Spring 2022 Capstone Project

Table Extraction via Eye Gaze Tracking

Second Progress Report

Shihang Wang (sw3275), Yeqi Zhang (yz3975), Yibai Liu (yl4616), Yijia Jin (yj2682), Yinqiu Feng (yf2579)

April 17, 2022

1. Progress Overview

As mentioned in the first progress report, we completed our initial borderless table extraction model based on the YoloV5 architecture in CV to detect all tables within an image-based document, and we also designed an Area Of Interest (AOI) prediction model to analyze eye gaze data collected with the Gazepoint equipment. In the second half of this project, we primarily focused on extending and finalizing models, integrating models into an End-to-End Eye Gaze event, and generating production-level code and a wrapper package for future use.

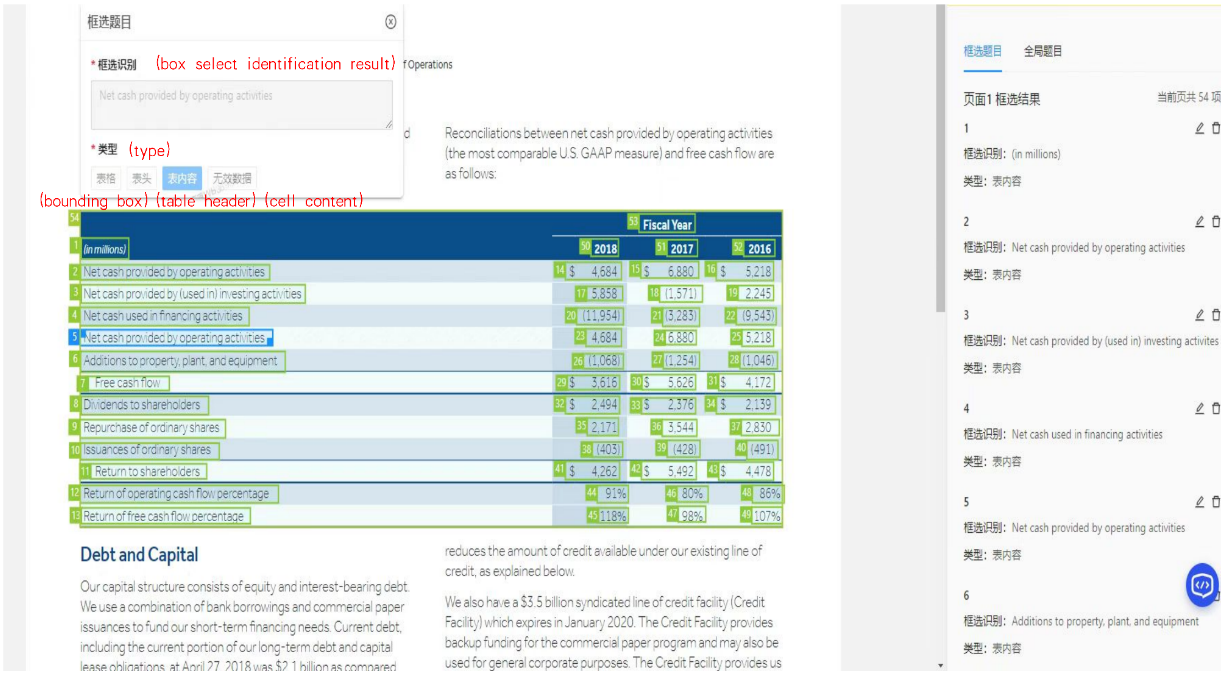
To develop and implement the OCR model, we retreated our dataset and added manual annotations to obtain the ground truths for tabular structures and texts inside tables.

For modeling, we retrained the table extraction CV model with more images with different characteristics and atypical structures to achieve a higher accuracy, and we added OCR as an extension of our model which automatically recognizes texts from the tables. Also, we successfully embedded both models in the eye gaze experiments so that within one experiment trial, the participant after collecting eye movements data can directly see the prediction results and evaluate the correctness.

For generating production-level code, we broke code into modules and developed a pipeline for preprocessing, predicting, and displaying results. We finalized the code for eye gaze experiments to enable an one-click automated trial, and we are currently working on wrapping up the source code into an open-source Python package.

2. Dataset Retreatment

To obtain annotations for structures and texts of the tables, we utilized Alicloud's crowd-source platform, which returns text content and coordinates based on our selections. The table header and the table (cell) content would also be labeled as two different categories. The exhibition of a sample annotation output is as below:



框选题目

* 框选识别 (box select identification result)

Net cash provided by operating activities

* 类型 (type)

表格 表头 [✓ 表内容] 无效数据

(bounding box) (table header) (cell content)

Reconciliations between net cash provided by operating activities (the most comparable U.S. GAAP measure) and free cash flow are as follows:

54		Fiscal Year 53	
1 (in millions)	50 2018	51 2017	52 2016
2 Net cash provided by operating activities	14 $ 4,684	15 $ 6,880	16 $ 5,218
3 Net cash provided by (used in) investing activities	17 5,858	18 (1,571)	19 2,245
4 Net cash used in financing activities	20 (11,954)	21 (3,283)	22 (9,543)
5 Net cash provided by operating activities	23 4,684	24 6,880	25 5,218
6 Additions to property, plant, and equipment	26 (1,068)	27 (1,254)	28 (1,046)
7 Free cash flow	29 $ 3,616	30 $ 5,626	31 $ 4,172
8 Dividends to shareholders	32 $ 2,494	33 $ 2,376	34 $ 2,139
9 Repurchase of ordinary shares	35 2,171	36 3,544	37 2,830
10 Issuances of ordinary shares	38 (403)	39 (428)	40 (491)
11 Return to shareholders	41 $ 4,262	42 $ 5,492	43 $ 4,478
12 Return of operating cash flow percentage	44 91%	46 80%	48 86%
13 Return of free cash flow percentage	45 118%	47 98%	49 107%

Debt and Capital

Our capital structure consists of equity and interest-bearing debt. We use a combination of bank borrowings and commercial paper issuances to fund our short-term financing needs. Current debt, including the current portion of our long-term debt and capital lease obligations at April 27, 2018 was $2.1 billion as compared

reduces the amount of credit available under our existing line of credit, as explained below.

We also have a $3.5 billion syndicated line of credit facility (Credit Facility) which expires in January 2020. The Credit Facility provides backup funding for the commercial paper program and may also be used for general corporate purposes. The Credit Facility provides us

框选题目 全局题目

页面1 框选结果 当前页共 54 项

1
框选识别: (in millions)
类型: 表内容

2
框选识别: Net cash provided by operating activities
类型: 表内容

3
框选识别: Net cash provided by (used in) investing activites
类型: 表内容

4
框选识别: Net cash used in financing activities
类型: 表内容

5
框选识别: Net cash provided by operating activities
类型: 表内容

6
框选识别: Additions to property, plant, and equipment
类型: 表内容

The aggregated annotation results were shown below:

	oss_path	abandon	abandonReason	box_select
0	http://education-annotate.oss-cn-beijing.aliyu...	NaN	NaN	{"container":{"page1":{"width":594,"height":77...
1	http://education-annotate.oss-cn-beijing.aliyu...	NaN	NaN	{"container":{"page1":{"width":594,"height":77...
2	http://education-annotate.oss-cn-beijing.aliyu...	NaN	NaN	{"container":{"page1":{"width":594,"height":77...
3	http://education-annotate.oss-cn-beijing.aliyu...	NaN	NaN	{"container":{"page1":{"width":594,"height":77...
4	http://education-annotate.oss-cn-beijing.aliyu...	NaN	NaN	{"container":{"page1":{"width":593,"height":77...

The annotations (the 'box_select' column in the dataframe) were stored in json format:

```
{'container': {'page1': {'width': 594, 'height': 774}},
 'annotations': [{'id': 1,
   'points': [[38.63636363636364, 235.15151977539065],
    [131.81818181818184, 235.15151977539065],
    [131.81818181818184, 243.78788341175428],
    [38.63636363636364, 243.78788341175428]],
   'label': 'Location Country or State',
   'page': 1,
   'result': {'type': 'table header'},
   'color': 'rgba(204, 110, 51, 1)'},
  {'id': 2,
   'points': [[292.72727272727275, 234.24242886629975],
    [380.4545454545455, 234.24242886629975],
    [380.4545454545455, 244.6969743208452],
    [292.72727272727275, 244.6969743208452]],
   'label': 'Square Feet (in thousands)',
   'page': 1,
   'result': {'type': 'table header'},
   'color': 'rgba(204, 110, 51, 1)'},
  {'id': 3,
   'points': [[42.727272727272734, 247.42424704811793],
    [87.27272727272728, 247.42424704811793],
    [87.27272727272728, 255.15151977539065],
    [42.727272727272734, 255.15151977539065]],
   'label': 'Connecticut',
   'page': 1,
   'result': {'type': 'cell content'},
   'color': 'rgba(145, 204, 51, 1)'},
```

3. Modeling

3.1 YoloV5-based Table Extraction model

The YOLO [2] network is mainly composed of three main components:

¨ ***Backbone***: The convolution neural network of image features is formed by aggregating and merging on different fine-grained images.

¨ ***Bottleneck***: A series of network layers that mix and combine image features, and transfer image features to the prediction layer

¨ ***Head***: Predict the image features, generate boundary boxes and predict categories.

Figure 1 below shows the general architecture of the object detection network.

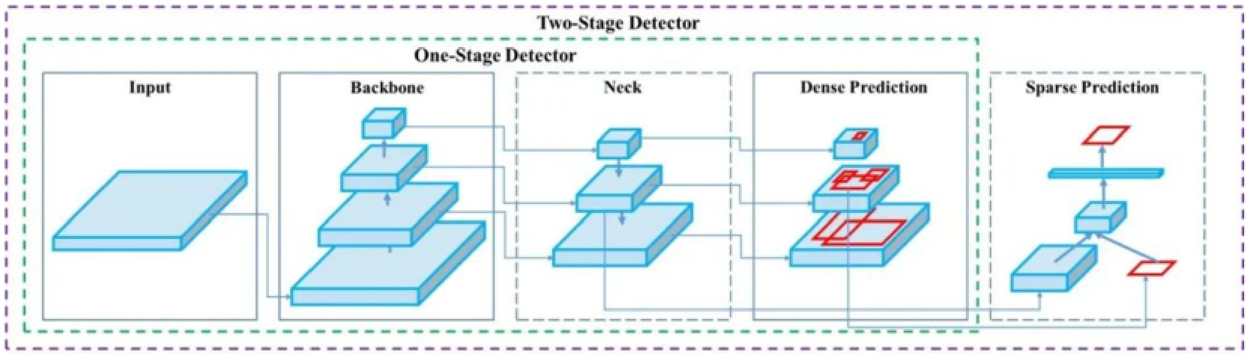


Figure 1: the general architecture of an object detection network

3.1.1 Training

The experiment is conducted using Pytorch framework and it is the second development of the source code ultralytics/yolov5: YOLOv5 🚀 in PyTorch > ONNX > CoreML > TFLite (github.com)

The whole process of the training are as following:

Annotate the dataset: the original dataset is unannotated, so we use labelImg tool to bound the table and generate an xml format file which contains the class label and the four coordinates of the bounding boxes.

Process the dataset: After an subset (about 100 images) of the full dataset set is annotated as the dataset for the later experiments, the dataset is transformed into a VOC dataset, which is one of the mostly used dataset format for detection tasks, and is split into train set and validation set, the ratio is 8:2

Train the network: The network was trained on 200 epochs, costing about 1001.62s on an RTX 2070

Finally, the mAP on the test set is 0.87. In Figure 2, training loss and validation loss curve have a steady descending trend, and the ascending curve of mAP, precision and recall shows the network has fit the dataset and extracted the features of the tables.

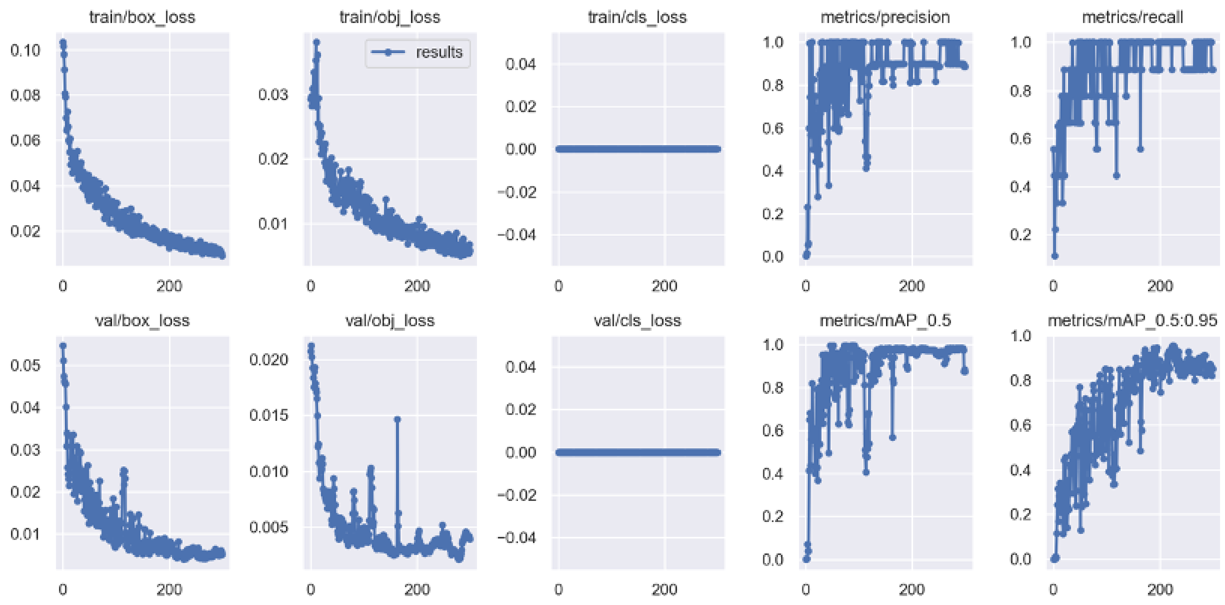


Figure 2: training and validation loss

The AUC value is 0.74, and the curve in Figure 3 shows the model have a high confidence on recognizing a table.

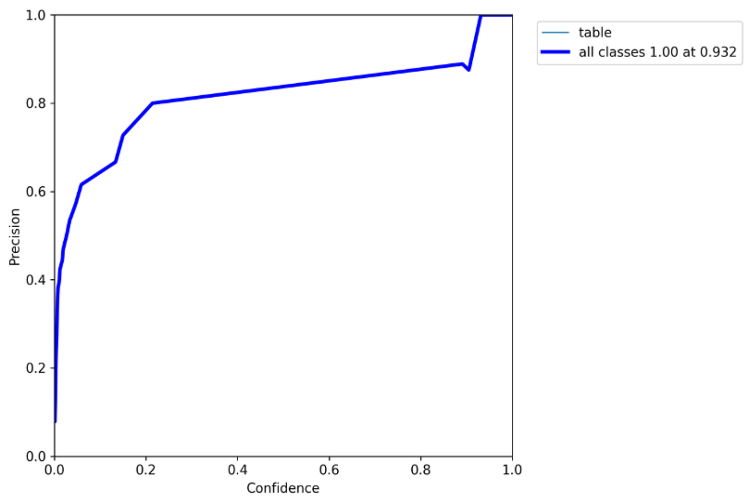


Figure 3: the AUC curve of the network

3.1.2 Results

In Figure 4, the prediction result on the test set shows the network has a good ability on recognizing borderless tables with different size, shape and format in average confidence of 0.9.

Almost all types of borderless tables could be detected precisely by the model as long as we train enough samples.

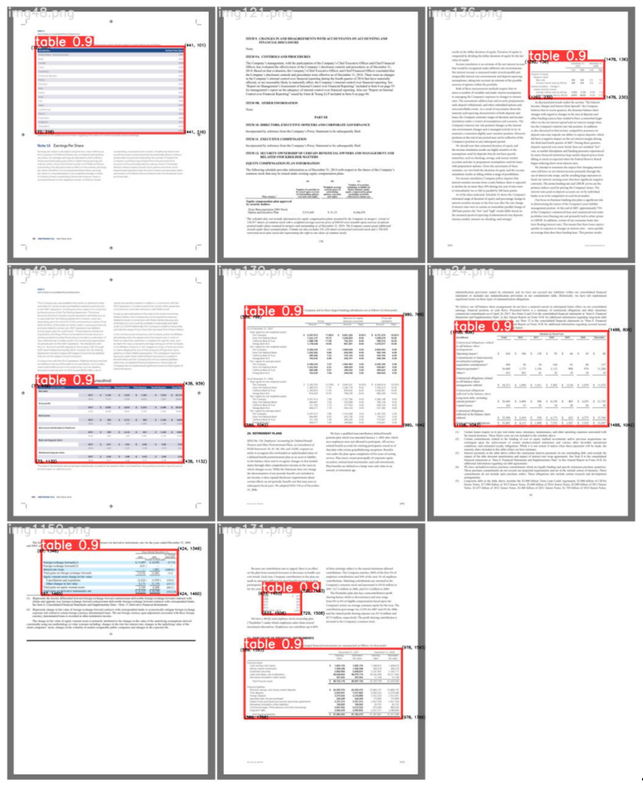


Figure 4: the prediction results on a batch of test images

3.2 PyGaze experiment: end-to-end

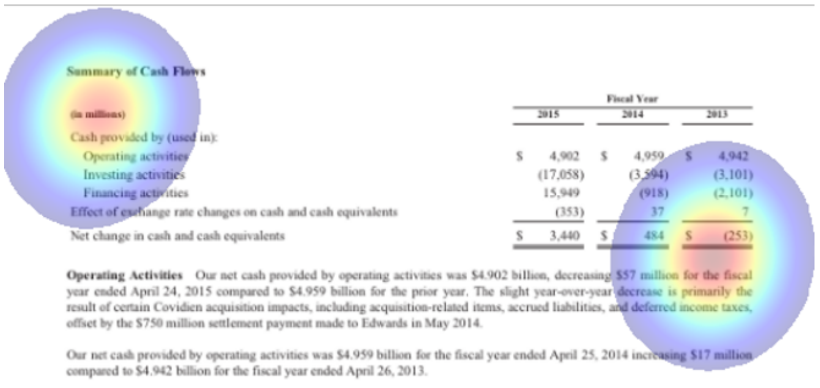
To collect eye-movement data with Gazepoint's eye tracker equipment, in the first phase of the project, we designed an experiment with the Python package PyGaze [3], which provides an API that establishes data connection to the tracker, calibrates the camera, displays several image-based documents in sequence, and then records eye gaze data and logs events in local files. In each experiment, it performs n trials corresponding to n test images, and in each trial it asks participants to fix their eyes on a table inside the image for 10 seconds. We adopted two approaches and developed several versions of the experiment that differentiate on the instructions, preprocessing process, and prediction models.

3.2.1 Baseline method: Eye gaze only

The baseline method of table extraction was to predict a bounding box for the table of interest solely based on the density distribution of eye gaze data. The rationale behind this approach was that if we capture the eye fixations densely falling around corners of a table, then we can estimate the AOI as a prediction of the table coordinates.

The experiment initially asked participants to stare at each corner of the table for 2 seconds, however, after processing and visualizing the tracking data, we realized that focusing on four corners has problems. First, the dataset contains primarily borderless tables, which, compared to normal tables with clear corner points, made it more difficult for participants to stare at a blank. Also, too much eye movement between the corners led to more noisy and inaccurate data points. Besides, spending approximately equal time on each corner was hard to achieve, which biased the density distribution.

After discussion, we came up with a more efficient way of stare similar to cropping a picture, which was to ask participants to only stare at the upper left corner and the lower right corner, each for 5 seconds. By calculation this method also gives the table coordinates but reduces noise.



Summary of Cash Flows

	Fiscal Year		
(in millions)	2015	2014	2013
Cash provided by (used in):			
Operating activities	$ 4,902	$ 4,959	$ 4,942
Investing activities	(17,058)	(3,594)	(3,101)
Financing activities	15,949	(918)	(2,101)
Effect of exchange rate changes on cash and cash equivalents	(353)	37	7
Net change in cash and cash equivalents	$ 3,440	$ 484	$ (253)

Operating Activities Our net cash provided by operating activities was $4.902 billion, decreasing $57 million for the fiscal year ended April 24, 2015 compared to $4.959 billion for the prior year. The slight year-over-year decrease is primarily the result of certain Covidien acquisition impacts, including acquisition-related items, accrued liabilities, and deferred income taxes, offset by the $750 million settlement payment made to Edwards in May 2014.

Our net cash provided by operating activities was $4.959 billion for the fiscal year ended April 25, 2014 increasing $17 million compared to $4.942 billion for the fiscal year ended April 26, 2013.

Figure 5: Heatmap of fixation points around two corners

Within a 10-second display time, the eye gaze data was recorded. Like shown in the heatmap above, after denoising the data points would cluster around two centroids, which are approximately the two corners of the table. Therefore we applied DBSCAN clustering to estimate the coordinates of two centroids and draw a bounding box based on their values. The preprocessing pipeline cut points falling outside the image edges and removed the unfixed eye movements, and the DBSCAN model with maximum intra-cluster distance of 50 pixels and 7 minimum samples per cluster could provide the centroids of two desired clusters. The coordinates of the centroids were used to form an AOI bounding box for the table.

3.2.2 Advanced method: Eye gaze + CV

The advanced method combines the CV object detection model and the eye gaze experiment to achieve a one-step solution that performs more precise table detection and extraction. The rationale behind this approach was: all documents have one or more table(s), and regardless of the table of interest, we could utilize CV to first extract all tables from a document and then perform eye gaze trials to determine the AOI and output the information about the desired table. The CV model was embedded in the experiment to predict and display results within the experiment, but they can also work independently with or without the eye tracker equipment, which makes this approach more flexible. In the future, if the equipment was not available or if we would rather recognize all tables within the image, the CV and OCR models could be adapted to different circumstances.

Compared to the baseline version, this method again changes the way of stare. Since the CV model provides more precise coordinates of the tables, the eye tracker only needs to collect information about the approximate location of AOI, so that participants were asked to fix their eyes on the center of the table during the display.

In the end-to-end eye gaze experiment, we ran the CV model first to obtain coordinates of the predicted bounding boxes for all tables, started the experiment environment to collect eye movement data, and calculated the fixation density in each bounding box and crop the table of interest.

3.3.3 Comparison

The evaluation of two methods focused on three aspects: time consumption, accuracy, and consistency of results.

- Time

For time consumption, a single experiment using either method took approximately the same amount of time: 4 test images took around 3 minutes 40 seconds in minimum, and 4 minutes on average. Adding an additional image would increase the experiment time by 15~20 seconds. The calibration step accounted for a large proportion of the time fluctuations since it might need several calibrations until getting accurate results, and the baseline method requires higher calibration accuracy than the advanced method. Repeating the calibration for another time would increase the experiment time by 30 seconds approximately.

The following two pie charts represent the time decomposition for a sample baseline experiment and a sample advanced experiment using 4 identical test images. In the sample experiments, the total time consumption for the baseline approach was 3 minutes 40 seconds and for the advanced approach was 3 minutes 36 seconds.

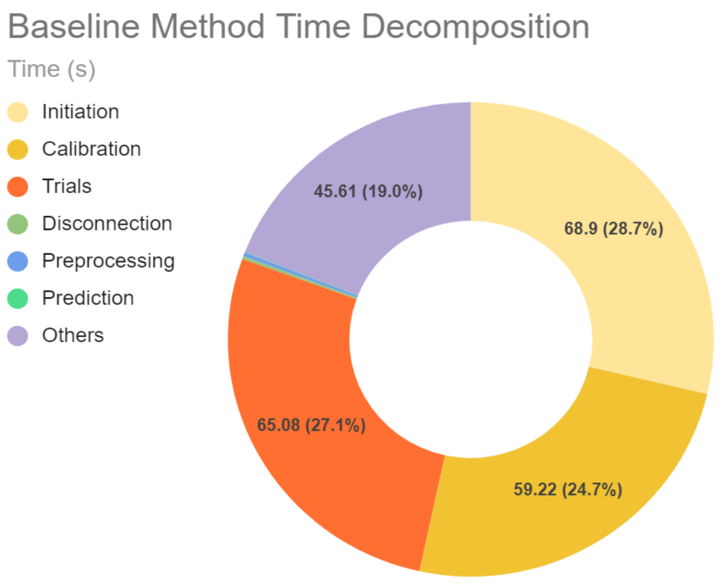


Figure 6: Time decomposition of a sample baseline experiment with 4 test images

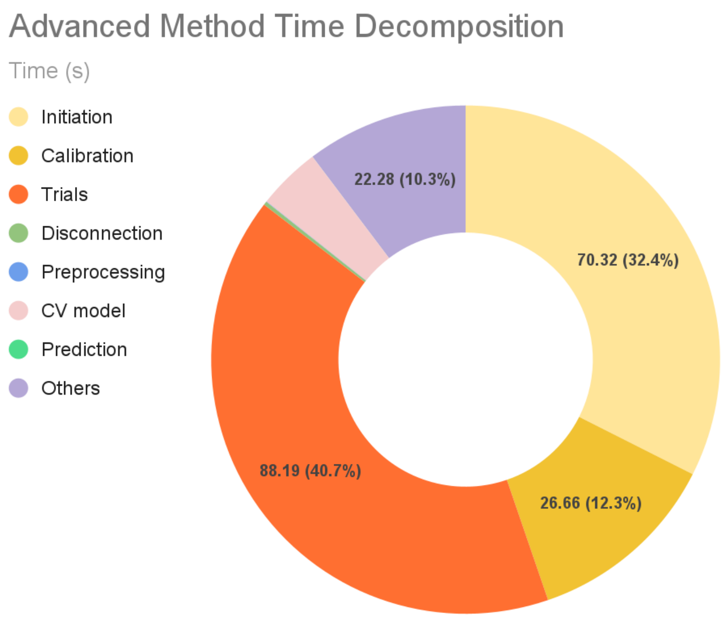


Figure 7: Time decomposition of a sample advanced experiment with 4 test images

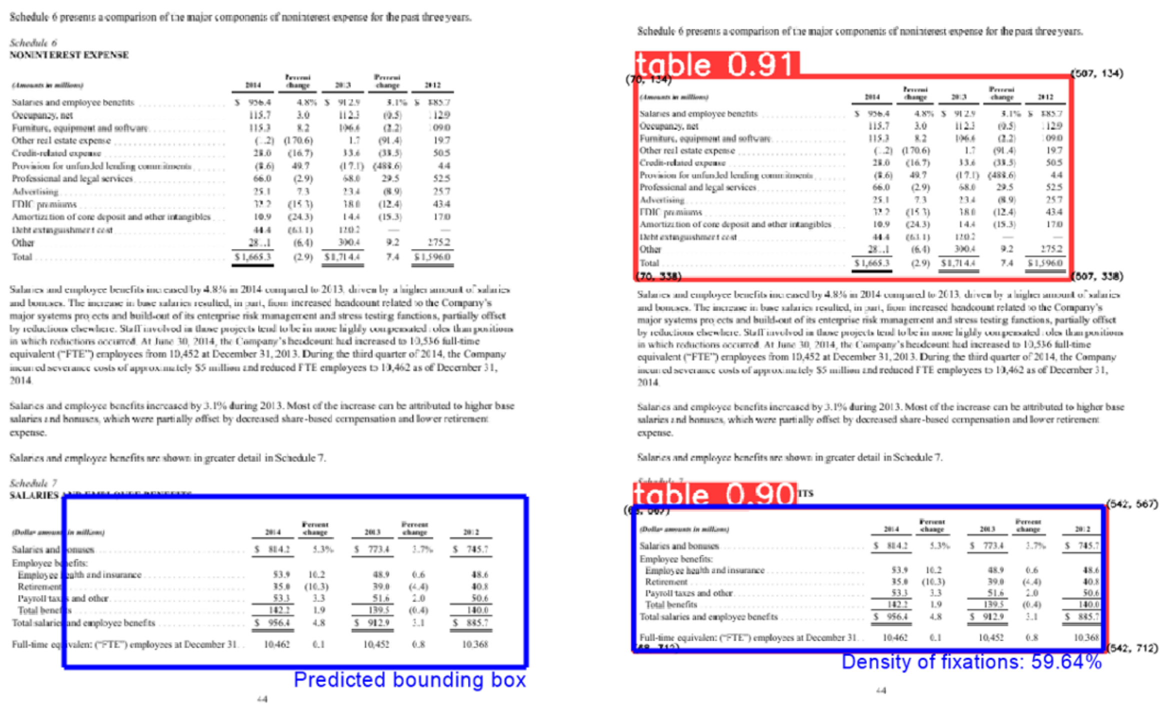
As shown above, the initiation (connection to device), calibration, and the experiment trials (display of images) took more than 80% of the time, whereas the preprocessing and prediction cost trivial time. The difference in time spent on initiation and trials between two approaches was caused by fluctuations related to requests, buffering, and logging, but the difference in time spent on calibration was based on the times of calibration attempts. In the samples, the baseline experiment carried out two calibrations while the advanced experiment only performed one. This simulated the real scenarios because the baseline prediction requires higher calibration accuracy.

In the advanced experiment, the CV model running time was only 8.73 seconds, taking 4% of the total. The ‘Others’ part in two experiments was primarily related to the user, and it included the time waiting for keyboard press or mouse clicks, as well as the time for manually displaying outputs after predictions.

- Accuracy

The accuracy of the baseline method fluctuates violently and greatly depends on the accuracy of calibration and recording because it counts on precise eye gaze data points to estimate the table coordinates. When one or both eyes deviate from the calibrated position, the clustering model would not be able to clearly differentiate clusters, thus the predicted table using centroid coordinates could be badly inaccurate.

However, the advanced approach only uses fixation data to calculate how much of the attention falls into each bounding box and determines the table of interest. The advanced method was more robust against inaccurate calibrations and also provided much more intuitive and accurate results that completed our major tasks.



Schedule 6 presents a comparison of the major components of noninterest expense for the past three years.

Schedule 6
NONINTEREST EXPENSE

(Amounts in millions)	2014	Percent change	2013	Percent change	2012
Salaries and employee benefits	$ 956.4	4.8%	$ 912.9	3.1%	$ 885.7
Occupancy, net	115.7	3.0	112.3	(0.5)	112.9
Furniture, equipment and software	115.3	8.2	106.6	(2.2)	109.0
Other real estate expense	(1.2)	(170.6)	1.7	(91.4)	19.7
Credit-related expense	28.0	(16.7)	33.6	(33.5)	50.5
Provision for unfunded lending commitments	(8.6)	49.7	(17.1)	(488.6)	4.4
Professional and legal services	66.0	(2.9)	68.0	29.5	52.5
Advertising	25.1	7.3	23.4	(8.9)	25.7
FDIC premiums	32.2	(15.3)	38.0	(12.4)	43.4
Amortization of core deposit and other intangibles	10.9	(24.3)	14.4	(15.3)	17.0
Debt extinguishment cost	44.4	(63.1)	120.2	—	—
Other	281.1	(6.4)	300.4	9.2	275.2
Total	$ 1,665.3	(2.9)	$ 1,714.4	7.4	$ 1,596.0

Salaries and employee benefits increased by 4.8% in 2014 compared to 2013, driven by a higher amount of salaries and bonuses. The increase in base salaries resulted, in part, from increased headcount related to the Company's major systems projects and build-out of its enterprise risk management and stress testing functions, partially offset by reductions elsewhere. Staff involved in those projects tend to be in more highly compensated roles than positions in which reductions occurred. At June 30, 2014, the Company's headcount had increased to 10,536 full-time equivalent ("FTE") employees from 10,452 at December 31, 2013. During the third quarter of 2014, the Company incurred severance costs of approximately $5 million and reduced FTE employees to 10,462 as of December 31, 2014.

Salaries and employee benefits increased by 3.1% during 2013. Most of the increase can be attributed to higher base salaries and bonuses, which were partially offset by decreased share-based compensation and lower retirement expense.

Salaries and employee benefits are shown in greater detail in Schedule 7.

Schedule 7
SALARIES AND EMPLOYEE BENEFITS

(Dollar amounts in millions)	2014	Percent change	2013	Percent change	2012
Salaries and bonuses	$ 814.2	5.3%	$ 773.4	3.7%	$ 745.7
Employee benefits:					
Employee health and insurance	53.9	10.2	48.9	0.6	48.6
Retirement	35.0	(10.3)	39.0	(4.4)	40.8
Payroll taxes and other	53.3	3.3	51.6	2.0	50.6
Total benefits	142.2	1.9	139.5	(0.4)	140.0
Total salaries and employee benefits	$ 956.4	4.8	$ 912.9	3.1	$ 885.7
Full-time equivalent ("FTE") employees at December 31	10,462	0.1	10,452	0.8	10,368

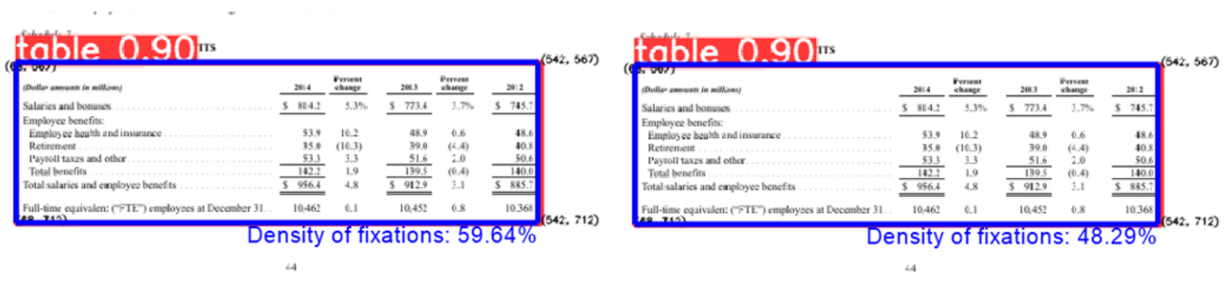
44

Figure 8: Table in the same image predicted by the baseline vs. advanced method

As shown above, the bounding box predicted by the advanced approach is much more accurate than the baseline approach, which did not crop the whole table.

- Consistency

The consistency of model predictions is significant. The quality of the eye gaze data was impacted by various factors, like device positioning, lighting, mobility of participants, calibration accuracy, quick blinks, fixation duration threshold, etc., and most noisy factors could not be eliminated by simple variable control or training. Comparing the two methods, the advanced method showed much higher consistency contributed by the well-trained CV results. Integrating deep learning models into the experiment largely reduced the information dependence on the eye tracker and thus led to much more stable performance and predictions.



(Dollar amounts in millions)	2014	Percent change	2013	Percent change	2012
Salaries and bonuses	$ 814.2	5.3%	$ 773.4	3.7%	$ 745.7
Employee benefits:					
Employee health and insurance	53.9	10.2	48.9	0.6	48.6
Retirement	35.0	(10.3)	39.0	(4.4)	40.8
Payroll taxes and other	53.3	3.3	51.6	2.0	50.6
Total benefits	142.2	1.9	139.5	(0.4)	140.0
Total salaries and employee benefits	$ 956.4	4.8	$ 912.9	3.1	$ 885.7
Full-time equivalent ("FTE") employees at December 31	10,462	0.1	10,452	0.8	10,368

Figure 9: Prediction of the same image in two independent experiments by the advanced method

Figure 9 shows the reliability of the advanced approach as it generates consistent prediction results when the same test image is displayed in multiple independent experiments.

4. Production

The final task of the project is to wrap up the source code for all processes mentioned above and write production-level code. We built a Python package `pygazeplotter` adapted from an open-source package named pygazeanalyser [4], which is a visualization library affiliated with PyGaze. We developed a preprocessing module that includes functions for data preprocessing, and we adjusted the original plotting functions to draw heatmaps, scanpaths, etc. of the eye gaze data. We automated all processes in the experiments and packaged all source code in a repository. It contains the environments, datasets, models, experiments in Jupyter Notebooks, and reports of the project so that the users can download the depository and run an instance on their local machine.

5. Further Improvements

We are currently at the 85% mark of our project. We have finished the data collection, including experiment design and execution, and modeling stage, including model development and validation. Our next steps will focus on the text information retrieval (OCR) and the end-to-end wrap-up to deliver a production-level solution.

5.1 Further Tasks

The problems and open tasks we are currently working on include the following:

5.1.1 Text Information Retrieval

The current output for our table recognition model is a bounding box for each single table, stored in json format per input image. For the purpose of use in the industry, for example, to improve efficiency of reading financial statements and annual reports for analysts, it would be better if we could deliver a structured table in textual forms. In order to deliver such information, our solution is to develop an OCR model, which transforms a bounded image of a table, with or without structure, to text-based table structure. Common choices for OCR engines include EasyOCR(lightweight but mostly for pdf conversion), tesseract(for image conversion) and ocropus(based on LSTM, less used nowadays), etc. With model selection, we've chosen Google's open source tesseract OCR engine [4] due to its good performance in high-resolution images. Currently, we are still in the process of model embedding and tuning for OCR. Upon finishing, we expect to deliver an integrated model with table detection, table recognition and information retrieval.

5.1.2 Dataset Annotations

One major difficulty that we have encountered during the model development stage is the discrepancy in annotation, both in terms of format of annotation and accuracy of hand labeling. Because the dataset of choice, FinTabNet, was initially published for table detection tasks, we do not have access to the ground truth labels for OCR, which requires text-based data per cell. Our enhanced model based on the SOTA TableNet model [5] also requires table structure data. Hand labeling for such is not only time-consuming, but, for cell structure recognition, the boundary of cells can also be ambiguous to define even for human bare eyes. Our current solution is to retreat to the text-base labeling for OCR, and we have just finished the human annotation. We are currently working on parsing the ground truth data, and then using the parsed label to fine-tune our model. If possible, we will also seek for possible solutions to retrieve table structural data in the future, to enhance the table recognition and to assist OCR modeling.

5.1.3 Unstable Data Quality of Eye Gaze Experiments

Regarding the eye gaze experiments, both the baseline method and the advanced method currently suffer from unstable data quality caused by occasionally poor calibration accuracy. When the device positioning or participant's sitting posture moved around, the equipment found it difficult to capture the true coordinates of eye movements. Also, the calibration took long time and repeated attempts to obtain accurate results.

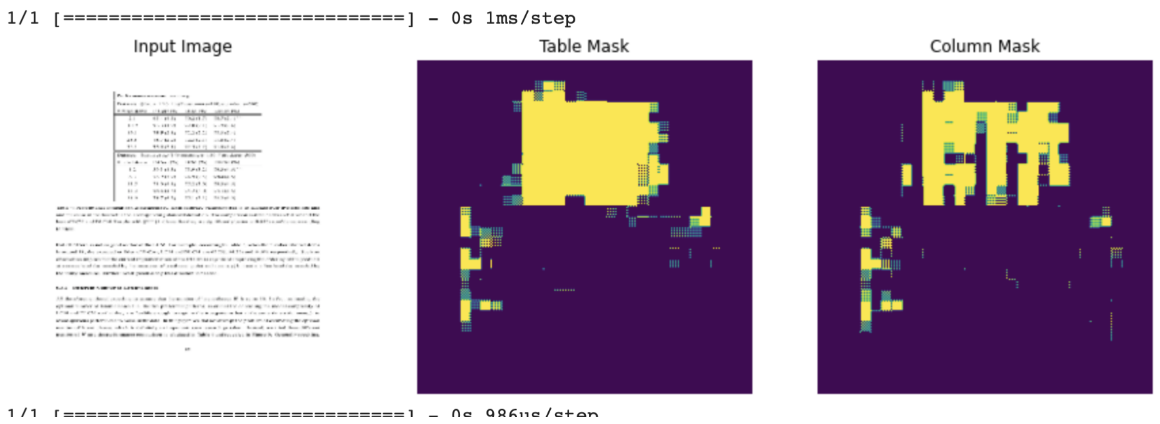
5.2 Further Steps

The further work for solving the previous problems include the following:

5.2.1 Fine Tuning OCR Model Based On Ground Truths

After obtaining the annotations for ground truths of all images, we would work on fine tuning and improving the OCR model. The annotations would contain the position of rows, columns, and cells, as well as texts inside the tables, and they would be in JSON format that requires parsing to be used as model inputs.

Our initial OCR model only supported recognizing tables with borders and a white background. We would work on expanding the model to borderless tables as well as tables with dark & noisy backgrounds, uncommon table layouts, multilingual texts, and international number & currency formats.


```
1/1 [==============================] - 0s 1ms/step
```

```
1/1 [==============================] - 0s 986us/step
```

Figure 11: Sample table recognition predicted by the initial OCR model

As shown above, the current OCR model is able to predict masks for the table and the columns. It had a total loss of 0.3 with a table prediction loss 0.15 and a column prediction loss 0.16.

5.2.2 Improving Time Efficiency and Data Quality of The Eye Gaze Experiments

Our current eye gaze experiments took from 3.5 minutes minimum up to 6 minutes maximum to complete with 4 images, which is not time efficient. After decomposing the time spent on each subtask inside the experiment, we believe that there is space for improvement. For example, we would work on adjusting the settings of calibration to address the unstable data quality and at the same time cut down the time consumption. Also, we could further simplify manual operations like mouse clicks and keyboard press, so that the waiting and responding time could be reduced.

5.2.3 Improving Production-Level Code and Package Build-up

Since the OCR model is still under development and training, we would continue to work on writing production-level code for all procedures and deliver an end-to-end solution for borderless table extraction and text recognition. Also, we would further standardize the Python package that we built, and add tutorials/instructions and configuration guidance to ease future use.

6. Contribution

Shihang Wang: Dataset retreatment and annotation acquisition

Yeqi Zhang: YoloV5-based Table Extraction CV model development and training

Yibai Liu: Eye Gaze experiments design and model integration, PyGazePlotter package build-up

Yijia Jin: TableNet-based table detection and structure recognition model development, Initial OCR model development

Yinqiu Feng: Initial OCR model development

JPMorgan Chase & Co. Columbia University Data Science Institute

References

[1] Ultranalytics (2021). YoloV5 in PyTorch, GitHub. Retrieved on Apr. 16, 2022 from https://github.com/ultralytics/yolov5

[2] Dalmaijer E. and Mathot, S. (2020). PyGaze, GitHub. Retrieved on Apr. 16, 2022 from https://github.com/esdalmaijer/PyGaze

[3] Dalmaijer E. and Mathot, S. (2014). PyGaze, GitHub. Retrieved on Apr. 16, 2022 from https://github.com/esdalmaijer/PyGazeAnalyser

[4] Anthony Kay. 2007. Tesseract: an open-source optical character recognition engine. Linux J. 2007, 159 (July 2007), 2.

[5] S. Paliwal, V. D, R. Rahul, M. Sharma and L. Vig, "TableNet: Deep Learning Model for End-to-end Table Detection and Tabular Data Extraction from Scanned Document Images," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, Sydney, Australia, 2019 pp. 128-133.

[6] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "*You Only Look Once: Unified, Real-Time Object Detection*"

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick, "*Mask R-CNN for Object Detection and Segmentation*"