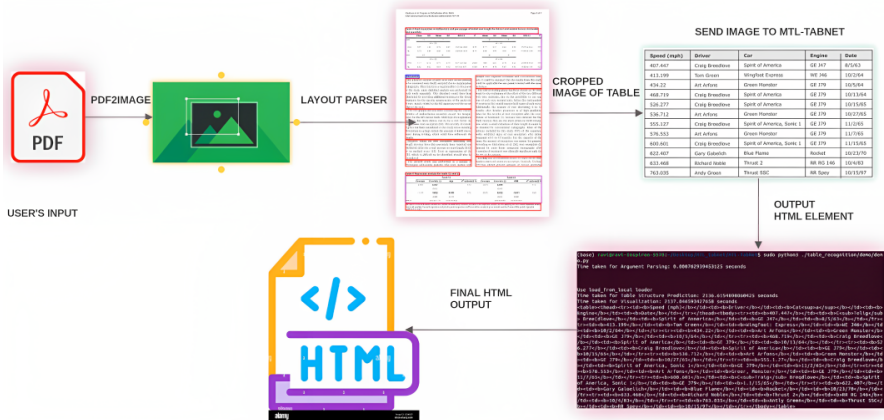
Figure 4 Example of Bordered Table Algorithm

Figure: Example of Bordered table working

Problem Statement

- 1 **Borderless tables are not identified:** There is a high chance that tables **without borders** will be present in PDFs, causing **screen readers** to **miss the table content**. This results in visually impaired individuals having partial knowledge and **difficulty understanding STEM material**, as they are unable to comprehend the borderless tables that describe the data statistics.
- 2 To identify the location of tables in document images and to extract the table data
- 3 **Debugging of bordered table algorithm** Different individuals contributed code for table analysis and content extraction, but the output HTML displayed extra paragraphs below the table. This **bug was been reported** and **resolved** in this work.

Workflow image :



Solution

- 1 The Layout parser is utilized to process the **input PDF file with multiple pages** and accurately **identify the regions containing tables** throughout the document.
- 2 **OpenCV** is employed to extract the specific table regions from the identified areas and save them as **separate images**. The corresponding coordinates (x1, x2, y1, y2) of each table region are recorded and **stored in a JSON file** for reference.
- 3 The Layout parser's output images are then **passed to the MTL-TabNet** algorithm. MTL-TabNet **processes each image and generates the corresponding HTML element** for the table. In case of **multiple images**, they are **processed in sequential order**.
- 4 The **JSON file is updated with the relevant image and its corresponding HTML table element**. This updated JSON file serves as a reference for **integrating** the table information into the **final HTML output**.
- 5 The extracted table data, along with the corresponding HTML table elements, are seamlessly **integrated into the main original HTML output**, ensuring accurate representation and **accessibility for visually impaired users**.

MTL-TabNet Introduction

- 1 MTL-TabNet is a multi-task learning-based model for **image based table recognition**
- 2 The model is designed to **recognize tables** in document images and **extract tabular data** from them
- 3 The model is trained on various **datasets** such as **PubTabNet**, **FinTabNet** and **Marmot**.

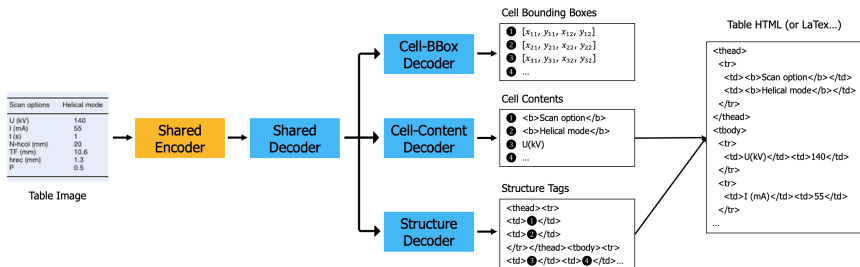


Figure: Working of MTL-TabNet

Why MTL-Tabnet?:

- 1 In this presentation, we will compare two popular methods for table recognition and extraction: MTL-Tabnet and TableMaster
- 2 These methods, developed in the field of document analysis and optical character recognition (OCR), have demonstrated their effectiveness in automatically detecting and understanding tables within documents.
- 3 We will explore their key features, advantages, and provide a comparative analysis.

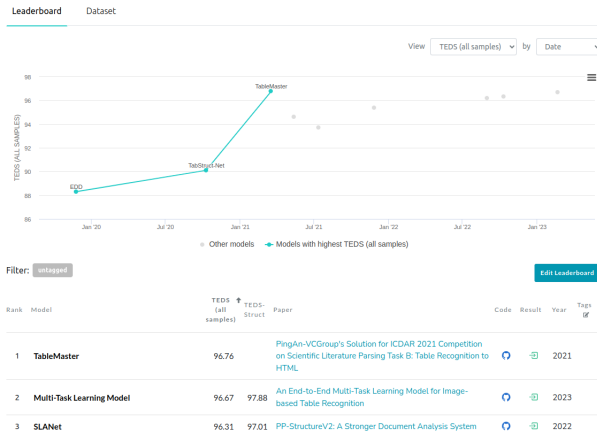
Table: Comparison between MTL-TabNet, Tablemaster, and Cascade TabNet

Approach	MTL-TabNet	Tablemaster	Cascade TabNet
Year	2023 (Latest Work)	2021 (Released)	2022 (Released)
Models	Transformer based model	PSENet + MASTER	Cascade RCNN + TabNet
Accuracy	0.9885	0.9676	0.9823
TEDS Score	96.67	0.9676	97.21
Other Differences			
Training Approach	End-to-End Multi-Task Learning with Tasks	Two-Stage Training Process	Cascade Training
Architecture	Transformer-based Model	CNN-based Model	Ensemble Model
Dependency	PyTorch Framework	TensorFlow Framework	PyTorch Framework

TEDS(Text Extraction from Document Structures) Score: Evaluation metric for text extraction algorithms, measures accuracy compared to reference text.

Accuracy: Overall correctness of a system or algorithm in text extraction.

Table Recognition on PubTabNet



MTL-Tabnet has been chosen over other approaches because of Latest,best approach in 2023 and big database(PubTabnet ,FinTabnet)

Implementaion of MTL-Tabnet

Input image:

Disability Category	Participants	Ballots Completed	Ballots Incomplete/Terminated	Results	
				Accuracy	Time to complete
Blind	5	1	4	34.5%, n=1	1199 sec, n=1
Low Vision	5	2	3	98.3% n=2 (97.7%, n=3)	1716 sec, n=3 (1934 sec, n=2)
Dexterity	5	4	1	98.3%, n=4	1672.1 sec, n=4
Mobility	3	3	0	95.4%, n=3	1416 sec, n=3

output image:

```
(base) ravi@ravi-Inspiron-5570:~/Desktop/MTP2/Approach1/MTL-TabNet$ <table id=
<tbody><tr><td rowspan="2" colspan="1" style="border: 1px solid black;">Disabi
ck;">Participants</td><td rowspan="2" colspan="1" style="border: 1px solid bla
solid black;">Ballots Incomplete/Terminated</td><td rowspan="1" colspan="2" st
n="1" style="border: 1px solid black;">Accuracy</td><td rowspan="1" colspan="1"
an="1" colspan="1" style="border: 1px solid black;">Blind</td><td rowspan="1"
="1" style="border: 1px solid black;">1</td><td rowspan="1" colspan="1" style=
: 1px solid black;">34.5%, n=1</td><td rowspan="1" colspan="1" style="border:
yle="border: 1px solid black;">Low Vision</td><td rowspan="1" colspan="1" sty
der: 1px solid black;">2</td><td rowspan="1" colspan="1" style="border: 1px so
ck;">98.3% n=2 (97.7%, n=3)</td><td rowspan="1" colspan="1" style="border: 1px
colspan="1" style="border: 1px solid black;">Dexterity</td><td rowspan="1" col
" style="border: 1px solid black;">4</td><td rowspan="1" colspan="1" style="bo
px solid black;">98.3%, n=4</td><td rowspan="1" colspan="1" style="border: 1px
yle="border: 1px solid black;">Mobility</td><td rowspan="1" colspan="1" style=
: 1px solid black;">3</td><td rowspan="1" colspan="1" style="border: 1px solid
">95.4%, n=3</td><td rowspan="1" colspan="1" style="border: 1px solid black;">
```

Examples of tables:

Showcasing the effectiveness of MTL-Tabnet in handling different types of tables:

output image after:



Speed (mph)	Driver	Car	Engine	Date
407.447	Craig Breedlove	Spirit of America	GE J47	8/5/63
413.199	Tom Green	Wingfoot Express	WE J46	10/2/64
434.22	Art Arfons	Green Monster	GE J79	10/5/64
468.719	Craig Breedlove	Spirit of America	GE J79	10/13/64
526.277	Craig Breedlove	Spirit of America	GE J79	10/15/65
536.712	Art Arfons	Green Monster	GE J79	10/27/65
555.127	Craig Breedlove	Spirit of America, Sonic 1	GE J79	11/2/65
576.553	Art Arfons	Green Monster	GE J79	11/7/65
600.601	Craig Breedlove	Spirit of America, Sonic 1	GE J79	11/15/65
622.407	Gary Gabelich	Blue Flame	Rocket	10/23/70
633.468	Richard Noble	Thrust 2	RR RG 146	10/4/83
763.035	Andy Green	Thrust SSC	RR Spey	10/15/97

Crystal	Type	Melting temp.	Notes
I	O N E	17 °C (63 °F) Soft, crumbly, melts too easily	
II	T W O	21 °C (70 °F)	Soft, crumbly, melts too easily
III	T H R E E	26 °C (79 °F) 28 °C (82 °F)	Firm, poor snap, melts too easily
IV	F O U R		Firm, good snap, melts too easily
V	F I V E	34 °C (93 °F)	Glossy, firm, best snap, melts near body temperature (37 °C) Hard, takes weeks to form
VI	S I	36 °C (97 °F)	



output image after:

```
(base) ravi@ravi-Inspiron-5570:~/Desktop/MTL_tabnet/MTL-TabNet$ sudo python3 ./table_recognition/demo/demo.py
Time taken for Argument Parsing: 0.000762939453125 seconds
```

Use load_from_local loader

Time taken for Table Structure Prediction: 2136.6154890060425 seconds

Time taken for Visualization: 2137.046593427658 seconds

```
<table><thead><tr><td><b>Speed (mph)</b></td><td><b>Driver</b></td><td><b>Cat<sup>a</sup></b></td><td><b>Engine</b></td><td><b>Date</b></td></tr></thead><tbody><tr><td><b>407.447</b></td><td><b>C<sub>Telig</sub> Bree(dlove</b></td><td><b>Spirit of Annerica</b></td><td><b>GE J47</b></td><td><b>8/5/63</b></td></tr><tr><td><b>413.199</b></td><td><b>Tom Green</b></td><td><b>Wingfoot: Express</b></td><td><b>WE J46</b></td><td><b>10/2/64</b></td></tr><tr><td><b>434.22</b></td><td><b>Art Arfons</b></td><td><b>Green Monster</b></td><td><b>GE J79</b></td><td><b>10/5/64</b></td></tr><tr><td><b>468.719</b></td><td><b>Craig Breedlove</b></td><td><b>Spirit of America</b></td><td><b>GE J79</b></td><td><b>10/13/64</b></td></tr><tr><td><b>526.277</b></td><td><b>Craig Breedlove</b></td><td><b>Spirit of America</b></td><td><b>GE J79</b></td><td><b>10/15/65</b></td></tr><tr><td><b>536.712</b></td><td><b>Art Arfons</b></td><td><b>Green Monster</b></td><td><b>GE J79</b></td><td><b>10/27/65</b></td></tr><tr><td><b>555.1.27</b></td><td><b>Craig Breedlove</b></td><td><b>Spirit of America, Sonic 1</b></td><td><b>GE J79</b></td><td><b>11/2/65</b></td></tr><tr><td><b>576.553</b></td><td><b>Art Arfons</b></td><td><b>Greer, Monster</b></td><td><b>GE J79</b></td><td><b>11/7/65</b></td></tr><tr><td><b>600.601</b></td><td><b>C<sub>Traig</sub> Breqdlove</b></td><td><b>Spirit of America, Sonic 1</b></td><td><b>GE J79</b></td><td><b>1.1/15/65</b></td></tr><tr><td><b>622.407</b></td><td><b>Gary Galoelich</b></td><td><b>Blue Flame</b></td><td><b>Rocket</b></td><td><b>10/23/70</b></td></tr><tr><td><b>633.468</b></td><td><b>Richard Noble</b></td><td><b>Thrust 2</b></td><td><b>RR RG 146</b></td><td><b>10/4/83</b></td></tr><tr><td><b>763.035</b></td><td><b>Antly Green</b></td><td><b>Thrust SSC</b></td><td><b>RR Spey</b></td><td><b>10/15/97</b></td></tr></tbody></table>
```

output image after:

```
<!DOCTYPE html>
<html>
<style>
table, th, td {
  border: 1px solid black;
}
</style>
<body>
```

```
<table><thead><tr><td><b>Crystal</b></td><td>
<b>Type</b></td><td><b>Melting temp.</b></td><td>
<b>Notes</b></td></tr></thead><tbody><tr><td><b>I</b></td>
<b><u>O</u></b></td><td><b>17 °C (63 °F)</b></td>
<b><u>Soft, crumbly, melts too easily</u></b></td></tr>
<tr><td><b>II</b></td><b><u>TTQWD</u></b></td><td><b>21 °C (70 °F)</b></td>
<b><u>Soft, crumbly, melts too easily</u></b></td></tr>
<tr><td><b>III</b></td><b><u>T HR</u></b></td><td><b>26 °C (79 °F)</b></td>
<b><u>Firm, (OOH Sizz, melts too @SI)</u></b></td></tr>
<tr><td><b>IV</b></td><b><u>FR</u></b></td><td><b>28 °C (82 °F)</b></td>
<b><u>Firm, QOd SinAp, melts too ezsily</u></b></td></tr>
<tr><td><b>V</b></td><b><u>FR</u></b></td><td><b>34 °C (93 °F)</b></td>
<b><u>GloSsy, firm, best snap, melts near body (emper)dblure (37 °C)</u></b></td></tr>
<tr><td><b>VI</b></td><b><u>SDI</u></b></td><td><b>36 °C (97 °F)</b></td>
<b><u>Hard, takes weeks to form</u></b></td></tr></tbody></table>
```

```
</body>
</html>
```

Crystal	Type	Melting temp.	Notes
I	O	17 °C (63 °F)	
II	TTQWD	21 °C (70 °F)	Soft, crumbly, melts too easily
III	T HR	26 °C (79 °F)28 °C (82 °F)	Firm, (OOH Sizz, melts too @SI)
IV	FR		Firm, QOd SinAp, melts too ezsily
V	FR	34 °C (93 °F)	GloSsy, firm, best snap, melts near body (emper)dblure (37 °C)
VI	SDI	36 °C (97 °F)	

Crystal	Type	Melting temp.	Notes
I	O N E	17 °C (63 °F)	Soft, crumbly, melts too easily
II	T W O	21 °C (70 °F)	Soft, crumbly, melts too easily
III	T H R E E	26 °C (79 °F) 28 °C (82 °F)	Firm, poor snap, melts too easily
IV	F O U R		Firm, good snap, melts too easily
V	F I V E	34 °C (93 °F)	Glossy, firm, best snap, melts near body temperature (37 °C) Hard, takes weeks to form
VI	S I X	36 °C (97 °F)	

output image after:

parameter	Squamous cell carcinoma (n = 229)	Adenocarcinoma (n = 150)	p-value
-Surgical complications (%)			
-General complications (%)	37.9	33.7	0.447
-Total complications (%)	40.7	28.0	0.012*
-Length of ICU – stay (d)	68.1	57.6	0.039*
-30-day-mortality (%)			
-Mortality (%)	11 (2–176)	8 (1–107)	0.0001*
	6.1	6.1	0.8569
	15.1	6.6	0.0001*

Speed (mph)	Driver	Car	Engine	Date
407.447	Craig Breedlove	Spirit of America	GE J47	8/5/63
413.199	Tom Green	Wingfoot Express	WE J46	10/2/64
434.22	Art Arfons	Green Monster	GE J79	10/5/64
468.719	Craig Breedlove	Spirit of America	GE J79	10/13/64
526.277	Craig Breedlove	Spirit of America	GE J79	10/15/65
536.712	Art Arfons	Green Monster	GE J79	10/27/65
555.127	Craig Breedlove	Spirit of America, Sonic 1	GE J79	11/2/65
576.553	Art Arfons	Green Monster	GE J79	11/7/65
600.601	Craig Breedlove	Spirit of America, Sonic 1	GE J79	11/15/65
622.407	Gary Gabelich	Blue Flame	Rocket	10/23/70
633.468	Richard Noble	Thrust 2	RR RG 146	10/4/83
763.035	Andy Green	Thrust SSC	RR Spey	10/15/97

Topics	Days for Class	Timing
Design For Ux	Monday	8:30 – 10:30 am
Design Thinking		12:30 – 2:30 pm
Empathy Map	Tuesday	9:30 – 11:30 pm
Emotional Intelligence	Wednesday	
Usability	Thursday	8:30 – 10:30 am
Utilitv	Fridav	8:30 – 10:30 am

parameter	Squamous cell carcinoma (n = 229)	Adenocarcinoma (n = 150)	p-value
-Surgical complications (%)			
-General complications (%)	37.9	33.7	0.447
-Total complications (%)	40.7	28.0	0.012*
-Length of ICU – stay (d)	68.1	57.6	0.039*
-30-day-mortality (%)			
-Mortality (%)	11 (2–176)	8 (1–107)	0.0001*
	6.1	6.1	0.8569
	15.1	6.6	0.0001*

Speed (mph)	Driver	Car	Engine	Date
407.447	Craig Breedlove	Spirit of America	GE J47	8/5/63
413.199	Tom Green	Wingfoot Express	WE J46	10/2/64
434.22	Art Arfons	Green Monster	GE J79	10/5/64
468.719	Craig Breedlove	Spirit of America	GE J79	10/13/64
526.277	Craig Breedlove	Spirit of America	GE J79	10/15/65
536.712	Art Arfons	Green Monster	GE J79	10/27/65
555.127	Craig Breedlove	Spirit of America, Sonic 1	GE J79	11/2/65
576.553	Art Arfons	Green Monster	GE J79	11/7/65
600.601	Craig Breedlove	Spirit of America, Sonic 1	GE J79	11/15/65
622.407	Gary Gabelich	Blue Flame	Rocket	10/23/70
633.468	Richard Noble	Thrust 2	RR RG 146	10/4/83
763.035	Andy Green	Thrust SSC	RR Spey	10/15/97

Topics	Days for Class	Timing
Design For Ux	Monday	8:30 – 10:30 am
Design Thinking		12:30 ± 2:30 pm
Empathy Map	Tuesday	
Emotional Intelligence	Wednesday	Q:30 = 11:30 gm
Usability	Thursday	8:30 – 10:30 am
Utility	Friday	8:30 = 10:30 am
Accessibility	Friday	8:30 – 10:30 am

Changes made in MTL-TabNet and Layout Parser Code

- ➊ MTL-TabNet has been enhanced to **process multiple images** by directly reading them from a **folder** in a **sequential manner**. This modification **improves efficiency** when dealing with a large number of images.
- ➋ Removal of unnecessary code like taking input images several times has been removed and edited the code of **MTL-Tabnet to make it more generalised**
- ➌ Removal of unnecessary code in **Layout parser** and used only for Layout analysis and **extract tables from a PDF** file. It detects the layout, **crops the images around the table** blocks, and **saves** them. It generates JSON data with table coordinates and writes it to a file
- ➍ **JSON data generation** has been **integrated** into the Layout parser, producing a file that contains the **coordinates of the extracted tables**. This information serves as a **reference** for further processing and **integration into the final output**.
- ➎ The updated code in both MTL-TabNet and the Layout parser removes redundancy and unnecessary steps ,These changes contribute to improved **overall performance** and code maintainability.

Connecting Layout Parser to MTL Tabnet

① Table Detection with Layout Parser

- Utilizes Layout Parser to detect tables within a document.
- Provides cropped images of table parts for further processing.
- Efficiently identifies table regions and extracts them.

② Automated Whole PDF Processing:

- Enables processing of entire PDF documents.
- Detects all tables present within the document. Allows for bulk processing and saves manual effort.
- Extracted tables are saved as cropped images in a designated folder.

③ Sending Cropped Images to MTL-Tabnet:

- ④ Cropped images obtained from Layout Parser are sent as input to MTL-Tabnet and MTL-Tabnet extracts HTML table element with order of tables
- ### Benefits

- ⑤ Enhanced table recognition accuracy with combined capabilities
- ⑥ processing of **multiple tables** within a document or **bulk pdf is easy**

Overall Output image after integration:

```

[1] ravi@ravi-Inspiron-5570: ~/D...  Q  E  -  O  X
=>rectangle(x_1=894.3217163085938, y_1=1486.537109375, x_2=1486.603759765025, y_2=1765.2281494140625), text=Non
e, l_id=None, type=Title, parent=None, next=None, score=0.208423304191890]], page_data({})
table_blockssss Layout(blocks=[TextBlock(blocks=Rec
tangle(x_1=193.32640075683594, y_1=1306.179443359375, x_2=807.1260986328125, y_2=1649.8363037109375), text=Non
e, l_id=None, type=Table, parent=None, next=None, score=0.819728434085846)], page_data({})
Shape of the actual image <PIL.PpmImagePlugin.PpmImageF
ile image mode=RGB size=1700x2200 at 0x7F874312FBB0>
Image written to file-system : True
table_1: 193.32640075683594 807.1260986328125 1306.17
9443359375 1649.8363037109375
f'inaa111J50NMN : [{"page": "1", "tableNumber": "1", "
tableData": {"x1": 193.32640075683594, "x2": 807.126098
6328125, "y1": 1306.179443359375, "y2": 1649.8363037109
375}}]

#page_1_table_1 {
  position: absolute;
  left: 193.32640075683594px;
  top: 1306.179443359375px;
  width: 613.7996978759766px;
  height: 343.6568683515625px;
}

started compiling image.py
completed running image.py subprogram
Use load_from_local_loader
Time taken for Table Structure Prediction: 344.76110774
04022 seconds
Time taken for Visualization: 344.7056927680969 seconds
<table><thead><tr><td><b>Month</b></td><td><b>Sale</b></td><td><b>Expenditure</b></td></tr><tbody><tr><td><b>Jan</b></td><td><b>6743</b></td><td><b>6764</b></td></tr><tr><td><b>Feb</b></td><td><b>3211</b></td><td><b>3234</b></td></tr><tr><td><b>Mar</b></td><td><b>236</b></td><td><b>234</b></td></tr><tr><td><b>Apr</b></td><td><b>8790</b></td><td><b>8743</b></td></tr><tr><td><b>May</b></td><td><b>654</b></td><td><b>654</b></td></tr><tr><td><b>Jun</b></td><td><b>324</b></td><td><b>324</b></td></tr><tr><td><b>Jul</b></td><td><b>8743</b></td><td><b>8743</b></td></tr><tr><td><b>Aug</b></td><td><b>324</b></td><td><b>324</b></td></tr><tr><td><b>Sep</b></td><td><b>8743</b></td><td><b>8743</b></td></tr><tr><td><b>Oct</b></td><td><b>324</b></td><td><b>324</b></td></tr><tr><td><b>Nov</b></td><td><b>8743</b></td><td><b>8743</b></td></tr><tr><td><b>Dec</b></td><td><b>324</b></td><td><b>324</b></td></tr></tbody></table>
Mapping updated successfully in output2.json.
(hack) ravi@ravi-Inspiron-5570: ~/Desktop/layo... Tabnet/NTL

[1] ravi@ravi-Inspiron-5570: ~/Desktop/layo... Tabnet/NTL
output.json [Read-Only]
~/Desktop/layo... Parse/cropped Save E - O X
< table_inference.py > inference.py > output.json > output.json >
1 [{"page": "1", "tableNumber": "1", "tableData": {"x1":
193.32640075683594, "x2": 807.1260986328125, "y1": 1306.179443359375,
"y2": 1649.8363037109375}}]

[1] ravi@ravi-Inspiron-5570: ~/Desktop/layo... Tabnet/NTL
output2.json [Read-Only]
~/Desktop/NTL_Subnet/NTL_Tabnet Save E - O X
1 [
2 {
3   "page": "1",
4   "tableNumber": "1",
5   "tableData": {
6     "x1": 193.32640075683594,
7     "x2": 807.1260986328125,
8     "y1": 1306.179443359375,
9     "y2": 1649.8363037109375
10  },
11   "htmlTable": "<table><thead><tr><td><b>Month</b></td><td><b>Sale</b></td><td><b>Expenditure</b></td></tr><tbody><tr><td><b>Jan</b></td><td><b>6743</b></td><td><b>6764</b></td></tr><tr><td><b>Feb</b></td><td><b>3211</b></td><td><b>3234</b></td></tr><tr><td><b>Mar</b></td><td><b>236</b></td><td><b>234</b></td></tr><tr><td><b>Apr</b></td><td><b>8790</b></td><td><b>8743</b></td></tr><tr><td><b>May</b></td><td><b>654</b></td><td><b>654</b></td></tr><tr><td><b>Jun</b></td><td><b>324</b></td><td><b>324</b></td></tr><tr><td><b>Jul</b></td><td><b>8743</b></td><td><b>8743</b></td></tr><tr><td><b>Aug</b></td><td><b>324</b></td><td><b>324</b></td></tr><tr><td><b>Sep</b></td><td><b>8743</b></td><td><b>8743</b></td></tr><tr><td><b>Oct</b></td><td><b>324</b></td><td><b>324</b></td></tr><tr><td><b>Nov</b></td><td><b>8743</b></td><td><b>8743</b></td></tr><tr><td><b>Dec</b></td><td><b>324</b></td><td><b>324</b></td></tr></tbody></table>"
12 }
13 ]

JSON Tab Width: 8 Ln 12, Col 6 INS
```

Show Live Terminal output

Future work

- ① Integration of Borderless table work to RAVI output HTML
- ② To identify where to place the table in the output HTML .