

SMLE-1 Assignment Deliverable

1. Code Submission

All the code for data preprocessing, model training, inference pipeline, and evaluation has been implemented in a structured Kaggle notebook. The notebook runs end-to-end without requiring any changes and includes all outputs for reproducibility. The Kaggle notebook URL is shared separately as part of the submission.

2. Inference Pipeline

An inference pipeline was built to allow users to input a PDF file or URL. Each page is converted into an image using PyMuPDF and passed through the trained model to detect table bounding boxes. The output returns page-wise table locations, enabling easy integration into real-world document workflows.

3. Time Taken

The complete solution took approximately 7 to 8 hours. This included understanding the dataset, building the training pipeline on Kaggle, developing inference functions, evaluating performance, and debugging practical issues such as memory crashes, storage limits, and GPU configuration problems.

4. Solution Overview

The PubTables-1M dataset was used for training the model. XML annotations were parsed and converted into COCO-style bounding boxes. A custom PyTorch Dataset and DataLoader were created to handle image loading and annotation formatting. A transformer-based object detection model was fine-tuned to detect tables in document images. The pipeline includes training, inference on both images and PDFs, and quantitative evaluation on a test dataset.

5. Model Used

Microsoft's Table Transformer (table-transformer-detection) model was used. It was selected because it is specifically designed for table detection in documents, uses transformer-based object detection for strong spatial understanding, and comes pretrained on large document datasets.

6. Challenges and Improvements

Major challenges included RAM crashes due to the large dataset size, Kaggle storage limitations while extracting image archives, and instability caused by multiple GPU setups which required forcing single-GPU training. Future improvements include training on larger portions of the dataset, applying data augmentation, fine-tuning for longer durations, and adding post-processing to refine overlapping detections.

7. Model Performance

Using the full test dataset of 2029 images, the model achieved a Mean IoU of 0.7151, Precision at IoU 0.50 of 0.7802, Precision at IoU 0.75 of 0.6540, and an average inference latency of 0.0979 seconds per image. A total of 2835 ground truth samples were evaluated.

8. Metric Choice

IoU-based metrics were chosen as they are the standard evaluation method for object detection tasks. Precision at different IoU thresholds reflects both general detection accuracy and strict localization quality. Latency was included to measure real-world usability of the pipeline.

9. Kaggle Notebook

The full implementation is available in a Kaggle notebook structured to run from start to end without code modifications. All outputs are preserved as required. The notebook link is shared separately.

Final Summary

A complete table detection pipeline was successfully developed using a transformer-based model. The solution includes training, PDF inference, and evaluation. Despite practical challenges related to memory, storage, and GPU configuration, a stable and effective system was built for detecting tables in PDF documents.