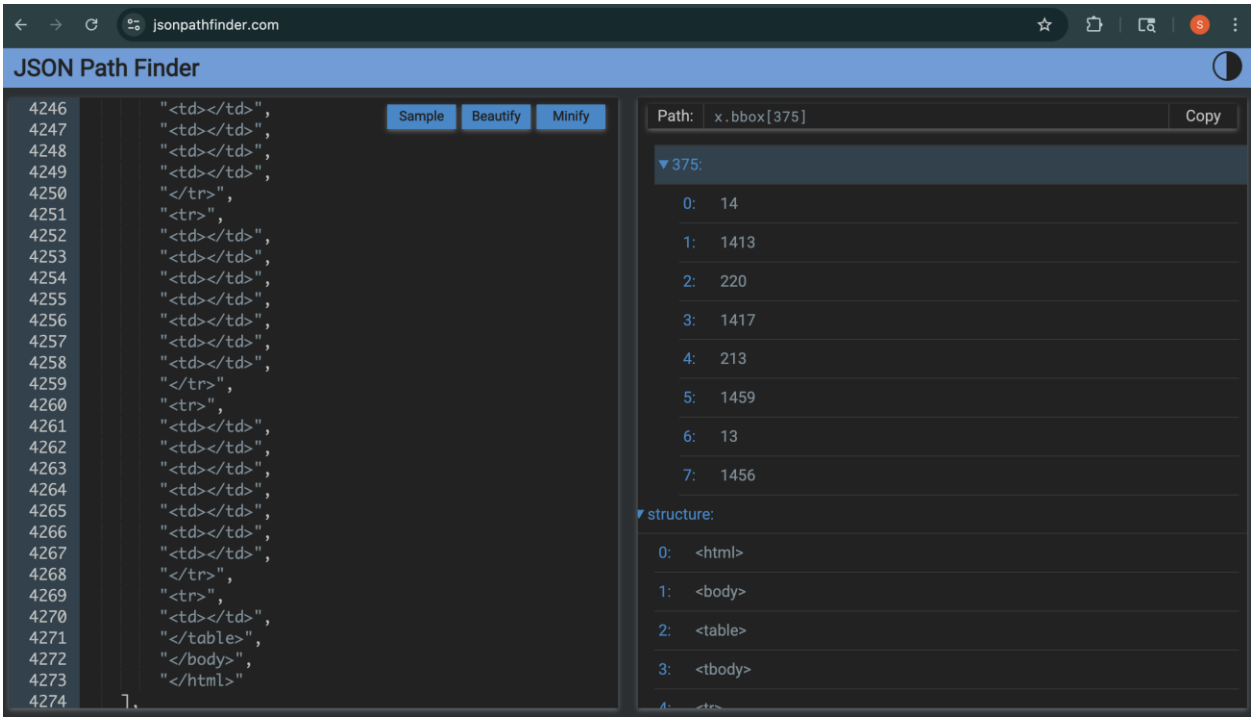
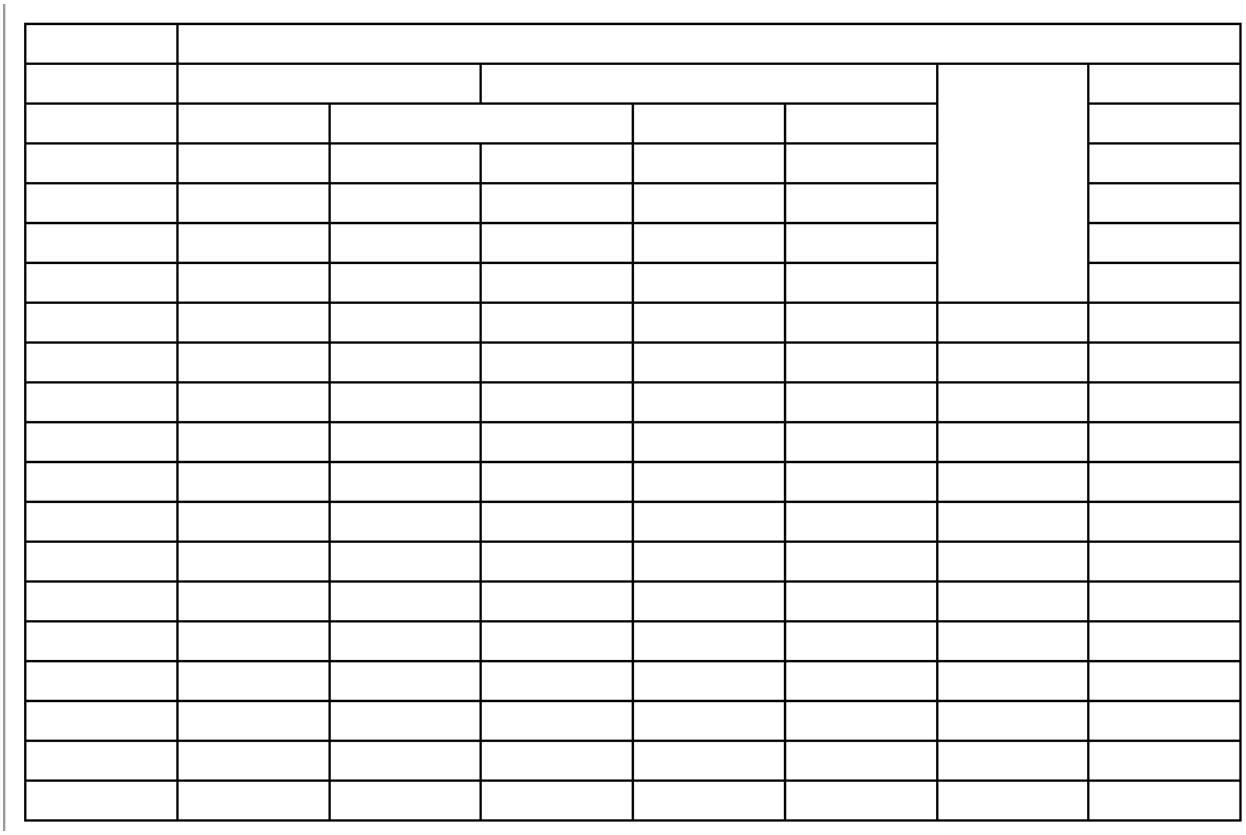
Weekly Work Report

1. URL Testing - PaddleOCR Table Structure Recognition:

This week, I tested the table structure recognition feature of PaddleOCR using the URL provided ("https://www.paddleocr.ai/main/en/version3.x/module_usage/table_structure_recognition.html"). The tool returns coordinates and an HTML representation of the detected table structure.



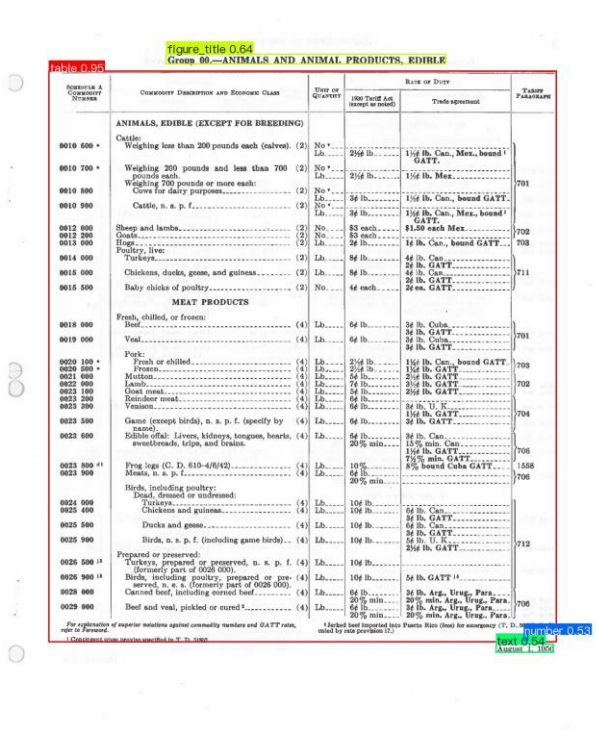
I attempted to map the HTML output, and here's a preview of the result:

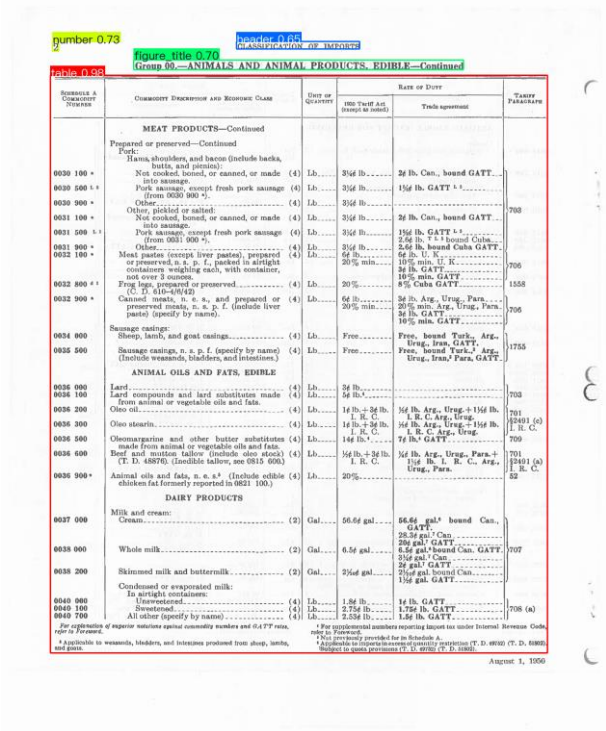


Upon comparing the results with the data from page 28 of the document, I found discrepancies in the cell count. The number of cells detected by PaddleOCR did not match the expected count, highlighting a potential issue with its table structure recognition. This misalignment suggests that the tool is either incorrectly identifying or misrepresenting the table structure in certain cases.

2. Layout Detection - PaddleOCR:

I also explored the layout detection feature of PaddleOCR, which helps identify the general structure of a document. However, due to the varying table sizes in the images, we found that dynamic allocation based on layout detection was not feasible. The tool could not reliably handle the varying dimensions of the tables, which made it unsuitable for our current use case



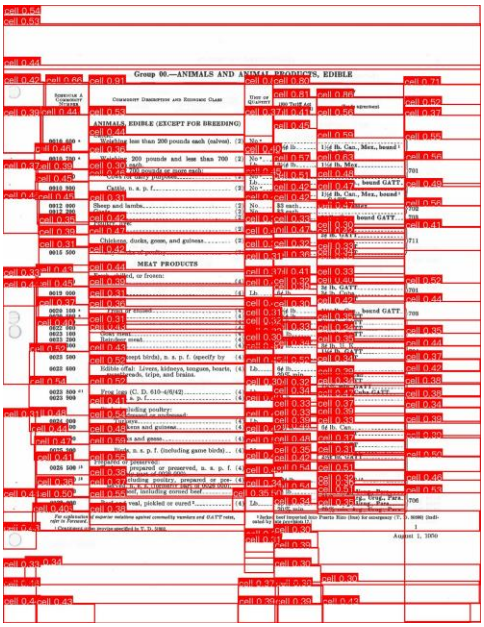


.

.

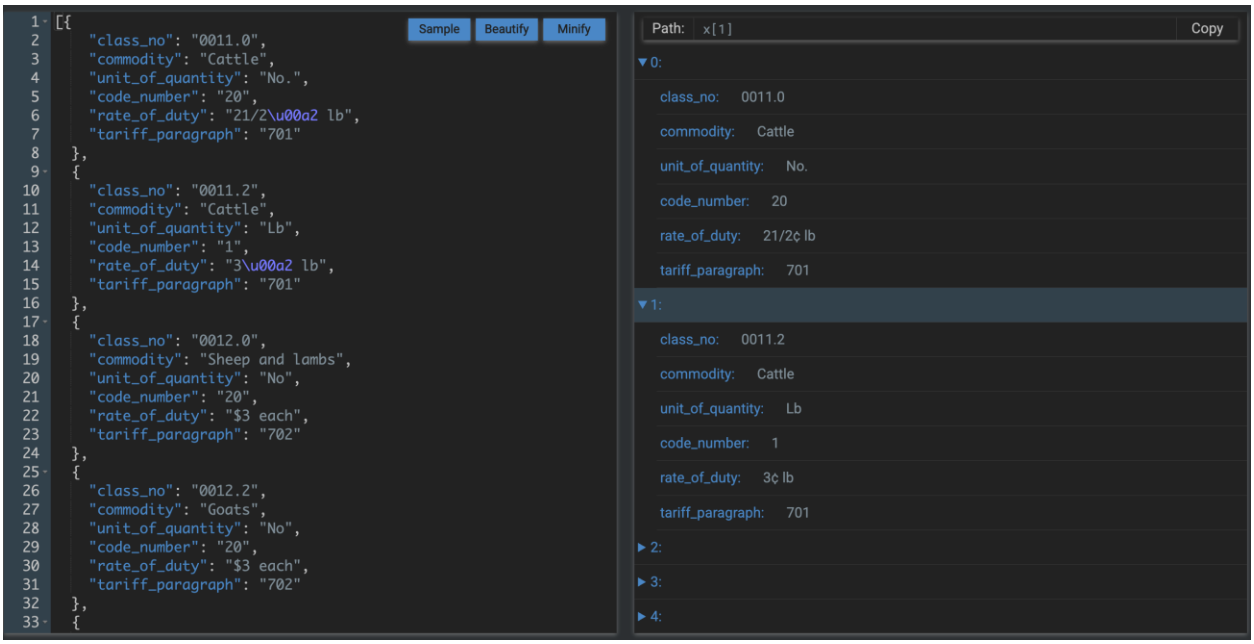
3. Table Cell Detection - PaddleOCR:

I tested the Table Cell Detection feature in PaddleOCR, and while it did provide the cell count, the results were not usable for our purposes. The detection did not align with the expectations we had for accurately identifying individual cells in a table, making it difficult to rely on for further processing.



4. Testing with Sparrow OCR LLM (PDF shared by Kristy):

I proceeded with testing the PDF shared by Kristy using Sparrow OCR LLM. The JSON response received from the OCR was as follows:




```
[{
  "class_no": "0011.0",
  "commodity": "Cattle",
  "unit_of_quantity": "No.",
  "code_number": "20",
  "rate_of_duty": "21/2\u00a2 lb",
  "tariff_paragraph": "701"
 },
 {
  "class_no": "0011.2",
  "commodity": "Cattle",
  "unit_of_quantity": "Lb",
  "code_number": "1",
  "rate_of_duty": "3\u00a2 lb",
  "tariff_paragraph": "701"
 },
 {
  "class_no": "0012.0",
  "commodity": "Sheep and lambs",
  "unit_of_quantity": "No",
  "code_number": "20",
  "rate_of_duty": "$3 each",
  "tariff_paragraph": "702"
 },
 {
  "class_no": "0012.2",
  "commodity": "Goats",
  "unit_of_quantity": "No",
  "code_number": "20",
  "rate_of_duty": "$3 each",
  "tariff_paragraph": "702"
 },
 {
```


This output provides structured data in a JSON format that includes product information like class number, commodity, unit of quantity, code number, rate of duty, and tariff paragraph. Although the data format is promising, additional mapping would be needed to

convert this information into a CSV file for further processing. This is something I plan to explore further.

5. I reached out to Suzie, a student of Kristy, to discuss possible improvements and strategies. Unfortunately, I have not received a response yet, but I will continue to follow up on this for further collaboration.

Next Steps:

- As of now, we are still using the QWEN LLM for OCR tasks. However, Sparrow OCR has recently released a new feature that utilizes local Llama models. I plan to test this new feature in the coming week and provide updates based on its performance. This could potentially offer improvements in accuracy and processing speed.