Creating a Historical Migration Dataset from Finnish Church Records, 1800–1920

Ari Vesalainen[a*†], Jenna Kanerva[b†], Aïda Nitsch[c], Kiia Korsu[c], Ilari Larkiola[b], Laura Ruotsalainen[a], Filip Ginter[b]

[a]University of Helsinki, Department of Computer Science, Helsinki, Finland

[b]University of Turku, Department of Computing, TurkuNLP, Turku, Finland

[c]University of Turku, Department of Biology, Turku, Finland

*Corresponding author: Ari Vesalainen; ari.vesalainen@helsinki.fi

† Equal contribution

Abstract

This article presents a large-scale effort to create a structured dataset of internal migration in Finland between 1800 and 1920 using digitized church moving records. These records, maintained by Evangelical-Lutheran parishes, document the migration of individuals and families and offer a valuable source for studying historical demographic patterns. The dataset includes over six million entries extracted from approximately 200,000 images of handwritten migration records.

The data extraction process was automated using a deep learning pipeline that included layout analysis, table detection, cell classification, and handwriting recognition. The complete pipeline was applied to all images, resulting in a structured dataset suitable for research.

The dataset can be used to study internal migration, urbanization, and family migration, and the spread of disease in preindustrial Finland. A case study from the Elimäki parish shows how local migration histories can be reconstructed. The work demonstrates how large volumes of handwritten archival material can be transformed into structured data to support historical and demographic research.

Keywords: handwritten text recognition; document layout analysis; historical data; migration records; Finnish; Swedish

Author roles: **Ari Vesalainen**; Software; Writing - original draft; Resources; Methodology; Validation; **Jenna Kanerva**: Software; Writing - original draft; Data annotation; Methodology; Validation; **Aïda Nitsch**: Data annotation; Validation; Writing - original draft; **Kiia Korsu**: Data annotation; **Ilari Larkiola**: Data annotation; Software; **Laura Ruotsalainen**: Writing - review & editing; **Filip Ginter**: Software; Writing - original draft; Data annotation; Methodology; Validation;

1 Introduction

Internal migration in 19th- and early 20th-century Finland, shaped by economic, social, and environmental factors, influenced population structures, the spread of ideas, and the transmission of diseases (Hietala, 1981; Pasanen et al., 2024; Pitkänen, 1980). Analyzing these movements can help to track migration patterns over time and clarify demographic and societal changes in historical Finland.

Historical tabular data, such as the church records of internal migration, presents several challenges that differ from modern structured documents. Layouts often vary, with hand-drawn or missing separators and inconsistent formatting across or even within records. Handwriting styles differ significantly and are often degraded by ink fading, paper wear, and image quality, complicating layout detection and text recognition. Many documents lack clear structural markers such as lines or consistent spacing, requiring models to infer structure from weak visual signals. In addition, the lack of annotated training data that reflects these conditions limits the performance of standard models needed to extract such data in quantities. Aligning recognized text with the correct table cells and minimizing the need for manual correction remain key challenges in large-scale processing.

Recent studies have introduced methods for layout detection, table recognition, and handwritten text recognition to process historical tabular documents (Blomqvist et al., 2022; Clinchant et al., 2018; Granell et al., 2023; Lehenmeier et al., 2020). These include techniques for segmenting table rows and columns, extracting structured information, and evaluating tools such as Transkribus (Colutto et al., 2019). While these approaches show promising results on selected datasets, many depend on dataset-specific configurations or require manual steps. Data extraction from historical tabular data collections therefore remains to some extent a case-by-case effort relying on unique features of each collection, without a single universally proven solution.

The contribution of this work is thus two-fold. Firstly, we develop and evaluate a pipeline for the extraction of data from the Finnish church migration records. Our pipeline applies machine learning throughout the workflow — image pre-processing, table structure detection, and text recognition — to convert data into a format suitable for research. Although most likely not constituting a universal solution, the individual steps and their evaluation will add to the spectrum of techniques available for historical tabular data extraction. Further, with machine learning playing an increasing role in cultural heritage research (Fiorucci et al., 2020), the methods developed here add to our understanding of the possibilities and limitations of these techniques to support research across the humanities and social sciences by linking archival sources with computational approaches.

Our second contribution is a dataset obtained using the pipeline we develop, capturing detailed information on individuals and families who moved within Finland between 1800 and 1920. In total, the dataset constitutes over six million entries extracted from approximately 200,000 images of church migration records. These records provide a basis for studying internal migration trends and examining the social, economic, and health-related effects of mobility in historical contexts. Furthermore, once the dataset has been linked with other historical sources — such as records of births, deaths, and diseases — it can contribute to research on how mobility contributed to the spread of infectious diseases like smallpox, measles, and pertussis (Briga et al., 2022; Ketola et al., 2021; Nitsch et al., 2025). It also supports research on broader topics such as economic development, urbanization, and transformations in rural communities. If combined with census data or tax registers, the dataset can enable longitudinal studies of demographic change. The dataset is openly available, aiming to support future research into population movement, demographic change, and regional development using historical records. It can also support reconstructing individual life courses and family histories.

In the following, we describe the full process of developing the pipeline and the dataset, from digitizing and transcribing historical sources to structuring and validating the data.

2 Data

Moving records were maintained by Evangelical-Lutheran parishes in Finland, and the Church Law of 1686 required each parish to record religious acts such as baptisms, marriages, burials, in- and out-migrants, and communion participation (Pitkänen, 1980). Finland's Family History Association (FFHA) has been digitizing these records since 2004, and the FFHA digital archive now contains approximately 200,000 images from 2,781 books from 468 parishes (Finland's Family History Association (FFHA), 2025). Each image captures one opening (i.e. a double-page) of a book. Half of these images are grayscale or binary scans from microfilms commissioned by the Genealogical Society of The Church of Jesus Christ of Latter-day Saints in the 1950s (Anonymous, 1953), while the other half consists of digital color photographs taken by FFHA volunteers from original church book sources during the 2010s. Although the Church Law mandated migration records starting in 1686, the earliest records in the FFHA digital archive date to the 1720s. Records originating from the 18th century are rare in the dataset. Out of the 2,781 books, only 100 include any entries from the 18th century, and just 50 contain records exclusively from that period.

The layout of the records varies across the books, ranging from free text to standardized, preprinted movement tables. In many cases, the layout also changes within a single book. The earliest books mostly contain handdrawn tables, occasionally accompanied by free text records, whereas by the late 19th century, the records were increasingly entered into preprinted tables. To gain insight into this variation, we enhanced the metadata for each book by annotating the layout of each of the 2,781 books using four main categories: *free text*, *half-table* (free text with a few structural columns), *handdrawn table*, and *preprinted table*. The *preprinted table* category is further subdivided into unique preprinted forms, with a total of 55 different forms used.

Table 1 summarizes the layout annotation statistics in terms of the number of images per main layout type. The majority of pages contain hand-drawn or preprinted tables, with these two categories accounting for more than 85% of the dataset. Because only a small portion of the pages lack a table-like structure, this work focuses exclusively on the two main categories.

In Figure 1, we further analyze the frequencies of the different preprinted forms and present the cumulative count for the 15 most common layouts. Although the total number of unique layouts is quite high (55 layouts), only four layouts accounts for 50% of the preprinted images, and the seven most common layouts cover 75% of the images, indicating a long tail of rarely occurring forms.

Typically, in- and out-migrations were recorded in separate books, or in separate tables within a single book, with one table per page of a book opening. Occasionally, these were combined into a single table with separate columns for in- and out-migrants. Some variations in table structure occurred depending on regional practices or the instructions followed by specific parishes. Figures 2 and 3 provide examples of typical migration tables. Figure 2 illustrates a book in which the moving-in and moving-out tables are separate. In this example, the left-hand page contains individuals moving into the Huittinen parish in 1878, while the right-hand page lists individuals

Layout type	Images	% of data
handdrawn	94,477	47.07%
preprinted	79,243	39.48%
half-table	14,162	7.06%
free text	9,457	4.71%
other	3,395	1.69%

Table 1: Layout statistics for the main categories. *Other* refers to all remaining categories, primarily including empty images or images that do not contain migration records (due to incorrect metadata or mixed data within a book).

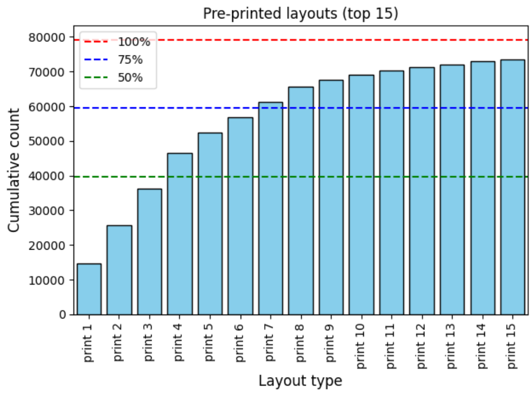


Figure 1: Cumulative counts for different preprinted layout types.

Data	Description
Reference number	An identifier for the record, which may represent a page reference, an order number within a specific year, or other context-dependent information.
Date	Date of the recording, not necessarily the actual moving date.
Occupation and name	Name of the person or main person of the family and his/her occupation.
Number of persons	Number of people moving, females and males separated.
Where to / Where from	Name of the new/old parish depending if moving-in or moving-out.
Reference to communion book	Reference to the page in the communion book where other details about the person are recorded.
Notes	Other related markings.

Table 2: Typical elements of migration tables.

moving out of the same parish. Figure 3 shows an example of a preprinted layout from Heinävesi in 1909, where both in- and out-migration records are combined into a single double-page table with separate columns.

The number of columns and the content present in the migration tables vary; however, the most common information is presented in Table 2 and Figure 4 shows one complete entry of an out-migration record. Occasionally, the same field may be split into two or more columns, for example, the date may appear as one field or be divided into separate columns for the month and the day of the month. Typically, one person is recorded on a single row of the table, with other members of the same family listed on subsequent rows. In older books, all members of the same household are sometimes grouped together in a single record entry. The language of the records also varies between books, with some of the records written in Finnish and others in Swedish because both are used as administrative languages in Finland. Furthermore, many parishes have Finnish and Swedish names (e.g. *Helsinki – Helsingfors*), and the orthography of these names in historical data may differ or include abbreviations (e.g. *Helsingfors – H:fors*). These, and other inconsistencies in data recording practices, as well as missing or incomplete entries, further complicate the extraction process.

2.1 Manual annotation for model training and evaluation

To train and evaluate the text recognition pipeline, a set of images was randomly sampled for manual annotation. Table 3 provides an overview of the annotated dataset used to develop the different models. The annotation process was conducted using Transkribus (Colutto et al., 2019) and Label Studio (Tkachenko et al., 2020-2025), which allowed both the transcription of textual content and the labeling of text regions. These annotations serve as ground-truth data for training and evaluation, allowing the models to be tested on accurately labeled examples.

We focused on a diverse selection of images from both preprinted and handdrawn books to capture variations in table layouts, handwriting styles, document quality, and historical printing techniques, improving the robustness of the recognition pipeline. Comprehensive annotation statistics are reported in Table 3 separately for each annotation type, as well by the training, development, and test sections.

In total, 1,632 images were annotated with at least table structure information. A subset of these was further annotated with de-skew key points, cell type classification, transcribed textual content, and year recognition (see Section 3.1 for more information about the pipeline components). Of all annotated images, 64% contain preprinted layouts, while the remaining 36% consist of handdrawn tables. However, the development and test sections are stratified to ensure 50/50 distribution between preprinted and handdrawn tables. The manually annotated data as

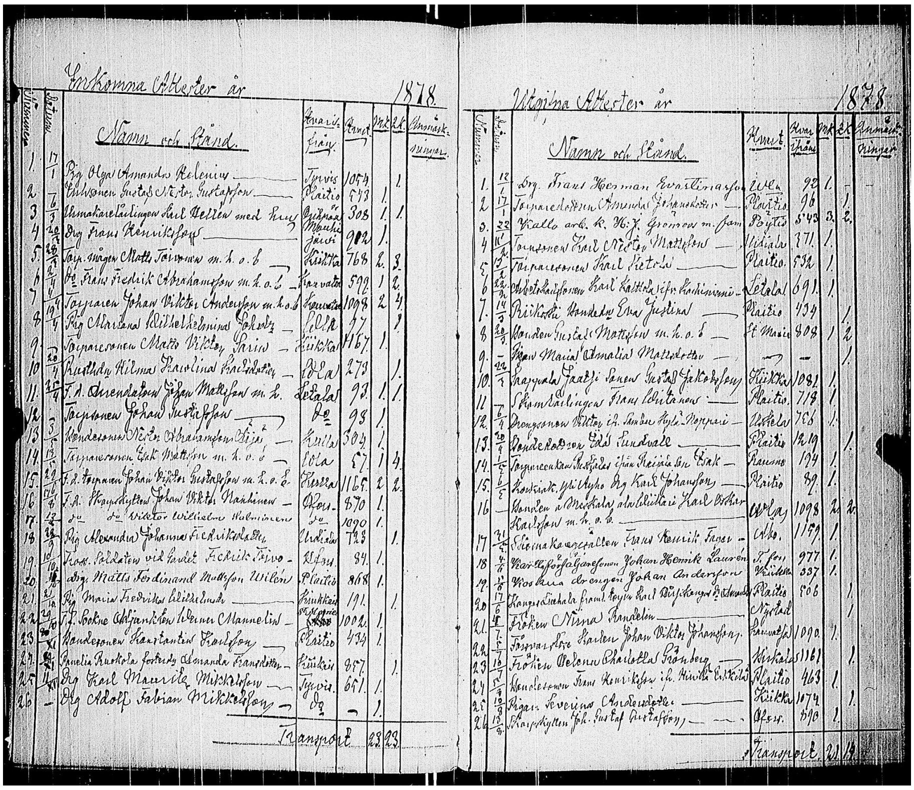


Figure 2: Example of handdrawn moving table (Huittinen 1878). FFHA's digital archive.

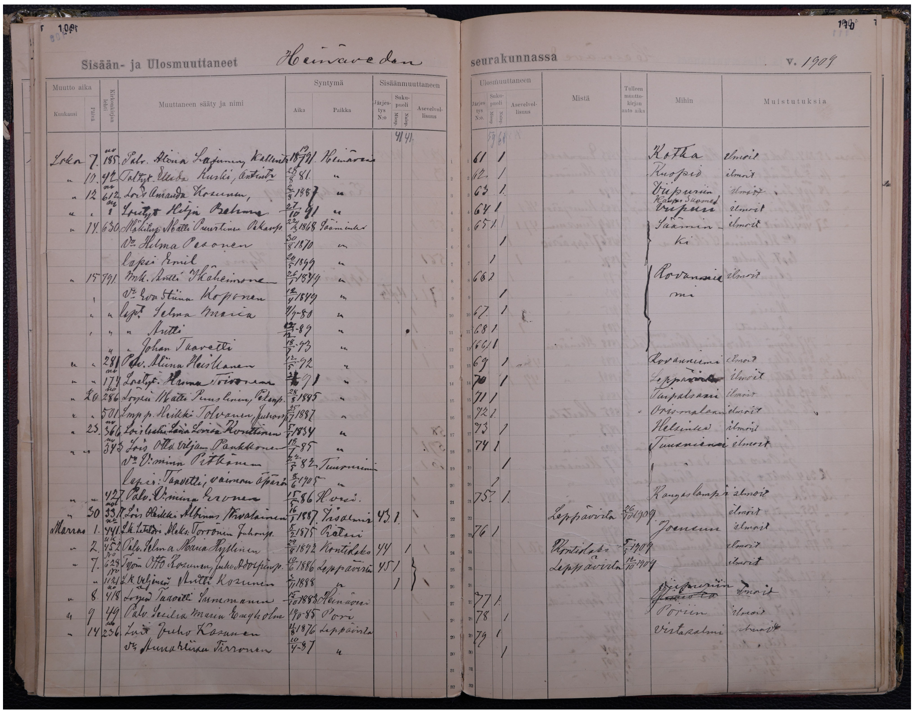


Figure 3: Example of preprinted moving table (Heinävesi 1909). FFHA's digital archive.

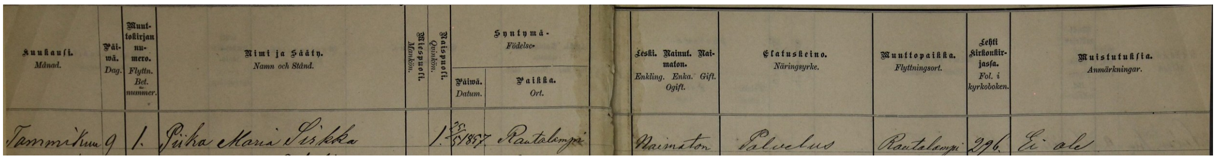


Figure 4: Details of a typical moving table entry from Hankasalmi: Maria Sirkka, a servant (piika), moved to Rautalampi on January 9th. She is female (naisenpuoli), born on March 25, 1857, in Rautalampi. Her marital status is single, and her occupation is servant (palvelus). Additional information can be found in the communion book on page 296. No further remarks are recorded.

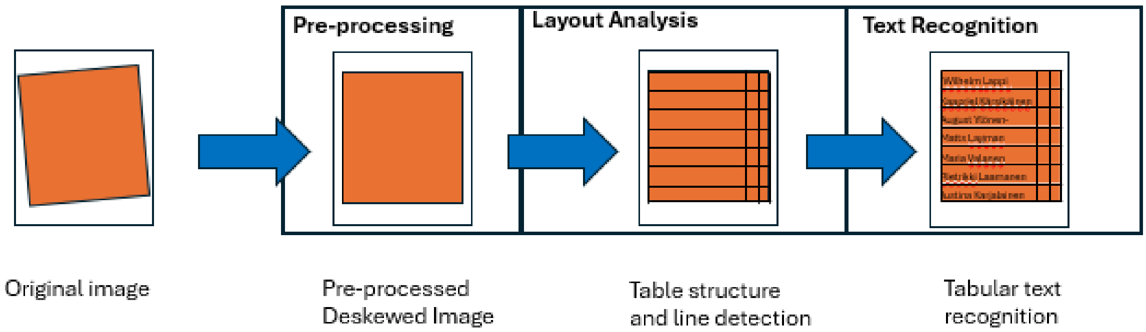


Figure 5: Text recognition for tabular data.

well as detailed annotation guidelines are available at https://github.com/TurkuNLP/finnish-migration-data.

Annotation type	Count of	Train	Dev	Test	Total
De-skew key points	images	900	190	200	1,290
Table structure	images	1,252	188	192	1,632
Cell type	images/cells	230/47K	47/14K	46/16K	323/77K
Text recognition	images/cells	—	41/1,947	39/2,277	80/4,224
Year recognition	images	1,026	188	192	1,326

Table 3: Summary of manually annotated data for the different stages of the pipeline, divided into training, development and test sections.

3 Method

The recognition of text in tabular data requires a structured approach that addresses the challenges posed by the layout and content of such documents. Deep learning methods have become effective tools for automating various stages of the text recognition pipeline (Nockels et al., 2022). Their ability to identify and adapt to patterns makes them suited for handling tabular layouts, text variability, and postprocessing needs.

This section describes the application of machine learning techniques across different stages of the pipeline, including preprocessing, document layout analysis, and text recognition. Preprocessing prepares input data for subsequent stages. Document layout analysis uses deep learning models to segment and classify structural elements such as rows, columns, and cells, enabling accurate localization of textual and non-textual components. The text recognition stage applies deep learning to transcribe text, addressing challenges like diverse writing styles and degraded text quality.

3.1 Text recognition pipeline for tabular data

Figure 5 outlines the main phases of the workflow, which relies on OCR and handwritten text recognition (HTR) techniques. During the preprocessing phase, the system converts image files into a uniform format and improves their quality for subsequent steps. Preprocessing includes tasks such as image de-skewing, resizing, normalization, noise reduction, and binarization.

Text recognition involves two main steps: document layout analysis and recognition. Layout analysis divides the image into smaller sections (e.g., text regions, tables, and lines) for further processing. Character and word recognition occur during the text recognition phase. In this step, the system identifies characters, and an XML (extensible markup language) file or similar format is generated, which includes references to object locations in the original image file and the recognized text.

While not shown in Figure 5, postprocessing typically follows the automated text recognition workflow. This stage focuses on improving the accuracy or usability of the output, and may include automatic correction methods such as comparing detected words with dictionaries to identify potential errors. The final analysis phase is guided by the researcher, who applies techniques such as data mining to explore patterns and extract insights relevant to the research questions.

3.2 Preprocessing

Our initial experiments have shown that de-skewing the pages can have a notable positive effect on the accuracy of table detection as it brings the rows and columns closer to horizontal/vertical, simplifying subsequent recognition. For illustration, an extreme case of a skewed page and its automatically de-skewed output are shown in Figure 6. This example also illustrates that the left- and right-hand side of the opening often need to be de-skewed independently, as they can be at a relative angle to each other, irrespective of the overall rotation of the book.

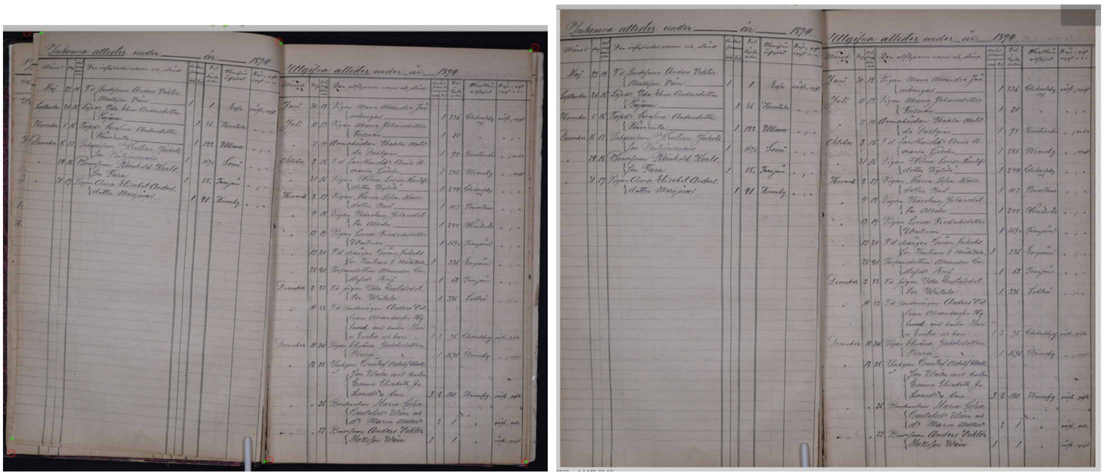


Figure 6: Extreme example of page skew (left) and the output of the de-skew process (right). Red circles mark stage-I corner recognition, green dots mark stage-II corner recognition.

We approach de-skew as an image recognition problem, where the objective is to recognize six key points on the image: the four corners of the opening, and the two ends of the middle division. The idealized process is illustrated in Figure 7: given the six key points A-F, two projective transforms can be induced (one by the points A-B-E-D and one by the points B-C-F-E) resulting in the de-skewed image.

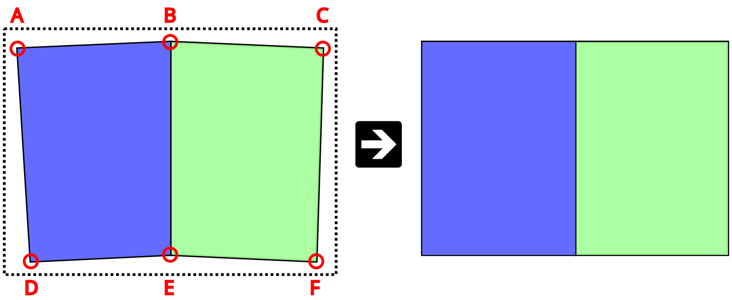


Figure 7: De-skew process. Two pages of the opening with the relevant six keypoints A-F and the image frame (dashed line).

In order to recognize the six key points, we manually annotate the points in 1,290 images (Table 3) and train the YOLO (You Only Look Once) model for the *pose detection* task (Deng et al., 2024; Huang et al., 2019; Ultralytics, 2024). Pose detection is a machine vision task, whereby an object is detected in the image, together with defined key points within (e.g. a person with a key point for every main joint, allowing the pose to be estimated). For improved accuracy, we apply a two-stage recognition. In stage-I, the model receives the image of the whole opening and detects the six key points. In preliminary experiments, we found that the global, image-wide features used at this stage do not allow for a precise-enough placement of the key points, but are sufficient to place the key points in the vicinity of their correct position. Therefore, in stage-II, we extract the

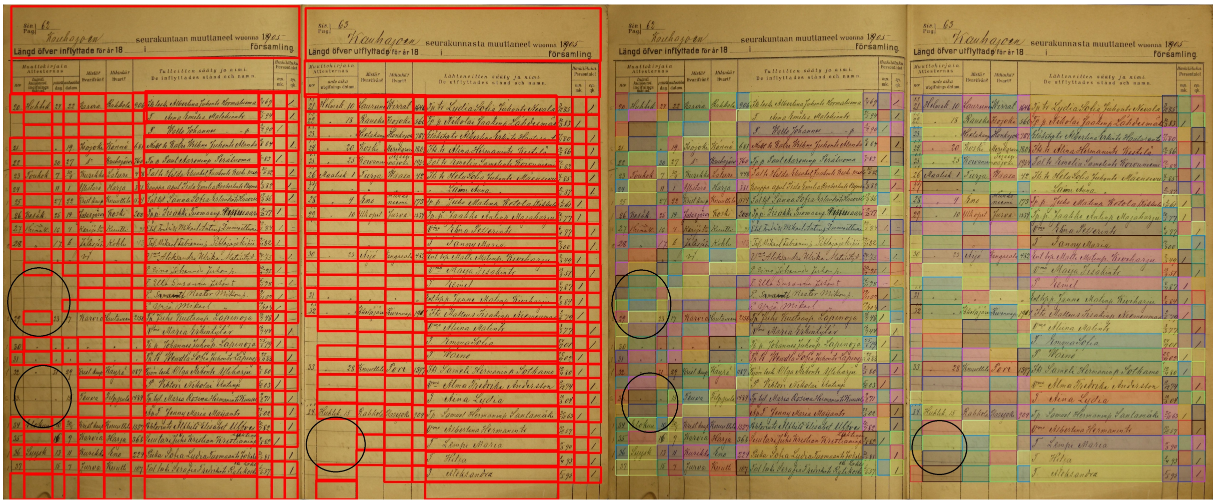


Figure 8: Example of how clustering improves results. In some cases, the table cell detection model fails to detect all cells in a table (black circles on the left-hand side). By applying a clustering method, these gaps can be filled (black circles on the right-hand side).

rectangular areas of max 15% of image size centered on each of the six stage-I points, and train a second YOLO pose detection model to recognize the precise placement of the single key point within these "zoomed-in" images. To provide for a more dense training data for the Stage II classifier, we mirror the right-hand patches along the vertical axis, and the lower patches along the horizontal axis such that the target keypoint is always in the top left corner of the patch. That way, the model does not need to learn separately the visually different features of e.g. top-left corners as opposed to bottom-right corners, since after the mirroring, these will look the same.

3.3 Table structure detection

Table structure recognition is a specific aspect of document layout detection, focusing on the identification and interpretation of tabular content within documents. Tables, with their rows, columns, and sometimes nested or merged cells, present unique challenges. Extracting information accurately from these tables involves distinguishing between their overall structure and the specific components within them. To address this, the process is divided into two tasks: detecting tables on the page, and identifying the internal lines that define rows and columns in each detected table.

During the Table Detection step, the system identifies the boundaries and overall geometry of tables within the document. The Line Detection step focuses on detecting the internal lines or separators, enabling the extraction of cell-level data. Separating these tasks allows for the use of specialized methods optimized for each, improving accuracy and adaptability to various table formats and layouts.

YOLO is used for Table Detection for its speed and accuracy (Deng et al., 2024; Huang et al., 2019; Ultralytics, 2024). It processes an entire image in a single forward pass, making it efficient for detecting multiple tables in a document. The model is trained to detect entire tables as well as individual table cells (Figure 8). Some cells may be missed during prediction. To address this, density-based clustering (DBSCAN) is applied as a postprocessing step. The detected cell borders serve as elements to be clustered—left and right borders for columns, and top and bottom borders for rows. This helps infer missing cells and reconstruct the complete table structure efficiently.

For Line Detection, Mask R-CNN, an instance segmentation model, is used as it can separate fine-grained instances within tables (He et al., 2017). Mask R-CNN's pixel-level precision allows accurate identification of lines, even in complex or degraded table images. Since most cells contain a single line of text, the full cell can be treated as a text line. This approach improves text recognition accuracy by preserving the entire text line and avoiding unnecessary cropping that could lead to loss of information.

By combining these two methods, the pipeline leverages YOLO's detection capabilities for identifying table structure and Mask R-CNN's segmentation for detailed line detection. This approach ensures accurate table structure recognition across a variety of document types and layouts.

3.4 Cell type classification

We use a YOLO-based image classification model to categorize table cells in historical moving records. Many cells are either empty or contain "repetition" marks, indicating that the information should be copied from the row above. Some cells include only a single line of text, while others have multiple lines, which affects how they should be processed. By identifying these types early, we avoid running text recognition on cells that contain no useful content. This improves both speed and accuracy, as our HTR model can produce incorrect results if applied to

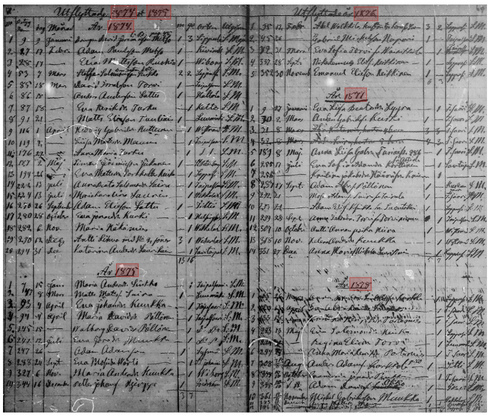


Figure 9: Example of an opening with several year mentions outside of the header area.

empty cells. Using the classifier helps direct text recognition only to cells that contain meaningful information. We categorize cells into four classes, *single-line*, *multi-line*, *repetition*, and *empty*.

3.5 Text recognition

Finally, text recognition is performed separately for each cell, either at the cell level for single line cells, or in the case of multi-line cells, separately for each individual line. Cells identified as empty by the cell type classifier are excluded at this stage to prevent the text recognition model from hallucinating content.

For this task, we use a handwriting recognition model (HTR) model trained by the National Archives of Finland[1]. The model is based on the TrOCR architecture introduced by Li et al. (2023), a transformer-based encoder-decoder model consisting of an image encoder and a text decoder. While the original TrOCR model was trained on English data, the National Archives of Finland fine-tuned it for handwritten, historical Finnish and Swedish. The fine-tuning dataset included text lines from a variety of sources spanning the 17th to 20th centuries, as well as tabular data to improve model's performance on table cells.

3.6 Year detection

In order to be able to utilize the migration records in downstream research, it is critical to also extract the year for each record. Practically without exception, years are not repeated in the records, but are stated on the page, typically as part of the header (e.g. *1878* in Figure 2 and *1909* in Figure 3), but in some cases also within the page in cases where a new page is not started at the beginning of each new year. An example of such page is shown in Figure 9.

The detection of the year is carried out in two phases. First, a YOLO model is trained on manually annotated data to identify year occurrences on each page. These are subsequently recognized using the Finnish National Archive HTR model (see Section 3.5). Using the coordinates from the page de-skew process, each occurrence is identified w.r.t. being on the left or right page of the opening. Considering the somewhat ad-hoc manner in which the years are written on each page, the recognition results are noisy. In a subsequent step, we therefore lean on the fact that the years form a sequence throughout the book, i.e. a mis-recognized year can be corrected based on the surrounding pages. This presupposes a combination of HTR postcorrection (e.g. *19/4* correction into *1914*) and inference of the overall logical sequence of years. To this end, we apply a capable large language model (specifically OpenAI's *GPT-4o-mini*) prompted to extract the most likely sequence of years, given the raw output of the recognition. In the evaluation (Section 3.7.5) we demonstrate that the large language model is indeed capable of considering the context of surrounding pages and improving the raw extraction results.

3.7 Evaluation

Traditional classification metrics are used to measure table structure detection by evaluating how accurately the model detects tables, rows, and columns. Cell detection accuracy is not assessed separately since cells are intersections of rows and columns. The same metrics are also used for cell classification and year detection.

Accuracy (Eq. 1) describes the ratio between correct and total predictions. Precision (Eq. 2) describes the proportion of correct predictions among all detected instances, reflecting the quality of detection. Recall (Eq. 3) describes the proportion of correct predictions compared to all actual instances, reflecting the quantity identified. These metrics depend on True Positives (TP), which indicate correct detections, False Positives (FP), indicating incorrect detections, and False Negatives (FN), indicating ground-truth instances missed by detection. True

[1] https://huggingface.co/Kansallisarkisto/tablecell-htr

Negatives (TN), indicating correct negative classifications, are not relevant for object detection. A detection counts as a True Positive if Intersection over Union (IoU) (Eq. 5) exceeds 0.5. IoU compares the area of the image detected by the model (A) to the area of the image specified as correct in the ground truth (B).

$$accuracy = \frac{TP}{TP + FP + FN} = \frac{TP}{all\ predictions} \tag{1}$$

$$precision = \frac{TP}{TP + FP} = \frac{TP}{all\ detections} \tag{2}$$

$$recall = \frac{TP}{TP + FN} = \frac{TP}{all\ ground\ truths} \tag{3}$$

Precision and recall are more informative when considered together. This is achieved using the F1-score (Eq. 4), which combines precision and recall through their harmonic mean. By balancing precision and recall, the F1-score provides a more comprehensive measure of performance.

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{4}$$

$$\text{IoU} = \frac{A \cap B}{A \cup B} \tag{5}$$

3.7.1 De-skew

We evaluate the de-skew step in terms of angle from vertical of the three important edges: the left edge of the left page, the center line dividing the two pages of the opening, and the right edge of the right page. The results are reported in Table 4, demonstrating that the de-skew substantially reduces the angle from vertical for both the left and right edges of the book opening. Further, the results demonstrate that the Stage II of the de-skew process is crucial, as it has a very substantial effect on the overall quality of the result.

	Left	Middle	Right
Base	0.33°± 0.78	-0.06°± 0.53	-0.28°± 0.77
Stage I	0.17°± 0.62	-0.15°± 0.43	-0.27°± 0.64
Stage II	0.08°± 0.59	0.06°± 0.82	-0.005°± 0.69

Table 4: Skew angle, in degrees difference from vertical, of the left, middle, and right borders, reported on the test set. The angles in the original image (Base) are calculated using the manual annotation of the test set images, and Stage I and II are the two stages of the de-skew algorithm.

3.7.2 Table structure detection

Tables 5, 6, and 7 present the accuracy results for detecting table structures in preprinted and handdrawn documents. Table detection performs slightly better on handdrawn pages compared to preprinted ones. This difference likely arises from the challenge the model faces with multiple-table layouts found more frequently on preprinted pages. Row detection accuracy is clearly higher on preprinted pages than on handdrawn pages, probably due to greater consistency in the layout of preprinted tables and more variation in rows within handdrawn tables. The accuracy differences for column detection between the two types of pages are minor, indicating column structures remain similarly defined and visible on both table types. The overall results indicate effective detection performance for diverse table formats.

Table type	Accuracy	Recall	Precision	F1-score
Preprinted	93.2	93.2	100.0	96.5
Handdrawn	95.4	95.4	100.0	97.6
All	94.2	94.2	100.0	97.0

Table 5: Table detection

3.7.3 Cell classification

Cell classification is evaluated on 46 test images, comprising 16,000 cells annotated for cell type. The distribution of cell types is skewed towards single-line cells, 60% are single-line, 23% are empty, 12% are repetition symbols, and only 5% are multi-line cells. The classification results are presented in Table 8, with Precision, Recall, and F1-score reported separately for each cell type.

Table type	Accuracy	Recall	Precision	F1-score
Preprinted	95.1	96.4	98.7	97.5
Handdrawn	87.9	93.7	93.4	93.6
All	91.4	95.1	96.0	95.5

Table 6: Row detection

Table type	Accuracy	Recall	Precision	F1-score
Preprinted	96.1	99.1	96.9	98.0
Handdrawn	92.4	98.3	93.9	96.1
All	94.4	98.7	95.6	97.1

Table 7: Column detection

As expected, the results reflect the underlying distribution, with the most frequent class, single-line cells, achieving the highest F1-score of 92%, while the least frequent multi-line cells yield the lowest F1-score of 69%. We also experimented with training using label weighting, where the class weights are adjusted in the loss function to give higher importance to underrepresented classes, but this did not lead to any notable improvement in performance.

3.7.4 Text recognition

The text recognition model is evaluated on 39 test images, comprising 2,277 annotated cells (excluding empty cells). Most cells contain a single line of text, with only 140 cells featuring multiple lines. In total, the test dataset includes 2,471 lines for text recognition. However, in some cases, poor image quality or difficult handwriting made it impossible for the human annotator to confidently transcribe all characters. In such instances, the unreadable portions were marked with question marks. A total of 342 lines contain question marks and are excluded from the evaluation, unreadable even to a trained human.

The overall performance of the model yields an Exact Match (EM) score of 49.9% and a Character Error Rate (CER) of 0.19. In Table 9, in addition to the overall results, we also present the performance separately for numerical lines (containing only numbers and punctuation) and textual lines (which include at least one letter). While the CER is comparable across both types, the Exact Match score is significantly higher for numerical lines. This reflects the fact that textual lines tend to be longer and are therefore more difficult to transcribe perfectly, without even a single error.

3.7.5 Year extraction

We evaluate the year extraction step separately for the left and right page of each opening, on manually annotated data (192 openings) in terms of unique years stated on the page. We focus the evaluation on such examples, where at least one occurrence of a year was annotated (168/192 openings). This is because an opening with no explicit year mention may come from a book where a year is recorded only once, when a new year starts. Since the LLM-corrected extraction method infers these years from the surrounding sequence, these predictions would be incorrectly counted as false positives. We evaluate in terms of precision (proportion of correctly predicted years out of all unique predictions for each page) and recall (proportion of correctly predicted years out of all unique years annotated for each page), and F1-score, their harmonic mean. As shown in Table 10, we see that the method's precision surpasses 91% and recall 83%; and additionally we see that the LLM-based correction improves the results both in terms of precision and recall.

Cell type	Precision	Recall	F1-score	Support
single-line	96.3	87.3	91.6	9829
empty	81.2	96.7	88.3	3692
repetition	79.4	87.1	83.1	2020
multi-line	67.9	69.6	68.7	744
accuracy			88.6	16285
macro avg	81.2	85.2	82.9	16285
weighted avg	89.5	88.6	88.8	16285

Table 8: Cell type classification performance with Precision, Recall, and F1-score reported separately for class.

	EM	CER	Avg. length	Support
textual	28.2%	0.19	12.2 chars	897
numeric	65.8%	0.18	3.2 chars	1,232
All	49.9%	0.19	7.0 chars	2,129

Table 9: Comparison of text recognition evaluation for numeric and textual lines.

year extraction method	Precision	Recall	F1-score
with LLM correction	91.6	83.1	87.2
without LLM correction	89.2	80.0	84.4

Table 10: Precision, Recall, and F1-score of per-page year mention extraction.

4 Results and error analysis

The pipeline was applied to the full dataset of moving record images on a supercomputer equipped with NVIDIA V100 GPUs (32 GB memory), parallelized on the 468 parishes. The total processing time per image averaged at 60 seconds, with 5% for detecting table structure, 30% for detecting text lines, and 65% for recognizing text content. Image de-skew processing time was negligible. If processed one image at a time, the complete run would have taken over four months. By distributing work across parishes and being able to maintain up to 80 jobs in parallel, the entire set of 200,000 images was completed in just four days. The total number of detected moving records was roughly 6.2 million.

So far, we have evaluated the pipeline components in isolation. To better understand the overall output quality and typical error types, we compare the full predicted pipeline output to the manually annotated data. We find that in 100 out of 192 test images (52%) the number of tables and the number of columns in each table match the annotations. This indicates that for these images the pipeline correctly predicted both the number of tables and their general layout. In the remaining 92 images (48%), common errors include:

1. Predicting the correct amount of tables but with one (28 images) or more (33 images) columns too few or too many (a total of 61 images). This may lead to losing some information from the migration record, however, the primary information (e.g. date, name, parish) is often unaffected.
2. Missing one or both of the two tables in the image (15 images). This naturally results in data loss if the missing table contains any records.
3. Splitting a double-page, full-opening table into two separate tables, causing information from a single migration event to be split into two entries (16 images).

Among these, errors (1) and (3) can potentially be addressed through post-processing, while error (2) would require improvements to the table recognition component.

Regarding rows, each row generally corresponds to one migration record, usually involving a single person, though in some cases it corresponds to an entire family. Excluding the 16 images where the pipeline split a double-page table into two separate tables (which do not provide reliable row count estimates), but including other discrepancies, the pipeline produced a total of 2,411 rows, compared to 2,648 rows in the manually annotated tables. This indicates that, excluding the double-page issues, approximately 91% of the rows were successfully extracted by the pipeline. In cases where the row count differs between the annotated and extracted tables, the discrepancy is typically small or due to a missing table.

5 Case study: Migrations of Elimäki

To illustrate a potential use case of the extracted migration data, we selected one parish, Elimäki, and combined data from all available migration books for that parish. This allows us to demonstrate the potential of quantifying in- and out-migration volumes for a specific parish over the study period, as well as to show the spatial patterns of migrations involving Elimäki. The Elimäki parish includes six migration books comprising a total of 405 images and 18,809 extracted migration records. Since the books differ in layout (two record only in-migration, two only out-migration, and two combine both), we first standardized the data to enable statistical analysis.

In this case study, for each row, we extracted information on the direction of migration (in or out), the year of migration, as well as the origin or destination of the individual (i.e., where from or where to the person migrated). For cells predicted to contain only repetition symbols (based on both cell type classification and text recognition results), we filled in the missing content using the nearest preceding cell containing actual data.

All six books use preprinted layouts, which allowed us to rely on our metadata annotations to determine whether a page contains in- or out-migration records (in books with mixed records, the left page corresponds to in-migration and the right to out-migration), and to identify the expected column for the parish name.[2]

Of the 18,809 predicted rows, 64% followed the expected layout; meaning the predicted tables had the correct number of columns, and the parish column content matched expected patterns based on simple heuristics involving data type and average text length. An additional 34% of the predicted rows had layout or content mismatches, but we were able to realign the columns using the same simple heuristics. For the remaining 2%, we were unable to reliably identify the parish column, leaving these fields empty.

This process resulted in over 3,000 unique parish names. However, this number includes both minor orthographic variations in the original writings as well as cases where the text recognition model misidentified some of the letters. To remove noise and normalize the names, we first applied LLM-based cleaning, where we prompted OpenAI's GPT-4o-mini[3] to map the predicted parish names to a given list of known Finnish parishes. Since the relationship between the predicted name and the actual parish was not always straightforward, the model was instructed to flag uncertain cases. While this approach resolved many of the simple cases, we manually reviewed the LLM output and further standardized it by matching each name to the known list of parishes. After standardization, 66% of the 3,363 parish name spellings could be linked to a specific parish.

After examination of the predictions, we noticed that two of the books were fully duplicated and the duplicated rows were removed from the dataset. This duplication may be due to the same book being photographed several times. This resulted in a dataset containing 15,597 rows. We further removed rows with missing information (either the direction of the movement, the year or the parish name, respectively 317, 200 and 1,178 rows) or when the parish name could not be linked to any known parish (2,874 rows, 15% of the initial records). The resulting dataset of movements in Elimäki contained a total of 11,295 usable records (60% of the 18,809 initial records), with 4,826 departures from Elimäki and 6,469 arrivals to Elimäki.

5.1 Case study results

This dataset enables us to quantify the departures from and arrivals to Elimäki across both time and space. For instance, it is possible to quantify annual departures (Mean = 135 departures ±59 s.d., Figure 10) and arrivals to the parish (Mean = 100 ±41 s.d., Figure 10). It is worth noting that there are no records of arrivals to Elimäki for the years 1914 and 1915. This is due to the year identification failing in few pages, and the records from these years are merged with records from 1916 and 1917.

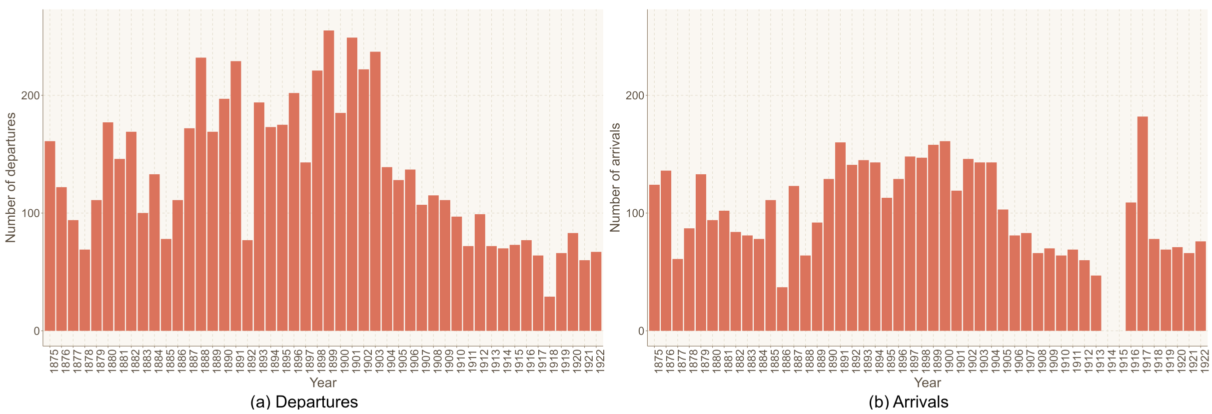


Figure 10: Histograms of departures from and arrivals to Elimäki between 1875 and 1922.

Spatially, the initial overview of the destinations and the origins of migration shows that most movements were local (see Figure 11 and Supplementary videos available on the project repository[4] for a yearly overview).

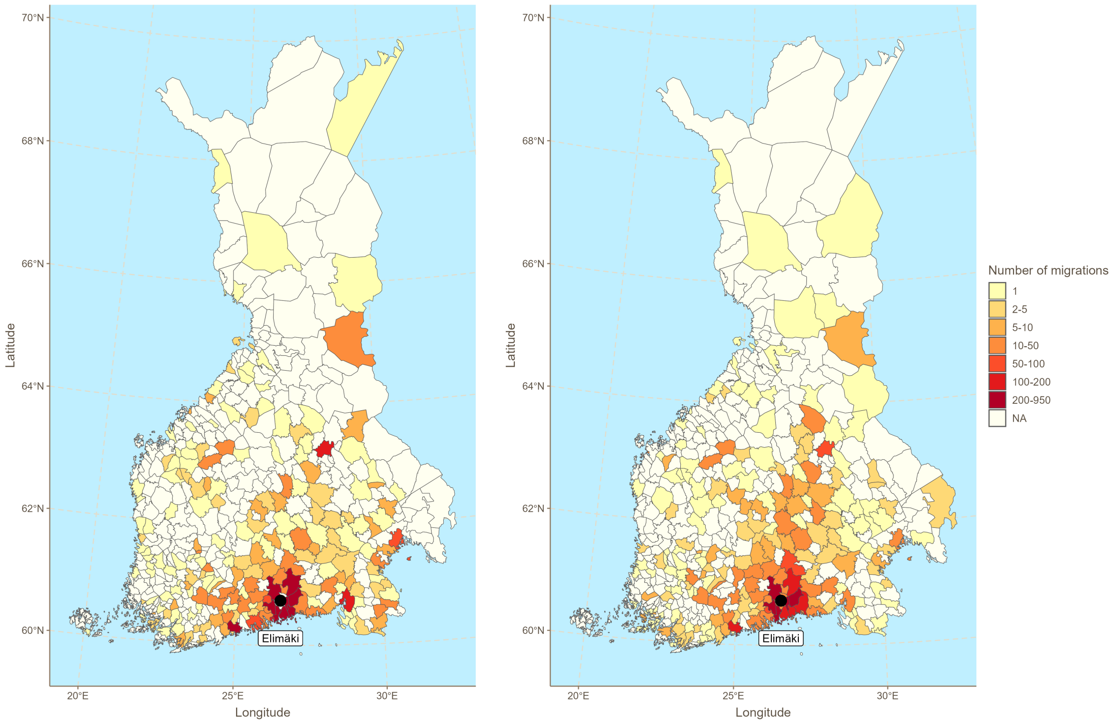
6 Conclusions and future work

This work demonstrates that large-scale, automated extraction of structured data from handwritten historical records is feasible, even though it remains technically challenging. Using digitized Finnish moving records, we created a dataset containing over six million entries from approximately 200,000 images dated between the 19th and early 20th centuries. These entries document individual and family migration between parishes and provide a valuable source for demographic and historical analysis. The resulting dataset has a potential to support studies of population movement, urbanization, family structures, and disease transmission in preindustrial Finland. In a

[2] We note that one reason we selected Elimäki as our case study parish is that all its books use preprinted forms, which simplifies the extraction process. Section 6 discusses our plans for extending this approach to books with handdrawn tables.

[3] gpt-4o-mini-2024-07-18

[4] https://github.com/TurkuNLP/finnish-migration-data



(a) Origins (b) Destinations

Figure 11: Maps showing the origins and destinations of migration to and from Elimäki between 1875 and 1922.

case study based on the Elimäki parish, we illustrated how migration volumes and patterns can be reconstructed locally.

Our pipeline combines deep learning methods for preprocessing, layout analysis, table detection, cell classification, and handwriting recognition. Its modular structure allows each stage to be improved independently. While we have already demonstrated the extraction output to be sufficient for certain quantitative applications, there are still challenges that need to be addressed in future development. First, we aim to enhance the extraction of document metadata, such as table type, table titles, and column headers. Additional metadata is needed for conducting studies similar to our case study, especially when working with handdrawn books, where the expected columns on a page are not known in advance. To address this, we intend to use LLM-based column header inference, where a language model will inspect the extracted data table and infer common column headers (e.g. date, name, and parish).

Second, we are developing LLM-based postprocessing tools to standardize variations in the extracted data. This will help to normalize e.g. names, places, and dates including both orthographic variation as well as misread letters into standardized spelling, as well as infer missing values where contextual patterns in the data suggest repetition or shared information.

In addition to these, we plan to further improve table detection accuracy through better line segmentation and layout modeling. We also intend to extend the pipeline support for varied table formats, with the aim of generalizing the pipeline to various types of historical tables beyond migration records available as scanned images through various historical archives.

The pipeline, the manually annotated data, as well as the extracted migration records are available through the project repository: https://github.com/TurkuNLP/finnish-migration-data.

Acknowledgements

We thank Finland's Family History Association for providing the original digitized images used in this study. We also extend our gratitude to the National Archives of Finland for supplying the text recognition model that supported the processing of historical records. This study is connected to the national DARIAH-FI infrastructure, which fosters collaboration among Finnish research groups. We are also grateful to the Helsinki Computational History Group (COMHIS) for their facilitation support. Computational resources were provided by CSC - IT Center for Science (HPC-HD project 2005072). Open access funded by Helsinki University Library.

Funding Statement

This work was supported by the Human Diversity consortium, Profi7 program by Research Council of Finland (grant 352727) as well as the HPC-HD Research Council of Finland general research grant (grant 347708).

Competing interests

The authors have no competing interests to declare.

References

Anonymous. (1953). Sukututkimusaineiston valokuvaaminen. *Genos*, *3*, 89. (In Finnish)

Blomqvist, C., Enflo, K., Jakobsson, A., & Åström, K. (2022). Joint handwritten text recognition and word classification for tabular information extraction. In *26th international conference on pattern recognition (ICPR)* (p. 1564-1570). DOI: 10.1109/ICPR56361.2022.9956282

Briga, M., Ketola, T., & Lummaa, V. (2022). The epidemic dynamics of three childhood infections and the impact of first vaccination in 18th and 19th century Finland. *medRxiv*. DOI: 10.1101/2022.10.30.22281707

Clinchant, S., Déjean, H., Meunier, J.-L., Lang, E. M., & Kleber, F. (2018). Comparing machine learning approaches for table recognition in historical register books. In *13th IAPR international workshop on document analysis systems (DAS)* (pp. 133–138).

Colutto, S., Kahle, P., Guenter, H., & Muehlberger, G. (2019). Transkribus. a platform for automated text recognition and searching of historical documents. In *15th international conference on escience (escience)* (p. 463-466). DOI: 10.1109/eScience.2019.00060

Deng, Q., Ibrayim, M., Hamdulla, A., & Zhang, C. (2024). The YOLO model that still excels in document layout analysis. *Signal, Image and Video Processing*, *18*(2), 1539–1548.

Finland's Family History Association (FFHA). (2025). *Finland's family history association website.* https://www.sukuhistoria.fi/sshy/index.htm. (Accessed: January 8, 2025)

Fiorucci, M., Khoroshiltseva, M., Pontil, M., Traviglia, A., Del Bue, A., & James, S. (2020). Machine learning for cultural heritage: A survey. *Pattern Recognition Letters*, *133*, 102-108. Retrieved from https://www.sciencedirect.com/science/article/pii/S0167865520300532 DOI: https://doi.org/10.1016/j.patrec.2020.02.017

Granell, E., Romero, V., Prieto, J. R., Andrés, J., Quirós, L., Sánchez, J. A., & Vidal, E. (2023). Processing a large collection of historical tabular images. *Pattern Recognition Letters*, *170*, 9-16. DOI: 10.1016/j.patrec.2023.04.007

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961–2969).

Hietala, K. (1981). Internal migration and technological development. *Finnish Yearbook of Population Research*. DOI: 10.23979/fypr.44751

Huang, Y., Yan, Q., Li, Y., Chen, Y., Wang, X., Gao, L., & Tang, Z. (2019). A YOLO-based table detection method. In *Proceedings of the 15th international conference on document analysis and recognition (ICDAR 2019)* (pp. 813–818).

Ketola, T., Briga, M., Honkola, T., & Lummaa, V. (2021). Town population size and structuring into villages and households drive infectious disease risks in pre-healthcare Finland. *Proc R Soc B*. DOI: 10.1098/rspb.2021.0356

Lehenmeier, C., Burghardt, M., & Mischka, B. (2020). Layout detection and table recognition – recent challenges in digitizing historical documents and handwritten tabular data. In *Digital libraries for open knowledge: 24th international conference on theory and practice of digital libraries (TPDL 2020)* (p. 229–242). DOI: 10.1007/978-3-030-54956-5_17

Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., ... Wei, F. (2023). TrOCR: Transformer-based optical character recognition with pre-trained models. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 37, pp. 13094–13102).

Nitsch, A., Lummaa, V., Ketola, T., Honkola, T., Vesakoski, O., & Briga, M. (2025). The spatial distribution of pertussis, but not measles or smallpox, in pre-industrial Finland matches dialects. *iScience*. DOI: 10.1016/j.isci.2025.112530

Nockels, J., Gooding, P., Ames, S., & Terras, M. (2022). Understanding the application of handwritten text recognition technology in heritage contexts: a systematic review of Transkribus in published research. *Archival science*, *22*(3), 367–392.

Pasanen, T.-M., Helske, J., Högmander, H., & Ketola, T. (2024). Spatio-temporal modeling of co-dynamics of smallpox, measles, and pertussis in pre-healthcare Finland. *PeerJ*, *12*, e18155.

Pitkänen, K. (1980). Registering people in a changing society-the case of Finland. *Finnish Yearbook of Population Research*, 60–79.

Tkachenko, M., Malyuk, M., Holmanyuk, A., & Liubimov, N. (2020-2025). *Label Studio: Data labeling software.* Retrieved from https://github.com/HumanSignal/label-studio (Open source software available from https://github.com/HumanSignal/label-studio)

Ultralytics. (2024). *Yolov11 implementation.* https://github.com/ultralytics/ultralytics. GitHub. (Retrieved from GitHub on 2024-12-01)