TABLESNAP

Table Detection

ASHLEY BREA, ASHLEY KIM, DIANE PARK

OVERVIEW

- *Fine-tuned a pre-trained YOLOv5 model to detect tables within our dataset*
- *Starting from baseline, extended training with adjusted hyperparameters to improve performance*
- *We evaluated the model using precision, recall, and F1 scores*

PREVIOUS SOLUTIONS

Deep learning for table detection and structure recognition: A survey *(Kasem et al., 2022)*

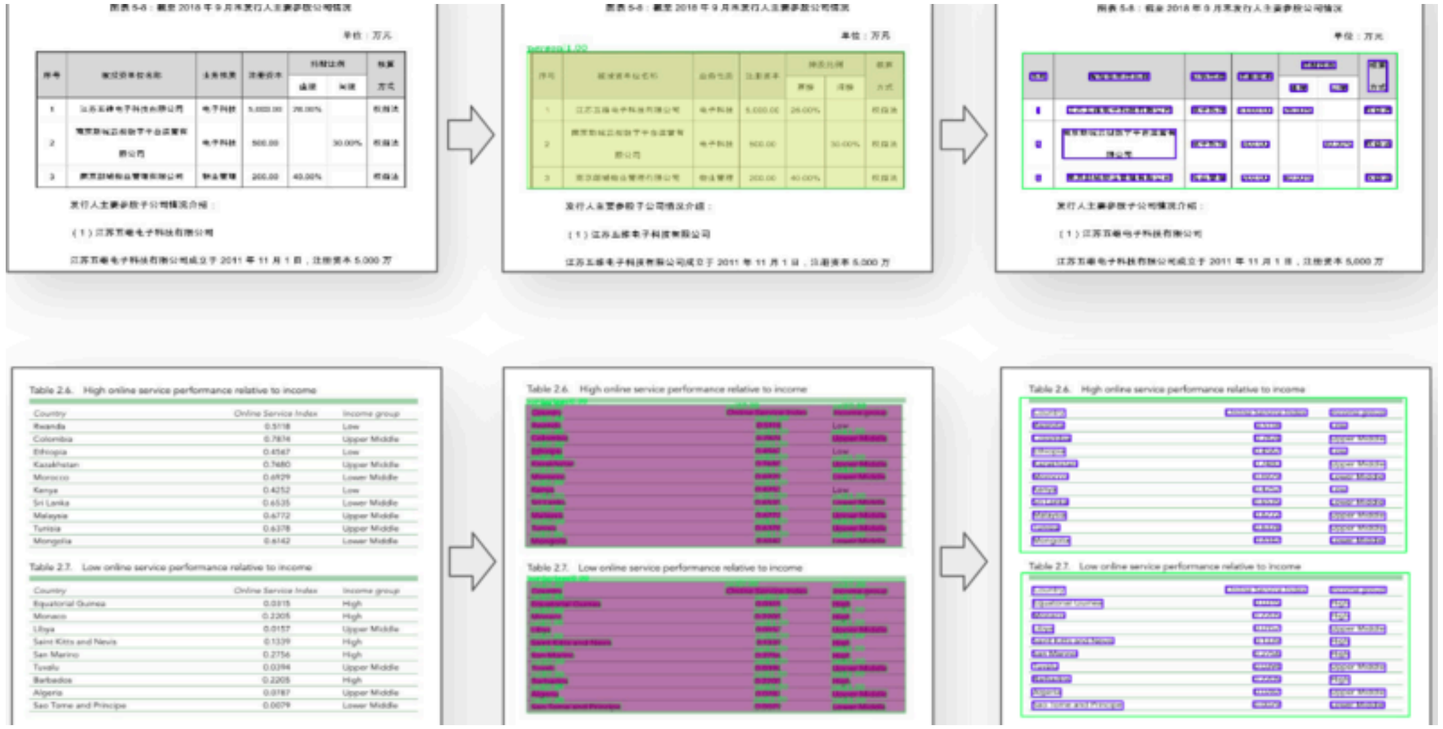
Prasad et al. *proposed* ***CascadeTabNet****, a R-CNN that identifies table areas along with individual table cell structures.*

Siddiqui et al. *created a network, a combined R-CNN and FPN that works on photos instead of just PDFs.*

ORIGINAL PLAN

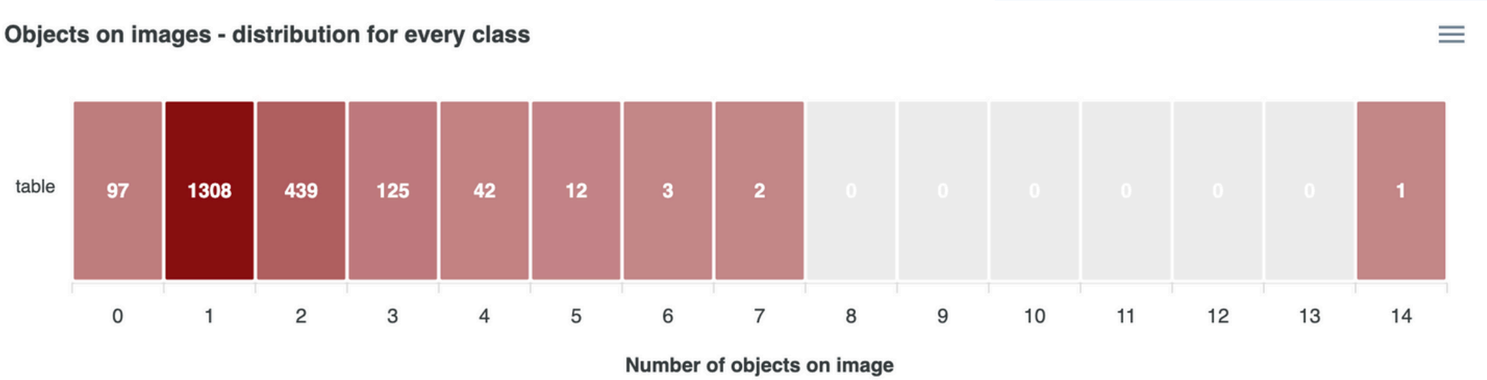
- *Table detection → Structure recognition (cols/rows) → Extracting info from each cell*
- *Ambitious initial goals → adjusted for a realistic timeline*
- *Wanted to ensure table detection was a solid starting model*

GENERAL TABLE DETECTION DATASET



DATA EXPLORATION

- A total of 2029 samples consisting of mixed file types, e.g. PNG, JPG, etc.



- However we only want to use the samples with 1 table per page, which reduces the set to 1308 samples

- The dataset is compiled of the train_annotated.csv and train_folds.csv, and train.csv

DATA PREPROCESSING

- *General Table Detection Dataset (1308 samples)*
- *X: Images*
- *Y: Bounding-Box values (x_min, y_min, bbox_width, bbox_height)*
  - *values are rescaled in terms of image width and height*
- *Split data*
  - *Training: 70%, Validation: 20%, Testing: 10%*
- *Created specific file pathing to be compatible with YOLOv5s*

BASELINE MODEL

- *YOLOv5s model → pre-trained (our dataset is not large enough)*
  - *Fast, accurate, and built-in support for image resizing, normalization, and padding.*
- *Trained on our preprocessed dataset with 1 table per image (YOLO format).*
- *The baseline model was trained for 10 epochs using this data and achieved reasonable performance in detecting tables.*

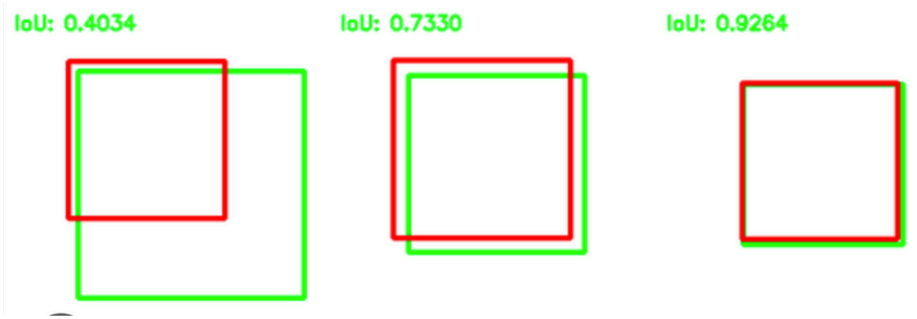
DEEP LEARNING MODEL

- *Applied transfer learning by fine-tuning the YOLOv5s model.*
- *Initialized training with the best weights from the baseline*
- *Trained model for 30 epochs with a lower learning rate*
    - *goal was to increase the model's sensitivity to table-specific patterns (e.g., ruled lines, spacing) while also avoiding overfitting.*

EVALUATION

Comparing Ground Truth Bounding box to Predicted Bounding Box.
Values are calculated with a IOU threshold of 0.5 and 0.9

- Precision
- Recall
- F1 score



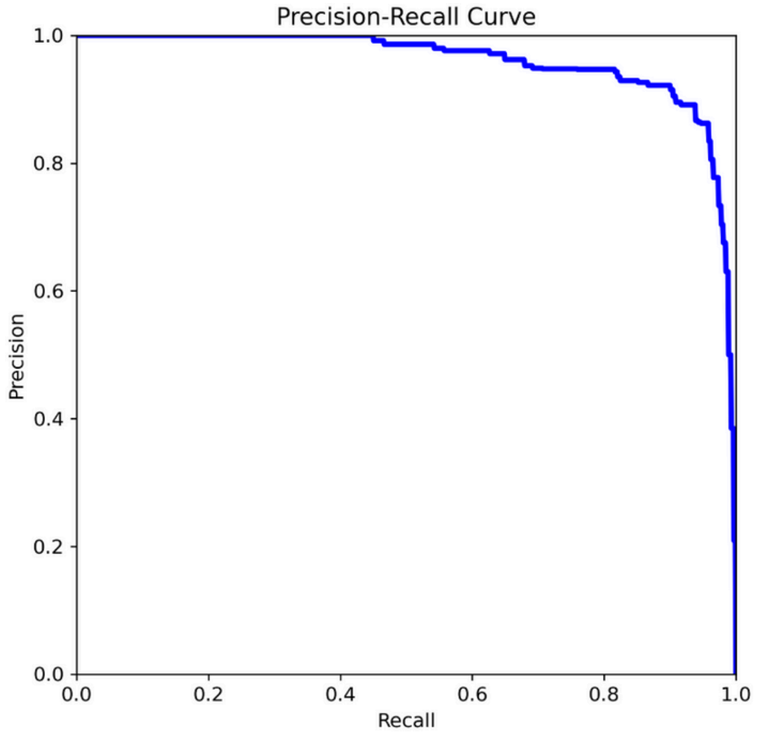
BASELINE MODEL

With 0.5 IOU Threshold:

Precision: 0.8820

Recall: 0.9240

F1 Score: 0.9025



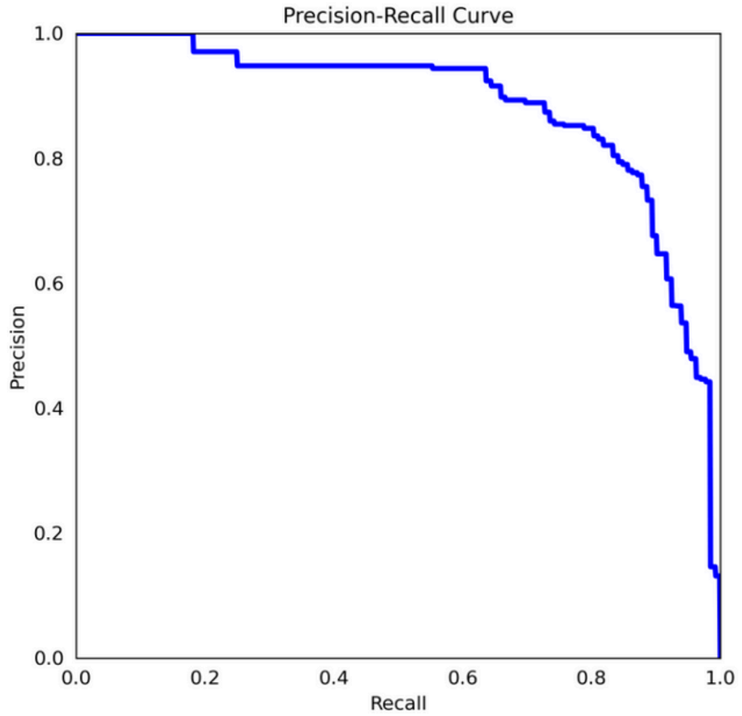
BASELINE MODEL

With 0.9 IOU Threshold:

Precision: 0.7900

Recall: 0.8480

F1 Score: 0.8180



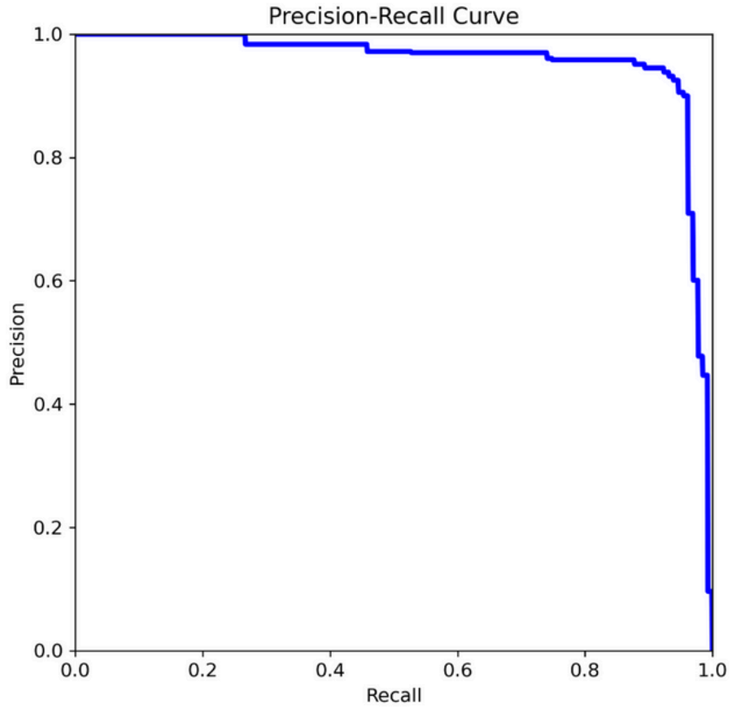
DEEP LEARNING MODEL

With 0.5 IOU Threshold

Precision: 0.9260

Recall: 0.9690

F1 Score: 0.9470



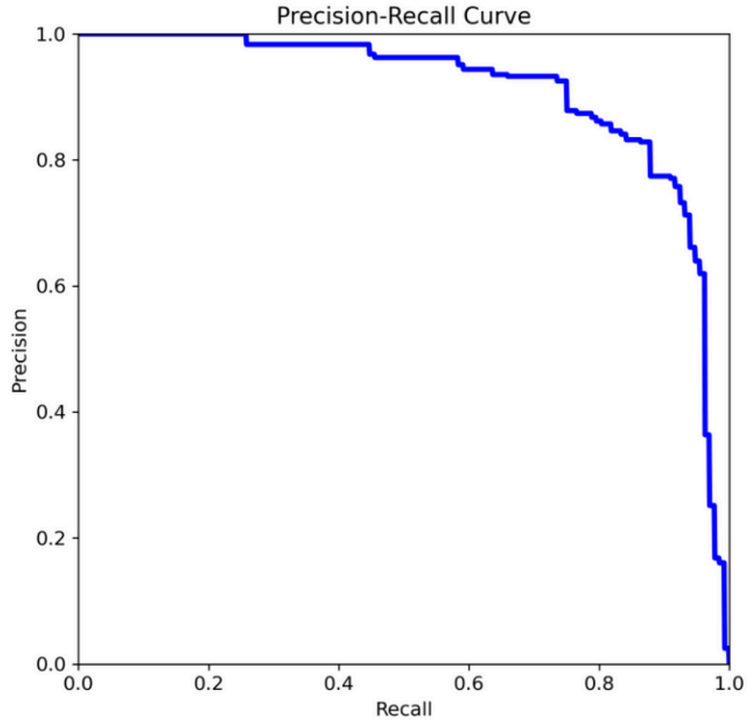
DEEP LEARNING MODEL

With 0.9 IOU Threshold

Precision: 0.8280

Recall: 0.8410

F1 Score: 0.8344



CHALLENGES

- Identifying tables with no borders
- No dataset to fit our original task (hand-written tables)
- Detecting multiple tables within an image
- Adjusted scope to fit available datasets (printed tables)

RESULTS & NEXT STEPS

- So Far: Able to discern a table from other text within a document.

- Visualize good and bad bounding boxes and analyze
- Continue with table structure recognition

THANK YOU!

ANY QUESTIONS ?