

**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 12.04.2024

	<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 Write your abstract here.

3 **KEYWORDS**

4 keywords, belong, here, with, commas, like, this

5 **GITHUB REPOSITORY**

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

7 **1 INTRODUCTION**

8 The Bank of Amsterdam was an early bank established in 1609, de-  
9 scribed by some as the first central bank [20]. The bank played an  
10 important role in the financial world of the 17th and 18th century.  
11 Transactions made at the bank from 1650 to 1800 were recorded in  
12 ledgers and are still preserved to this day. These ledgers contain  
13 information about transactions, both debit and credit, of customers  
14 of the bank. The ledgers can thus be very important in analyzing  
15 the money flow at the time. Digitizing these ledgers may help in  
16 preserving them in a much easier to analyze format which would  
17 allow for more research into the contents of the ledgers. However,  
18 these ledgers are all handwritten making it hard to retrieve the  
19 information out of them on large scale. Current efforts to retrieve  
20 the information are done by using Handwritten Text Recognition  
21 (HTR) tools to extract the text in the ledgers. The problem is that  
22 the currently used HTR techniques have too many inaccuracies to  
23 be used effectively. Because of this, the handwritten text cannot  
24 be recognized correctly and can thus not be digitized properly. To  
25 improve digitization efforts, it is crucial to find out why some hand-  
26 written text cannot be recognized, what kind of errors are made  
27 during the HTR process and how the HTR process can be optimized  
28 to improve these inaccuracies. The research question we wish to  
29 answer is:

30  
31 *How can the categorization of errors in handwritten ledger analysis  
32 be used to enhance the identification and resolution of text recognition  
33 inaccuracies?*

34 Subquestions belonging to this research question are:

- 35
- What factors contribute to text recognition errors in hand-  
written ledgers?
  - What are the different types of errors encountered in hand-  
written text recognition?
  - How can specific preprocessing methods improve text detec-  
tion and segmentation in handwritten ledgers?
  - What strategies can be implemented to mitigate common  
text recognition errors in handwritten ledgers?

44 **2 RELATED WORK**

45 The research gap being addressed in this project revolves around the  
46 challenges associated with the digitization of handwritten ledgers.  
47 The proposed research will contribute in closing the research gap  
48 within the HTR domain, specifically looking at historical handwrit-  
49 ten documents. The digitization of handwritten ledgers contains

50 several steps. In this section, we will look at key research papers  
51 that discuss different approaches to the steps of our problem.

52 **2.1 Document Layout Analysis**

53 The first challenge in our digitization task is being able to find the  
54 correct word order from the HTR output. Seuret describes layout  
55 analysis and finding the correct word order as one of the main  
56 challenges in HTR [21]. Several approaches to solve this problem  
57 have been tried. These approaches are typically divided in either  
58 deep learning methods or more classical computer vision methods.  
59 We will show recent advancements made in both approaches.

60 Kastanas et al. show the use of different deep learning archi-  
61 tectures for layout analysis [8]. Transformer-based, graph-based  
62 models, and convolutional neural networks (CNN) are compared.  
63 The CNN-based model YOLOv5 and transformed-based model Lay-  
64 outLMv3 both show promising results for identifying most elements  
65 on documents, with YOLOv5 performing slightly better [8].

66 Multiple other studies describe the use of transformer based  
67 methods to perform layout analysis and optical character recog-  
68 nition (OCR) [10, 13, 24]. Furthermore, Smock et al. show that  
69 transformer-based object detection models can produce excellent re-  
70 sults for the tasks of detection, structure recognition, and functional  
71 analysis of tables [22]. While these approaches show promising  
72 results, they are not tested for our specific task at hand.

73 Oliveira et al. propose dhSegment, an open-source implemen-  
74 tation of a CNN-based pixel-wise predictor for document segmen-  
75 tation. The approach aims to address various document processing  
76 tasks simultaneously, including page extraction, baseline extraction,  
77 layout analysis, and illustration and photograph extraction. They  
78 show that a single CNN architecture can be used across tasks with  
79 competitive results [2].

80 Drobny et al. propose a holistic method that applies Mask R-CNN  
81 for text line extraction in historical documents. Their work achieved  
82 state-of-the-art results on well known historical datasets [6]. While  
83 their work does not directly transfer to our task, it does show the  
84 potential for document segmentation and possibly table recogni-  
85 tion abilities of Mask R-CNN.

86 The aforementioned studies show that the use of deep learning  
87 methods can be used to perform layout analysis as well as table  
88 recognition tasks in historical documents. However, these methods  
89 often require massive amounts of training data to be used effectively.  
90 To use these methods, we should look whether pre-trained models  
91 transfer well to our dataset and are able to detect the layout of our  
92 documents. Other approaches use more classical computer vision  
93 techniques to segment a document in order to recognize the layout.  
94 These methods often require homogeneity of the documents to  
95 work effectively.

96 Lehenmeier et al. show a method that combines various state-  
97 of-the-art methods for OCR pro- cessing to detect layout on a  
98 document [9]. To perform table recognition, they make use of tech-  
99 niques such as binarization, denoising, and Hough line detection

100 to find relevant parts of the tables. By taking the intersection of  
101 relevant horizontal and vertical Hough lines they are able to find  
102 table cells and thus determine the layout of the table [9].

103 Bulacu et al. introduce a method based on contour tracing that  
104 generates curvilinear separation paths between text lines in order to  
105 preserve the ascenders and descenders of text lines to determine the  
106 layout in handwritten historical documents [3]. With this approach  
107 they are able to determine separate elements of tables in historical  
108 documents.

109 Liu et al. describe the use of several key steps, including skew  
110 correction, region segmentation, and page layout analysis on his-  
111 torical Tibetan documents [12]. Skew correction is achieved by  
112 leveraging baseline features and Hough transform to determine  
113 the document's skew angle. Subsequently, region segmentation is  
114 accomplished by identifying borders through a series of preproces-  
115 sing steps, including median filtering, Gaussian smoothing, Sobel  
116 edge detection, and removal of small area regions. These processes  
117 collectively enable accurate positioning of document borders for  
118 further analysis and structure extraction [12].

119 Liang et al. describe the use of Hessian and Gabor filters in  
120 order to find table structures on a document [11]. They show the  
121 possibility for column segmentation for the tables using the found  
122 lines.

123 Prieto et al. address the challenge of document image understand-  
124 ing in documents with complex layouts like tables. They compare  
125 two approaches and show promising results [19]. Their approaches  
126 take the text lines outputted by HTR systems and use machine  
127 learning methods to group these text lines and classify cells based  
128 on these grouped text lines in order to find the tabular structure.

129 It can be seen that depending on the dataset and the homogene-  
130 ity of the documents, different approaches for layout analysis/table  
131 recognition may be suitable. When a similar table structure is clearly  
132 drawn on all the documents, approaches using basic image process-  
133 ing techniques could be suitable to determine the structure of the  
134 table. When the structure is not easily identifiable on the page, it  
135 might be better to look at approaches that use the output of HTR  
136 systems such as text lines or words locations to classify table cells.  
137 Machine learning based approaches could also be useful in these  
138 cases.

## 139 2.2 HTR Processing Techniques

140 Typical HTR pipelines consist of pre-processing an image to make  
141 it ready for HTR, running the image through the HTR system and  
142 post-processing the output of the HTR system [9].

143 Studies have shown that pre-processing images can lead to better  
144 results when using them as input for document layout analysis or  
145 OCR. Thus, we want to optimize our image material such that the  
146 document layout analysis and HTR can be performed more accu-  
147 rately. There are different methods for doing this such as adjusting  
148 the contrast to remove artifacts and noise or fixing the image align-  
149 ment [1, 5, 9]. It is crucial to find out which pre-processing steps  
150 might be useful to our problem at hand. Chen et al. show that the

151 use of several pre-processing steps including character segmenta-  
152 tion, tilt correction, offset correction, size normalization and image  
153 thinning lead to better HTR output results [4].

154 Post-processing OCR and HTR output typically consists of trying  
155 to correct misidentified words or characters. One strategy to do  
156 this is to create a dictionary of candidate characters and use the  
157 Levenshtein distance to calculate the distance between predicted  
158 and dictionary characters [18]. The use of a dictionary for allowed  
159 characters is shown to be a good strategy to eliminate HTR errors  
160 [9]. Other studies show the use of language models to correct word  
161 output and spelling in HTR systems [15, 18].

## 162 2.3 HTR Evaluation Strategies

163 Typically, HTR systems are evaluated using the word error rate  
164 (WER) and character error rate (CER). However, the evaluation  
165 of page-level HTR output faces challenges primarily due to the  
166 Reading Order (RO) problem [23]. In their study, they define a bag-  
167 of-words Word Error Rate (bWER) to accurately measure page-level  
168 RO-independent word errors as well as a regularized version of the  
169 Hungarian Algorithm (HA) to compute word- and character-level  
170 RO-independent recognition accuracy [23]. Another strategy is  
171 the use of the intersection over union (IoU) score which matches  
172 ground truth and HTR output boxes based on their coordinates  
173 on the page [17]. It is then possible to use the WER and CER to  
174 calculate the accuracy.

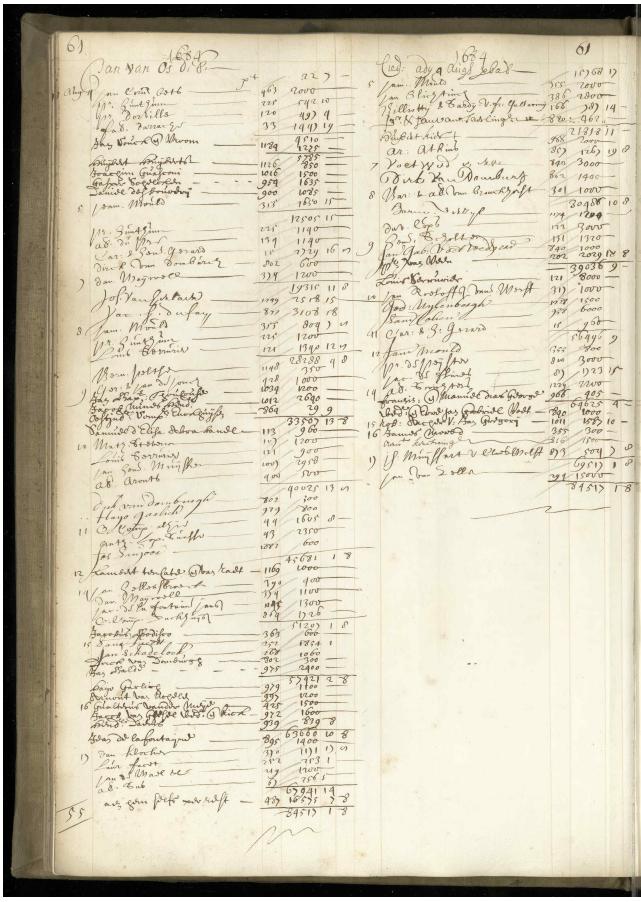
## 175 3 METHODOLOGY

176 This section describes the data and methods used to conduct this  
177 research.

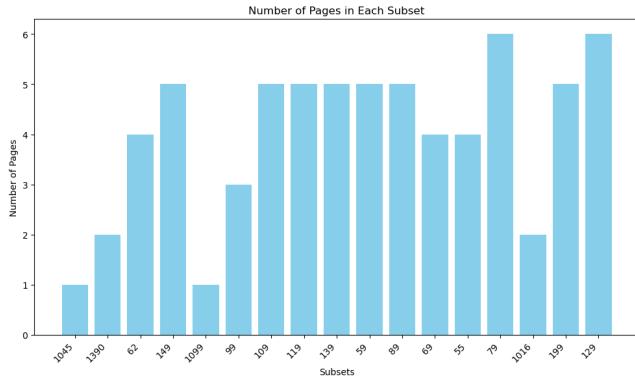
### 178 3.1 Data Description

179 The dataset used in this study consists of pages from the collection  
180 of ledgers from the Bank of Amsterdam. The full collection can be  
181 found on the [website](#) of The Amsterdam City Archives. In the series  
182 of ledgers, the current accounts of the customers were maintained.  
183 Each customer had one or more pages, with smaller traders having  
184 a portion of a page. An alphabetical list of the traders and the page  
185 of their current account can be found in the index. The ledgers kept  
186 track of the amounts that were debited and credited from and to a  
187 customer. The ledger pages consist of several columns containing  
188 information such as the date, account holder name, account number  
189 and amount debited/credited. Figure 1 shows an example of what  
190 a page looks like. Our dataset is split into multiple subsets that all  
191 have slightly varying pages, while still having structural similarity.  
192 Figure 2 shows the amount of pages for each subset in the dataset.  
193 The total dataset consists of 68 pages. A word count for each page  
194 is shown in Figure 3

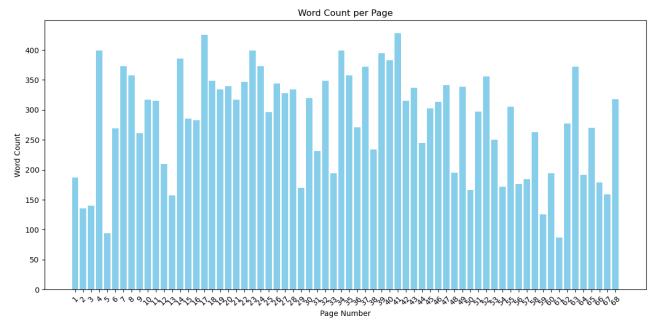
195 The ground truth and HTR system output both use a [PageXML](#)  
196 format to store values on the page. These PageXML store the coor-  
197 dinates for each item (e.g. textregion, tablerregion, etc.) on the page.  
198 We can plot the values in the PageXML file over our image to better  
199 show this. Figure 4 shows the PageXML data of the ground truth  
200 and HTR system plotted on the original image for part of one page  
201 in our dataset.



**Figure 1: An example page from the ledger collection.**



**Figure 2: The amount of pages for each subset in our dataset.**



**Figure 3: The word count for each page in our dataset.**

206     3.2.1 *Analyzing the HTR System.* Before attempting to improve  
 207     the HTR system, we will first conduct an extensive analysis to identify  
 208     where the current system fails or has flaws. This will involve  
 209     creating a categorization scheme that shows for each category the  
 210     nature of the error and a possible solution. Based on this scheme,  
 211     we can then try out various steps to improve the HTR system.

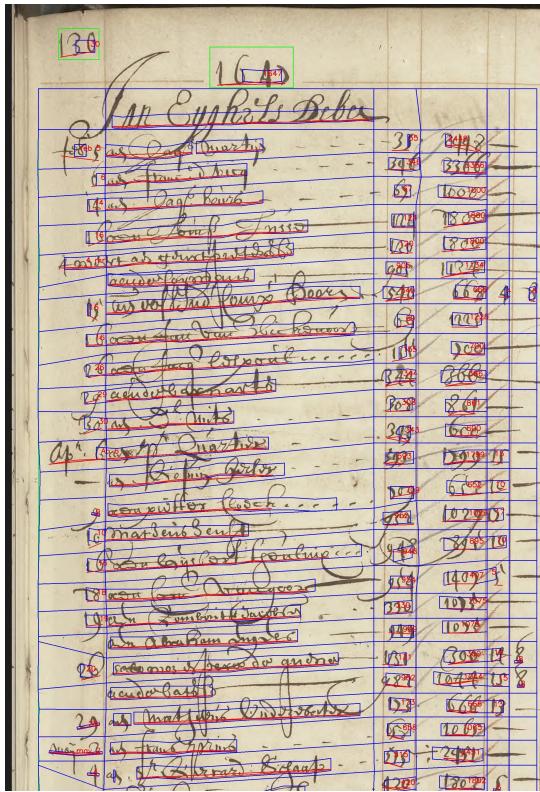
212     To obtain the accuracy of the HTR system, we will need to  
 213     compare the HTR output to the ground truth values. This in itself  
 214     is not straightforward, since the ground truth values do not contain  
 215     all the words in the image while the HTR output does. Furthermore,  
 216     since the layout of the page is not recognized by the HTR system,  
 217     the order of the words is completely different in the ground truth  
 218     and HTR output. To still compare the ground truth values and HTR  
 219     output, we will use an approach that matches the words based on  
 220     their bounding box coordinates on the image. To do this we make  
 221     use of the IoU score. This is a technique that can be used to find how  
 222     similar two bounding boxes are [17]. By matching each bounding  
 223     box in the ground truth with a corresponding bounding box in the  
 224     HTR output based on the highest IoU score, we can compare the  
 225     ground truth values and HTR output on a word to word basis.

226     Once we have aligned the words in the ground truth and the HTR  
 227     output properly, we can start to identify the flaws of the HTR system.  
 228     We will make use of the Levenshtein distance, which can be used  
 229     to find the minimum number of single-character edits (insertions,  
 230     deletions, or substitutions) required to change one string into the  
 231     other [7]. By applying this on our ground truth values and HTR  
 232     output on a word to word basis, we can track how many character  
 233     insertions, deletions and substitutions are done in the HTR output  
 234     for each word in the ground truth. We will show this for each subset  
 235     of our data to also see whether some subsets are more difficult for  
 236     the HTR system. We will also keep track of what characters are  
 237     inserted, deleted or substituted so we can find what characters  
 238     seem to be most problematic for the HTR system. The results of  
 239     this analysis is shown Section 4.1.

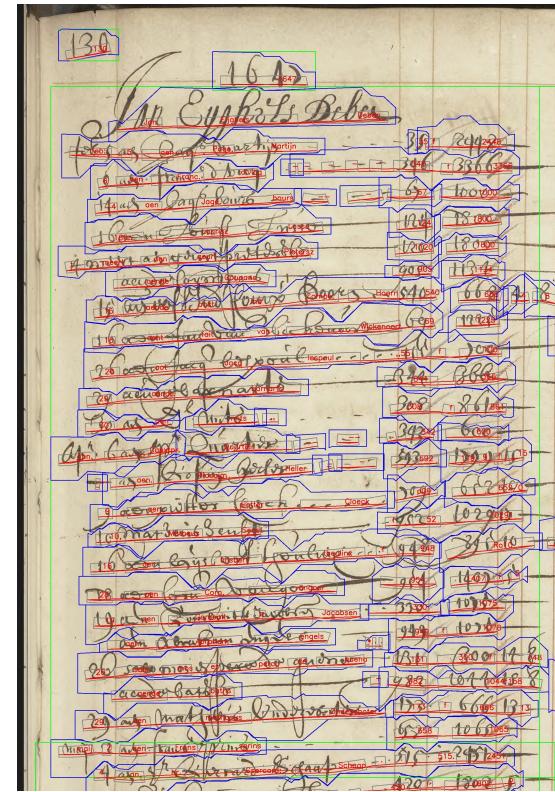
240     By creating this analysis, we can also better inspect our image  
 241     by visually showing the words that the HTR system can or cannot  
 242     match. An example of this visual representation is shown in Figure  
 243     13. Using this visual representation, we can inspect the image to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

## 3.2 Method

203     The methodology for this task is divided into two main parts: ana-  
 204     lyzing the current HTR system to identify its flaws, and performing  
 205     steps to mitigate the errors made by the HTR system.



(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 4:** 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.

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 findings of this analysis are shown in a categorization scheme in Table 1, showing the errors and possible solutions. The approach for these solutions is outlined in the next sections.

analysis to recover the right characters. The effectiveness of these post-processing techniques in improving the accuracy of the HTR output is shown in Section 4.2.

**3.2.2 Post-processing.** Post-processing in our experimental framework focuses on refining the output generated by the HTR system. One such post-processing step involves the removal of characters from the output strings that do not align with the ground truth dictionary [9, 18]. Analysis of the HTR output (as depicted in Figure 10a) 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 10c. 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

**3.2.3 Layout Analysis.** The final part of the methodology involves analyzing the layout of the documents. Recognizing the layout is the most important step in digitizing the ledger pages, since without it there is no good way to use the output of the HTR system because it is hard to tell what the structure of the page is. Since the ledger pages in our dataset show this structural similarity, we can use an approach using image processing techniques. For this, we will use the [OpenCV \(cv2\)](#) library to perform the necessary processing steps.

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 reducing the computational complexity and reduces high frequency noise [14].

289 Next, we apply two types of thresholding to the blurred image:  
 290 Otsu's thresholding and Adaptive Gaussian thresholding. Otsu's  
 291 thresholding only captures the most clear markings on the page,  
 292 such as the text, while the more thin and faint column lines are not  
 293 captured [3]. The Adaptive Gaussian threshold captures both the  
 294 text as well as the column lines. By subtracting the first threshold  
 295 from the latter, we are left with the column lines and some noise.

296 Following thresholding, we perform morphological operations,  
 297 specifically dilation and erosion [16]. We use specific kernels to  
 298 remove noise while retaining the vertical lines.

299 We then apply the Hough Line Transform to detect the lines  
 300 in the image. This can be used to detect (nearly) straight lines in  
 301 images [9]. We filter the detected lines based on their length and  
 302 position. For each detected line, we calculate its slope and intercept,  
 303 and then draw an extended line from the top to the bottom of the  
 304 foreground area (which contains the ledger page) in the image.  
 305 This is done using basic principles of line equations in coordinate  
 306 geometry.

307 Finally, we perform additional dilation and erosion to merge  
 308 close lines and remove final noise. We then find contours in the  
 309 image and draw a line from the top to the bottom of each contour.  
 310 We are then left with an image containing the contours of each  
 311 column section, which can then be used to create bounding boxes  
 312 for the different columns. The whole process is shown in Figure 5.

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

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

327 We then take a list of boxes (either left boxes or right boxes)  
 328 and group them into rows. This is done by sorting the boxes by  
 329 their x-coordinates (from left to right) and then grouping boxes  
 330 with similar y-coordinates together. For each box, we find the best  
 331 row to append the box to, based on the minimum difference in  
 332 y-coordinates. If no suitable row is found, we create a new row. The  
 333 result of these steps are shown in Figure 6.

334 Once we have found the rows, we can extend the bounding boxes  
 335 from the left side of the ledger page to the middle for the left boxes  
 336 and from the middle to the right side of the page for the right boxes.  
 337 We are now left with both the column and row bounding boxes.

338 Once the row and column bounding boxes are determined, we  
 339 can use the intersection of these bounding boxes to determine the  
 340 table cells. Figure 7 shows these found table cells.

### 3.3 Evaluation

341 The evaluation of the methodology will involve calculating various  
 342 metrics that measure the accuracy and effectiveness of the HTR  
 343 system and the layout analysis methods.

344 For the HTR system, metrics such as WER, CER, and Levenshtein  
 345 Distance will be calculated. These metrics measure the minimum  
 346 number of operations (substitutions, insertions, deletions) needed  
 347 to transform the HTR output into the ground truth. The formula  
 348 for these metrics is:

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

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

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

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

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

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

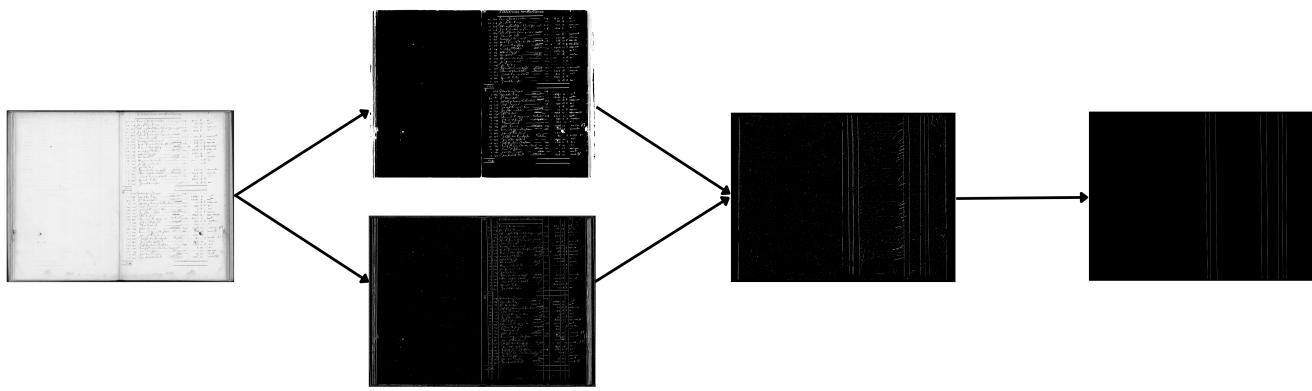
366 The recall and precision are then:

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

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

367 For the layout analysis methods, we take the ground truth table  
 368 cell bounding boxes and compare them to our method's predicted  
 369 table cell bounding boxes. We do this by matching each ground truth  
 370 cell to a predicted cell based on the highest IoU score between the  
 371 two. Only one ground truth cell can be matched to one predicted cell  
 372 and vice versa. We then determine whether cells are a true positive  
 373 by setting an IoU threshold. If ground truth and predicted cells are  
 374 matched with an IoU score higher than this threshold, we consider  
 375 it a true positive. Ground truth cells that are not matched with an  
 376 IoU score above the threshold are considered a false negative and  
 377 predicted cells that are not matched with an IoU score above the  
 378 threshold are considered a false positive. We will use a threshold  
 379 of 0.5, 0.25 and >0. We can then determine the recall and precision  
 380 scores, using the same formula as above. Additionally, we calculate  
 381 the F1-score:

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



**Figure 5:** 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 6:** 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.

**Figure 7:** The bounding boxes of the found table cells after taking the intersection of the rows and columns.

## 383 4 RESULTS

### 384 4.1 Analysis of HTR System Accuracy

385 The accuracy of the current HTR system is shown in Table 2. To  
386 calculate these accuracies we used the IoU method to match ground  
387 truth and HTR output words. A more in-depth analysis about the  
388 amount character insertions, deletions, substitutions and matches  
389 made by the HTR system can be seen in Figure 9 and Figure 8.  
390 Figure 10 shows the frequencies of characters inserted, substituted  
391 and deleted. Using these analyses, we can see more about the nature  
392 of the errors that the HTR system makes. We can see for instance  
393 that the characters '7' and '1' are often substituted. Furthermore,  
394 we can notice that there are ground truth annotations that contain  
395 spaces, where as this is not possible in the HTR output, because  
396 we select individual words. A selection of the most common errors  
397 and possible solutions is shown in Table 1.

416 and >0 are shown in Table 4, Table 5 and Table 6 respectively. The  
417 comparison of the total mean scores can be seen in Figure 12.

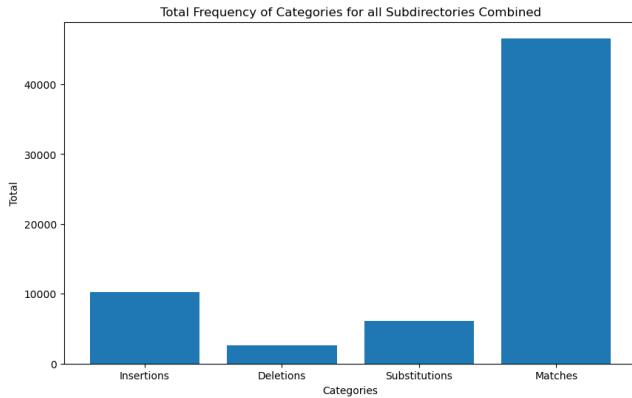


Figure 8: The total amount of insertions, deletions, substitutions and matches made by the HTR system compared to the ground truth values. The values are on a character level and made word per word.

### 398 4.2 Post-Processing Effectiveness

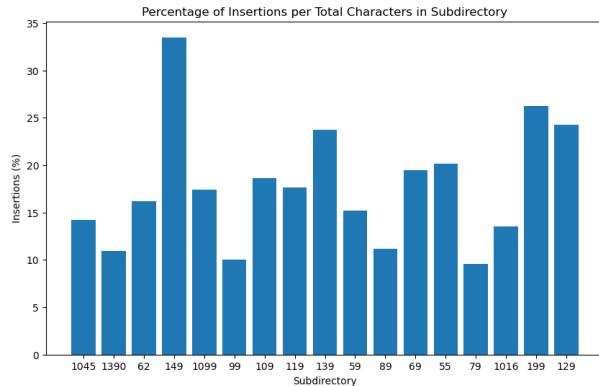
399 We compare the post-processed HTR output to the original HTR  
400 output to see the different accuracy scores. The original HTR output  
401 accuracy scores are shown in Table 2. The post-processed HTR  
402 output accuracy scores are shown in Table 3. We match HTR output  
403 words with ground truth words by taking their coordinates on the  
404 page and finding the best matching words based on the IoU score.  
405 The comparison of the performances is shown in Figure 11. We see  
406 a slight decrease in recall while all other metrics improve.

### 407 4.3 Layout Detection Accuracy

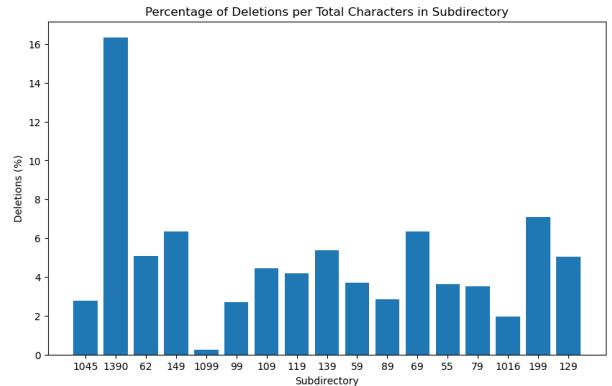
408 We measure our layout detection accuracy by comparing the IoU  
409 score of the table cells found by our method compared to the ground  
410 truth table cells. We create pairings for a predicted cell and a ground  
411 truth cell based on the highest IoU score and measure the accuracy  
412 scores for different IoU thresholds. The 'IoU Score' indicates the  
413 mean score for the true positives found at this IoU threshold. We  
414 show the performance both at the subset level as well as the total  
415 mean scores. The performances using IoU thresholds of 0.5, 0.25

**Table 1: Categorization of HTR System Errors**

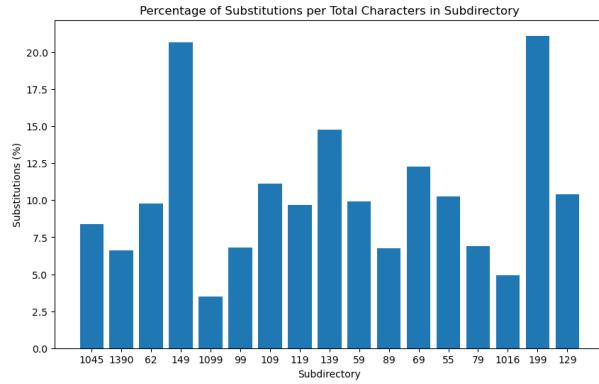
Error Type	Description	Possible Solution
Deletions	Characters not recognized at all (false negative)	Pre-process image, Improve HTR system
Insertions	Characters recognized that are not there (false positives)	Pre-process image, Improve HTR system
Substitutions	Characters recognized as another character	Pre-process image, Improve HTR system
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-process string
Invalid Characters	Characters that cannot occur	Post-process string



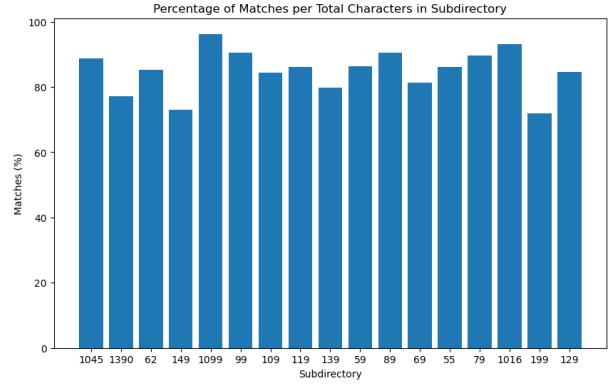
(a) Amount of insertions made by the HTR system per subdirectory.



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

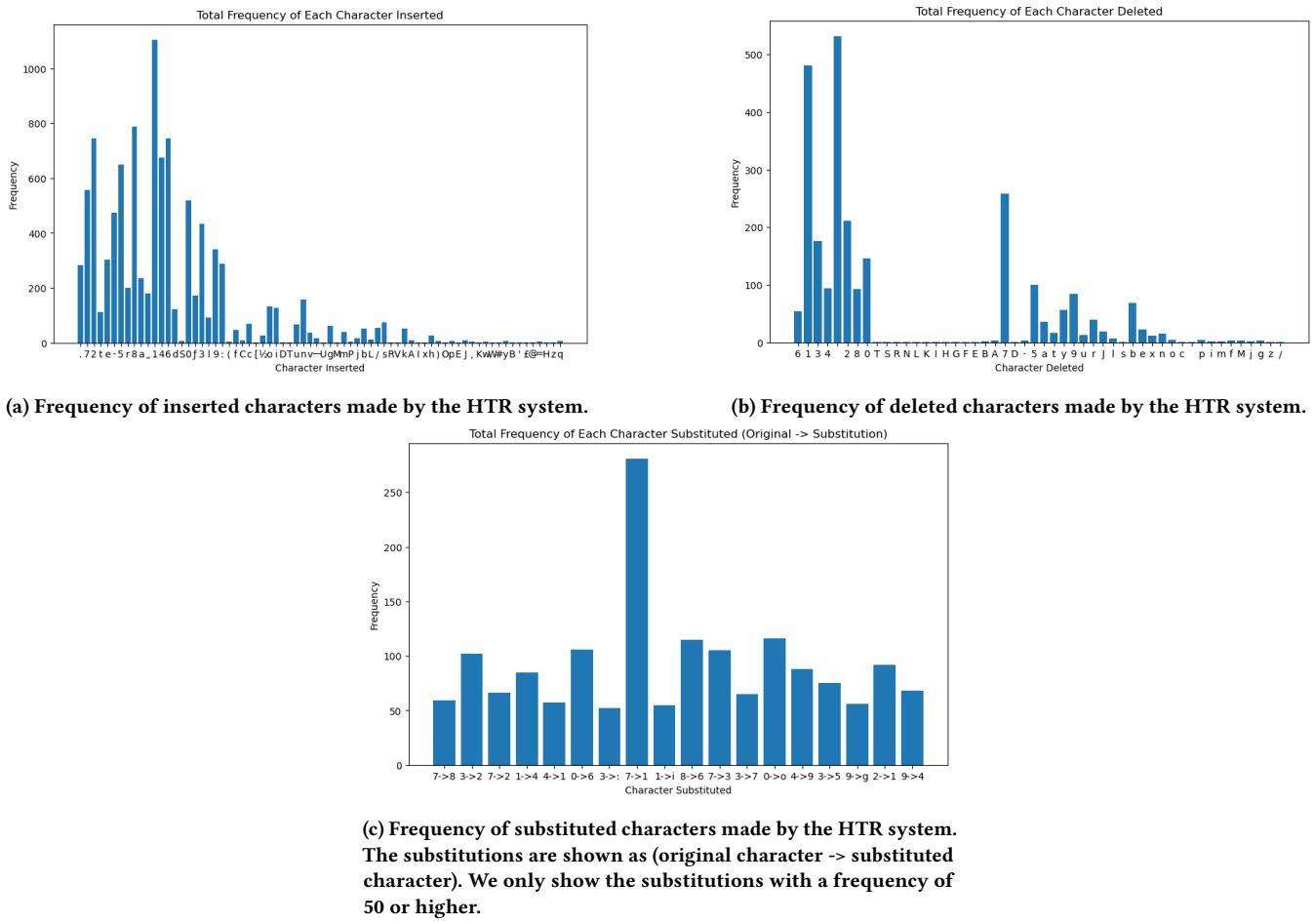


(c) Amount of substitutions made by the HTR system per subdirectory.



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

**Figure 9: The 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 and as a percentage of total characters in ground truth.**



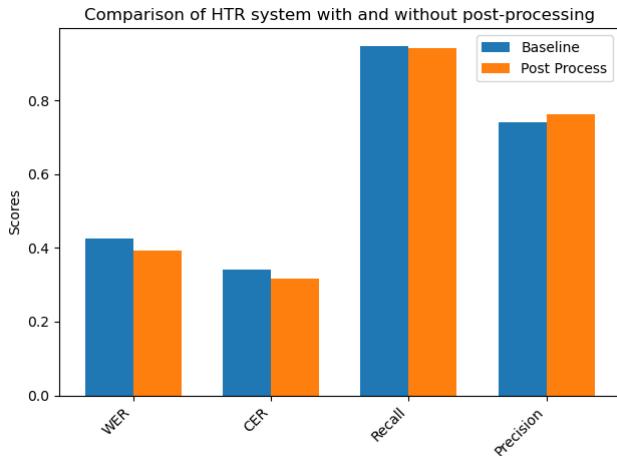
**Figure 10: The frequencies of inserted, deleted and substituted characters made by the HTR system.**

**Table 2: 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

**Table 3: Accuracy scores for HTR system with post-processing techniques applied. We matched the ground truth words with HTR output words based on highest IoU value.**

Metric	Value
WER	0.3916
CER	0.3169
Recall	0.9418
Precision	0.7622
Total Edit Distance	17508



**Figure 11: The WER, CER, precision, and recall scores for the HTR system with and without post-processing.**

**Table 4: Table Cell Performance Metrics at IoU Threshold 0.5.**

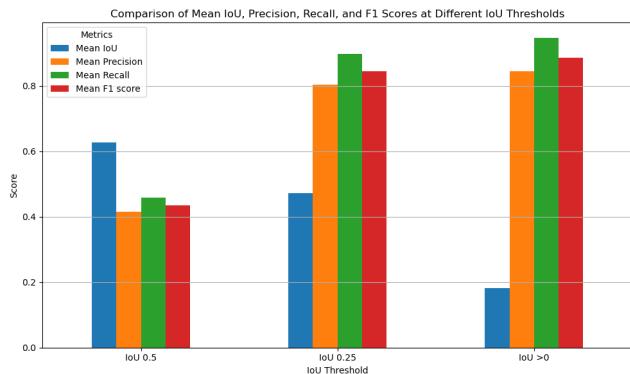
Subdir	IoU Score	Precision	Recall	F1
1045	0.638	0.674	0.674	0.674
1390	0.641	0.303	0.468	0.363
62	0.627	0.437	0.487	0.460
149	0.608	0.310	0.383	0.342
1099	0.621	0.532	0.499	0.515
99	0.638	0.525	0.504	0.514
109	0.645	0.520	0.552	0.534
119	0.617	0.422	0.437	0.429
139	0.626	0.435	0.475	0.453
59	0.664	0.555	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>	<b>0.627</b>	<b>0.416</b>	<b>0.459</b>	<b>0.434</b>

**Table 5: 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>	<b>0.473</b>	<b>0.804</b>	<b>0.898</b>	<b>0.844</b>

**Table 6: Table Cell Performance Metrics at IoU Threshold >0**

Subdir	IoU Score	Precision	Recall	F1
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>	<b>0.182</b>	<b>0.845</b>	<b>0.946</b>	<b>0.887</b>



**Figure 12: The mean IoU, precision, recall and F1 scores for ground truth and predicted table cell bounding boxes for different IoU thresholds.**

## 5 DISCUSSION

Write your discussion here. Do not forget to use sub-sections. Normally, the discussion starts with comparing your results to other studies as precisely as possible. The limitations should be reflected upon in terms such as reproducibility, scalability, generalizability, reliability and validity. It is also important to mention ethical concerns.

## 6 CONCLUSION

Write your conclusion here. Be sure that the relation between the research gap and your contribution is clear. Be honest about how limitations in the study qualify the answer on the research question.

## REFERENCES

- [1] Yasser Alginahi. 2010. *Preprocessing Techniques in Character Recognition*. <https://doi.org/10.5772/9776>
- [2] 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>
- [3] 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>
- [4] Tieming Chen, Guangyuan Fu, Hongqiao Wang, and Yuan Li. 2020. Research on Influence of Image Preprocessing on Handwritten Number Recognition Accuracy. In *The 8th International Conference on Computer Engineering and Networks (CENet2018)*, Qi Liu, Mustafa Misir, Xin Wang, and Weiping Liu (Eds.). Springer International Publishing, Cham, 253–260.
- [5] Sergio Correia and Stephan Luck. 2023. Digitizing historical balance sheet data: A practitioner's guide. *Explorations in Economic History* 87 (2023), 101475. <https://doi.org/10.1016/j.eeh.2022.101475> Methodological Advances in the Extraction and Analysis of Historical Data.
- [6] Ahmad Drobj, 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>
- [7] 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>
- [8] 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]
- [9] Constantin Lehenmeier, 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)
- [10] 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]
- [11] 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>
- [12] 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>
- [13] Tengchao Lv, Yupan Huang, Jingye Chen, Lei Cui, Shuming Ma, Yaoyao Chang, Shaohan Huang, Wenhui Wang, Li Dong, Weiyao Luo, Shaoxiang Wu, Guoxin Wang, Cha Zhang, and Furu Wei. 2023. Kosmos-2.5: A Multimodal Literate Model. arXiv:2309.11419 [cs.CL]
- [14] 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 Jialbo He (Eds.). Gulf Professional Publishing, 289–314. <https://doi.org/10.1016/B978-0-12-817736-5.00010-7>
- [15] 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>
- [16] 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.
- [17] 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>
- [18] 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>
- [19] 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>
- [20] 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.
- [21] 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)
- [22] 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.
- [23] 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>
- [24] Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, and Furu Wei. 2021. LayoutXML: Multimodal Pre-training for Multilingual Visually-rich Document Understanding. arXiv:2104.08836 [cs.CL]

514 Appendix A FIRST APPENDIX

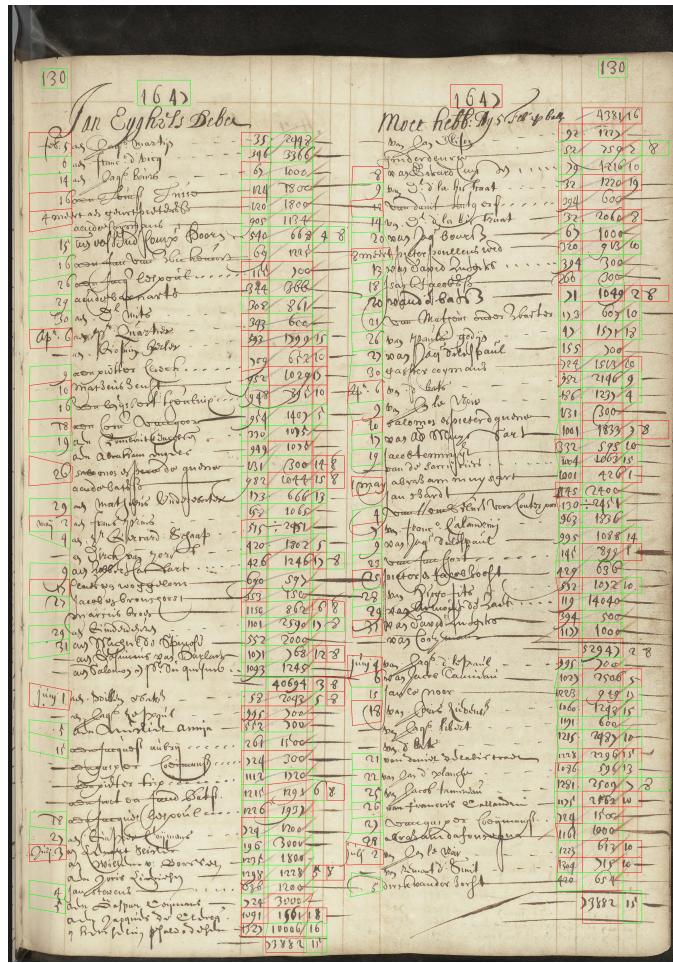


Figure 13: An example page showing the words correctly matched by the HTR system in green and the words wrongly matched in red. This can be helpful in visually inspecting the page to see whether there are common characteristics on parts of the page that could cause mistakes.

515 Put your appendices here.