

Ribeiro, Leonardo Costa; Andersson, Jonatan; Skoglund, William; Molinder, Jakob;
Önnerfors, Martin

Working Paper

Automated historical census digitization using image augmentation and transformer-based methods

EHES Working Paper, No. 298

Provided in Cooperation with:

European Historical Economics Society (EHES)

Suggested Citation: Ribeiro, Leonardo Costa; Andersson, Jonatan; Skoglund, William; Molinder, Jakob; Önnerfors, Martin (2026) : Automated historical census digitization using image augmentation and transformer-based methods, EHES Working Paper, No. 298, European Historical Economics Society (EHES), s.l.

This Version is available at:

<https://hdl.handle.net/10419/339327>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



European
Historical
Economics
Society

EHES Working Paper | No. 298 | February 2026

Automated historical census digitization using image augmentation and transformer-based methods

Leonardo Costa Ribeiro¹
Federal University of Minas Gerais

Jonatan Andersson
Uppsala University

William Skoglund
Lund University

Jakob Molinder
Uppsala University

Martin Önnersfors
Uppsala University

<https://www.ehes.org/>

Automated historical census digitization using image augmentation and transformer-based methods

Leonardo Costa Ribeiro¹
Federal University of Minas Gerais

Jonatan Andersson
Uppsala University

William Skoglund
Lund University

Jakob Molinder
Uppsala University

Martin Önnarfors
Uppsala University

Abstract

A large literature in economic history uses digitized census data to study individual-level outcomes in history. Although many census records have been digitized manually, the process is extremely labor-intensive, and substantial material remains unprocessed in archives. Recent advances in machine learning offer the potential to automate large part of this work. We demonstrate an end-to-end digitization pipeline based on the transformer-based Donut model, trained on hand-annotated data and enhanced with image augmentation, to extract information from the 1955 Stockholm tax and census records. The resulting output attains high accuracy across multiple evaluation metrics.

JEL Codes: N01

Keywords: Digitization, Census, OCR, Transformers

¹ Correspondance: Leonardo Costa Ribeiro (lcr@cedeplar.ufmg.br)

The computations and data handling were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725. We are also grateful to the Stockholm City Archives and SwedPop for providing us with the original source material. We recognize funding support from Jan Wallander och Tom Hedelius stiftelse. Thank you to Auke Rijpma for the inspiration as well as Erik Andersson Perdahl and Evin Sayfoor for research assistance.

Notice

The material presented in the EHES Working Paper Series is property of the author(s) and should be quoted as such. The views expressed in this Paper are those of the author(s) and do not necessarily represent the views of the EHES or its members.

1 Introduction

Digitized censuses have for a long time been indispensable sources for economic historians studying outcomes at the individual level (Ferrie, 1999; Williamson, 1990). Over the last few years, the use of census data has gained even greater popularity. This is not the least due to the adoption of machine learning tools in economic history, which have allowed researchers to link individuals across censuses from various countries using probabilistic methods (Abramitzky et al., 2012; Collins and Wanamaker, 2014; Connor, 2019; Eriksson, 2015; Modalsli, 2017; Pérez, 2017; Long, 2005). Nevertheless, while linking already digitized records can be performed cheaply, the extraction of text and numeric characters from historical source material into machine-readable format remains costly. This is a consequence of the extraction process being extremely laborious and time-consuming. Thus, the efforts of digitizing historical censuses have for a long time relied on the collaboration of large teams of researchers and research assistants. For instance, digitizing the 1881 census of Canada and Great Britain and the 1880 census of the United States required thousands of workers who together spent 11 million work hours entering and verifying the information (Roberts et al., 2003). However, recent advancements in Optical Character Recognition (OCR) and Handwritten Text Recognition (HTR) mean that machine-learning methods can now substitute for labour with high efficiency. This creates opportunities for researchers to compile novel rich datasets quickly and at low cost.

Although OCR methods have been a revelation for historical studies, extracting data from documents with tables has long been a challenge. To read such materials, methodologies often employ a step-wise OCR procedure, starting with layout classifications, image cell recognition and finally OCR:ing (Dahl et al., 2023). While layout recognition is not always possible, traditional OCR/HTR approaches can instead require significant segmentation to recognize relevant text blocks (Blomqvist et al., 2023). Moreover, traditional OCR models often require excellent images and, therefore, struggle to produce accurate output when the ink is smudged, the text is bleeding through from the previous page, or when the paper is degraded.

In this paper, we demonstrate how to employ the transformer-based Donut model to bypass traditional OCR methods when digitizing historical census records. Specifically, we describe the strategies used to digitize the 1955 tax and census records (Swedish: *mantalslängder*) of Stockholm’s roughly 750 000 people using automated methods. From a digitization perspective, the tax and census records have some interesting attributes. The census portions of the records (e.g., occupational information, name, birth date, or birth place) were documented using stamps, giving the characters a typewritten appearance that are often slightly smudged and have noisy backgrounds. The columns containing information pertaining to taxes and income were, by contrast, documented in handwriting. The final dataset includes individual-level information on variables like income by source,

address, rent, employer, occupation, place of birth, year of birth, marital status, gender, and relation to other people in the census.

2 The Donut Model: End-to-End Document Understanding Without OCR

This section provides a conceptual overview of Document Understanding Transformer (Donut)—a core technical innovation enabling this project (Kim et al., 2022). Fundamentally, Donut is an artificial neural network—a sophisticated pattern-recognition algorithm—trained to interpret document images and extract their textual information and structure in a single, integrated step.

The model represents a significant conceptual shift from traditional automated document processing. Conventional methods rely on a multi-stage pipeline that begins with Optical Character Recognition (OCR), which first identifies individual characters or words to produce raw text. A separate process must then parse this text to identify relevant fields and impose structure. This pipeline is particularly fragile for historical documents, as OCR engines often struggle with degraded paper, unusual typefaces, and handwriting. Errors in initial character recognition propagate irrecoverably, and the separation of “reading” from “understanding” ignores the holistic visual context where spatial layout conveys critical meaning. Donut’s key innovation is to bypass explicit OCR entirely. Instead of learning to recognize characters, it learns a direct mapping from the entire visual representation of a document page to a structured textual representation of its content. It effectively treats document parsing as a task of “visual language translation”, converting the “language” of document images directly into the “language” of structured data strings.

Architecturally, Donut consists of two main components, functionally analogous to a human expert reading a form. The first is the Vision Encoder, a component known as a Swin Transformer, which acts as the “perceiver”. It processes the raw pixel data of the input image not by seeking characters, but by breaking the image into patches to create a compact numerical summary of the page’s visual features (Liu et al., 2021). This summary, or latent representation, encodes textures, stroke patterns of handwriting, spatial arrangements of text blocks, and the presence of checkboxes—capturing what the document looks like and how its information is organized. The second component is the Text Decoder, which functions as the “interpreter”. Based on a BART Transformer (Lewis et al., 2020), it takes the visual summary from the encoder and autoregressively generates a text sequence describing the document’s content in a structured format. It is trained to output this sequence directly as JSON (JavaScript Object Notation), a standard data format that can be easily converted to a spreadsheet or database row. For instance,

it might generate: {"name": "Andersson, Karl", "occupation": "Snickare", "taxable income": "12500"}. The decoder produces this sequence token-by-token, considering both the encoder’s comprehensive visual context and the text it has generated so far.

The model’s application involves distinct training and inference phases. Training requires a large dataset of document images paired with corresponding “ground truth” JSON annotations. During supervised learning, the model adjusts its internal parameters to ensure that the JSON output from its decoder matches the provided ground truth. For our specific application, we fine-tuned a pre-trained Donut model on a curated dataset of several thousand annotated 1955 Stockholm tax forms, thereby teaching it the specific fields, handwriting styles, and layout peculiarities of these documents. In the inference phase, the trained model performs a single forward pass for each form image in the archive. The image is fed into the encoder; the decoder generates the JSON string, which is then directly parsed and appended to our growing database. This end-to-end process proves significantly more robust to noise, variable handwriting, and layout shifts than traditional OCR-based pipelines.

For the economic historian, this approach offers several critical advantages. It demonstrates notable robustness to degraded sources such as smudged, faded, or skewed scans common in archival material. It capably handles mixed inputs, seamlessly processing documents containing both printed text and handwritten entries by learning from the overall visual pattern. The model develops an implicit understanding of structure by learning from the global document context; it can infer, for example, that a number written next to the printed label “Taxerad inkomst:” represents “taxable income”, even with poor handwriting, thereby resolving ambiguities that would stump a standard OCR engine. Finally, it delivers data pre-structured into defined fields, eliminating the need for complex and error-prone post-processing scripts. In essence, Donut functions as an automated, trainable research assistant. Once trained on a representative sample, it can extract structured data from vast collections of similar forms with high accuracy, transforming images of historical records into analyzable data and serving as the technical cornerstone of our pipeline, ranging from the 1955 Stockholm taxation forms to a comprehensive population database.

Having established Donuts end-to-end philosophy, a closer look at its two core components—the Swin Transformer encoder and the BART decoder—clarifies the technical execution of its “visual translation” capability. These architectures, while sophisticated in their full implementation, can be understood by their core design principles and the specific problems they solve within the document understanding pipeline.

2.1 The Vision Encoder: Swin Transformer

The Swin (Shifted Window) Transformer ([Liu et al., 2021](#)) is responsible for converting the raw, two-dimensional pixel grid of a document image into a coherent, compressed numerical representation that the decoder can interpret. Its design directly addresses a fundamental challenge: standard Transformers, originally designed for one-dimensional text sequences, are computationally inefficient for high-resolution images, where each pixel can be treated as a token.

The Swin Transformer introduces an elegant, hierarchical solution. Imagine analysing a complex document not by scrutinising every letter at once, but by first examining local windows—a group of words in a form field—and then progressively integrating information from adjacent windows to understand the page’s overall layout. This is the essence of Swin’s mechanism. It first divides the image into small, non-overlapping patches. Self-attention, the mechanism that allows a model to weigh the importance of different parts of the input, is computed within these local windows. This is highly efficient. In subsequent layers, it employs a “shifted window approach”, in which the window boundaries are shifted, enabling cross-window communication and building a broader understanding. Crucially, as the network deepens, adjacent patches are merged, creating a hierarchical representation that progresses from fine-grained details (stroke textures, character shapes) to larger structures (text lines, field boundaries, the spatial relationship between a label and its handwritten answer). This hierarchical, windowed attention makes Swin exceptionally adept at modelling the precise layout and long-range dependencies in a form, which is paramount for distinguishing, for example, income entered in a personal field from a total written in a summary box.

2.2 The Text Decoder: BART

The BART (Bidirectional and Auto-Regressive Transformer) ([Lewis et al., 2020](#)) model serves as the generative interpreter. It is a sequence-to-sequence model pre-trained on a massive corpus of text, giving it a deep, probabilistic understanding of language structure, syntax, and semantics. In Donut, its role is adapted from text-to-text translation to vision-to-text generation.

BART is conditioned on the visual “context” provided by the Swin encoder. Its task is to autoregressively generate the JSON string, one token (a subword unit) at a time. Each token it produces is based on two streams of information: first, the full visual context from the encoder, and second, the sequence of tokens it has already generated. This autoregressive property is essential—it allows the model to build a coherent, syntactically correct JSON structure, where the appearance of a closing brace depends on an opening brace, and the value for “occupation” follows its key. BART’s pre-training on denoising and text reconstruction tasks makes it particularly robust at generating plausible and

consistent sequences even when the visual input is ambiguous. When encountering a smudged salary figure, the model can leverage its learned priors from the training data to generate a probable, contextually appropriate number, guided by the surrounding visual cues from the forms layout.

2.3 Synergy in the Donut Framework

The synergy between Swin Transformer and BART is what enables Donuts OCR-free paradigm. The Swin encoder does not produce text; it outputs a rich, spatially aware model of the document’s visual grammar. The BART decoder is then trained to “query” this visual model to produce textual descriptions. It learns to ask, implicitly, “Given this visual pattern that typically surrounds a surname in the training data, what character sequence is most probable?” This bypasses the error-prone step of character segmentation and recognition, allowing the model to use all available contextual and visual information simultaneously to infer the content. For the historian, this means the system is not simply reading characters in isolation but is interpreting the document as a whole, much as a human researcher would, leading to greater accuracy and robustness when dealing with the heterogeneous and challenging nature of historical forms.

3 Building the Training Dataset: From Archival Scans to Annotated Pairs

The Donut model operates as a supervised learning algorithm. This means its capability to parse documents accurately is not inherent, but learned through exposure to examples where the correct output is known. Consequently, its performance is fundamentally contingent on the quality and representativeness of its training dataset, a carefully constructed collection of image-annotation pairs. Each pair consists of a digitized form image and its corresponding, manually verified structured annotation (in JSON format), which serves as the ground truth for the model to learn from.

Given the unique and historically specific nature of the 1955 Stockholm taxation and census forms—with their distinct layout, terminology, and a mix of printed and contemporary handwriting—no suitable pre-existing dataset was available. It was therefore necessary to construct a dedicated training corpus from the archival source material itself. This process required a multi-stage pipeline of preprocessing, annotation, and validation to transform raw, heterogeneous scans into a clean, consistent set of instructional examples for the model. The workflow for creating this essential training dataset is documented in the following subsections.

3.1 Image Acquisition and Denoising

The construction of the training dataset began with the preprocessing of raw digital scans from the Stockholm City Archives. This first stage aimed to standardise image quality and enhance legibility, forming a crucial foundation for all subsequent manual annotation. While the Donut model is designed to be robust to noise, systematic preprocessing reduces visual variance and minimises artefacts that could confuse the model or increase cognitive load for human annotators, thereby improving the efficiency and consistency of the ground-truth labelling process.

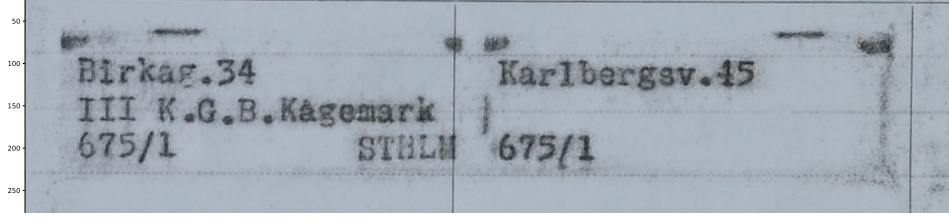
All image operations were performed programmatically using the OpenCV (CV2) library in Python, ensuring a reproducible and scalable pipeline. Each scan was first converted from its native colour space to the BGR (Blue, Green, Red) format required by OpenCV’s core functions. The primary challenge addressed was complex image denoising, targeting characteristic artefacts of historical documents such as paper degradation, ink bleed-through, foxing, and scanning artefacts.

To effectively remove this noise while preserving critical text information, we employed the `fastNlMeansDenoisingColored` function from OpenCV. This algorithm is particularly suited for historical documents, as it operates on the principle of non-local means: for a given pixel, it identifies similar patches across the entire image and uses a weighted average of these patches to compute a denoised value. This approach is superior to local filters for removing structured noise, such as stains or bleed-through, because it distinguishes random noise from the coherent, repeating patterns of text strokes. The function’s key parameters were empirically optimized on a representative subset. The optimization balanced aggressive noise suppression with the imperative to preserve the fine details of both printed and handwritten text, avoiding the blurring of character edges.

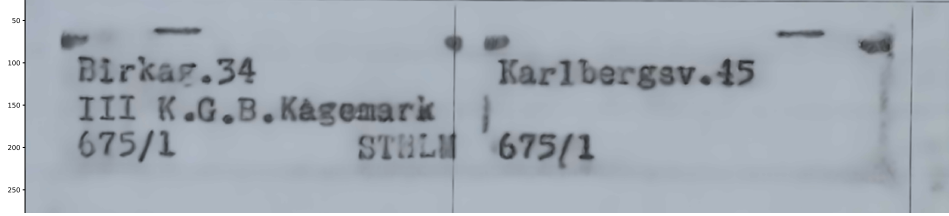
The output of this stage was a curated set of denoised document images with a uniform, clean background and enhanced text contrast. Figure 1 provides a visual demonstration of this denoising effect, displaying a representative crop from a 1955 taxation form before and after processing. The raw scan (Panel a) exhibits characteristic background mottle and faint ink bleed-through. The processed image (Panel b) shows a significantly cleaner background, while textual information is preserved with enhanced clarity. This qualitative improvement provided an optimal visual substrate for both the subsequent OCR-guided segmentation and the manual annotation tasks.

3.2 Strategic Segmentation for Targeted Model Training

Following initial denoising, a critical structural challenge was addressed. Each full-page taxation form presents a complex, dense layout with three distinct logical sections: a standardized header (with information on residential building identification) and a main table with columns detailing employer, personal information, income sources, deductions,



(a) Raw scan crop.



(b) Processed image after `fastNlMeansDenoisingColored`. Noise is suppressed while text structure is preserved.

Figure 1: Visual effect of the image denoising pre-processing step. (a) A representative raw crop from a 1955 Stockholm taxation form scan, exhibiting typical archival noise. (b) The same region after denoising with OpenCV’s non-local means algorithm. The preprocessing enhances text legibility and provides a standardized input for subsequent segmentation and annotation.

and other items. Presenting the entire, information-rich page as a single input to the Donut model risked overwhelming its capacity to learn precise spatial relationships between labels and values, potentially reducing extraction accuracy for individual fields.

To mitigate this, a strategic segmentation protocol was implemented. Each preprocessed form image was programmatically split into three consistent regions of interest (ROIs): (1) the header section, (2) the left-side page, and (3) the right-side page. This segmentation was guided by the known, fixed template of the 1955 forms, using coordinate boundaries. Consequently, three specialized Donut models were developed: Donut-Header, Donut-Left, and Donut-Right.

This modular approach conferred significant advantages. Each model could be trained to specialise in the specific graphical structure and vocabulary of its assigned section, learning a more precise mapping between the visual cues in that region and the corresponding JSON keys and values. The final structured data for each individual was later synthesized by algorithmically merging the JSON outputs from the three models, based on a unique form identifier parsed from the rows.

To achieve precise, consistent segmentation of each form into the three defined regions (header, left column, right column), a coordinate-based cropping strategy was employed. This step was necessary because the scanned images exhibited minor but consequential positional variance. Factors such as the book’s alignment on the scanner, the curvature of pages near the binding, and differences in cropping during digitization meant that the

target form content did not appear at fixed pixel coordinates across the image corpus.

To dynamically locate the form within each image, we utilized a targeted Optical Character Recognition (OCR) step, which is applied only to the top-left quadrant of the document, to identify invariant printed keywords that functioned as anchor points. Specifically, the model searched for fixed-header phrases such as “Adress” (address) and “Arbetsgivare” (Employer), which were consistently formatted across all forms. The pixel coordinates of these text elements were used to calculate a relative offset that defines the top-left corner of the standardized form template.

Once the anchor point was established, the three regions of interest were programmatically cropped using predefined pixel dimensions relative to that coordinate. For instance, the header was defined as a rectangle with a fixed width and height, extending from the anchor. The main body columns were then calculated as rectangles beginning a fixed distance below the header. This method ensured that, regardless of the form’s position in the scanned image, the cropped sections provided a normalized, consistent input to their respective Donut models. This preprocessing segmentation, distinct from the Donut model’s parsing operation, was essential for creating a homogeneous training set and enabling reliable inference, thereby standardizing the input space for subsequent specialized models.

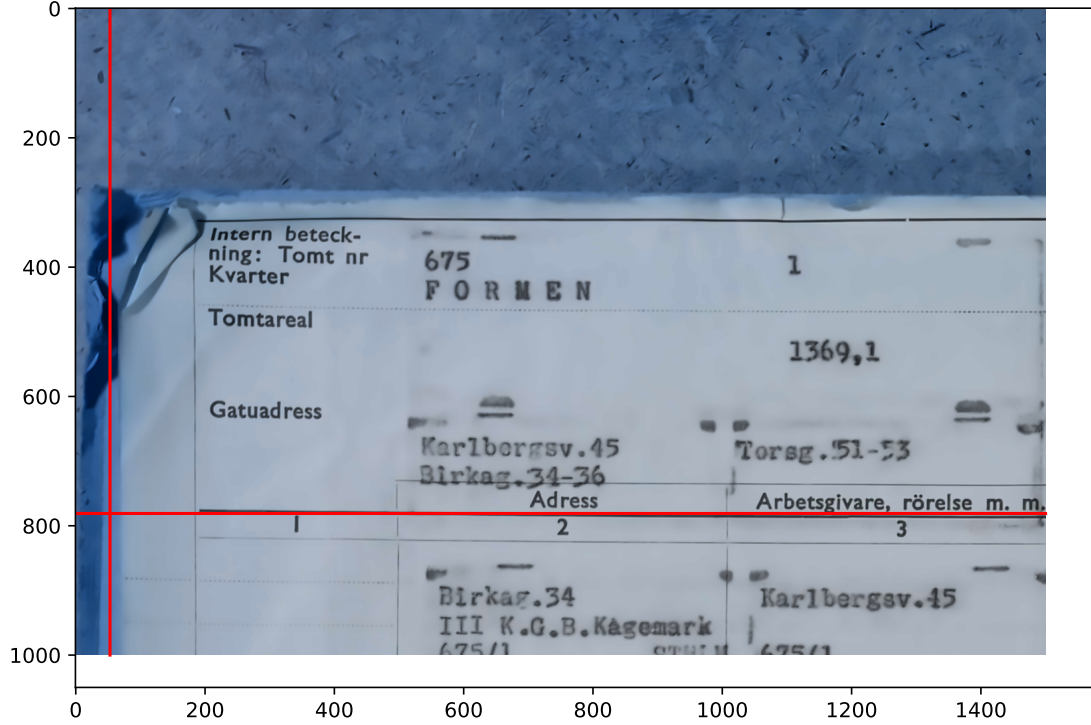
Figure 2 illustrates this segmentation logic. The located anchor points—the start of the main header line and the beginning of the tabular data body—are marked as the red reference lines in the figure. Based on these dynamically identified positions, the bounding boxes for the three ROIs (Header, Left Column, Right Column) were calculated using relative vertical and horizontal offsets. These offsets were empirically determined from a representative sample of forms and then applied uniformly across the entire corpus, ensuring robust and repeatable segmentation despite scan-to-scan variation.

3.3 Table Row Segmentation and Deskewing

Building on the column-based segmentation, a further refinement was necessary to handle the complex tabular structure of the income and deduction forms. For both the left and right columns, the decision was made to train the Donut models on individual table rows rather than on the entire column image. This approach addresses the same fundamental challenge of data complexity: presenting the model with a smaller, more consistent visual context—a single individual’s entry within its row boundaries—allows for more precise learning of the mapping between field positions and their values. Furthermore, this granularity significantly reduces the required training dataset size, as each row constitutes a distinct training example, effectively increasing the number of samples from a single form page.

The process of isolating these rows required accurate detection of the horizontal

Figure 2: Schematic of the OCR-guided segmentation process



Note: An example of segmentation for the main header line.

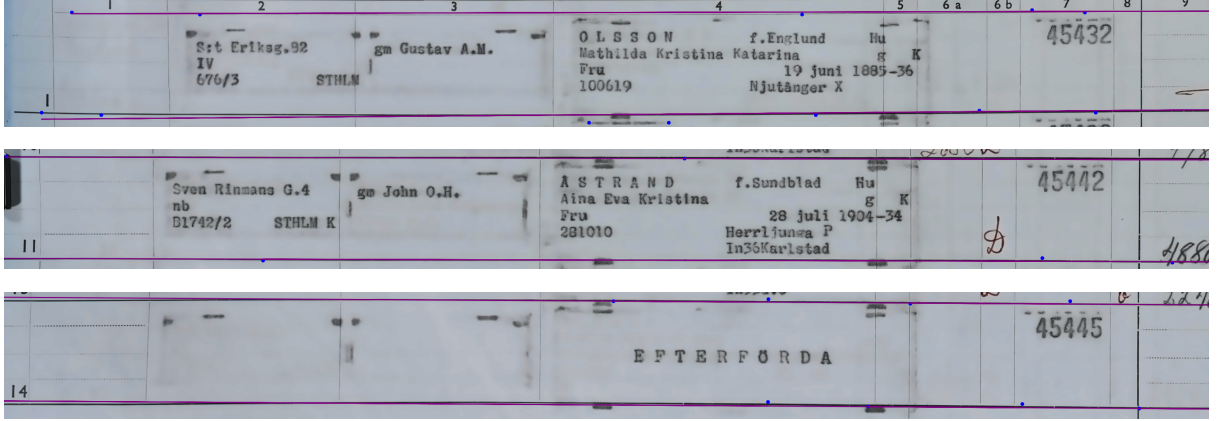
lines separating each entry. We began with the pre-cropped left- and right-ROI images. To identify the row separators, the Hough Line Transform algorithm (`HoughLinesP` in OpenCV) was applied. This algorithm detects line segments in an image by aligning edge pixels. For these forms, it effectively identified the discontinuous printed lines (and often their visual continuations through handwritten text) that demarcate the rows.

The output of `HoughLinesP` is a set of discrete line segments for each horizontal rule. To reconstruct a single, continuous separator line for each row boundary, the endpoints of all line segments associated with a given separator were aggregated. Points were sampled along these segments, and a linear regression was fitted to the resulting point cloud. This step was critical because it provided a robust mathematical representation of the separator line that compensated for gaps, ink smudges, and minor curvature.

Figure 3 illustrates this process. The blue circles denote the sampled points from the detected line segments for a given row separator, and the resulting fitted regression line is drawn in purple. Crucially, the figure demonstrates that the inclination (skew) of these lines is not constant across the page. Due to the curvature of the bound ledger book during scanning, the top rows exhibit a positive inclination, the middle rows are nearly horizontal, and the bottom rows show a negative inclination. This non-uniform skew precludes applying a single, global deskewing correction to the entire column.

Therefore, a row-specific deskewing procedure was implemented. For each row i , defined as the region between the fitted separator line L_i (top) and L_{i+1} (bottom), the

Figure 3: Example of Row Separator Line Detection



Note: Detection and modeling of row separator lines in the left column ROI. The light cyan circles represent points sampled from line segments detected by the Hough Transform algorithm for three horizontal rules. The purple lines show the linear regression fitted to each set of points. The variation in line inclination—positive at the top, near-zero in the middle, and negative at the bottom—illustrates the non-uniform skew caused by book curvature during scanning, necessitating row-specific deskewing.

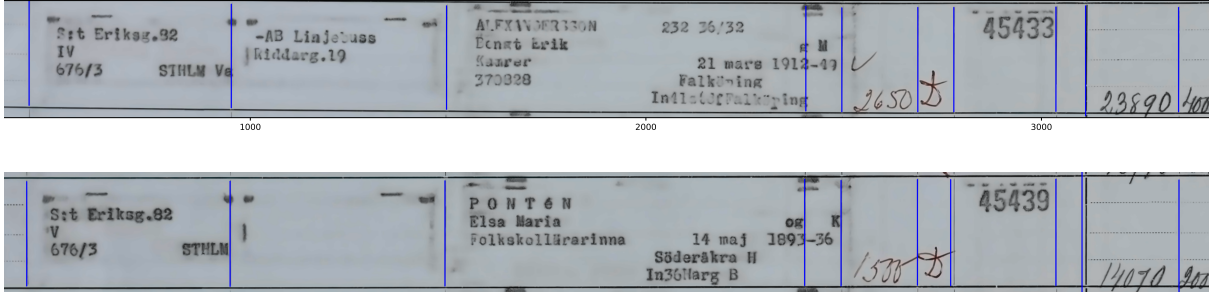
inclination angle θ_i of the top line L_i was calculated from its regression coefficients. The image region containing the row was then cropped using the y-coordinates defined by these lines and rotated by $-\theta_i$ around its center, effectively aligning the row’s baseline to horizontal. This localized correction ensured that the text and numerical values in each row were presented to the Donut model in a standardized, rectified orientation, dramatically improving the consistency of the visual input and the model’s subsequent ability to learn spatial relationships among the row’s fields.

Figure 4 presents a sample of the final, pre-processed inputs derived from the row segmentation and deskewing pipeline. The figure displays a sequence of cropped and rectified individual table rows. As the output demonstrates, the horizontal text baselines and the printed ruling lines within each row are now consistently aligned. The visual variation introduced by the page’s global curvature—manifest as divergent row inclinations in the raw scan—has been successfully transferred to the top and bottom margins of the individual row images. This margin acts as a buffer zone that contains the residual, rotated ends of the separator lines, ensuring that the critical data-containing center of each row is presented in a normalized, horizontal orientation.

3.4 Vertical Column Delineation within Rows

Following the horizontal deskewing of individual rows, the final preprocessing step involved identifying vertical lines that delineate the distinct data columns within each table row. This delineation is essential to provide the Donut model with spatially normalized inputs, in which the position of each data field (e.g., occupation, income amount, taxable

Figure 4: Example of Cropped and Deskewed Table Rows



Note: Sample of cropped and deskewed individual table rows. After row-specific rotation, the horizontal ruling lines and text baselines within the data region of each row are consistently aligned. The residual skew from the original page curvature is absorbed into the top and bottom margins of each row image, creating a standardized, rectified input for the Donut model. This normalization ensures consistent spatial relationships between field labels and handwritten values across all training examples.

income) is consistent.

To detect these vertical separators, we again employed the Probabilistic Hough Line Transform (`HoughLinesP`), but with parameters optimized for detecting near-vertical features. Due to the prior row-specific deskewing, these lines appear almost perfectly vertical in the processed row images, simplifying their detection. The algorithm successfully identified the majority of the printed vertical rules.

However, due to variations in print quality, ink fading, or scanning contrast, some faint vertical lines were not detected by the Hough Transform. To ensure a complete and consistent column structure across all rows, we implemented a rule-based inference procedure. For each missing line, its position was inferred based on the average inter-column distance calculated from a large, representative sample of successfully parsed rows. This defined a canonical, average column template for the form. If a line was missing in a particular row, its x-coordinate was estimated by applying this average offset from the nearest reliably detected vertical line or from the row’s left edge.

The resulting column boundaries for a sample deskewed row are visualized in Figure 4. The blue vertical lines overlaid on the row image indicate the final, complete set of column separators—combining those directly detected by `HoughLinesP` and those inferred from the average template.

3.5 Hybrid OCR Pipeline for Typed and Handwritten Text Recognition

With individual rows cropped, deskewed, and their column boundaries defined, the pre-processing step involved extracting machine-readable text from each column to create

the structured annotations required for training the Donut models. Given the mixed nature of the forms—containing both pre-printed (typed) text and handwritten entries—a purpose-built, hybrid Optical Character Recognition (OCR) pipeline was implemented. This pipeline operated on the column-segmented regions of the deskewed row images to maximise accuracy.

For columns containing typewritten Swedish text, we employed a dual-engine approach using Tesseract OCR and EasyOCR. Both engines were applied to the same column region, and their outputs were compared. Based on preliminary quantitative tests on a held-out validation set of typed text, Tesseract, when it returned a high-confidence result, demonstrated superior accuracy for the specific typeface and linguistic content of these forms. Therefore, a confidence-weighted decision rule was implemented: the Tesseract output was prioritized and selected as the initial annotation. Only when Tesseract failed to return a result above a defined confidence threshold did the pipeline default to EasyOCR output, serving as a robust fallback.

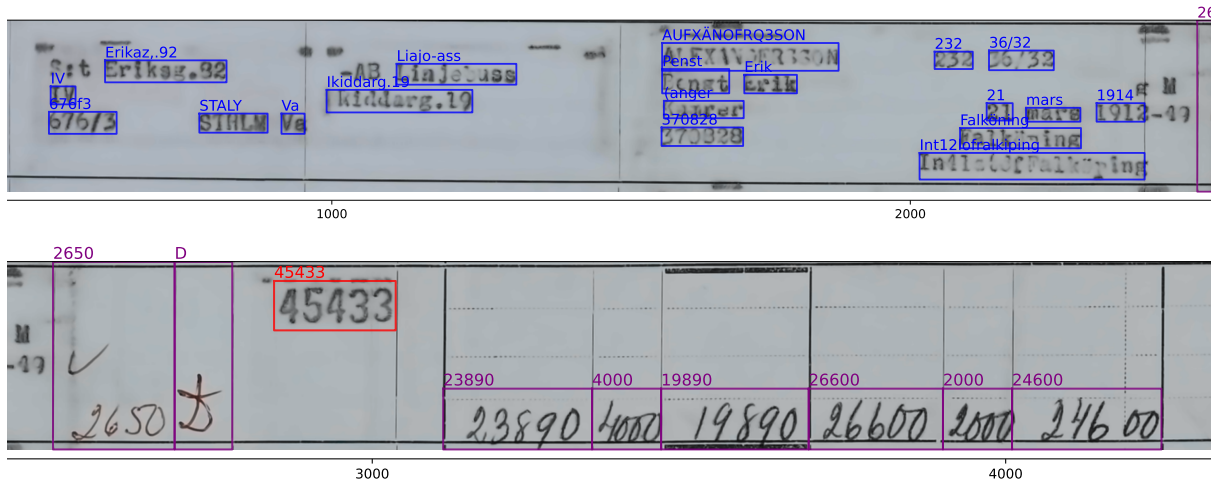
For columns containing handwritten entries, we utilized a specialized model: the HUMLA model, a fine-tuned Swedish TrOCR (Transformer-based OCR) described in [Skoglund et al. \(2025\)](#). This model, based on the Vision Encoder-Decoder architecture and pre-trained on extensive Swedish handwritten text datasets, is specifically optimized for the morphological characteristics of Swedish cursive and block handwriting, offering significantly higher accuracy than generic handwriting recognition engines.

Figure 5 illustrates the output of this complete OCR pipeline on a sample row. The visualization overlays the recognition results on the original row image: blue bounding boxes and text denote the Tesseract output for typed fields; red bounding boxes and text (visible where Tesseract confidence was low) denote the EasyOCR output; and purple text annotations show the transcribed handwritten text from the Humla TrOCR model.

3.6 From OCR Output to Structured Annotations

An essential step in constructing the training dataset is to synthesise structured annotations from the OCR output. Since the precise pixel coordinates of each text recognition and the canonical layout of the table—defined by the detected horizontal and vertical lines—are known, we can algorithmically map each recognized text string to its corresponding semantic field. This mapping is governed by a coordinate-based rule system: the bounding box of each OCR result is compared against predefined regions for each field in the column template. When the centroid of a text bounding box falls within a specific field’s region, the recognized text is assigned to that field’s key in the JSON object. This process creates the final image-annotation pair: the preprocessed row image serves as the input, and the generated JSON object, containing all field-value mappings for that row, serves as the ground truth annotation for training the Donut model.

Figure 5: Multi-Model Output for Training Data



Note: Visualization of the hybrid OCR pipeline applied to a deskewed table row. Blue bounding boxes and text indicate successful Tesseract OCR recognition of typed fields. Red bounding boxes and text indicate EasyOCR’s recognition, used when Tesseract’s confidence fell below the threshold. Purple text annotations display the output of the specialized Humla Swedish TrOCR model for handwritten entries. This multi-engine approach ensures robust text extraction from both printed and handwritten content, generating the structured ground truth annotations required for Donut model training.

Figure 6 illustrates a complete example of this pairing. The top panel shows the preprocessed row image with column boundaries. The bottom panel displays the corresponding JSON annotation generated by the mapping process, structured for clarity. The JSON keys correspond to the standardized field names used in our database schema.

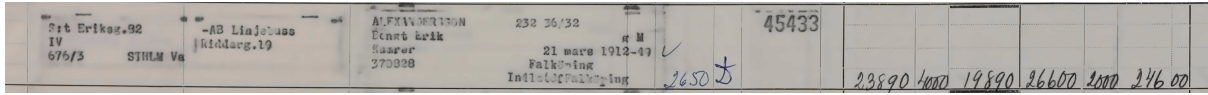
This automated pipeline—from denoising and segmentation to hybrid OCR and coordinate mapping—enabled the efficient creation of thousands of such annotated pairs associated with the left- and right-side pages and the header.

3.7 Manual Verification and Correction of Ground Truth Annotations

As illustrated in Figure 6, the automated OCR pipeline, while efficient, inevitably introduces recognition errors—character misreads, word segmentation mistakes, and misinterpretations of handwritten text. Directly using these raw OCR outputs as the ground truth for training the Donut models would lead to the undesirable propagation and reinforcement of these errors within the model’s learning process. To ensure the highest possible data quality, the raw annotations underwent mandatory manual verification and correction before being finalized as training targets.

This correction process was facilitated by a custom Python application specifically developed for this project. The application presented a dynamic interface that displayed,

Figure 6: Example of a Training Image With Model Output



```
{street_address: "Erikaz,.92",      complement_address: "IV 676f3",
  place_name: "STALY Va",           employer_row1: "Liajo ass",
  employer_row2: "Ikiddarg.19",     employer_row3: "",
  surname: "AUFXÄNOFRQ3SON",       maiden_name: "232 36/32",
  wife: "",                         first_name: "Penst Erik",
  marital_status: "og",             occupation: "anger",
  date_birth: "21 1914 mars",       house_hold_id: "370828",
  place_birth: "Falköning",         nationality: "",
  moved_in: "Int12lofralkiping",    gender: "",
  personal_id: "",                  rent: "2650",
  tax_submitted: "D",               register_id: "45433",
  state_taxed_income: "23890",       state_deduction: "4000",
  state_taxable_income: "19890",     municipality_taxed_income: "26600",
  municipality_deduction: "2000",    municipality_taxable_income: "24600"}
```

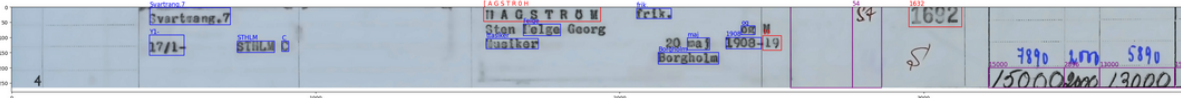
Note: Example of a final training data pair. Top: The preprocessed row image with identified column boundaries. Bottom: The corresponding structured JSON annotation, generated by mapping the OCR output from Figure 5 to field keys based on spatial coordinates. This image, JSON pair constitutes one training example for the Donut model.

for each row, the original preprocessed image alongside the raw OCR text for each field, arranged according to the established database schema. Two independent researchers systematically reviewed each data pair, comparing the OCR-suggested text with the image’s visual evidence. They corrected any errors, such as misreading “sångare” as “anger” or parsing address components incorrectly, directly within the interface. Corrections were saved upon validation, and the application then advanced to the next image-annotation pair, ensuring a streamlined and auditable workflow. This dual-reviewer protocol improved accuracy and consistency, establishing a high-fidelity ground-truth dataset.

Figure 7 provides a screenshot of this annotation correction application. The interface clearly juxtaposes the source image with the editable annotation fields, demonstrating the practical workflow that enabled the creation of a cleaned, reliable training corpus. This meticulous manual step was critical for transforming the output of the automated preprocessing pipeline into a trustworthy foundation for supervised model training, ultimately ensuring that the Donut models learned from accurate representations of the historical records rather than from the artefacts of the OCR process.

Figure 7: Manual Verification and Correction Tool in Python

Current Image Position: 1, Left Images: 49, Image Name: SE_SSA_0031_06_G_1_BA_154_002_033_row_04.jpg



Previous Im... Save Labels Next Image ☒ Show OCR Output Image Status: Corrected by Erik

Street_Address: Svartmang.7 ☐ Complement_Ad... 17/1- ☐

Place_Name: STHLM C ☐ Employer Row1: ☐

Employer Row2: ☐ Employer Row3: ☐

Surname: HAGSTRÖM ☐ Maiden_name: frik. ☐ Wife: ☐

First_Name: Sten Helge Georg ☐ Marital_Status: og ☐

Occupation: Musiker ☐ Date_Birth: 20 maj 1908 ☐

House_Hold_ID: ☐ Place_Birth: Borgholm ☐

Nacionality: ☐ Moved_in: ☐

Gender: ☐ Personal_ID: 19 ☐

Rent: ☐ Tax_Submitted: ☐ Register_ID: 1632 ☐

State_Taxed_Income: 7890|15000 ☐ State_Deduction: 2000|2000 ☐ State_Taxable_Income: ☐

Municipality_Taxed_Income: 8850|15000 ☐ Municipality_Deduction: 1400|1400 ☐ Municipality_Taxable_Inco: ☐

Note: Screenshot of the custom Python application for manual verification and correction of ground truth annotations. The interface displays the preprocessed row image alongside editable fields populated with the raw OCR output. Researchers compare the visual information in the image with the suggested text, correct errors directly in the form fields, and save the validated annotations. This step ensures the creation of a high-fidelity training dataset, preventing OCR errors from propagating to the Donut models during training.

4 Training the Donuts Models

With the curated and verified dataset of image-annotation pairs for each segment, the next phase involved training the three specialized Donut models: one for the Left-side page (personal and income information), one for the Right-side page (income origin), and one for the Header (blocks information). The training followed a standard supervised learning procedure, optimized for the specific characteristics of our historical document data. Crucially, the right-side page contains a summary of incomes, allowing us to compare the output for individual income fields to the sum in order to test the precision of our model.

For each model type, the corresponding dataset was first randomly split into a training set (90% of the pairs) and a held-out test set (10%). The test set was reserved exclusively for the final evaluation of model performance and was never used during training or validation.

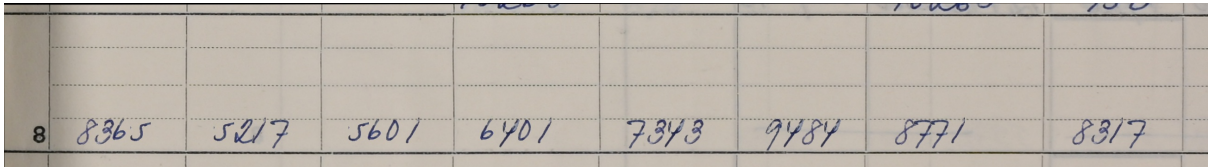
To improve model generalization and robustness to imaging variations not fully ad-

dressed in preprocessing, we applied data augmentation to the training images. This technique artificially expands the training dataset by creating modified versions of the original images. Our augmentation pipeline included random adjustments to brightness and contrast (simulating different lighting or scan conditions) and the addition of subtle, plausible noise (simulating minor paper degradation or scanning artefacts).

To improve model efficiency, another image augmentation process was also implemented, developed after tests showed an increased error rate specifically for rare registry fields. Those errors were overwhelmingly concentrated in fields related to agricultural income and acquisition deductions, which were relatively rare in the 1955 Stockholm urban population and were consequently severely underrepresented in our original training dataset. The model, having seen few examples, failed to learn robust visual features for these specific numerical entries.

To mitigate this data imbalance, we devised a novel, targeted image augmentation strategy. We identified scanned forms with blank fields, i.e., no income was registered to that individual. We then programmatically extracted numerical sub-images from forms where the fields were populated. Then, the backgrounds of these number patches were removed, and we pasted them onto the corresponding empty cells in the target images and pasted over the empty registers. This process generated synthetic, yet visually authentic, training examples that filled all numerical columns with plausible values. The visual fidelity of these augmented images is such that they are virtually indistinguishable from genuine entries to the human eye, as illustrated in Figure 8.

Figure 8: Synthetic Image Example.



8	8365	5217	5601	6401	7343	9484	8771	8317
---	------	------	------	------	------	------	------	------

Note: Example of synthetically augmented row image of the right-side page created to address underrepresented field types. All fields were filled with random numbers.

We generated approximately 300 such augmented images, representing a 10% increase to the existing training set, ensuring balanced representation across all field types. Adding this synthetically enriched data to the training corpus and retraining the model yielded a significant improvement. The mismatch rate dropped to about 4%.¹ Furthermore, the remaining errors were predominantly minor, involving single-digit discrepancies. When applying a more stringent validation criterion—considering only cases where the difference between the sum and the total exceeded 10% of the total—the error rate fell to 2%. This

¹The mismatch rate in this case is the mismatch between the summary income field and the sum of all other fields.

level of accuracy is comparable to, or surpasses, the error rate expected from manual data entry.

This targeted augmentation methodology represents a powerful and efficient tool for historical document processing. It dramatically reduces the dependency on large volumes of rare, naturally occurring examples in the source material. By programmatically creating a vast combinatorial space of plausible entries from a limited set of base images and value patches, we can generate a comprehensive and balanced training dataset. This approach will be instrumental in efficiently scaling our pipeline to process taxation forms from other years, where the distribution of economic activities may differ, without the prohibitive cost of annotating thousands of new rare-case examples by hand.

The final number of manually verified image-annotation pairs available for training each model after the train-test split is summarized in Table 1. The Left-side page model, which parses the most extensive and complex table, was trained on the largest corpus. The Right-side page and Header models, dealing with more compact structures, were trained on proportionally smaller but sufficient datasets.

Table 1: Number of unique, verified image-annotation pairs used for training each Donut model after the 90%-10% train-test split and augmentation.

Left-side page	Right-side page	Header
2,198	4,234	696

Each Donut model was trained for 3 epochs using the following key hyperparameters, implemented within the PyTorch Lightning framework:

- Learning rate: 3×10^{-5}
- Training batch size: 16
- Validation batch size: 16
- Gradient clipping value: 1.0 (to prevent exploding gradients)

To assess the performance of the trained Donut models, a comprehensive evaluation was conducted on the held-out test sets. Given the structured nature of the model’s output—a JSON (or equivalently, XML-like) representation comprising both field identifiers (keys) and their associated text values (strings)—two complementary evaluation metrics were employed. These metrics measure accuracy at different levels of the output structure.

Jaccard similarity for Structural Accuracy: This set-based metric evaluates the correctness of the extracted field identifiers JSON tags. For each parsed document, the set of predicted keys is compared to the set of ground truth keys. The Jaccard similarity

coefficient J is computed as the size of the intersection divided by the size of the union of the two sets: $J(A, B) = |A \cap B| / |A \cup B|$. A high Jaccard similarity indicates that the model successfully identifies which fields are present in the document.

Jaro-Winkler Distance for Content Accuracy: This string-based metric assesses the accuracy of the extracted text values within the identified fields. The Jaro-Winkler distance measures the similarity between two strings (the predicted and ground-truth values), considering character matches and transpositions, and assigning greater weight to common prefixes—a useful property for names, addresses, and other entries where initial characters are particularly informative. The distance ranges from 0 (identical strings) to 1 (completely dissimilar). For each correctly identified field (i.e., fields present in both the prediction and ground truth sets), the Jaro-Winkler distance is computed between the corresponding text strings, providing a fine-grained measure of the model’s textual extraction fidelity.

Levenshtein distance: The Levenshtein distance is a standard distance metric that measures the edit distance between two strings, taking into account the position of characters. Comparing two strings with ten characters but no similarity would have an edit distance of ten, conversely two strings that match perfectly have an edit distance of zero. By dividing the edit distance with the number of characters in the string we achieve a character error rate representing the percentage of characters that are erroneous.

Together, these metrics provide a twofold assessment: the Jaccard similarity measures the model’s ability to understand the document’s schema (which fields exist), while the Jaro-Winkler and Levenshtein distances measure its precision in transcribing the content within those fields. This combination is essential for a complete evaluation of a document understanding system, as it separates structural recognition errors from textual transcription errors. We display the Jaccard, Levenshtein and Jaro-Winkler metrics in Table 2. In Figure 9 we look at error heterogeneity by comparing the Levenshtein and Jaro-Winkler across cells and observations.

5 Results

The performance of each specialized model on its respective test set is quantified in Table 2. The reported values are the mean distances across all test documents.

The results demonstrate high performance across all models. The Jaccard scores indicate near-perfect identification of the relevant fields, with all models achieving a structural accuracy above 0.96. The low Jaro-Winkler distances confirm that the text content extracted for those fields is highly accurate, with an average character-level dissimilarity of only 2-2.4%. This validates the effectiveness of both the preprocessing pipeline and the Donut architecture for the task of extracting structured data from historical taxation forms.

Table 2: Model precision metrics on the testing set.

	Left-side page	Right-side page	Header
Jaccard Similarity	0.984	0.989	0.973
Jaro-Winkler Distance	0.0227	0.0102	0.0170
Levenshtein Distance	0.0185	0.0132	0.0262

Note: Jaccard distance measures structural field identification accuracy (higher is better, 1 = perfect). Jaro-Winkler distance measures textual content similarity (lower is better, 0 = perfect). Levenshtein distance is edit distance between two strings divided by the length of the string (0 = perfect)

Table 3 presents a representative example of a complete inference from the Left-side page model on a test-set document (Prediction), along with its ground-truth annotation (Real). This example was selected because its Jaro-Winkler distance (0.02) closely matches the model’s global average precision (0.0215, as reported in Table 2), making it representative of typical performance.

Table 3: Example inference from the Left-side page Donut model.

Field	Prediction	Real
Munic. Taxed Income	24970	24970
State Taxed Income	22360	22360
Gender	M	M
Register ID	1910	1910
Maiden name	frik.	frik.
Occupation	Konstnär	Konstnär
Tax Submitted	D	D
Place Birth	Kristine W	Kristine W
Marital Status	gm	g
Place Name	STHLM	STHLM
Surname	NORELIUS	NORELIUS
Munic. Taxable Income	22970	22970
Street Address	Köpmang.11	Köpmang.11
Municipality Deduction	2000	2000
Rent	1900	1900
Date Birth	20 juli 1900	20 juli 1900
First Name	Einar Vilhelm Gottfrid	Einar Vilhelm Gottfrid
Complement Address	IV 17/18	IV 17/18
Personal ID	65	65
State Deduction	4000	4000
House Hold ID	250412	230412
State Taxable Income	18360	18360
Moved in	In42St.Kopparb. W	In42St.Kopparb. W

Note: The table shows example output for the left-side of the Stockholm census records.

The Jaro-Winkler distance for this record is 0.02. Two character-level errors are highlighted in red. All other fields are perfectly transcribed.

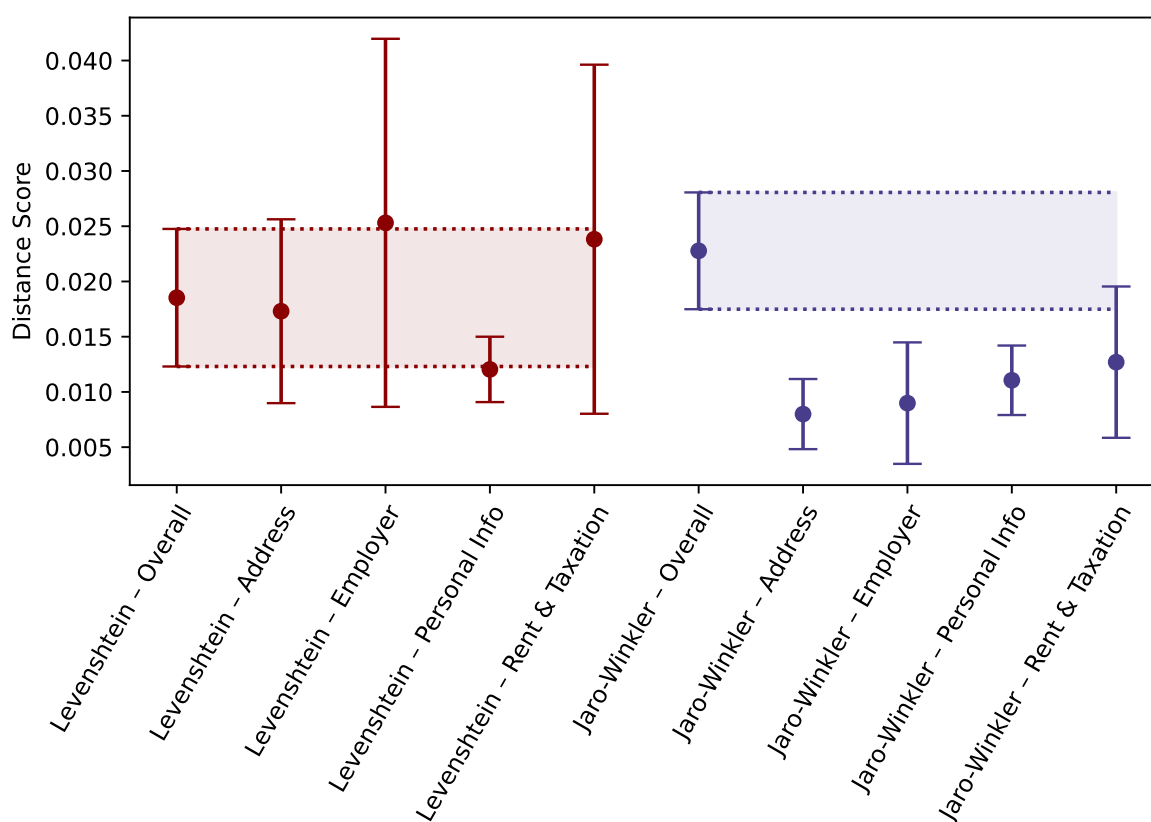
The inference demonstrates the model’s high practical utility. Of the 23 fields, 21 are transcribed perfectly. The two errors are minor character substitutions, highlighted in red: 1) an extra character ‘m’ appended to the single-letter marital status field (‘g’ predicted as ‘gm’), and 2) a single-digit error in the Household ID (‘5’ predicted instead of ‘3’).

In Figure 9 we show a detailed breakdown of the distribution of errors across observations and types of cells. Cells with more standardized information, i.e., personal information (names, place of birth, birth year etc.) display little variation in the error rate across observations as measured by the Levenshtein distance. However, information regarding employers and numerical values for rent and income perform relatively worse.

These strings are both less standardized and often very long, resulting in a more heterogeneous error rate. The Jaro-Winkler similarity, which emphasizes similarity at the start of strings, shows remarkably low edit distances across all cells, with little variation across observations.²

Figure 9 further shows that there are only negligible differences in the performance of our model between hand- and typewritten cells. The rent and taxation values are all handwritten but show a similar error rate to the rest of the cells which are typewritten. Not only does the model handle both types of writing but it does it simultaneously and with similar precision.

Figure 9: Model Performance Across Cells



Note: Figure shows text similarity between model output and corrected text across cells and observations. Points show the average score, confidence intervals at the 95% level.

²The median combined employer string has a length of 17 characters whereas the median name and surname has a length of 13 and 8 respectively.

6 Conclusions

Digitized censuses have become vital data sources for research in economic history. Nevertheless, the costs associated with digitizing census records have restrained the availability of new datasets. Recent developments in machine learning, however, create possibilities to extract information rapidly, at low cost, and without compromising on quality.

In this article, we have demonstrated the advantages of the transformer-based Donut model in extracting information from census records where the characters are either machine-written (stamps) or handwritten. Even though the text is smudged, there is bleed through between pages, and the cells are noisy, the model produces excellent output at a low error rate. As we display in Figure 9 the errors produced by the model are homogeneous across cells, performing slightly worse on less-standardized and longer strings. Performance does not seem to differ across types of written material, as the handwritten text has similar precision to that of the typewritten. Our overall character error rate of 1.3%-1.8%, minimizes the need for post-inference cleaning, further lowering the costs of creating databases.

A clear strength of the donut-approach is the minimal image- and line segmentation necessary in the pre-processing. This flexibility is especially important where text has been crossed-out and/or replaced with new text. For less information-dense materials the model can be applied without any segmentation. This advantage compared to traditional OCR makes Donut models especially valuable for complex historical source materials with varying layouts.

Since the model can be retrained on other historical source material, it is feasible to scale up the production of historical micro data using the strategy proposed in this article. A clear future avenue of research is the use of existing digitized censuses to train a Donut model that can push the frontier of digitized historical microdata.

Acknowledgments

The computations and data handling were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725. We are also grateful to the Stockholm City Archives and SwedPop for providing us with the original source material. We recognize funding support from *Jan Wallander och Tom Hedelius stiftelse*. Thank you to Auke Rijpma for the inspiration as well as Erik Andersson Perdahl and Evin Sayfoor for research assistance.

References

- Abramitzky, R., Boustan, L. P., and Eriksson, K. (2012). Europe’s tired, poor, huddled masses: Self-selection and economic outcomes in the age of mass migration. *American Economic Review*, 102(5):1832–1856.
- Blomqvist, C., Enflo, K., Jakobsson, A., and Åström, K. (2023). Reading the ransom: Methodological advancements in extracting the Swedish Wealth Tax of 1571. *Explorations in Economic History*, 87:101470.
- Collins, W. J. and Wanamaker, M. H. (2014). Selection and economic gains in the Great Migration of African Americans: New evidence from linked census data. *American Economic Journal: Applied Economics*, 6(1):220–252.
- Connor, D. S. (2019). The cream of the crop? geography, networks, and Irish migrant selection in the age of mass migration. *The Journal of Economic History*, 79(1):139–175.
- Dahl, C. M., Johansen, T. S. D., Sørensen, E. N., Westermann, C. E., and Wittrock, S. (2023). Applications of machine learning in tabular document digitisation. *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, 56(1):34–48.
- Eriksson, B. (2015). *Dynamic Decades: A Micro Perspective on Late Nineteenth-Century Sweden*. PhD thesis, Lund University.
- Ferrie, J. P. (1999). *Yankees Now: Immigrants in the Antebellum US, 1840–1860*. Oxford University Press.
- Kim, G., Hong, Y., Yun, S., and Kim, J. (2022). Donut: Document understanding transformer without OCR. In *European Conference on Computer Vision (ECCV)*. Springer.

- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Long, J. (2005). Rural–urban migration and socioeconomic mobility in victorian britain. *The Journal of Economic History*, 65(1):1–35.
- Modalsli, J. (2017). Intergenerational mobility in norway, 1865–2011. *The Scandinavian Journal of Economics*, 119(1):34–71.
- Pérez, S. (2017). The (South) american dream: Mobility and economic outcomes of first- and second-generation immigrants in nineteenth-century argentina. *The Journal of Economic History*, 77(4):971–1006.
- Roberts, E., Ruggles, S., Dillon, L. Y., Gardarsdottir, Ó., Oldervoll, J., Thorvaldsen, G., and Woollard, M. (2003). The North Atlantic Population Project: An overview. *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, 36(2):80–88.
- Skoglund, W., Molinder, J., Ericsson, J., and Berger, T. (2025). Labor market concentration during industrialization. Working paper.
- Williamson, J. (1990). Migrant earnings in britain’s cities in 1851: Testing competing views of urban labour market absorption. *Journal of European Economic History*, 19(1):163.

EHES Working Paper Series

Recent EHES Working Papers

- EHES 297** The Jamaica Life Expectancy Paradox and Hookworm Eradication
Elisabeth Preyer, Eric Strobl
- EHES 296** Historical social tables: advantages, methodology, and problems
Philipp Erfurth, María Gómez León, Giacomo Gabbuti, Branko Milanović
- EHES 295** Physical Memories of the Past and Support for the Far-Right: Evidence from Inter-War Denmark
Lasse Aaskoven, Christian Vedel
- EHES 294** The Early Evolution of Segregation and Neighborhood Disadvantage: Evidence from Stockholm During Industrialization, 1878–1950
Jakob Molinder, Martin Önnersfors
- EHES 293** „Vinum regnum, rex vinorum” Royal and Cameral Vineyard Possession in the Tokaj Estate in the 18th Century (Reforms, Viticulture, Wine Treatment, and Their Implications)
Attila Ulrich
- EHES 292** Unions and the Great Leveling: Evidence from Across the Atlantic
William Skoglund, Jakob Molinder,
- EHES 291** The Great Inner Divergence: TFP and Manufacturing Dualism in Industrializing Empires before WWI
Guillem Blasco-Piles
- EHES 290** Mitigating the impact of extreme temperatures: the role of public health interventions in Germany, 1888-1913
Daniel Gallardo-Albarrán
- EHES 289** Top Income Concentration in Early Modern Spain, 1574-1799
Carlos Álvarez-Nogal, Leandro Prados de la Escosura

All papers may be downloaded free of charge from: <https://www.ehes.org/>

The European Historical Economics Society is concerned with advancing education in European economic history through study of European economies and economic history. The society is registered with the Charity Commissioners of England and Wales number: 1052680.