

# DIGITIZATION OF HANDWRITTEN LEDGERS

## AN INTEGRATED APPROACH TO IMPROVE HANDWRITTEN TEXT RECOGNITION

SUBMITTED IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTER OF SCIENCE

RODERICK MAJOOR  
12852724

MASTER INFORMATION STUDIES  
DATA SCIENCE  
FACULTY OF SCIENCE  
UNIVERSITY OF AMSTERDAM

SUBMITTED ON 17.06.2024

	<b>UvA Supervisor</b>
<b>Title, Name</b>	Dr. N.J.E. (Nanne) van Noord
<b>Affiliation</b>	MultiX Lab
<b>Email</b>	<a href="mailto:n.j.e.vannoord@uva.nl">n.j.e.vannoord@uva.nl</a>



## 1 ABSTRACT

2 Handwritten text recognition is a crucial step in digitizing historical  
3 documents. In this work, we will explore the challenges associated  
4 with a handwritten text recognition model using historical ledger  
5 pages from the Bank of Amsterdam. We first conduct an error analy-  
6 sis of the model, and based on that make suggestions on where and  
7 how the performance of the model could be improved. The main  
8 concern is the model's inability to capture the layout of the pages.  
9 We perform post-processing steps to improve the output of the  
10 model and we create a layout analysis method, which aims to cap-  
11 ture the distinctive tabular layout found in the ledger pages. While  
12 earlier research mainly focuses on capturing different elements on  
13 documents, such as text, tables and images, our work specifically  
14 focuses on capturing table cells and contents from a tabular struc-  
15 ture. Our layout analysis method shows promising results, yet it  
16 faces challenges in accurately matching the ground truth bounding  
17 boxes. This discrepancy shows the difficulties in evaluating com-  
18 plex document structures. Lastly, we propose a method for training  
19 a deep learning model using pseudo-annotations which shows that  
20 the quantity of pseudo-annotations can compensate for the noise  
21 present in them and achieve promising results.

## 22 KEYWORDS

23 HTR, Layout analysis, Computer vision, Historical documents, Ob-  
24 ject detection

## 25 GITHUB REPOSITORY

26 <https://github.com/roderickmajoer/Thesis>

## 27 1 INTRODUCTION

28 The Bank of Amsterdam was an early bank established in 1609, de-  
29 scribed by some as the first central bank [18]. The bank played an  
30 important role in the financial world of the 17th and 18th century.  
31 Transactions made at the bank from 1650 to 1800 were recorded in  
32 ledgers and are still preserved to this day. These ledgers contain  
33 information about transactions, both debit and credit, of customers  
34 of the bank. The ledgers can thus be very important in analyzing  
35 the money flow at the time. Digitizing these ledgers may help in  
36 preserving them in a much easier to analyze format which would  
37 allow for more research into the contents of the ledgers. However,  
38 these ledgers are all handwritten making it hard to retrieve the  
39 information out of them on large scale. Current efforts to retrieve  
40 the information are done by using Handwritten Text Recognition  
41 (HTR) tools to extract the text in the ledgers. The problem is that  
42 the currently used HTR techniques have too many inaccuracies to  
43 be used effectively. Because of this, the handwritten text cannot  
44 be recognized correctly and can thus not be digitized properly. To  
45 improve digitization efforts, it is crucial to find out why some hand-  
46 written text cannot be recognized, what kind of errors are made  
47 during the HTR process and how the HTR process can be optimized  
48 to improve these inaccuracies. The research question we wish to  
49 answer is:

51 *To what extent can the categorization of errors in handwritten ledger  
52 analysis be used to enhance the identification and resolution of HTR  
53 model inaccuracies?*

54 Subquestions belonging to this research question are:

- 56 • What factors contribute to text recognition errors in hand-  
57 written ledgers?
- 58 • What are the different types of errors encountered in hand-  
59 written text recognition?
- 60 • How can different processing methods improve HTR output  
61 in handwritten ledgers?

62 Since we perform a number of different experiments, our report  
63 follows the following structure: we start by comparing some related  
64 work. Then we will provide a description of the dataset. After that  
65 we look at the different experiments, where for each we provide our  
66 methodology including evaluation steps as well as results and dis-  
67 cussion of that particular experiment. We end with a more general  
68 discussion and conclusion.

## 69 2 RELATED WORK

70 The research gap being addressed in this project revolves around the  
71 challenges associated with the digitization of handwritten ledgers.  
72 The proposed research will contribute in closing the research gap  
73 within the HTR domain, specifically looking at historical handwrit-  
74 ten documents. The digitization of handwritten ledgers contains  
75 several steps. In this section, we will look at key research papers  
76 that discuss different approaches to the steps of our problem.

### 77 2.1 Document Layout Analysis

78 The biggest challenge in our digitization task is being able to find the  
79 correct word order from the HTR output. Seuret describes layout  
80 analysis and finding the correct word order as one of the main  
81 challenges in HTR [19]. Several approaches to solve this problem  
82 have been tried. These approaches are typically divided in either  
83 deep learning methods or more classical computer vision methods.  
84 We will show recent advancements made in both approaches.

85 Kastanas et al. show the use of different deep learning archi-  
86 tectures for layout analysis [6]. Transformer-based, graph-based  
87 models, and convolutional neural networks (CNN) are compared.  
88 The CNN-based model YOLOv5 and transformed-based model Lay-  
89 outLMv3 both show promising results for identifying most elements  
90 on documents, with YOLOv5 performing slightly better [6].

91 Multiple other studies describe the use of transformer based  
92 methods to perform layout analysis and optical character recogni-  
93 tion (OCR) [8, 11, 23]. Furthermore, Smock et al. show that transformer-  
94 based object detection models can produce excellent results for the  
95 tasks of detection, structure recognition, and functional analysis of  
96 tables [21]. While these approaches show promising results, they  
97 are not tested for our specific task at hand.

98 Oliveira et al. propose dhSegment, an open-source implemen-  
99 tation of a CNN-based pixel-wise predictor for document segmenta-  
100 tion. The approach aims to address various document processing

101 tasks simultaneously, including page extraction, baseline extraction,  
102 layout analysis, and illustration and photograph extraction. They  
103 show that a single CNN architecture can be used across tasks with  
104 competitive results [1].

105 Droby et al. propose a holistic method that applies Mask R-CNN  
106 for text line extraction in historical documents. Their work achieved  
107 state-of-the-art results on well known historical datasets [4]. While  
108 their work does not directly transfer to our task, it does show the  
109 potential for document segmentation and possibly table recognition  
110 abilities of Mask R-CNN.

111 The aforementioned studies show that the use of deep learning  
112 methods can be used to perform layout analysis as well as table  
113 recognition tasks in historical documents. However, these methods  
114 often require massive amounts of training data to be used effectively.  
115 To use these methods, we should look whether pre-trained models  
116 transfer well to our dataset and are able to detect the layout of our  
117 documents. Other approaches use more classical computer vision  
118 techniques to segment a document in order to recognize the layout.  
119 These methods often require homogeneity of the documents to  
120 work effectively.

121 Lehnenmeier et al. show a method that combines various state-of-  
122 the-art methods for OCR processing to detect layout on a document  
123 [7]. To perform table recognition, they make use of techniques such  
124 as binarization, denoising, and Hough line detection to find relevant  
125 parts of the tables. By taking the intersection of relevant horizontal  
126 and vertical Hough lines they are able to find table cells and thus  
127 determine the layout of the table [7].

128 Bulacu et al. introduce a method based on contour tracing that  
129 generates curvilinear separation paths between text lines in order to  
130 preserve the ascenders and descenders of text lines to determine the  
131 layout in handwritten historical documents [2]. With this approach  
132 they are able to determine separate elements of tables in historical  
133 documents.

134 Liu et al. describe the use of several key steps, including skew  
135 correction, region segmentation, and page layout analysis on his-  
136 torical Tibetan documents [10]. Skew correction is achieved by  
137 leveraging baseline features and Hough transform to determine  
138 the document's skew angle. Subsequently, region segmentation is  
139 accomplished by identifying borders through a series of preprocess-  
140 ing steps, including median filtering, Gaussian smoothing, Sobel  
141 edge detection, and removal of small area regions. These processes  
142 collectively enable accurate positioning of document borders for  
143 further analysis and structure extraction [10].

144 Liang et al. describe the use of Hessian and Gabor filters in order  
145 to find table structures on a document [9]. They show the possibility  
146 for column segmentation for the tables using the found lines.

147 Prieto et al. address the challenge of document image understand-  
148 ing in documents with complex layouts like tables. They compare  
149 two approaches and show promising results [17]. Their approaches  
150 take the text lines outputted by HTR systems and use machine  
151 learning methods to group these text lines and classify cells based  
152 on these grouped text lines in order to find the tabular structure.

153 It can be seen that depending on the dataset and the homogene-  
154 ity of the documents, different approaches for layout analysis/table

155 recognition may be suitable. When a similar table structure is clearly  
156 drawn on all the documents, approaches using basic image process-  
157 ing techniques could be suitable to determine the structure of the  
158 table. When the structure is not easily identifiable on the page, it  
159 might be better to look at approaches that use the output of HTR  
160 systems such as text lines or words locations to classify table cells.  
161 Machine learning based approaches could also be useful in these  
162 cases but often require many training data.

## 163 2.2 HTR Post-Processing Techniques

164 Post-processing OCR and HTR output typically consists of trying  
165 to correct misidentified words or characters. One strategy to do  
166 this is to create a dictionary of candidate characters and use the  
167 Levenshtein distance to calculate the distance between predicted  
168 and dictionary characters [16]. The use of a dictionary for allowed  
169 characters is shown to be a good strategy to eliminate HTR errors  
170 [7]. Other studies show the use of language models to correct word  
171 output and spelling in HTR systems [13, 16].

## 172 2.3 HTR Evaluation Strategies

173 Typically, HTR systems are evaluated using the word error rate  
174 (WER) and character error rate (CER). However, the evaluation  
175 of page-level HTR output faces challenges primarily due to the  
176 Reading Order (RO) problem [22]. A strategy to mitigate this is  
177 the use of the intersection over union (IoU) score which matches  
178 ground truth and HTR output boxes based on their coordinates  
179 on the page [15]. It is then possible to use the WER and CER to  
180 calculate the accuracy.

## 181 3 DATA DESCRIPTION

182 The dataset used in this study consists of pages from the collection  
183 of ledgers from the Bank of Amsterdam. The full collection can be  
184 found on the [website](#) of The Amsterdam City Archives. In the series  
185 of ledgers, the current accounts of the customers were maintained.  
186 Each customer had one or more pages, with smaller traders having  
187 a portion of a page. An alphabetical list of the traders and the page  
188 of their current account can be found in the index. The ledgers kept  
189 track of the amounts that were debited and credited from and to a  
190 customer. The ledger pages consist of several columns containing  
191 information such as the date, account holder name, account number  
192 and amount debited/credited. Figure 7 shows an example of what  
193 a page looks like. Our dataset is split into multiple subsets that  
194 come from different books and that all have slightly varying pages,  
195 while still having structural similarity. Figure 1 shows the amount  
196 of pages for each subset in the dataset. The total dataset consists of  
197 68 pages. A word count for each page is shown in Figure 2

198 The ground truth and HTR system output both use a [PageXML](#)  
199 format to store values on the page. These PageXML store the coor-  
200 dinates for each item (e.g. textregion, tablerregion, etc.) on the page.  
201 We can plot the values in the PageXML file over our image to better  
202 show this. Figure 8 shows the PageXML data of the ground truth  
203 and HTR system plotted on the original image for part of one page  
204 in our dataset.

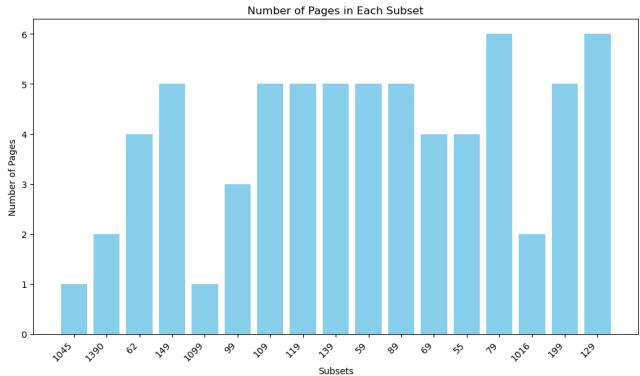


Figure 1: The amount of pages for each subset in our dataset.

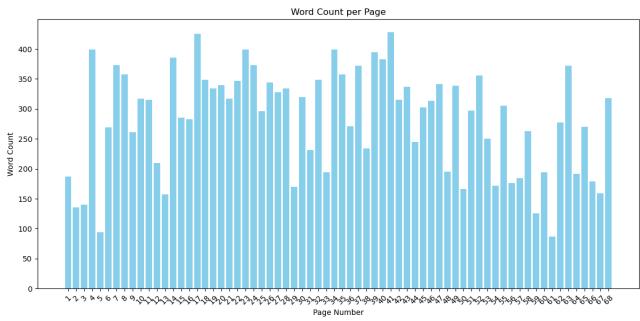


Figure 2: The word count for each page in our dataset.

## 4 HTR SYSTEM ERROR ANALYSIS

Before attempting to improve the HTR system, we will first conduct an extensive analysis to identify where the current system fails or has flaws. This will involve creating a categorization scheme that shows for each category the nature of the error and a possible solution. Based on this scheme, we can then try out various steps to improve the HTR system. This section thus lays the basis for further sections, where we will explore strategies to improve the HTR system.

### 4.1 Method

To obtain the accuracy of the HTR system, we will need to compare the HTR output to the ground truth values. This in itself is not straightforward, since the ground truth values do not contain all the words in the image while the HTR output does. Furthermore, since the layout of the page is not recognized by the HTR system, the order of the words is completely different in the ground truth and HTR output. To still compare the ground truth values and HTR output, we will use an approach that matches the words based on their bounding box coordinates on the image. To do this we make use of the IoU score. This is a technique that can be used to find how similar two bounding boxes are [15]. By matching each bounding box in the ground truth with a corresponding bounding box in the HTR output based on the highest IoU score, we can compare the ground truth values and HTR output on a word to word basis.

Once we have aligned the words in the ground truth and the HTR output properly, we can start to identify the flaws of the HTR system. We will make use of the Levenshtein distance, which can be used to find the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one string into the other [5]. By applying this on our ground truth values and HTR output on a word to word basis, we can track how many character insertions, deletions and substitutions are done in the HTR output for each word in the ground truth. We will show this for each subset of our data to also see whether some subsets are more difficult for the HTR system. We will also keep track of what characters are inserted, deleted or substituted so we can find what characters seem to be most problematic for the HTR system.

By creating this analysis, we can also better inspect our image by visually showing the words that the HTR system can or cannot match. An example of this visual representation is shown in Figure 9. Using this visual representation, we can inspect the image to see whether there are any characteristics that might cause the HTR system to make errors. This could be noise on parts of the page, words being close together or even overlapping etc.

Upon completing our analysis, we can conduct a list of main errors made by the HTR system and decide on what steps could be taken to mitigate these errors. The approach for these possible solutions is outlined in further sections.

### 4.2 Evaluation

For the evaluation of the HTR system, metrics such as WER, CER, and Levenshtein Distance will be calculated. These metrics measure the minimum number of operations (substitutions, insertions, deletions) needed to transform the HTR output into the ground truth. The formulas for these metrics are:

$$WER = \frac{S_w + D_w + I_w}{N_w} \quad (1)$$

Where  $S_w, D_w, I_w$  are the number of word substitutions, deletions and insertions respectively and  $N_w$  is the total number of words in the ground truth.

$$CER = \frac{S_c + D_c + I_c}{N_c} \quad (2)$$

Where  $S_c, D_c, I_c$  are the number of character substitutions, deletions and insertions respectively and  $N_c$  is the total number of characters in the ground truth.

We will also calculate recall and precision scores. To calculate those for HTR output, we first need to define what constitutes true positives (TP), false positives (FP), and false negatives (FN) in our context.

- **TP (True Positive):** Characters correctly recognized by the HTR system.
- **FP (False Positive):** Characters incorrectly recognized by the HTR system (i.e., system identifies text where there is none or incorrectly identifies text).
- **FN (False Negative):** Characters that are present in the ground truth but not recognized by the HTR system.

The recall and precision are then:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

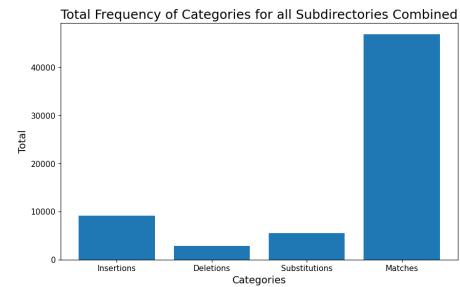
$$Precision = \frac{TP}{TP + FP} \quad (4)$$

### 277 4.3 Results

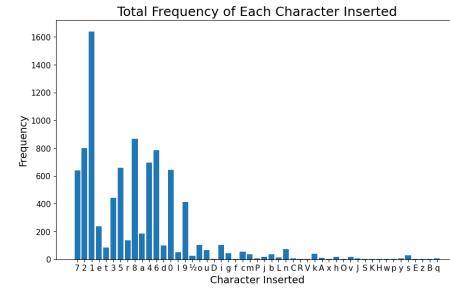
278 The accuracy of the current HTR system using the aforementioned  
 279 metrics is shown in Table 1. As can be seen, the accuracy of the base  
 280 model is not flawless. The recall is quite high, showing that not a  
 281 lot of false negatives are found. However, the precision is lower,  
 282 showing that the models incorrectly captures text. Figure 3 shows  
 283 the total amount of character insertions, deletions, substitutions  
 284 and matches made by the HTR system for the complete dataset.  
 285 Here, we can clearly see that substitutions and mainly insertions of  
 286 characters are quite high while deletions are relatively low. This fur-  
 287 ther shows that the HTR system captures a relatively high amount  
 288 of false positives, meaning that there are a lot of characters cap-  
 289 tured that are not actually on the page. Figure 5 shows this in more  
 290 detail for each of our subdirectories. Here, we have 4 subfigures  
 291 showing the insertions, deletions, substitutions and matches as  
 292 percentage of total characters in each subdirectory. This shows that  
 293 the accuracy of the HTR model can vary quite a lot per subdirectory.  
 294 Furthermore, it shows that the amount of insertions and substitu-  
 295 tions are consistently higher than deletions except for one outlier in  
 296 subdirectory 1390, where there are a lot more deletions. In order  
 297 to improve the model, it is thus crucial to find a way to mainly  
 298 reduce the insertions and substitutions made by the model. Figure  
 299 4 shows the frequencies of different characters inserted, substituted  
 300 and deleted. Here we can see that a lot of ‘invalid’ characters are  
 301 inserted. These are non-alphanumeric characters that should not  
 302 be part of our dataset. Furthermore, we see that many numbers  
 303 and letters that look alike (such as i and 1) are substituted. We  
 304 use these analyses and the fact that the HTR system is not able  
 305 to capture the layout of the pages to conduct a list of errors made  
 306 by the HTR system. A selection of the most common errors and  
 307 possible solutions is shown in Table 2. In further sections, we will  
 308 use the results of this section to improve the model.

**Table 1: Baseline scores for the currently used HTR system. We matched the ground truth words with HTR output words based on highest IoU value.**

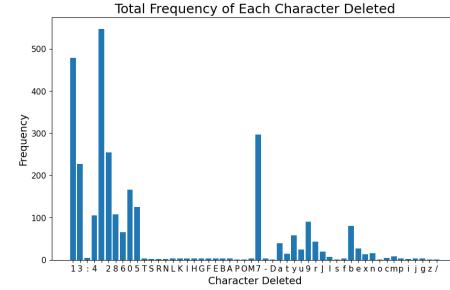
Metric	Value
WER	0.4243
CER	0.3422
Recall	0.9475
Precision	0.7403
Total Edit Distance	18920



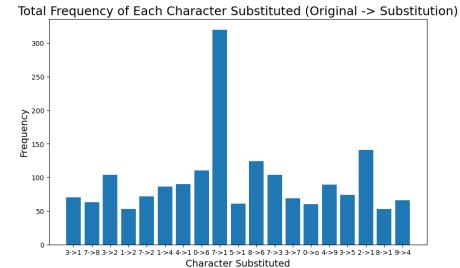
**Figure 3: The total amount of insertions, deletions, substitutions and matches made by the HTR system compared to the ground truth values. The values are on a character level and made word per word.**



**(a) Frequency of inserted characters made by the HTR system.**

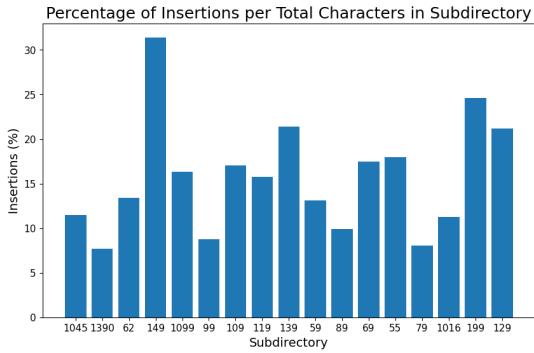


**(b) Frequency of deleted characters made by the HTR system.**

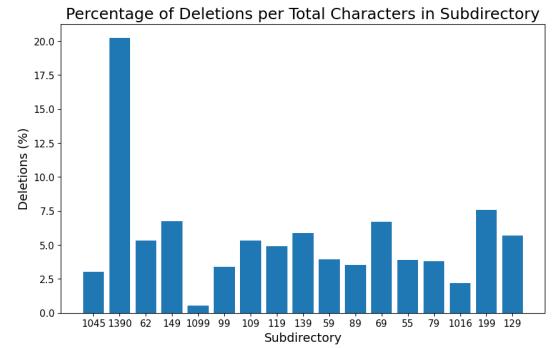


**(c) Frequency of substituted characters made by the HTR system. The substitutions are shown as (original character -> substituted character). We only show the substitutions with a frequency of 50 or higher.**

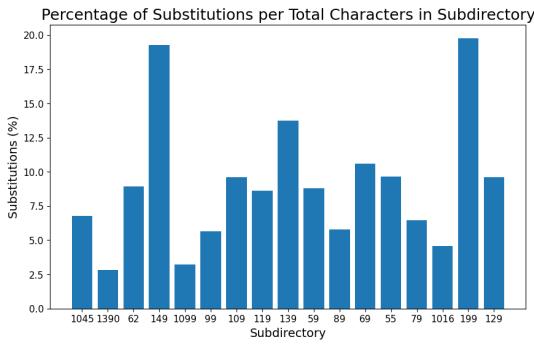
**Figure 4: The frequencies of inserted, deleted and substituted characters made by the HTR system.**



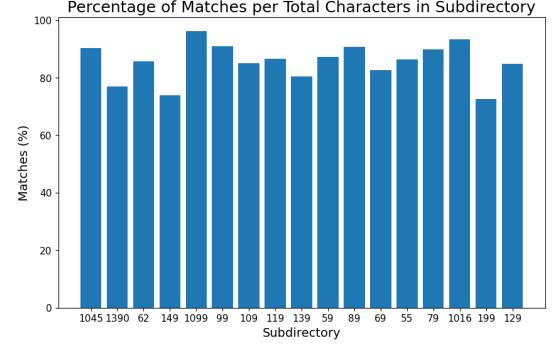
(a) Amount of insertions made by the HTR system per subdirectory.



(b) Amount of deletions made by the HTR system per subdirectory.



(c) Amount of substitutions made by the HTR system per subdirectory.



(d) Amount of matches made by the HTR system per subdirectory.

Figure 5: The amount of insertions, deletions, substitutions and matches made by the HTR system compared to the ground truth values per subdirectory. The values are on a character level and made word per word and as a percentage of total characters in ground truth.

Table 2: Categorization of HTR System Errors

Error Type	Description	Possible Solution
Deletions	Characters not recognized at all (false negative)	Improve HTR system
Insertions	Characters recognized that are not there (false positives)	Improve HTR system, post-processing
Substitutions	Characters recognized as another character	Improve HTR system, post-processing
Layout - Word Split	Single word is split into separate words	Layout analysis
Layout - Word Merge	Multiple words are recognized as one word	Layout analysis
Layout - Data Split	Data wrongly split (e.g., 'feb 5' in GT becomes 'feb' and '5' in HTR)	Layout analysis
Layout - Word Order	Word order not recognized	Layout analysis
HTR - Bounding Box	Wrong location bounding box vs actual location word	Improve HTR system
Case Sensitivity	Case sensitivity issues	Post-processing
Invalid Characters	Characters that cannot occur	Post-processing

## 5 POST-PROCESSING

From our error analysis, we find that using post-processing techniques may lead to improved accuracy. Post-processing in our experimental framework focuses on refining the output generated by the HTR system. By using suitable post-processing techniques on the HTR output, we are able to improve the accuracy of the model.

### 5.1 Method

We have seen that insertions and substitutions are the main problems of our model. We will employ strategies to reduce these errors while trying to keep the deletions at a similar rate. One such post-processing strategy involves the removal of characters from the output strings that do not align with the ground truth dictionary [7, 16]. Analysis of the HTR output (as depicted in Figure 4a) reveals

several insertions that are not valid letters or numeric characters. To address this, we employ regular expressions (regex) to filter out invalid characters from the HTR output. We are then left with alphanumeric characters only. Furthermore, the HTR system often substitutes letters for numbers, as can be seen in Figure 4c. We can undo these substitutions in some cases. For instance, when a single character is found to be a letter by the HTR system, we can expect this to be wrong, since single character words are never letters in the ground truth. Also, numbers that contain a letter (such as '11i0') are probably wrong. We use these facts and the most common made substitutions as found in our earlier analysis to recover the right characters. We use the same evaluation metrics as before to compare our method to the baseline scores.

## 5.2 Results

We compare the post-processed HTR output to the original HTR output to see the different accuracy scores. The accuracy scores for both the baseline and post-processed HTR output are shown in Table 3. We see a very slight decrease in recall while all other metrics improve. This shows that our post-processing strategy is successful in eliminating some of the false positives (insertions and substitutions) of the original model while maintaining almost the same rate of false negatives (deletions). However, we can still see that the precision is relatively low compared to the recall, showing that there is still a good amount of false positives that our method did not catch.

**Table 3: WER, CER, precision and recall scores for the HTR system output with and without post-processing techniques applied.**

Metric	Baseline	Post-Process
WER	0.4243	0.3916
CER	0.3422	0.3169
Recall	0.9475	0.9418
Precision	0.7403	0.7622
Total Edit Distance	18920	17508

## 6 LAYOUT ANALYSIS

Up until this point, we have been comparing the HTR output to the ground truth values by matching their bounding boxes based on the highest IoU score. We did this because the HTR system currently is not able to capture the layout of the tabular structure of the ledger pages and thus we could not just match the output directly, since the word order would be wrong.

The final part of the methodology involves analyzing the layout of the documents. Recognizing the layout is a crucial step in digitizing the ledger pages, since without it there is no good way to use the output of the HTR system because the correct word order would not be known. We will perform the layout analysis by using image processing techniques as well as making use of the HTR output. For this, we will use the [OpenCV \(cv2\)](#) library to perform the necessary processing steps.

### 6.1 Method

To detect the tabular structure on the ledger pages, we must find a way to detect the columns and rows of the table to get the table cells. The ledger pages contain physical lines, which separate the different columns. We start by trying to detect these lines.

The first step is to pre-process our image. We do this by converting our image to grayscale and applying a Gaussian blur on the grayscale image. This helps to reduce the computational complexity and reduces high frequency noise [12].

Next, we apply two types of thresholding to the blurred image: Otsu's thresholding and Adaptive Gaussian thresholding. Otsu's thresholding only captures the most clear markings on the page, such as the text, while the more thin and faint column lines are not captured [2]. The Adaptive Gaussian threshold captures both the text and the column lines. By subtracting the first threshold from the latter, we are left with the column lines and some noise.

Following thresholding, we perform morphological operations, specifically dilation and erosion [14]. We use specific kernels to remove noise while retaining the vertical lines.

We then apply the Hough Line Transform to detect the lines in the image. This can be used to detect (nearly) straight lines in images [7]. We filter the detected lines based on their length and position. For each detected line, we calculate its slope and intercept, and then draw an extended line from the top to the bottom of the foreground area (which contains the ledger page) in the image. This is done using basic principles of line equations in coordinate geometry.

Finally, we perform additional dilation and erosion to merge close lines and remove final noise. We then find contours in the image and draw a line from the top to the bottom of each contour. We are then left with an image containing the contours of each column section, which can then be used to create bounding boxes for the different columns. The whole process is shown in Figure 11.

To detect the rows present in the ledger pages, we have to use a different strategy, since there are no physical markings for row lines on the pages. We will use the word bounding boxes which are outputted by the HTR system in order to find the row bounding boxes.

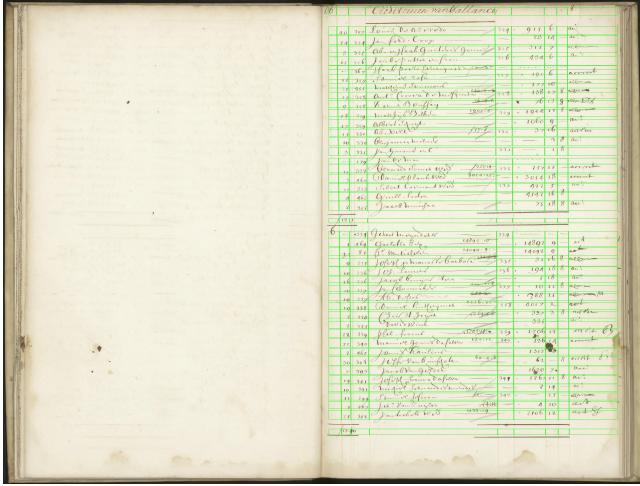
We start by extracting the bounding boxes of the words outputted by the HTR system. We will only use the word bounding boxes that are kept after performing the post-processing. Since the images in our dataset typically consist of two tables (debit and credit) we have to determine where both tables start and end. We do this by finding the middle column line from the list of column lines we found by our method above. We then separate the bounding boxes into two groups: left boxes and right boxes, based on their x-coordinates relative to the middle column line.

We then take a list of boxes (either left boxes or right boxes) and group them into rows. This is done by sorting the boxes by their x-coordinates (from left to right) and then grouping boxes with similar y-coordinates together. For each box, we find the best row to append the box to, based on the minimum difference in

414 y-coordinates. If no suitable row is found, we create a new row. The  
 415 result of these steps are shown in Figure 12.

416 Once we have found the rows, we can extend the bounding boxes  
 417 from the left side of the ledger page to the middle for the left boxes  
 418 and from the middle to the right side of the page for the right boxes.  
 419 We are now left with both the column and row bounding boxes.

420 Once the row and column bounding boxes are determined, we  
 421 can use the intersection of these bounding boxes to determine the  
 422 table cells. Figure 6 shows an example of these found table cells.



456 **Figure 6: The bounding boxes of the predicted table cells after**  
**taking the intersection of the rows and columns.**

## 423 **6.2 Evaluation**

424 For the layout analysis methods, we take the ground truth table cell  
 425 bounding boxes and compare them to our method's predicted table  
 426 cell bounding boxes. We do this by matching each ground truth cell  
 427 to a predicted cell based on the highest IoU score between the two.  
 428 Only one ground truth cell can be matched to one predicted cell  
 429 and vice versa. We then determine whether cells are a true positive  
 430 by setting an IoU threshold. If ground truth and predicted cells are  
 431 matched with an IoU score higher than this threshold, we consider  
 432 it a true positive. Ground truth cells that are not matched with an  
 433 IoU score above the threshold are considered a false negative and  
 434 predicted cells that are not matched with an IoU score above the  
 435 threshold are considered a false positive. We will use a threshold  
 436 of 0.5, 0.25 and >0. We can then determine the recall and precision  
 437 scores, using the same formula as before. Additionally, we calculate  
 438 the F1-score:

$$439 F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (5)$$

440 Furthermore, after matching the cells as described above, we  
 441 further verify the correctness of the match by comparing the text  
 442 content of the predicted table cells against the text content in the  
 443 ground truth table cells. This done by taking the words from the  
 444 HTR output that match location with the predicted cells. The text

445 comparison is done using a normalized Levenshtein distance with  
 446 various thresholds: 0 (exact match), 0.25, 0.5, and 0.75. We use  
 447 these thresholds because we cannot be sure that words from the  
 448 HTR output match exactly with ground truth words because of the  
 449 imperfections of the HTR system. For each threshold, we calculate  
 450 the mean accuracy of the word matches. Additionally, we calculate  
 451 the CER for all the words in the ground truth cells and the words  
 452 in the predicted cells. If the cells are indeed matched correctly,  
 453 we would expect to see good mean accuracy of the word matches  
 454 (especially for the lower thresholds) and we would expect the CER  
 455 to be close to the value found in Section 4.

## 456 **6.3 Results**

457 The mean performances using IoU thresholds of 0.5, 0.25 and >0  
 458 are shown in Table 4. The 'IoU Score' indicates the mean score  
 459 for the true positives found at this IoU threshold. As can be seen,  
 460 when using a higher IoU threshold, the accuracy scores are low and  
 461 when using a lower IoU threshold, the accuracy scores are better.  
 462 This means that when using a lower IoU threshold, a lot of the  
 463 ground truth table cells are matched with predicted cells while this  
 464 is not the case when using a higher IoU threshold. One of the main  
 465 reasons for this behaviour is the fact that the bounding boxes of our  
 466 predicted cells are much more tight than the ground truth boxes.  
 467 Furthermore, the bounding boxes for our predicted table cells are  
 468 often placed slightly lower than the ground truth bounding boxes.  
 469 This is because we use the word bounding boxes from the HTR  
 470 model to determine the rows, and these bounding boxes are almost  
 471 always found to be slightly lower than the actual location of the  
 472 words on the image. This causes the IoU score of a predicted cell  
 473 and ground truth cell to be relatively low, even when both capture  
 474 the same contents of the page. When taking a IoU threshold of 0.25,  
 475 we see the best ratio of IoU scores and accuracy scores. Here we see  
 476 that matched cells have a mean IoU score of 0.473, which is quite  
 477 close to the usual threshold of 0.5. Furthermore, we see a score  
 478 of 0.804, 0.898 and 0.844 for mean precision, recall, and F1-score  
 479 respectively. This shows that we have a good rate of false positives  
 480 and false negatives, as well as having a reasonable area overlap for  
 481 predicted and ground truth cells. To get a more detailed picture of  
 482 these results on subdirectory level, we show the results for the IoU  
 483 thresholds 0.5, 0.25 and >0 in Appendix C in Table 8, Table 9 and  
 484 Table 10 respectively.

485 The accuracy of comparison of text contents for ground truth  
 486 and predicted cells can be seen in Table 5. Here, 'LT 0' until 'LT  
 487 0.75' means how much percent of the text in the ground truth and  
 488 predicted cells differ. So for the first row, 0.5369 means that 53.69%  
 489 of texts are matched exactly, 0.5996 means that 59.96% of texts are  
 490 matched with at most 25% of characters changed etc. If we look  
 491 at the IoU threshold of 0.25, which seemed to perform well in the  
 492 previous result, we can see that about 85% of words are matched  
 493 with at most 75% of the characters differing. Furthermore we see  
 494 a CER of 0.3859. This is higher than the CER found in Section 4.  
 495 However, this can partly be explained due to the fact that in the first  
 496 experiment, we calculated CER by matching one HTR word with a  
 497 ground truth cell text. That means that we discarded some of the  
 498 false positives (HTR words that were not matched). In this case, we  
 499 first append all HTR words into a predicted cell and then compare

499 predicted cell content to ground truth cell content, meaning that  
500 in this case we do keep these false positives when comparing the  
501 texts. Keeping this in my mind, we can determine that most of the  
502 predicted cells and ground truth cells that are matched by IoU, are  
503 also matched based on text contents, showing the success of our  
504 method.

**Table 4: Mean table cell performance metrics at different IoU Thresholds. 'IoU Score' indicates the mean score for the true positives found at this IoU threshold.**

IoU Threshold	IoU Score	Precision	Recall	F1
>0	0.182	0.845	0.946	0.887
0.25	0.473	0.804	0.898	0.844
0.5	0.627	0.416	0.459	0.434

**Table 5: Mean accuracy and CER for matching ground truth table cells with predicted table cells using different IoU thresholds and comparing their text contents using normalized Levenshtein distance thresholds. Values for LT 0, 0.25, 0.50, and 0.75 indicate the mean accuracy at which the normalized Levenshtein distance between the ground truth and predicted texts is within 0 (total match), 25%, 50%, and 75% respectively. CER indicates the average character error rate for the text content in the matched cells.**

IoU Thresh.	LT 0	LT 0.25	LT 0.50	LT 0.75	CER
>0	0.5369	0.5996	0.7333	0.8242	0.4290
0.25	0.5555	0.6213	0.7563	0.8469	0.3859
0.50	0.5898	0.6586	0.7984	0.8770	0.3362

526 for enough pages, we aim to get a more accurate and robust model,  
527 that might perform better on these outliers.

528 We will test this by training two models, one with a smaller  
529 pseudo-annotated train set consisting of 691 pages and one with  
530 a larger train set consisting of 1861 pages. We will test our model on  
531 the same hand-annotated set as before. Since our pseudo-annotations  
532 are not flawless, this experiment can help us determine if the sheer  
533 volume of train data can compensate for the noise introduced by  
534 inaccuracies in the annotations and create a more generalizable  
535 model.

536 For this experiment, we will make use of the Layout Parser Tool  
537 [20], which consists of multiple pre-trained layout detection mod-  
538 els. Specifically, we will use the mask\_rcnn\_X\_101\_32x8d\_FPN\_3x  
539 model trained on the PubLayNet dataset and use transfer learning  
540 for our training. The training and evaluation is performed using  
541 a NVIDIA Tesla T4 GPU. For the training and evaluation of the  
542 model, we started with the basic configuration and made several  
543 adjustments to optimize performance for our specific use case. The  
544 changed hyperparameters are shown in Table 6. An example of the  
545 output created by this model (trained on the large dataset) is shown  
546 in Appendix A in Figure 10.

**Table 6: Hyperparameter settings for model training.**

Hyperparameter	Value
DATALOADER.NUM_WORKERS	2
SOLVER.IMS_PER_BATCH	2
MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE	256
MODEL.ROT_HEADS.NUM_CLASSES	1
SOLVER.WEIGHT_DECAY	0.0005
SOLVER.BASE_LR	0.001
SOLVER.LR_SCHEDULER_NAME	"WarmupMultiStepLR"
SOLVER.STEPS	(333, 666)
SOLVER.WARMUP_ITERS	100
SOLVER.MAX_ITER	1000

## 7 LAYOUT PARSER

505 The final experiment of our research involves the use of a deep  
506 learning model to detect the columns on our ledger pages. If we are  
507 able to create an accurate model to detect and segment different  
508 columns, we can use this to run our HTR model for each column.  
509 This could potentially improve the HTR output by eliminating  
510 overlapping word bounding boxes on different columns as well as  
511 allowing the model to run in numeric or alphabetic mode depending  
512 on the column. This could potentially improve the accuracy of the  
513 model.

### 7.1 Method

514 Creating annotations for these pages is a tedious task and since we  
515 only have a relatively small dataset, we cannot use this to create a  
516 train-test split. Instead, we will use an approach similar to the one  
517 used by Chen et al. [3]. This means we will use our approach to de-  
518 tect columns as described in Section 6 to create pseudo-annotations  
519 for columns and use these as the training set for our deep learn-  
520 ing model. As shown in the previous section, the accuracy of this  
521 rule based method can vary between the different subdirectories  
522 so there exist certain outliers where the method does not perform  
523 well. By creating pseudo-annotations using our previous method  
524

## 7.2 Evaluation

547 For the evaluation of this experiment, we will use the standard  
548 Common Objects in Context (COCO) evaluation metrics. We will  
549 use the average precision (AP) calculated over a range of IoU thresh-  
550 old from 0.50 to 0.95 as well as show the average precision at IoU  
551 threshold 0.5 (AP50) and 0.75 (AP75). We will show the results both  
552 for the bounding box as well as the segmentation (mask) of the  
553 columns.

### 7.3 Results

554 The results are shown in Table 7. We can see that at iteration  
555 1000, the AP stops improving much or even decreases. We see that  
556 initially the model trained on the small dataset seems to perform  
557 better at lower iterations. After iteration 600, the performance of  
558 the model trained on the small dataset decreases whereas the model  
559 trained on the large dataset improves further. This suggests that  
560 the model trained on the small dataset converges faster initially.

563 The best scores for the model trained on the large dataset are at  
 564 iteration 1000 with AP 57.750, AP50 88.515 and AP75 62.306. For  
 565 the model trained on the smaller dataset, the best performance is  
 566 at iteration 600 with AP 56.366, AP50 84.020 and AP75 61.875.

567 The large dataset likely contains more diverse examples, enabling  
 568 the model to generalize better and avoid overfitting, even with ex-  
 569 tended training. This is reflected in the sustained improvement in  
 570 metrics up to iteration 1000. The small dataset might lack sufficient  
 571 diversity, leading to quicker initial convergence but also to over-  
 572 fitting with longer training. This shows that when supplemented  
 573 with a larger volume of data, the training process is improved. The  
 574 large dataset might help mitigate some of the noise present in the  
 575 pseudo-annotations, leading to better overall performance.

**Table 7: Comparison of evaluation results for bbox and segmentation metrics of columns between large and small datasets at different iterations.**

Iteration	Dataset	Metric	AP	AP50	AP75
200	Large	BBox	41.937	78.181	39.855
		Segm	39.311	76.370	36.690
	Small	BBox	43.818	75.341	47.760
		Segm	43.382	75.232	44.457
400	Large	BBox	54.308	86.919	58.568
		Segm	51.570	86.647	53.775
	Small	BBox	53.752	83.035	58.756
		Segm	51.680	82.202	56.379
600	Large	BBox	56.889	89.015	62.029
		Segm	53.364	87.851	57.041
	Small	BBox	56.366	84.020	61.875
		Segm	53.025	83.152	59.184
800	Large	BBox	57.702	88.816	62.442
		Segm	53.945	88.389	56.905
	Small	BBox	54.212	83.443	58.479
		Segm	51.661	82.009	56.954
1000	Large	BBox	57.750	88.515	62.306
		Segm	53.948	88.186	57.151
	Small	BBox	54.022	83.344	58.046
		Segm	51.926	81.720	57.215

## 8 DISCUSSION

577 In this work, we have analyzed a HTR system and found what  
 578 the main errors of the system are. We have then proposed and  
 579 implemented various methods for the different set of errors in the  
 580 system.

### 8.1 Analysis of Findings

581 *What factors contribute to text recognition errors in handwritten  
 582 ledgers?* The primary factors contributing to errors in text recogni-  
 583 tion include the variability in handwriting styles, the quality of the  
 584 original documents and the presence of noise such as stains and  
 585 faded ink. Our analysis revealed that line spacing and overlapping  
 586 text further worsen the recognition accuracy.

588 *What are the different types of errors encountered in handwritten  
 589 text recognition?* We identified several categories of main errors in  
 590 handwritten text recognition:

- 591 • Substitution Errors: Where one character is incorrectly rec-  
 592 ognized as another.
- 593 • Insertion Errors: Where extra characters are included in the  
 594 output.
- 595 • Deletion Errors: Where characters are omitted in the output.
- 596 • Segmentation Errors: Where the system fails to accurately  
 597 distinguish between distinct text regions or words or the  
 598 bounding boxes of found words are inaccurate.
- 599 • Layout Error: The layout of the page is not detected which  
 600 causes the word order to be wrong.

601 *How can different processing methods improve HTR output in hand-  
 602 written ledgers?* Our research proposed various methods to mitigate  
 603 the identified errors:

604 **Post-processing** We implemented a post-processing technique  
 605 where we created a dictionary which we used to filter out excessive  
 606 characters. Furthermore, we showed that it is possible to undo sub-  
 607 stitutions of characters in specific situations. These steps allowed  
 608 us to get a better recognition output of the HTR model.

609 **Layout Analysis** We implemented a rule-based method that is  
 610 able to extract columns and rows from the ledger pages and thus  
 611 recognize the table cells with a mean F1-score of 0.844 and mean  
 612 IoU value of 0.473 when using an IoU threshold of 0.25. This method  
 613 improves the layout error where the word order was not recognized  
 614 at all by the HTR system. We show a CER of 0.3859 when match-  
 615 ing predicted table cells with ground truth cells based on an IoU  
 616 threshold of 0.25, which is not far of the CER found in the initial  
 617 error analysis, showing the effectiveness of our layout analysis.

618 We have also shown that a deep learning model such as Layout  
 619 Parser can be effective in recognizing the columns of our ledger  
 620 pages. We have created pseudo-annotations based on our rule-based  
 621 method and used these to train our model. We have shown that  
 622 even though noise is present in these pseudo-annotations, they can  
 623 still be effective in training a model when using a large quantity.  
 624 The model showed to have promising accuracy on the test set.

### 8.2 Limitations

626 Our rule-based method requires document homogeneity to work  
 627 effectively. Variations in document structure and quality can signif-  
 628 icantly impact the performance of the method used. While almost  
 629 all documents follow the same structure, there can exist outliers.

630 Furthermore, the evaluation of the table cell layout detection in  
 631 Section 6 proved to be difficult using a standard IoU threshold of 0.5  
 632 due to the fact that our rule-based method did not always accurately  
 633 match the ground truth table cells. Our method did perform well  
 634 using a lower threshold, showing a mean IoU value of 0.473 for the  
 635 matched cells, just below the usual threshold of 0.5. While we tried  
 636 to support our evaluation by also comparing text contents, this in  
 637 itself is again based on a different threshold. While this still gives  
 638 a good idea of our method's performance, it does not completely  
 639 validate the accuracy of the cell layout detection.

640 Additionally, In Section 7, we only tested the accuracy of the  
 641 models up until iteration 1000. While this shows the comparison

well between the models, we can see that the model trained on the large dataset has not fully converged yet, meaning that we might get a better accuracy when trained more. Since we use a batch size of 2, this model will only have seen the complete train set once, whereas the model trained on the small dataset will have already seen the train set three times. We chose to stop training at iteration 1000 since we were mainly interested in the comparison between the model's training phases on the pseudo-annotations, and not necessarily in getting the best possible accuracy.

### 8.3 Future Work

This work aims to detect common errors in a HTR model and lays a foundation to solve some of these errors. Future research could focus on the integration of the layout detection methods into the actual HTR model to create a structured pipeline for text detection. This could also involve testing whether running the HTR model on each column separately (either in numeric or alphabetic mode) would improve the accuracy of the output of the HTR model. Furthermore, it could be worthwhile to investigate whether models using a different architecture perform better on a layout detection task like this one. Lastly, the deep learning model trained with pseudo-annotations could be fine-tuned with hand annotated data to further improve its performance.

## 9 CONCLUSION

This work aimed to address the challenges associated with the digitization of handwritten ledgers, with a specific focus on error identification and resolution within HTR systems. Our main research question aimed to explore the extent to which categorizing errors in handwritten ledger analysis can enhance the identification and resolution of the HTR model inaccuracies.

Through a series of experiments, we identified key factors contributing to HTR errors and proposed various methods to improve the output of the model. Our analysis revealed several types of errors, including issues related to layout detection and text recognition. By implementing post-processing techniques and layout analysis methods, we observed notable improvements in the accuracy of the HTR output.

Despite these results, our study faced several limitations. One significant limitation was the difficulty in evaluating the table cell layout detection using a standard IoU threshold of 0.5. Our rule-based method did not consistently match the ground truth table cells, performing better with a lower IoU threshold. This indicates that while our method provides a good approximation of performance, it does not fully validate the accuracy of the cell layout detection.

In conclusion, this work contributes to the ongoing efforts to digitize historical handwritten documents by providing insights into the causes of HTR errors and proposing methods to mitigate them. While our findings show promising results for improving HTR accuracy, further research is necessary to fully overcome the challenges identified. By addressing the limitations and possibly exploring new methodologies, future work can build upon this work to improve the performance of HTR systems in digitizing historical documents.

## REFERENCES

- [1] Sofia Ares Oliveira, Benoit Seguin, and Frederic Kaplan. 2018. dhSegment: A Generic Deep-Learning Approach for Document Segmentation. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE. <https://doi.org/10.1109/icfhr-2018.2018.00011>
- [2] Marius Bulacu, Rutger van Koert, Lambert Schomaker, and Tijn van der Zant. 2007. Layout Analysis of Handwritten Historical Documents for Searching the Archive of the Cabinet of the Dutch Queen. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, Vol. 1. 357–361. <https://doi.org/10.1109/ICDAR.2007.4378732>
- [3] Hsiang-Chieh Chen and Zheng-Ting Li. 2021. Automated Ground Truth Generation for Learning-Based Crack Detection on Concrete Surfaces. *Applied Sciences* 11, 22 (2021). <https://doi.org/10.3390/app112210966>
- [4] Ahmad Drobly, Berat Kurar Barakat, Reem Alaasam, Boraq Madi, Irina Rabaev, and Jihad El-Sana. 2022. Text Line Extraction in Historical Documents Using Mask R-CNN. *Signals* 3, 3 (2022), 535–549. <https://doi.org/10.3390/signals3030032>
- [5] Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications* 165 (2021), 113679. <https://doi.org/10.1016/j.eswa.2020.113679>
- [6] Sotirios Kastanas, Shaomu Tan, and Yi He. 2023. Document AI: A Comparative Study of Transformer-Based, Graph-Based Models, and Convolutional Neural Networks For Document Layout Analysis. arXiv:2308.15517 [cs.CL]
- [7] Constantin Lehennmeier, Manuel Burghardt, and Bernadette Mischka. 2020. *Layout Detection and Table Recognition – Recent Challenges in Digitizing Historical Documents and Handwritten Tabular Data*. Springer International Publishing, 229–242. [https://doi.org/10.1007/978-3-030-54956-5\\_17](https://doi.org/10.1007/978-3-030-54956-5_17)
- [8] Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. 2022. TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models. arXiv:2109.10282 [cs.CL]
- [9] Xusheng Liang, Abbas Cheddad, and Johan Hall. 2021. Comparative Study of Layout Analysis of Tabulated Historical Documents. *Big Data Research* 24 (2021), 100195. <https://doi.org/10.1016/j.bdr.2021.100195>
- [10] Huaming Liu, Xuehui Bi, and Weilan Wang. 2019. Layout analysis of historical Tibetan documents. In *2019 2nd International Conference on Artificial Intelligence and Big Data (ICAIBD)*. 74–78. <https://doi.org/10.1109/ICAIBD.2019.8837040>
- [11] Tengchao Lv, Yuyan Huang, Jingye Chen, Lei Cui, Shuming Ma, Yaoyao Chang, Shaohan Huang, Wenhui Wang, Li Dong, Weiyao Luo, Shaoliang Wu, Guoxin Wang, Cha Zhang, and Furu Wei. 2023. Kosmos-2.5: A Multimodal Literate Model. arXiv:2309.11419 [cs.CL]
- [12] Siddharth Misra and Yaokun Wu. 2020. Chapter 10 - Machine learning assisted segmentation of scanning electron microscopy images of organic-rich shales with feature extraction and feature ranking. In *Machine Learning for Subsurface Characterization*, Siddharth Misra, Hao Li, and Jiabo He (Eds.). Gulf Professional Publishing, 289–314. <https://doi.org/10.1016/B978-0-12-817736-5.00010-7>
- [13] Arthur Flor de Sousa Neto, Byron Leite Dantas Bezerra, and Alejandro Héctor Toselli. 2020. Towards the Natural Language Processing as Spelling Correction for Offline Handwritten Text Recognition Systems. *Applied Sciences* 10, 21 (2020). <https://doi.org/10.3390/app10217711>
- [14] OpenCV. 2024. Morphological Transformations. [https://docs.opencv.org/3.4/dd/d7/tutorial\\_morph\\_lines\\_detection.html](https://docs.opencv.org/3.4/dd/d7/tutorial_morph_lines_detection.html) Accessed: 2024-04-18.
- [15] Rafael Padilla, Sergio Netto, and Eduardo da Silva. 2020. A Survey on Performance Metrics for Object-Detection Algorithms. <https://doi.org/10.1109/IWSSIP48289.2020.910020>
- [16] Rémi Petitpierre, Marion Kramer, and Lucas Rappo. 2023. An end-to-end pipeline for historical censuses processing. *International Journal on Document Analysis and Recognition (IJDAR)* 26, 4 (March 2023), 419–432. <https://doi.org/10.1007/s10032-023-00428-9>
- [17] Jose Ramón Prieto, José Andrés, Emilio Granell, Joan Andreu Sánchez, and Enrique Vidal. 2023. Information extraction in handwritten historical logbooks. *Pattern Recognition Letters* 172 (2023), 128–136. <https://doi.org/10.1016/j.patrec.2023.06.008>
- [18] Stephen Quinn and William Roberds. 2009. *An economic explanation of the early Bank of Amsterdam, debasement, bills of exchange and the emergence of the first central bank*. Cambridge University Press, 32–70.
- [19] Mathias Seuret. [n.d.]. *Layout Analysis in Handwritten Historical Documents*. Chapter Chapter 4, 45–65. [https://doi.org/10.1142/9789811203244\\_0004](https://doi.org/10.1142/9789811203244_0004) arXiv:[https://www.worldscientific.com/doi/pdf/10.1142/9789811203244\\_0004](https://www.worldscientific.com/doi/pdf/10.1142/9789811203244_0004)
- [20] Zejiang Shen, Ruochen Zhang, Melissa Dell, Benjamin Charles Germain Lee, Jacob Carlson, and Weining Li. 2021. LayoutParser: A Unified Toolkit for Deep Learning Based Document Image Analysis. arXiv:2103.15348 [cs.CV]
- [21] Brandon Smock, Rohith Pesala, and Robin Abraham. 2022. PubTables-1M: Towards Comprehensive Table Extraction From Unstructured Documents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4634–4642.
- [22] Enrique Vidal, Alejandro H. Toselli, Antonio Ríos-Vila, and Jorge Calvo-Zaragoza. 2023. End-to-End page-Level assessment of handwritten text recognition. *Pattern Recognition* 142 (2023), 109695. <https://doi.org/10.1016/j.patcog.2023.109695>

- 772 [23] Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha  
773 Zhang, and Furu Wei. 2021. LayoutXLM: Multimodal Pre-training for Multilingual  
774 Visually-rich Document Understanding. arXiv:[2104.08836](https://arxiv.org/abs/2104.08836) [cs.CL]

775 Appendix A SUPPLEMENTARY FIGURES

Jan Van Os d. 8.		Cred. ady 4 Augt 1684	
Buy	van Cruij Potte	1684	1684
Mr. Bontje	225	2000	2000
Mr. Dordt	120	592 10	2800
Mr. E. Parus	33	497 9	14
Mr. G. Verwoert	33	1947 19	462
Mr. H. Verwoert	1184	9710	21318 11
Mr. J. Bontje	1126	5285	2000
Brueghel Graecken	1016	850	1263
Graecken Soverijn	959	1635	2800
Hendrik van Dordt	900	1075	1000
Jan. Mondt	313	1630 15	108
Mr. Bontje	225	1250 15	1200
Mr. Bontje	139	1140	1320
Car. & Sons Gerard	15	2729 16 0	1000
Dick van Dordt	302	600	202
Jan. Mondt	314	1200	8
Jos. Van Geraard	1099	19318 11 0	39036 9
Jan. Mondt	877	2518 15	1000
Jan. Mondt	315	3108 18	1200
Mr. Bontje	225	809 10	6000
Wm. Bontje	121	1340 12 0	950
Wm. Bontje	1148	28288 9 0	56496 9
Wm. Bontje	128	350	300
Wm. Bontje	1039	1000	3000
Wm. Bontje	1012	2600	123 15
Wm. Bontje	369	9 9	2000
Samuel de la Brae Land	113	33507 13 8	6425 4
Matz Stobbe	107	960	1000
Louis Goris	121	900	1583 10
Jan. Son. Minster	103	2958	300
as. Arout	400	500	300
Jan. van Dordt	702	9025 13 0	5042 10
Floro Garlick	979	300	1500
Floro Garlick	44	800	1000
Floro Garlick	43	1600	300
Floro Garlick	1087	2350	300
Floro Garlick	600	126	1000
Floro Garlick	363	51200 1 0	1000
Floro Garlick	352	1854 1	1000
Floro Garlick	268	1060	1000
Floro Garlick	302	2000	1000
Floro Garlick	985	2400	1000
Floro Garlick	979	51921 2 8	1000
Floro Garlick	992	1100	1000
Floro Garlick	425	1500	1000
Floro Garlick	372	1600	1000
Floro Garlick	939	839 8	1000
Floro Garlick	395	63660 10 8	1000
Floro Garlick	310	1011 10 0	1000
Floro Garlick	252	1053 1	1000
Floro Garlick	149	1000	1000
Floro Garlick	61	2565	1000
Floro Garlick	437	62941 14 8	1000
Floro Garlick	437	16575 2 8	1000
Floro Garlick	84513	1 8	1000

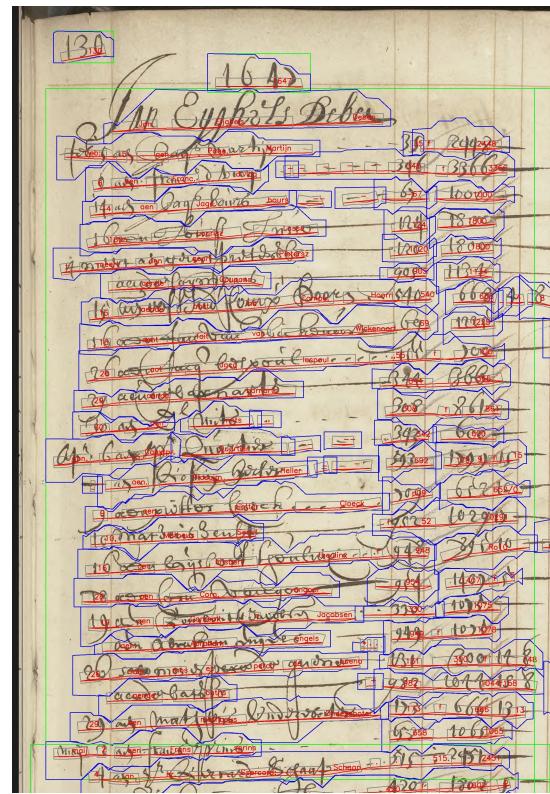
Figure 7: An example page from the ledger collection.

130 164

*Mr. Ephah's Barber*

1. Mr. Ephah's Barber	310	2442
2. Dan & Martin	2449	35662
3. Paul Kowalski	67	160495
4. Tom & Tom's Finest	171	150903
5. Mr. Ephah's Barber	129	17399
6. account payable	67	112729
7. Mr. Ephah's Barber Books	520	16294
8. Mr. Ephah's Barber Books	69	10118
9. Mr. Ephah's Barber Books	108	2060
10. account payable	824	2664
11. account payable	803	861
12. Tom & Tom's Finest	243	6595
13. Mr. Ephah's Barber	232	1294911
14. Dan & Martin	159	610910
15. account payable	102907	
16. Mr. Ephah's Barber	142	34919
17. Mr. Ephah's Barber Books	124	140731
18. Mr. Ephah's Barber Books	239	10739
19. Mr. Ephah's Barber Books	129	1092
20. Mr. Ephah's Barber Books	131	108146
21. Mr. Ephah's Barber Books	229	1044115
22. Mr. Ephah's Barber Books	123	66673
23. Mr. Ephah's Barber Books	128	10693
24. Mr. Ephah's Barber Books	119	25917
25. Mr. Ephah's Barber Books	120	170295

(a) The PageXML data of our ground truth plotted on part of the original image.



(b) The PageXML data of the HTR system plotted on part of the original image.

Figure 8: The PageXML data of the ground truth and HTR system plotted on the original image. The tabular structure of the ground truth values can be seen where this is not the case in the HTR output.

130	164)	130
Mr Eynh's Dibbs	More Reddys Corvinae	438116
130 299 2	92 922	
140 336 6	52 259 2 8	
150 100 1	29 246 16	
160 780 0	31 1220 19	
170 180 0	24 600	
180 112 4	22 266 2 2	
190 66 8 4 8	65 160 0	
200 200 0	20 93 16	
210 120 0	30 300	
220 200 0	26 200	
230 100 0	21 1049 2 8	
240 200 0	12 651 10	
250 660 1	15 151 18	
260 122 15	13 5 200	
270 102 9 6	28 151 18	
280 210 0	32 2146 9	
290 140 1	48 133 1 4	
300 109 5 1	12 300	
310 107 8	101 1839 2 8	
320 130 1 18	32 5 10	
330 104 1 18	104 400 9 15	
340 666 12	101 2 6 1	
350 106 0	115 2400	
360 240 1	12 1049 18	
370 130 2	963 133 6	
380 100 2 1	29 108 14	
390 100 2 1	141 329 1	
400 100 2 1	109 133 6	
410 100 2 1	152 133 2 10	
420 100 2 1	119 144 0	
430 124 6 1 8	104 500	
440 57 1	112 100	
450 150 1	29 294 2 8	
460 100 2 1	291 20 0	
470 100 2 1	102 250 5	
480 100 2 1	102 9 9 15	
490 100 2 1	166 124 8 15	
500 100 2 1	199 600	
510 100 2 1	125 242 10	
520 100 2 1	123 219 6 15	
530 100 2 1	126 19 6 13	
540 200 0	121 219 6 13	
550 100 2 1	124 219 6 13	
560 100 2 1	125 219 6 13	
570 100 2 1	126 219 6 13	
580 100 2 1	127 219 6 13	
590 100 2 1	128 219 6 13	
600 100 2 1	129 219 6 13	
610 100 2 1	130 219 6 13	
620 100 2 1	131 219 6 13	
630 100 2 1	132 219 6 13	
640 100 2 1	133 219 6 13	
650 100 2 1	134 219 6 13	
660 100 2 1	135 219 6 13	
670 100 2 1	136 219 6 13	
680 100 2 1	137 219 6 13	
690 100 2 1	138 219 6 13	
700 100 2 1	139 219 6 13	
710 100 2 1	140 219 6 13	
720 100 2 1	141 219 6 13	
730 100 2 1	142 219 6 13	
740 100 2 1	143 219 6 13	
750 100 2 1	144 219 6 13	
760 100 2 1	145 219 6 13	
770 100 2 1	146 219 6 13	
780 100 2 1	147 219 6 13	
790 100 2 1	148 219 6 13	
800 100 2 1	149 219 6 13	
810 100 2 1	150 219 6 13	
820 100 2 1	151 219 6 13	
830 100 2 1	152 219 6 13	
840 100 2 1	153 219 6 13	
850 100 2 1	154 219 6 13	
860 100 2 1	155 219 6 13	
870 100 2 1	156 219 6 13	
880 100 2 1	157 219 6 13	
890 100 2 1	158 219 6 13	
900 100 2 1	159 219 6 13	
910 100 2 1	160 219 6 13	
920 100 2 1	161 219 6 13	
930 100 2 1	162 219 6 13	
940 100 2 1	163 219 6 13	
950 100 2 1	164 219 6 13	
960 100 2 1	165 219 6 13	
970 100 2 1	166 219 6 13	
980 100 2 1	167 219 6 13	
990 100 2 1	168 219 6 13	
1000 100 2 1	169 219 6 13	
1010 100 2 1	170 219 6 13	
1020 100 2 1	171 219 6 13	
1030 100 2 1	172 219 6 13	
1040 100 2 1	173 219 6 13	
1050 100 2 1	174 219 6 13	
1060 100 2 1	175 219 6 13	
1070 100 2 1	176 219 6 13	
1080 100 2 1	177 219 6 13	
1090 100 2 1	178 219 6 13	
1100 100 2 1	179 219 6 13	
1110 100 2 1	180 219 6 13	
1120 100 2 1	181 219 6 13	
1130 100 2 1	182 219 6 13	
1140 100 2 1	183 219 6 13	
1150 100 2 1	184 219 6 13	
1160 100 2 1	185 219 6 13	
1170 100 2 1	186 219 6 13	
1180 100 2 1	187 219 6 13	
1190 100 2 1	188 219 6 13	
1200 100 2 1	189 219 6 13	
1210 100 2 1	190 219 6 13	
1220 100 2 1	191 219 6 13	
1230 100 2 1	192 219 6 13	
1240 100 2 1	193 219 6 13	
1250 100 2 1	194 219 6 13	
1260 100 2 1	195 219 6 13	
1270 100 2 1	196 219 6 13	
1280 100 2 1	197 219 6 13	
1290 100 2 1	198 219 6 13	
1300 100 2 1	199 219 6 13	
1310 100 2 1	200 219 6 13	
1320 100 2 1	201 219 6 13	
1330 100 2 1	202 219 6 13	
1340 100 2 1	203 219 6 13	
1350 100 2 1	204 219 6 13	
1360 100 2 1	205 219 6 13	
1370 100 2 1	206 219 6 13	
1380 100 2 1	207 219 6 13	
1390 100 2 1	208 219 6 13	
1400 100 2 1	209 219 6 13	
1410 100 2 1	210 219 6 13	
1420 100 2 1	211 219 6 13	
1430 100 2 1	212 219 6 13	
1440 100 2 1	213 219 6 13	
1450 100 2 1	214 219 6 13	
1460 100 2 1	215 219 6 13	
1470 100 2 1	216 219 6 13	
1480 100 2 1	217 219 6 13	
1490 100 2 1	218 219 6 13	
1500 100 2 1	219 219 6 13	
1510 100 2 1	220 219 6 13	
1520 100 2 1	221 219 6 13	
1530 100 2 1	222 219 6 13	
1540 100 2 1	223 219 6 13	
1550 100 2 1	224 219 6 13	
1560 100 2 1	225 219 6 13	
1570 100 2 1	226 219 6 13	
1580 100 2 1	227 219 6 13	
1590 100 2 1	228 219 6 13	
1600 100 2 1	229 219 6 13	
1610 100 2 1	230 219 6 13	
1620 100 2 1	231 219 6 13	
1630 100 2 1	232 219 6 13	
1640 100 2 1	233 219 6 13	
1650 100 2 1	234 219 6 13	
1660 100 2 1	235 219 6 13	
1670 100 2 1	236 219 6 13	
1680 100 2 1	237 219 6 13	
1690 100 2 1	238 219 6 13	
1700 100 2 1	239 219 6 13	
1710 100 2 1	240 219 6 13	
1720 100 2 1	241 219 6 13	
1730 100 2 1	242 219 6 13	
1740 100 2 1	243 219 6 13	
1750 100 2 1	244 219 6 13	
1760 100 2 1	245 219 6 13	
1770 100 2 1	246 219 6 13	
1780 100 2 1	247 219 6 13	
1790 100 2 1	248 219 6 13	
1800 100 2 1	249 219 6 13	
1810 100 2 1	250 219 6 13	
1820 100 2 1	251 219 6 13	
1830 100 2 1	252 219 6 13	
1840 100 2 1	253 219 6 13	
1850 100 2 1	254 219 6 13	
1860 100 2 1	255 219 6 13	
1870 100 2 1	256 219 6 13	
1880 100 2 1	257 219 6 13	
1890 100 2 1	258 219 6 13	
1900 100 2 1	259 219 6 13	
1910 100 2 1	260 219 6 13	
1920 100 2 1	261 219 6 13	
1930 100 2 1	262 219 6 13	
1940 100 2 1	263 219 6 13	
1950 100 2 1	264 219 6 13	
1960 100 2 1	265 219 6 13	
1970 100 2 1	266 219 6 13	
1980 100 2 1	267 219 6 13	
1990 100 2 1	268 219 6 13	
2000 100 2 1	269 219 6 13	
2010 100 2 1	270 219 6 13	
2020 100 2 1	271 219 6 13	
2030 100 2 1	272 219 6 13	
2040 100 2 1	273 219 6 13	
2050 100 2 1	274 219 6 13	
2060 100 2 1	275 219 6 13	
2070 100 2 1	276 219 6 13	
2080 100 2 1	277 219 6 13	
2090 100 2 1	278 219 6 13	
2100 100 2 1	279 219 6 13	
2110 100 2 1	280 219 6 13	
2120 100 2 1	281 219 6 13	
2130 100 2 1	282 219 6 13	
2140 100 2 1	283 219 6 13	
2150 100 2 1	284 219 6 13	
2160 100 2 1	285 219 6 13	
2170 100 2 1	286 219 6 13	
2180 100 2 1	287 219 6 13	
2190 100 2 1	288 219 6 13	
2200 100 2 1	289 219 6 13	
2210 100 2 1	290 219 6 13	
2220 100 2 1	291 219 6 13	
2230 100 2 1	292 219 6 13	
2240 100 2 1	293 219 6 13	
2250 100 2 1	294 219 6 13	
2260 100 2 1	295 219 6 13	
2270 100 2 1	296 219 6 13	
2280 100 2 1	297 219 6 13	
2290 100 2 1	298 219 6 13	
2300 100 2 1	299 219 6 13	
2310 100 2 1	300 219 6 13	
2320 100 2 1	301 219 6 13	
2330 100 2 1	302 219 6 13	
2340 100 2 1	303 219 6 13	
2350 100 2 1	304 219 6 13	
2360 100 2 1	305 219 6 13	
2370 100 2 1	306 219 6 13	
2380 100 2 1	307 219 6 13	
2390 100 2 1	308 219 6 13	
2400 100 2 1	309 219 6 13	
2410 100 2 1	310 219 6 13	
2420 100 2 1	311 219 6 13	
2430 100 2 1	312 219 6 13	
2440 100 2 1	313 219 6 13	
2450 100 2 1	314 219 6 13	
2460 100 2 1	315 219 6 13	
2470 100 2 1	316 219 6 13	
2480 100 2 1	317 219 6 13	
2490 100 2 1	318 219 6 13	
2500 100 2 1	319 219 6 13	
2510 100 2 1	320 219 6 13	
2520 100 2 1	321 219 6 13	
2530 100 2 1	322 219 6 13	
2540 100 2 1	323 219 6 13	
2550 100 2 1	324 219 6 13	
2560 100 2 1	325 219 6 13	
2570 100 2 1	326 219 6 13	
2580 100 2 1	327 219 6 13	
2590 100 2 1	328 219 6 13	
2600 100 2 1	329 219 6 13	
2610 100 2 1	330 219 6 13	
2620 100 2 1	331 219 6 13	
2630 100 2 1	332 219 6 13	
2640 100 2 1	333 219 6 13	
2650 100 2 1	334 219 6 13	
2660 100 2 1	335 219 6 13	
2670 100 2 1	336 219 6 13	
2680 100 2 1	337 219 6 13	
2690 100 2 1	338 219 6 13	
2700 100 2 1	339 219 6 13	
2710 100 2 1	340 219 6 13	
2720 100 2 1	341 219 6 13	
2730 100 2 1	342 219 6 13	
2740 100 2 1	343 219 6 13	
2750 100 2 1	344 219 6 13	
2760 100 2 1	345 219 6 13	
2770 100 2 1	346 219 6 13	
2780 100 2 1	347 219 6 13	
2790 100 2 1	348 219 6 13	
2800 100 2 1	349 219 6 13	
2810 100 2 1	350 219 6 13	
2820 100 2 1	351 219	

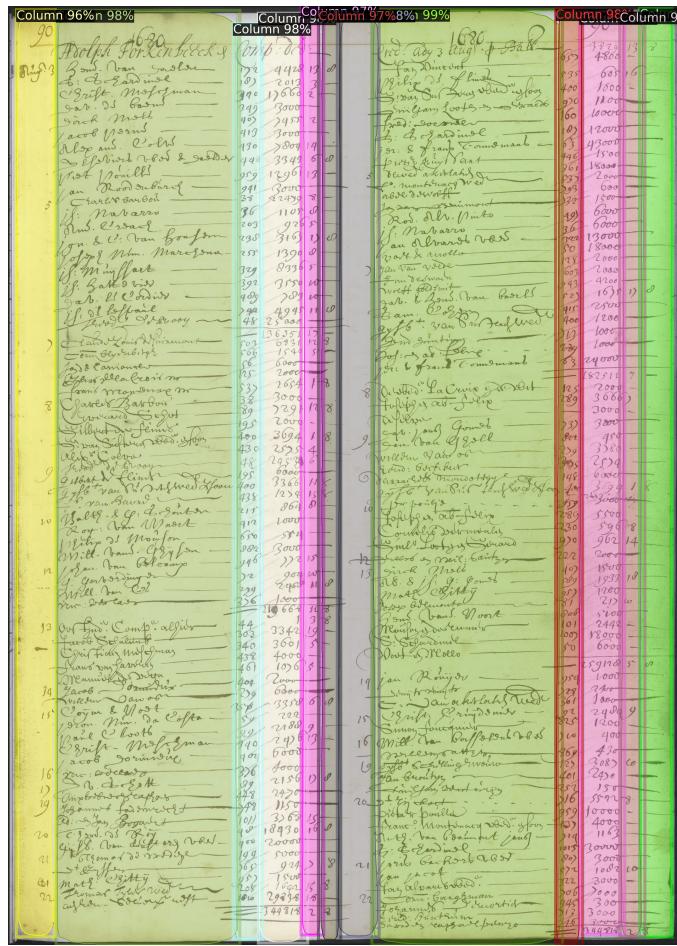


Figure 10: An example page showing a prediction of columns made by the Layout Parser model trained on our large dataset with pseudo-annotations.

776 Appendix B DETAILED METHODOLOGY

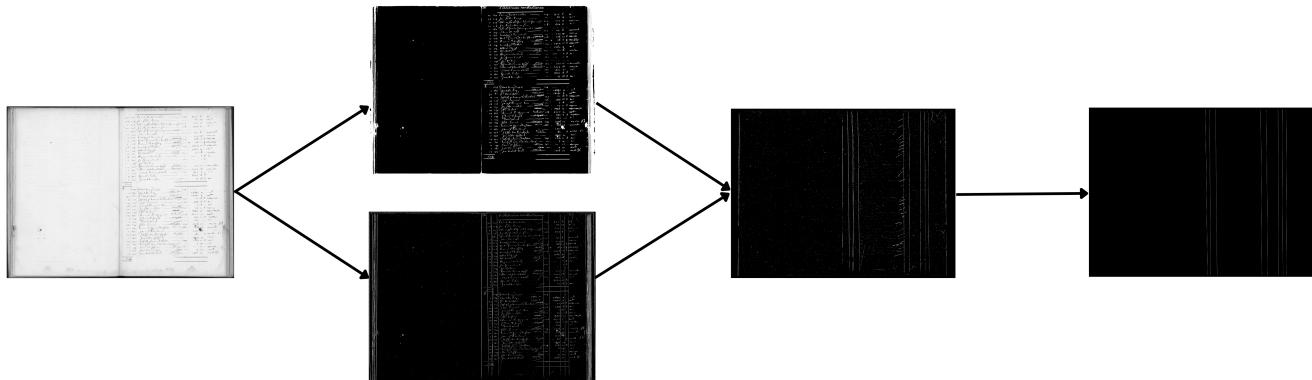


Figure 11: The process of the extraction of column lines. We start with the blurred grayscale image. From there we create two threshold images, one containing the faint column lines and one that does not contain them. We subtract the images from each other, remove noise and use Hough line detector to find the columns lines.



Figure 12: The process of extracting the rows from the ledger page. We use the HTR system to determine bounding boxes of words from the original page. We then determine the row bounding boxes based on the y-coordinates of the word bounding boxes.

777 **Appendix C COMPLEMENTARY RESULT TABLES**

**Table 8: Table Cell Performance Metrics at IoU Threshold 0.5.**

Subdir	IoU Score	Precision	Recall	F1
1045	0.638	0.674	0.674	0.674
1390	0.641	0.303	0.468	0.363
62	0.627	0.437	0.487	0.460
149	0.608	0.310	0.383	0.342
1099	0.621	0.532	0.499	0.515
99	0.638	0.525	0.504	0.514
109	0.645	0.520	0.552	0.534
119	0.617	0.422	0.437	0.429
139	0.626	0.435	0.475	0.453
59	0.664	0.555	0.541	0.548
89	0.632	0.466	0.519	0.490
69	0.626	0.391	0.456	0.420
55	0.631	0.287	0.319	0.302
79	0.614	0.389	0.439	0.412
1016	0.661	0.637	0.635	0.636
199	0.594	0.243	0.339	0.278
129	0.617	0.342	0.404	0.369
<b>Mean</b>	0.627	0.416	0.459	0.434

**Table 9: Table Cell Performance Metrics at IoU Threshold 0.25.**

Subdir	IoU Score	Precision	Recall	F1
1045	0.546	0.990	0.990	0.990
1390	0.498	0.513	0.785	0.613
62	0.471	0.854	0.959	0.902
149	0.440	0.710	0.879	0.784
1099	0.473	0.939	0.881	0.909
99	0.504	0.916	0.878	0.896
109	0.508	0.848	0.894	0.867
119	0.460	0.869	0.901	0.885
139	0.474	0.819	0.894	0.853
59	0.525	0.910	0.888	0.899
89	0.475	0.862	0.961	0.907
69	0.464	0.821	0.963	0.884
55	0.451	0.726	0.811	0.765
79	0.452	0.829	0.929	0.875
1016	0.553	0.835	0.831	0.832
199	0.432	0.639	0.894	0.731
129	0.453	0.738	0.878	0.801
<b>Mean</b>	0.473	0.804	0.898	0.844

**Table 10: Table Cell Performance Metrics at IoU Threshold >0**

<b>Subdir</b>	<b>IoU Score</b>	<b>Precision</b>	<b>Recall</b>	<b>F1</b>
1045	0.196	1.000	1.000	1.000
1390	0.152	0.603	0.952	0.729
62	0.190	0.876	0.984	0.925
149	0.169	0.762	0.943	0.841
1099	0.206	0.945	0.886	0.915
99	0.177	0.960	0.920	0.939
109	0.188	0.885	0.933	0.905
119	0.178	0.906	0.939	0.922
139	0.185	0.860	0.938	0.896
59	0.192	0.946	0.923	0.934
89	0.186	0.873	0.974	0.919
69	0.195	0.839	0.984	0.903
55	0.172	0.820	0.911	0.862
79	0.182	0.863	0.967	0.911
1016	0.232	0.845	0.840	0.841
199	0.169	0.690	0.962	0.788
129	0.164	0.802	0.952	0.869
<b>Mean</b>	0.182	0.845	0.946	0.887