**Table Extraction from a PDF document**

**Open Source Tools Tried:**

1. **Tabula** (tables detected but not converted to accurate dataframe)
2. **Camelot** (No tables detected)
3. **Unstructured** (tables detected but not converted to accurate dataframe)
4. **PdfPlumber** (No tables detected)
5. **Converting PDF to HTML** (tables detected but not converted to accurate dataframe)

These tools performed very poorly as our pdf had very complicated tables (although simple for our naked eye, for the tools it was as if it was just text)

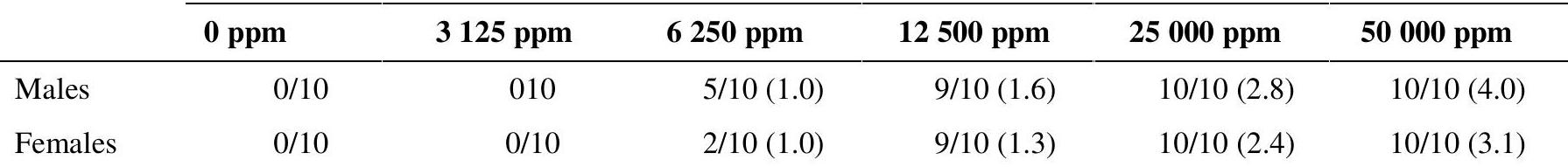
Hence, I had to employ more complex solutions and came up with the following algorithm:

**High Level Algorithm:**

1. Convert each page in the pdf to an image.
2. Use a YOLOV8 model trained on borderless table images to detect tables from each page image.
3. Extract those bounding boxes into snippets of images so we have a folder containing only images of tables accurately cropped out.
4. Use GPT4o vision to convert the table snippets into a **structured text format.**

**e.g.**

**Input Image from pdf:**

****

**Output from GPT4o:**

0 ppm | 3125 ppm | 6250 ppm | 12500 ppm | 25000 ppm | 50000 ppm

--- | --- | --- | --- | --- | ---

Males | 0/10 | 0/10 | 5/10 (1.0) | 9/10 (1.6) | 10/10 (2.8) | 10/10 (4.0)

Females | 0/10 | 0/10 | 2/10 (1.0) | 9/10 (1.3) | 10/10 (2.4) | 10/10 (3.1)

**More Technical details:**

1. Convert each page in the pdf to an image.
   1. We use a python library called **pdf2image** [1]
2. Use a **YOLOV8** [2] model trained on borderless table images to detect tables from each page image.

YOLOv8, or “You Only Look Once” version 8, is a state-of-the-art real-time object detection algorithm developed by Ultralytics. It builds upon the advancements of previous YOLO versions to offer enhanced accuracy, speed, and efficiency.

**Key Features of YOLOv8**:

**High Accuracy**: YOLOv8 consistently outperforms other object detection models on popular benchmarks like COCO.

**Speed**: As a single-stage detector, YOLOv8 predicts bounding boxes and class probabilities in a single forward pass, making it ideal for real-time applications.

**Advanced Architecture**: It employs a hybrid design combining convolutional neural networks (CNNs) with spatial attention mechanisms, improving feature extraction and object detection performance.

**Versatility**: YOLOv8 supports various tasks including object detection, instance segmentation, pose/keypoints detection, oriented object detection, and classification.

**Ease of Use**: It offers a range of pre-trained models and is compatible with frameworks like Darknet and PyTorch, making it accessible for different use cases.

1. Extract those bounding boxes into snippets of images so we have a folder containing only images of tables accurately cropped out.
   1. This is done with the use of a library called **pillow** [4]
2. Use GPT4o vision to convert the table snippets into a **structured text format**
   1. We first encode the image into a python library called **base64** and pass it to the llm which is define using langchain’s wrapper for Azure Open AI, called **AzureChatOpenAI**.
   2. The following prompt was used:
3. """
4. I have a table snippet that I need to convert to text.
5. Output Format:
6. Column 1 | Column 2 | Column 3
7. Data 1 | Data 2 | Data 3
8. etc
9. Output ONLY the table with no additional comments.
10. """

**References:**

[1] Belval, A. (2024). pdf2image (Version 1.16.3) [Computer software].

<https://pypi.org/project/pdf2image/>

[2] Ultralytics. (2023). YOLOv8: Real-Time Object Detection. Retrieved

from <https://github.com/ultralytics/ultralytics>

[3] Keremberke. (2023). YOLOv8m Table Extraction. Retrieved from

<https://huggingface.co/keremberke/yolov8m-table-extraction>

[4] Clark, A. (2015). Pillow (PIL Fork) Documentation. Retrieved from

https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf