TABLESNAP

Table Detection

ASHLEY BREA, ASHLEY KIM, DIANE PARK

OVERVIEW

- *Originally wanted to be able to digitize handwritten data tables*
- *To work our way there, we will first focus with table structure and object detection*

Object Detection

Row and Column Detection

Run OCR on the handwriting

Present a fully digitized data table

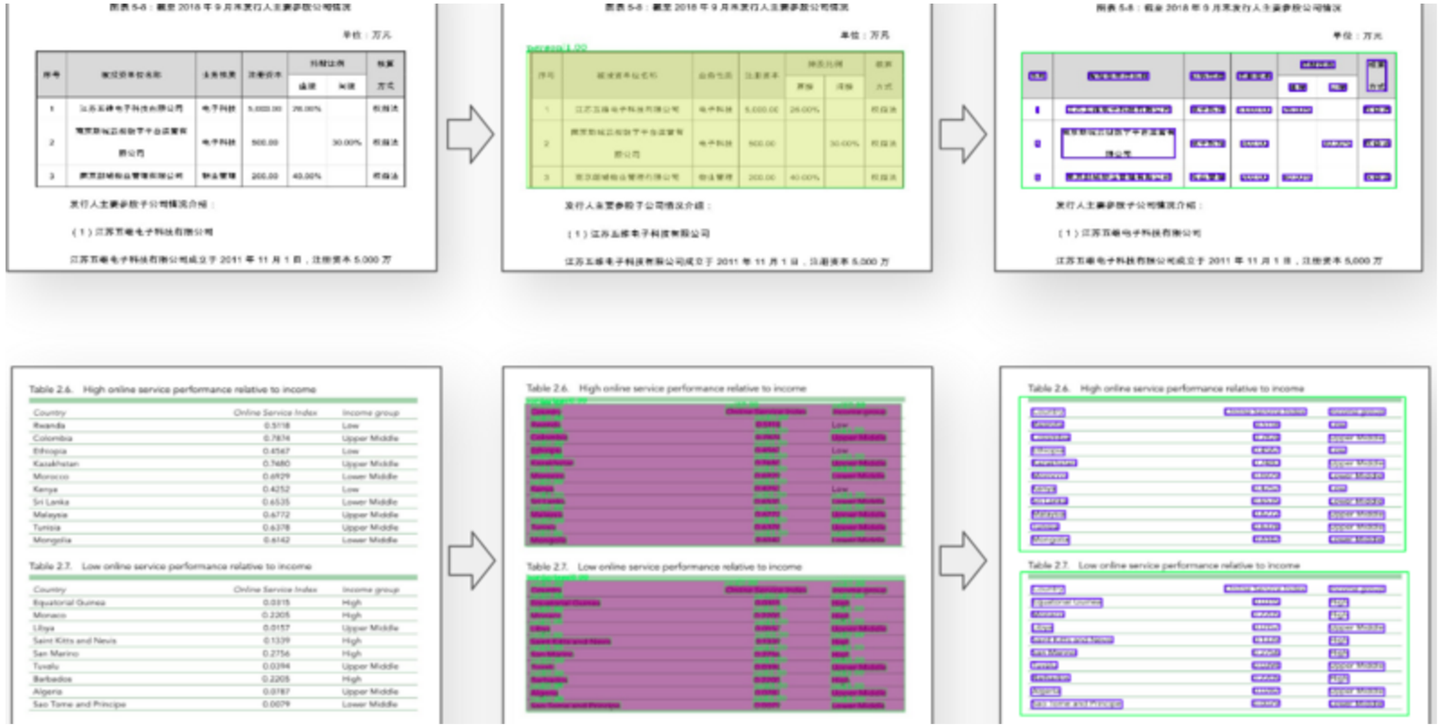
PREVIOUS SOLUTIONS

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.*

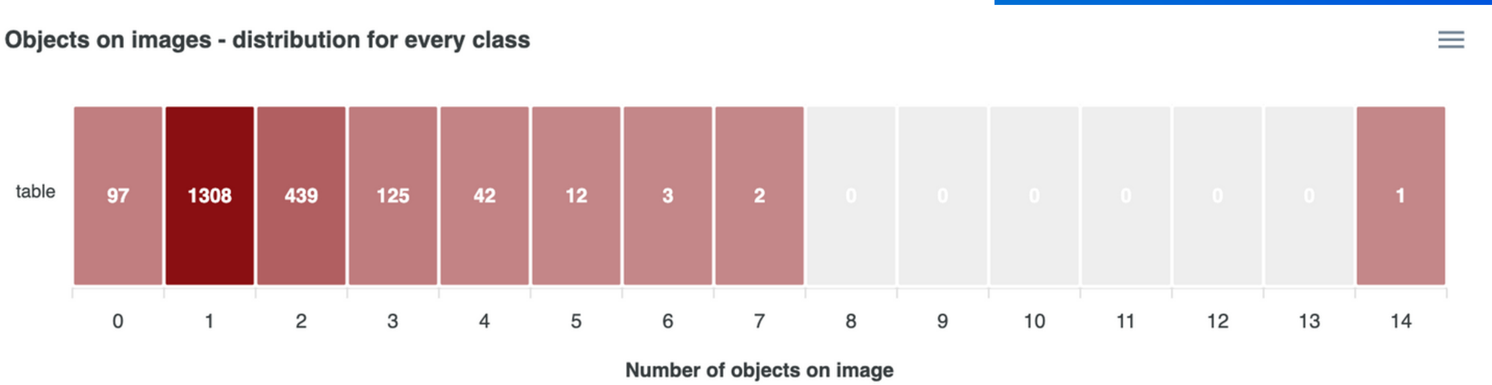
Siddiqui et al. *created a network, a combined R-CNN and FPN that works on photos instead of just PDFs.*

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 examples 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 CLEANSING

- Using train_annotations.csv and train_folds.csv
  - train_annotations.csv: boundingbox values in a 4-tuple
  - train_folds.csv: number or tables found in image
- Only want to focus on images with 1 table
  - Filtered data within train_folds, then within train_annotations
  - Left us with 1308 values for single tables

DATA PREPARATION

- X: Image ID
- Y: Bounding-Box values (x_min, y_min, bbox_width, bbox_height)
- Split Data:
  - Training: 70%
  - Validation: 20%
  - Testing: 10%

CHALLENGES

- Identifying tables with no borders
- Using a small dataset so our training may struggle to converge.
- Different file formats: .jpg and .png files

THANK YOU!

ANY QUESTIONS ?