TableSnap: Table Detection Neural Network

Ashley Kim
Harvey Mudd College
askim@g.hmc.edu

Diane Park
Harvey Mudd College
dipark@g.hmc.edu

Ashley Brea
Vassar College
abrea@vassar.edu

***Abstract*—Table detection in document images is a critical task in document analysis, facilitating the extraction of structured data from unstructured sources. In this project, we focus on detecting tables from images, with the long-term goal of digitizing handwritten data tables from images into CSV format. To achieve this, our current focus is on detecting table structures using object detection techniques.**

Building on prior work such as CascadeTabNet (Prasad et al.) [1] and DeCNT (Siddiqui et al.) [2], we leverage deep learning approaches for table detection. We used the General Table Detection dataset [3], initially training on a filtered subset of 1,308 images that contained a single table per page. After validating our approach, we retrained on the full dataset of 2,835 images, including pages with multiple tables. We preprocessed the data by extracting the bounding-box values and normalizing them relative to the image dimensions. Also, the dataset is split into training (70

We then use YOLOv5 [4] as our baseline model for object detection. This was motivated by YOLO's capacity to pad, resize, and normalize images, as well as its pretrained features. We continue to use YOLOv5 as our deep learning model, where we transfer learning and fine-tune our model by increasing the number of epochs and decreasing the learning rate. We decided to fine-tune our model since we can build on our baseline that has already learned general features from the large dataset of YOLOv5. Additionally, our dataset for table detection is relatively limited in size, making fine-tuning a more suitable approach. Both the baseline model and the deep learning model are evaluated, and we do see a 13.7

As next steps, we aim to extend our model to perform row and column detection and structure recognition, ultimately enabling full table digitization.

I. Introduction

Tabular data plays a critical role in numerous domains such as healthcare, finance, and government, where structured information is often recorded and communicated through tables. A significant portion of this data exists in non-digital formats, including scanned documents and handwritten forms, making it inaccessible for automated processing and analysis. Manual digitization of such tabular content can be labor-intensive and error-prone. To address this, machine learning techniques, particularly in computer vision, have become increasingly relevant due to their ability to learn visual patterns and generalize across diverse document types. Table detection is a natural task in this pipeline, as it involves the localization of table structures within document images. This enables further tasks such as table structure recognition and data extraction. In this project, we aim to develop a deep learning model that detects table structures in document images. We begin with a baseline YOLOv5 object detection model and enhance its performance through fine-tuning on a dataset of digitized tables. Our goal is to automate the identification of table bounding box boundaries with high accuracy, enabling more efficient digitization of tabular data as we continue work on this project.

II. Previous Work

Table detection in documents has been explored through a range of approaches, with recent advances emphasizing deep learning for improved accuracy and generalization. In a comprehensive survey, Kasem et al. reviewed various deep learning methods applied to table detection and structure recognition, highlighting the evolution from rule-based systems to convolutional neural networks (CNNs). Prasad et al. proposed CascadeTabNet, a model built on R-CNN architecture that not only detects table regions but also identifies the structure of individual table cells, making it well-suited for downstream tasks such as cell extraction and semantic parsing. Siddiqui et al. extended this direction by designing a model that combines R-CNN with a Feature Pyramid Network (FPN), enabling it to perform effectively on natural images and photographs, not just scanned PDFs. These works demonstrate the growing success of deep learning models in understanding complex table layouts, motivating the development of our own table detection pipeline using YOLOv5.

III. Dataset

We used the General Table Recognition Dataset from Kaggle, which includes scanned document images and corresponding annotations for table regions in the form of bounding boxes, as well as image width and height details. These annotations define the position and size of each table using the standard format: `(x_min, y_min, width, height)` in pixel values.

To simplify the detection task during initial development, we filtered the dataset to include only images containing a single table. We achieved this by cross-referencing `train_annotated.csv` (which contains bounding box information) and `train_folds.csv` (which specifies the number of tables per image), selecting only those with one table per page. After verifying our method, we expanded to include the full dataset, which contains images with multiple tables per page.

Next, we normalized the bounding box values by image width and height, rescaling the coordinates to fall within the range [0, 1]. This normalization mitigates problems caused by the wide variation in document sizes and ensures stable training performance.

We then split the dataset into training (70%), validation (20%), and test (10%) sets, maintaining consistent label formatting across subsets. Images and labels were organized into the YOLOv5-compatible directory structure with `.txt` files for each image, storing bounding box coordinates in the format: `class_id, x_center, y_center, width, height`. This structure allows for seamless integration with YOLOv5's data loader during training and evaluation.

IV. Method

We adopted the YOLOv5s architecture as our baseline for object detection due to its lightweight design, real-time performance, and integrated handling of image padding and resizing, features particularly beneficial for processing document images of varying dimensions. We cloned the official YOLOv5 repository and initialized the model with pretrained weights ("yolov5s.pt"). For our training, we cross-referenced "image_id" entries from both "train_annotated.csv" and "train_folds.csv" files, and retained only images that were present in both files. This filtering step ensured that all images used for training have corresponding fold metadata, which may be important for future exploration of the data (for example, if users want to examine only a specific number of tables per page). We trained this baseline model for 10 epochs with a batch size of 16 and an image size of 640×640 pixels. The dataset was formatted according to YOLO specifications, including .txt annotation files and a data.yaml file describing paths and class labels.

To enhance model performance, we implemented transfer learning by fine-tuning the baseline model. We resumed training from the best-performing checkpoint of the baseline model, increasing the number of epochs to 30 and reducing the learning rate using a custom YOLOv5 hyperparameter configuration file (hyp.scratch-low.yaml). This strategy allowed the model to retain learned spatial features, while adapting more precisely to the domain of our dataset. Both baseline and fine-tuned weights were exported and saved to a shared drive for reproducibility and evaluation.

We decided to fine-tune our model instead of starting with a new deep learning model because YOLOv5's pretrained weights are already trained on a large and diverse dataset (COCO). This enabled the model to learn generalizable low-level features such as edges, contours, and shapes. Additionally, given the relatively small size of our table detection dataset, fine-tuning allows us to leverage this prior knowledge, reducing training time and the risk of overfitting. Building a new model from scratch would also require significantly more data and computational resources to reach comparable levels of performance. So, fine tuning offered us an effective path to improve detection accuracy, while preserving resource efficiency.

V. Evaluation

We evaluated both models on the test set using standard object detection metrics: Precision, Recall, F1 Score, and mAP@0.5. The models were evaluated by numerical scores using an Intersection over Union (IoU) threshold of 0.9 to assess high-confidence bounding box alignment, and a 0.5 threshold for PR curves. YOLOv5's built-in validation pipeline was used to generate results and visualize Precision-Recall (PR) curves.

The IoU is calculated as the area of the overlap divided by the area of the union between the ground truth and the predicted bounding box. YOLO has a default value of 0.5. In this case, the prediction on the left will be classified as a false positive, since its IoU is lower than the required 0.5 (see Fig. 1). The other predictions are classified as true positives, since their IoU's are higher than 0.5. We decided to use a more strict threshold of 0.9 for table detection in calculating our Precision, Recall, and F1 Score. Then making the rightmost prediction the only true positive from Figure 1, which more precisely classifies a bounding box.

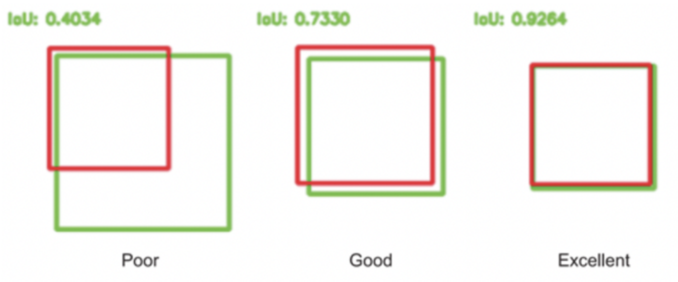


Fig. 1. Illustration of bounding boxes with different IoU values.

The F1 score was computed from the measured Precision and Recall values using the formula:

$$F1 = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

For the baseline model, the evaluation returned a precision of 0.7440, recall of 0.7840, and an F1 score of 0.7635. The generated PR curve showed a relatively stable trade-off between precision and recall, with an mAP@0.5 of 0.803 (see Fig. 2). This is a nice PR curve. High precision is maintained until about 0.8 recall, so our model keeps a precision above 0.8 until it has already recalled about 80% of the true positives. There is also no severe overfitting or weird jaggedness, and our shape is expected. The mAP@0.5 value of 0.803 indicates that the mean average precision at an IoU threshold of 0.5 is 80.3%, which is strong for object detection tasks.

After fine-tuning, the model exhibited substantial improvements across all metrics. Precision increased to 0.8710, recall to 0.8810, and the F1 score reached 0.8760. The PR curve for the fine-tuned model (see Fig. 3) also reflected this improvement, with a higher mAP@0.5 of 0.913, indicating better generalization and tighter box predictions on table data. For this PR curve, the precision remains near 0.9–1.0 until

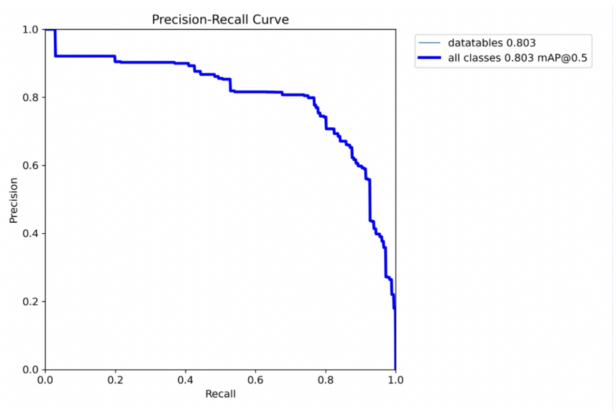


Fig. 2. Precision-Recall (PR) curve for baseline YOLOv5 model (mAP@0.5 = 0.803).

about 0.75 recall. Even after 0.75 recall, precision stays about 0.9 until about 0.9 recall, showing a good balance. Only in the final stretch toward full recall does the precision drop steeply, which is expected. Compared to our baseline results which had mAP@0.5=0.803, our fine tuned model has mAP@0.5=0.913, which is a +13.7% relative improvement in performance. For visual analysis, we selected random images from the test set and overlaid predicted bounding boxes using OpenCV (see Fig. 2 and Fig. 3). This helped verify the quality of predictions beyond numerical metrics and identify common failure cases.

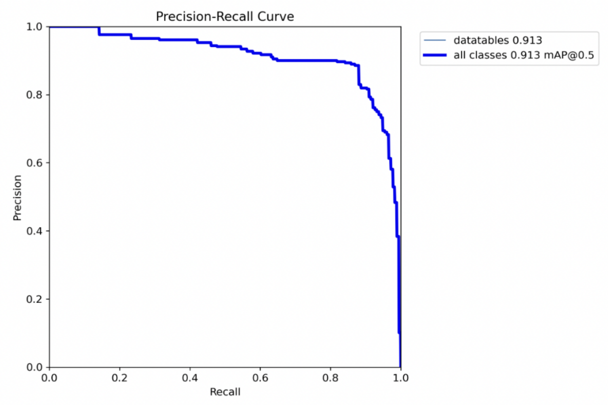
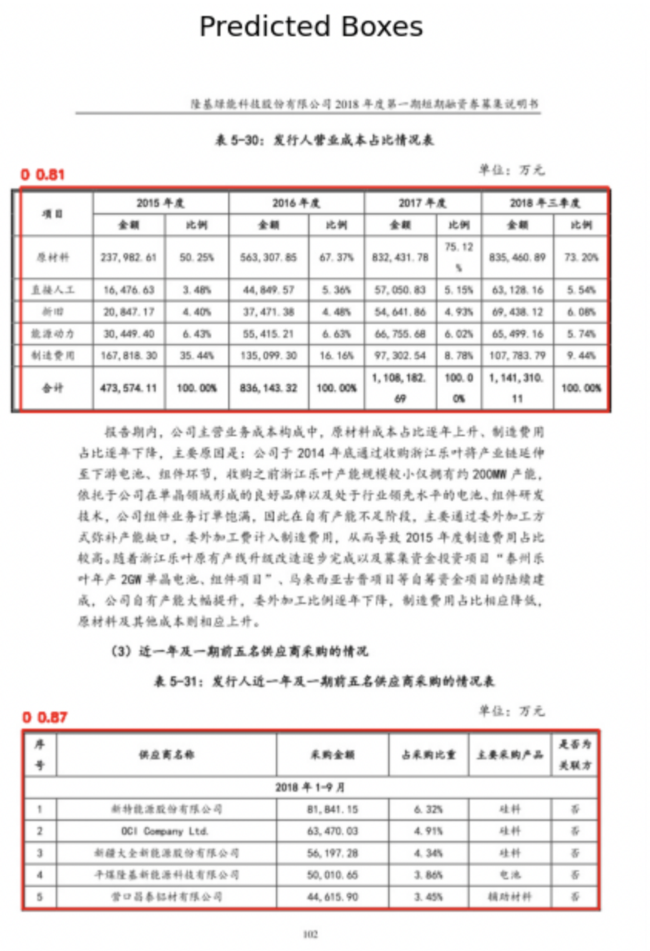


Fig. 3. Precision-Recall (PR) curve for fine-tuned YOLOv5 model (mAP@0.5 = 0.913).

VI. Results

For visual analysis, we randomly selected two images from the test set and overlaid the predicted bounding boxes using OpenCV (see Fig. 4 and Fig. 5). This allowed us to manually inspect prediction accuracy beyond numerical metrics.

For the figures, we see that the red boxes indicate detected table regions and display a confidence level score in red. For Figure 4, we see the detected table regions, with confidence scores of 0.81 and 0.87 respectively. In Figure 5, we see a single table detected region, with confidence score 0.92. We can see that our model correctly identifies and separates



Predicted Boxes

隆基绿能科技股份有限公司 2018 年度第一期短期融资券募集说明书

表 5-30：发行人营业成本占比情况表

单位：万元

0 0.81

项目	2015 年度		2016 年度		2017 年度		2018 年三季度	
	金额	比例	金额	比例	金额	比例	金额	比例
原材料	237, 982. 61	50. 25%	563, 307. 85	67. 37%	832, 431. 78	75. 12 %	835, 460. 89	73. 20%
直接人工	16, 476. 63	3. 48%	44, 849. 57	5. 36%	57, 050. 83	5. 15%	63, 128. 16	5. 54%
折旧	20, 847. 17	4. 40%	37, 471. 38	4. 48%	54, 641. 86	4. 93%	69, 438. 12	6. 08%
能源动力	30, 449. 40	6. 43%	55, 415. 21	6. 63%	66, 755. 68	6. 02%	65, 499. 16	5. 74%
制造费用	167, 818. 30	35. 44%	135, 099. 30	16. 16%	97, 302. 54	8. 78%	107, 783. 79	9. 44%
合计	473, 574. 11	100. 00%	836, 143. 32	100. 00%	1, 108, 182. 69	100. 0 0%	1, 141, 310. 11	100. 00%

报告期内，公司主营业务成本构成中，原材料成本占比逐年上升、制造费用占比逐年下降，主要原因是：公司于 2014 年底通过收购浙江乐叶将产业链延伸至下游电池、组件环节，收购之前浙江乐叶产能规模较小仅拥有约 200MW 产能，依托于公司在单晶领域形成的良好品牌以及处于行业领先水平的电池、组件研发技术，公司组件业务订单饱满，因此在自有产能不足阶段，主要通过委外加工方式弥补产能缺口，委外加工费计入制造费用，从而导致 2015 年度制造费用占比较高。随着浙江乐叶原有产线升级改造逐步完成以及募集资金投资项目“泰州乐叶年产 2GW 单晶电池、组件项目”、马来西亚古晋项目等自筹资金项目的陆续建成，公司自有产能大幅提升，委外加工比例逐年下降，制造费用占比相应降低，原材料及其他成本则相应上升。

（3）近一年及一期前五名供应商采购的情况

表 5-31：发行人近一年及一期前五名供应商采购的情况表

单位：万元

0 0.87

序号	供应商名称	采购金额	占采购比重	主要采购产品	是否为关联方
2018 年 1-9 月					
1	新特能源股份有限公司	81, 841. 15	6. 32%	硅料	否
2	OCI Company Ltd.	63, 470. 03	4. 91%	硅料	否
3	新疆大全新能源股份有限公司	56, 197. 28	4. 34%	硅料	否
4	平煤隆基新能源科技有限公司	50, 010. 65	3. 86%	电池	否
5	营口昌泰铝材有限公司	44, 615. 90	3. 45%	辅助材料	否

102

Fig. 4. Predicted bounding boxes on test image 1 using fine-tuned model.

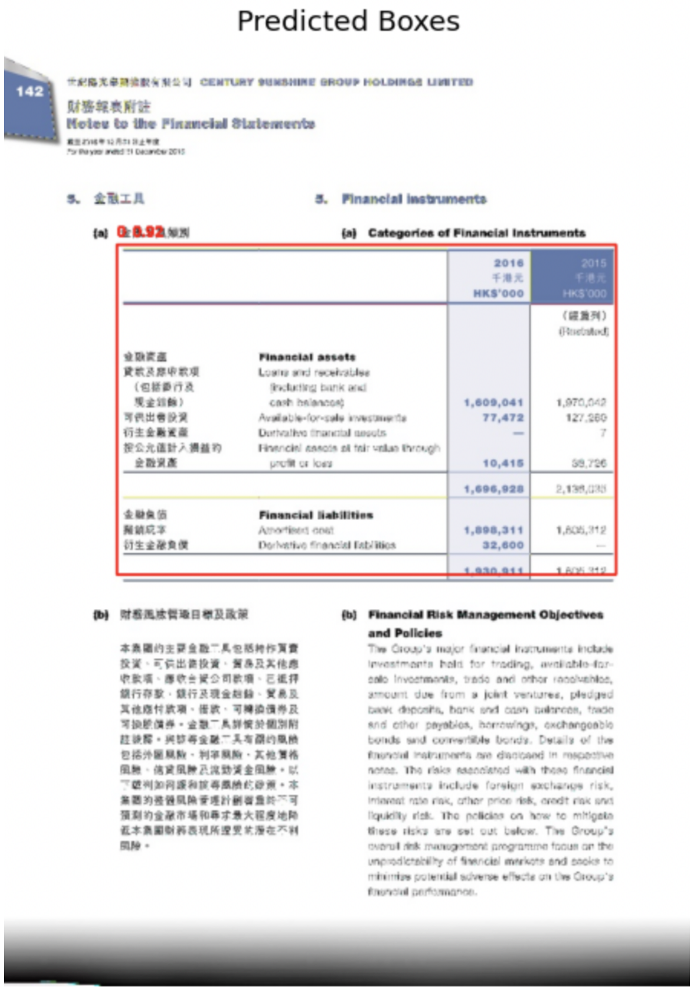
multiple tables within a single image, which highlights the effectiveness of our fine-tuned YOLOv5 model on real-world document layouts. We can also observe that the detected bounding boxes align well with the actual tables, generally enclosing the relevant content with high precision.

Despite these successes, the visualizations also highlight room for improvement. In both figures, we observe slight over-padding at the top of the bounding boxes, where the predicted boxes extend beyond the actual table boundaries. Additionally, under-padding is noticeable along the sides of the boxes in Figure 4, suggesting that some table content may be partially excluded. Addressing these issues could help further refine detection accuracy and improve performance on downstream tasks such as table structure recognition or OCR.

VII. Discussion

A. Model Performance Assessment

Our visual and quantitative results demonstrate that the fine-tuned YOLOv5s model performs well in detecting tables across diverse document layouts. The model not only identifies single tables but also effectively locates multiple tables within the same page, an important feature given the variability


Predicted Boxes
0.92

142 世紀陽光集團控股有限公司 CENTURY SUNSHINE GROUP HOLDINGS LIMITED

財務報表附註
Notes to the Financial Statements

截至2016年12月31日止年度
For the year ended 31 December 2016

5. 金融工具 5. Financial instruments

(a) 金融工具類別 (a) Categories of Financial Instruments

		2016 千港元 HK$'000	2015 千港元 HK$'000
			(經重列) (Restated)
金融資產	**Financial assets**		
貸款及應收款項（包括銀行及現金結餘）	Loans and receivables (including bank and cash balances)	1,609,041	1,970,042
可供出售投資	Available-for-sale investments	77,472	127,260
衍生金融資產	Derivative financial assets	—	7
按公允值計入損益的金融資產	Financial assets at fair value through profit or loss	10,415	38,726
		1,696,928	2,136,035
金融負債	**Financial liabilities**		
攤銷成本	Amortised cost	1,898,311	1,606,312
衍生金融負債	Derivative financial liabilities	32,600	—
		1,930,911	1,606,312

(b) 財務風險管理目標及政策

本集團的主要金融工具包括持作買賣投資、可供出售投資、貿易及其他應收款項、應收合營公司款項、已抵押銀行存款、銀行及現金結餘、貿易及其他應付款項、借款、可轉換債券及可換股債券。金融工具詳情於個別附註披露。與該等金融工具有關的風險包括外匯風險、利率風險、其他價格風險、信貸風險及流動資金風險。以下載列如何減輕該等風險的政策。本集團的整體風險管理計劃著重於不可預測的金融市場和尋求最大程度地降低本集團財務表現所受潛在不利影響。

(b) Financial Risk Management Objectives and Policies

The Group's major financial instruments include investments held for trading, available-for-sale investments, trade and other receivables, amount due from a joint ventures, pledged bank deposits, bank and cash balances, trade and other payables, borrowings, exchangeable bonds and convertible bonds. Details of the financial instruments are disclosed in respective notes. The risks associated with these financial instruments include foreign exchange risk, interest rate risk, other price risk, credit risk and liquidity risk. The policies on how to mitigate these risks are set out below. The Group's overall risk management programme focus on the unpredictability of financial markets and seeks to minimise potential adverse effects on the Group's financial performance.

Fig. 5. Predicted bounding boxes on test image 2 using fine-tuned model.

in real-world document structures. The confidence scores, ranging from 0.81 to 0.92 in our sampled figures, reflect strong model certainty and consistent performance.

B. Accuracy Analysis

The bounding boxes generally encapsulate the table regions accurately, with minimal overlap with surrounding text, validating our preprocessing and annotation strategy. However, we also observed minor imperfections in the form of over-padding, particularly on the top edges, and slight under-padding on the sides. These subtle issues suggest that the model may still struggle with precisely delimiting the spatial extent of certain table structures, especially when they are closely surrounded by dense text.

C. Limitations

One limitation of our current approach lies in the reliance on fixed image sizes and anchor boxes. Tables vary significantly in shape and layout, and this variability may not always be captured optimally using YOLO's default configurations. Incorporating adaptive bounding strategies or exploring architectures designed specifically for document layout analysis could help mitigate these challenges.

D. Conclusion

Nonetheless, our results highlight the potential of using lightweight models like YOLOv5s in table detection tasks, especially when paired with fine-tuning and well-structured annotation data. The model's ability to generalize across varying table formats highlights the model's improved ability to identify table structures accurately and confidently. The use of a stricter IoU threshold (0.9) during evaluation provides a robust measure of localization accuracy, further validating the fine-tuned model's effectiveness.

VIII. Summary

In this project, we developed a deep learning pipeline to detect table structures in document images using YOLOv5. We trained and evaluated both a baseline model and a fine-tuned model on a labeled dataset of printed tables. The fine-tuned model demonstrated clear performance gains, with improved F1 score and mAP values, validated through PR curve visualizations and visual inspection. Our results suggest that fine-tuning pretrained models is a viable approach for improving table detection in documents, with strong potential for deployment in real-world document digitization tasks.

IX. References

References

[1] Prasad, D. K., Zitnick, C. L., Zhang, H., Sakurai, K. (2020). "CascadeTabNet: An approach for end-to-end table detection and structure recognition from image-based documents." *arXiv preprint arXiv:2004.00220*.

[2] Siddiqui, T., Shafait, F., Dengel, A. (2019). "DeCNT: Deep learning-based table detection and structure recognition using conditional neural networks." *International Conference on Document Analysis and Recognition (ICDAR)*, IEEE.

[3] Kasem, A., Tamine, L., Al-Salman, A. (2022). "Deep learning for table detection and structure recognition: A survey." *Journal of Information Science and Engineering*, vol. 38, no. 3, pp. 587–609.

[4] Jocher, G., et al. (2020). "YOLOv5 by Ultralytics." [Online]. Available: https://github.com/ultralytics/yolov5

[5] Li, Z., Zhong, X., Cui, J., Govindaraju, V. (2019). "TableBank: A Benchmark Dataset for Table Detection and Recognition." *International Conference on Document Analysis and Recognition (ICDAR)*, IEEE.