CM3070 Final Project

KETUVIM: Machine learning for the Jewish records from the Russian Empire

Final report

Template: 'Gather Your Own Dataset' (CM3015 Machine Learning and Neural Networks)

by Sergey Yanovskiy

Github repository: https://github.com/kaneepston/ketuvim

Word count: 9353 words

Table of contents

Introduction

Tracing one's family history has become a global endeavour, yet for many individuals of Eastern European Jewish descent, this pursuit is often hindered by the scarcity and inaccessibility of historical documents. Unlike genealogical research relying on systematically maintained civil registries or readily digitised sources, inquiries into Eastern European Jewish ancestry must navigate fragile materials, non-standardised record-keeping, and linguistic complexity.

Domain

The administrative diversity of the Russian Empire contributed to irregular documentation practices, resulting in a wide range of sources—religious community registers, population censuses, birth, marriage, and death records from synagogues and kehillas (Jewish communal organisations), as well as various lists drafted in Cyrillic script by non-native Russian speakers.

Many such records, composed in Cyrillic under the Russian Empire, are now scattered across different countries and archival repositories. Over time, these documents have suffered fading ink, irregular handwriting, and outdated orthographic conventions, rendering even basic transcription a challenge.

This difficulty is further compounded by the fact that, of the approximately 25.5 million Jews worldwide [The Jewish Agency 2023], and many of them do not speak Russian or can read Cyrillic. As a result, countless descendants find themselves at an impasse, unable to reconcile their families' past with the present. Ensuring that these invaluable records are accurately transcribed and made accessible is a crucial act of cultural preservation, personal identity formation, and the safeguarding of collective memory.

Users

- **Amateur family historians:** Often descendants of immigrants, these individuals lack the linguistic and palaeographic skills to interpret archaic Cyrillic handwriting. Their need for direct, intelligible transcriptions is underscored by the scale of interest within genealogy communities. For example, the largest Facebook group “Tracing The Tribe” [2] includes approximately 80,800 members actively engaged in exploring their roots.
- **Professional genealogists and historians:** Scholars and commissioned researchers require high-fidelity tools that can accurately capture the subtle orthographic variations and handwriting inconsistencies found in archival materials. Such precision is essential for producing reliable, contextually nuanced historical analyses.
- **Community organisations and cultural preservationists:** Jewish genealogical societies, archivists, and museum professionals seek scalable solutions to digitise, preserve, and make accessible fragile documents. Their goal is to create searchable digital archives that maintain cultural heritage for future generations. The International Association of Jewish Genealogical Societies [3] lists around 84 Jewish genealogy societies worldwide, each addressing the distinctive needs of its local or thematic community.

Project concept

KETUVIM (Hebrew: "Writings" or "Scriptures") enters a field where the research community has already made significant strides in transcribing historical Cyrillic texts. Machine learning-based models for Handwritten Text Recognition (HTR) have already produced encouraging outcomes across various historical Cyrillic-script contexts. For example, Achim Rabus [4] reported character error rates as low as 3–4% when applying such models to 19th-century Church Slavonic manuscripts, demonstrating their capacity to handle complex orthographic conventions, including superscript characters and abbreviations. Another study tackled Romanian Cyrillic sources ranging from the 16th century onwards [Rabus 2019], successfully developing both “generic” and “smart” models capable of transliterating Cyrillic into Latin scripts—evidence of these approaches’ flexibility and applicability to diverse periods and linguistic environments.

Building on existing progress, KETUVIM aims to adapt and refine machine learning methods to meet the distinctive challenges posed by late 19th-century Jewish communal records. These sources, recorded in Cyrillic script, often display distinct letterforms, archaic orthographic conventions, and a variety of layout complexities, representing a domain that remains largely underexplored. Although improvements have been made in reducing character error rates and enhancing accuracy, further work is required to develop models capable of generalising across diverse

handwriting styles, incorporating semantic nuances, and maintaining methodological transparency.

To address these aims, KETUVIM's pipeline incorporates several key strategies:

- **Leveraging transfer learning and pre-trained models:** Employing neural networks initially trained on analogous Cyrillic corpora, then fine-tuning them for the orthographic and linguistic complexities inherent in late 19th-century Jewish communal texts.
- **Refined data handling and augmentation:** Implementing targeted preprocessing and augmentation techniques to accommodate ink fading, irregular letterforms, specialised Jewish vocabularies, and complex page layouts, ensuring that training data more accurately reflects real archival conditions.
- **Comprehensive evaluation metrics:** Moving beyond basic accuracy by including word error rates, coverage metrics, and qualitative assessments. This broader evaluative framework ensures that performance measures align more closely with scholarly, genealogical, and historical research objectives.

By employing these integrated approaches, KETUVIM aspires to produce more accurate and contextually sensitive transcriptions, ultimately enriching genealogical investigations and contributing to a deeper understanding of historical Jewish communities.

Project motivation

The motivation behind KETUVIM is both culturally significant and timely. Genealogy research has recently experienced a remarkable surge in global interest. Since the introduction of direct-to-consumer genetic ancestry tests in 2000, more than 40 million individuals have undergone DNA testing [Ferguson 2024], reflecting a growing desire for tools that facilitate the discovery of ancestral roots. This trend is mirrored in the genealogy products and services sector, which reached a value of US$3.5 billion in 2022 and is projected to expand at a rate of 11.2% annually [Polaris Market Research 2023].

The growing interest in uncovering historical records coincides with expanding applications of handwriting recognition (HWR) technologies. Valued at over US$1,340 million in 2020, the global HWR market is forecast to reach approximately US$4,291 million by 2027, with legal documents and cultural heritage materials emerging as key areas of importance [Credence Research 2021]. Developments in artificial intelligence (AI) and machine learning (ML) have significantly enhanced HWR systems, allowing them to learn from user feedback, adapt to various

handwriting styles, and process increasingly complex inputs [Growth Market Reports 2023].

Project template

The project is based on the “Gather Your Own Dataset” template from the Machine Learning and Neural Networks module. The aim of KETUVIM is to collect and annotate historical Jewish records from the Russian Empire, leverage pre-trained models for feature extraction, and iteratively refine both the dataset and model. This mirrors the template’s guidelines, ensuring a systematic path from data gathering to improved transcription accuracy.

Literature review

The field of historical document transcription has garnered significant academic and industry interest due to its potential for preserving cultural heritage and facilitating genealogical research. Advances in machine learning, particularly in handwriting recognition and natural language processing, have enabled more accurate and scalable transcription solutions. This literature review explores existing methods, technologies, and frameworks applied to transcribing historical records, with a focus on identifying best practices and uncovering research gaps relevant to the development of an ML-driven transcription system for historical Jewish records from the Russian Empire.

Methodology

The methodology for the literature review involved systematically identifying and analysing relevant academic, charitable, and commercial projects addressing challenges in transcribing historical records. The process began with foundational papers provided within the project framework, serving as a primary reference point.

To expand the search, academic databases such as Google Scholar, Academia.edu, and research aggregators like Perplexity Search were employed. Searches were conducted using key terms related to historical handwriting recognition, document transcription, and genealogical record digitisation. Citation tracking from initial key papers helped uncover additional relevant works. Additionally, Transkribus, a widely recognised platform for historical document transcription, was reviewed for its methodologies, evaluation techniques, and reported results.

Priority was given to peer-reviewed journal articles, reputable conference proceedings, and well-regarded academic theses to ensure research rigour and reliability.

Analysis

Rabus' research [4] was published in *Scripta & e-Scripta*, a peer-reviewed journal from the Bulgarian Academy of Sciences noted for its focus on Slavic linguistics and medieval manuscripts. By placing Church Slavonic handwriting into a state-of-the-art machine learning framework, Rabus provides a credible, rigorously vetted baseline for recognising complex historical scripts.

His study underscores the complexities of working with historical Slavic manuscripts due to their diverse handwriting styles, inconsistent orthography, and the use of scriptura continua, where words are written without spaces. By training models using the HTR+ engine of Transkribus, Rabus achieved Character Error Rates (CER) as low as 3-4% on test sets, demonstrating the potential for automated transcription of historically challenging scripts.

The paper offers several noteworthy contributions to the field of historical text recognition. First, it validates the effectiveness of transfer learning for historical handwriting recognition, demonstrating that models trained on a wide range of data sources can achieve high accuracy. This insight underscores the importance of diverse training datasets in developing robust recognition models. Second, Rabus discusses the challenge of overfitting, a frequent issue in machine learning, where models trained for too many epochs perform poorly on unseen data. He addresses this concern by developing combined models trained on multiple sources, resulting in models that generalize better across various scripts.

However, Rabus' work is not without limitations. The study focuses exclusively on Church Slavonic manuscripts, leaving unexplored the applicability of its methods to other historical contexts, such as Jewish records written in Cyrillic. Additionally, the scope of the research is limited to transcriptions using Transkribus, which, while powerful, may not encompass other advanced neural network frameworks or custom models. The reliance on CER as the primary evaluation metric, while common, could have been supplemented by additional performance indicators such as Word Error Rate (WER) or language model integration. Lastly, the study's emphasis on formal, structured texts with established paleographic standards limits its relevance to less formal or mixed-language documents that might present more varied challenges.

Arkhipov et al. [9] also published their findings in *Scripta & e-Scripta*, but with a distinct emphasis on bilingual Evenki–Russian manuscripts from the archive of Konstantin Rychkov. Using Transkribus and comparing its HTR+ engine with the PyLaia recognition engine, the researchers achieved a CER as low as 2.9%. Despite these promising results, they encountered numerous challenges related to layout analysis, recognition of diacritics, and complex bilingual data structures.

The authors note that separating the Evenki and Russian texts during layout analysis was particularly difficult due to the manuscripts' lack of clear column boundaries. Additionally, the automated system struggled with distinguishing similar-looking Cyrillic letters and dealing with complex diacritics such as stress marks and palatalization symbols. The comparison of recognition models revealed that HTR+ performed better with standard letters when stress marks were excluded from the training data, while PyLaia excelled in recognizing diacritics and punctuation.

While the study demonstrates the feasibility of applying HTR to complex bilingual manuscripts, it also highlights significant limitations. The authors found that even after extensive training, the models produced many errors related to diacritics, punctuation, and complex letterforms. Moreover, the reliance on manually annotated ground truth data required substantial human effort, limiting scalability. The researchers conclude by advocating for more collaborative development of publicly available HTR models for Cyrillic-based documents.

Abdallah et al. [10] published their study in the peer-reviewed *Journal of Imaging*, a reputable venue specialising in imaging and OCR advancements, thus underscoring the significance of their approach within the broader field of Cyrillic handwriting recognition. The article presents a novel handwriting recognition model based on an attention-gated convolutional neural network (CNN) and bidirectional gated recurrent unit (BGRU). Designed for Kazakh and Russian handwritten text recognition, this model achieved a CER of 4.5% and WER of 19.2% on test datasets.

Their system integrates multiple components: a fully gated CNN for feature extraction, a BGRU for sequence prediction, and an attention mechanism to enhance context-based decoding. The researchers demonstrate how this architecture outperformed other state-of-the-art models such as Bluche and Puigcerver in recognizing Cyrillic handwritten text.

Despite its effectiveness, the study reveals several technical challenges. The authors faced difficulties in recognizing visually similar Cyrillic characters, handling complex handwriting styles, and addressing overfitting. The absence of pre-trained models for Cyrillic-based handwriting forced them to train models from scratch using the newly introduced HKR dataset. Their findings suggest that improved data augmentation, broader datasets, and multi-language training could further boost the model's performance.

Kalken and Jantayev [11] published their comprehensive survey in the peer-reviewed *Proceedings of IYSC (2022 International Young Scholars' Conference)*, making it a credible and frequently cited resource on neural network architectures for Cyrillic OCR.

The paper categorizes existing OCR methods into convolutional, recurrent, and attention-based neural networks. It discusses various datasets, including the Handwritten Kazakh and Russian (HKR) dataset, which contains 64,943 samples of Kazakh and Russian handwritten text, achieving a CER as low as 4.13% using the Attention-Gated-CNN-BGRU model. Additionally, the Kazakh Offline Handwritten Text Dataset (KOHTD), with 140,335 samples, yielded a CER of 6.52% using the Flor model, indicating strong OCR potential for complex scripts.

A critical limitation in the reviewed works is the over-reliance on CNN architectures without integrating newer models like Transformers or hybrid CNN-RNN (Recurrent Neural Network) architectures. Although the authors acknowledge CRNNs' effectiveness in Cyrillic OCR, their reliance on models such as Bidirectional Long-Term Short-Term Memory (BiLSTM) and multidimensional LSTMs (MDLSTMs) exposes computational inefficiencies. For example, while MDLSTMs offer robust context capture, their high computational cost often limits real-world applications.

Data diversity is another major concern. Many studies used databases collected from limited sources, such as university exams (e.g., Toiganbaeva et al. [12], with 140,000 images) or mathematical Olympiads (Liepieshov et al. [13], with 82,061 samples). This limited representation could reduce OCR effectiveness when applied to broader contexts, such as historical records.

Evaluation metrics were inconsistently reported across studies. While CER and WER were commonly used, other important metrics such as Specificity and Sequence Error Rate (SER) appeared less frequently. Abdallah et al. [10] achieved the lowest CER at 4.13% using HKR with an attention-based model, outperforming models like Bluche, Puigcerver, and Flor.

Cristea et al. [14] published their work in the *Memoirs of the Scientific Sections of the Romanian Academy*, a respected academic series that rigorously evaluates AI-driven contributions to cultural heritage, underscoring the credibility of their large-scale, methodologically rigorous pipeline. The study is centred on old Cyrillic Romanian materials, presenting an insightful application of AI for historical document transcription. However, several methodological concerns arise. The pipeline design, though logically modular, operates in a linear fashion that limits adaptability. A more dynamic system would allow feedback from later stages to adjust earlier processes. Additionally, data annotation relied heavily on manual work, with 202,042 graphical objects annotated, creating a clear bottleneck. Crowdsourcing or semi-supervised learning could alleviate this issue. Preprocessing techniques were similarly constrained by older methods like Gaussian filtering and the Hough Transform, lacking integration with modern deep learning approaches such as RotNet or document-specific CNNs.

Model limitations are also evident. The use of YOLO (You Only Look Once) and CNNs without Transformer-based models like LayoutLM or Document AI limits scalability. Evaluation metrics reported character classification precision peaking at 95.65% with grayscale preprocessing, but recall and F1 scores were absent, raising concerns about completeness. The system's language model was restricted to Romanian dictionaries, limiting multilingual applications. Word segmentation algorithms struggled with compound words such as "cineva" being split into "cine" + "va," which could be resolved using contextual models like BERT.

Despite these concerns, valuable techniques from the paper can be adapted. For preprocessing, document deskewing methods could be enhanced using modern tools like RotNet, and noise reduction could leverage DnCNN (Denoising Convolutional Neural Network) models. Character recognition could combine YOLO for initial detection and Transformers for sequence recognition, while transfer learning from models like Tesseract or Kraken could be refined for historical scripts.

Pinche and Stokes [15] published in the peer-reviewed, open-access *Journal of Data Mining and Digital Humanities*, widely read by interdisciplinary teams for bridging humanities research with computational methods, ensuring the paper's broad relevance to historical text analysis and Automatic Text Recognition (ATR) solutions. They surveyed key progressions in ATR for historical documents, while also highlighting areas needing further exploration. The paper offers a detailed timeline of ATR's development from the Reading Optophone (1914) to modern CNN and RNN models, showcasing technological progression. Platforms like eScriptorium, Kraken, and Transkribus are well described, with specific features highlighted. For example, Kraken supports right-to-left scripts such as Arabic and Hebrew, while eScriptorium provides an intuitive GUI. Community-driven projects such as HTR-United and SegmOnto stand out, emphasizing collaborative data sharing and model improvement through centralized repositories.

However, some limitations persist. While metrics like CER and WER are frequently mentioned, detailed evaluation results for specific datasets are sparse. For example, CER metrics in various models range from 2% to 10%, but a breakdown by language or historical script is lacking. Although HTR-United centralizes data, the paper emphasises the need for diverse historical datasets. For instance, CREMMALab focuses on Western European manuscripts but could expand into underrepresented languages like Old Church Slavonic or Hebrew. Additionally, descriptions of technical innovations such as adaptive thresholding, attention-based models, and CNN-BGRU architectures are too brief, limiting deeper insight into what makes these approaches effective.

A concise comparative overview of these works, including key methods, reported error rates, and identified gaps, is provided in Table 1.

Reference	Focus / Script(s)	Approach / Model(s)	Performance Metrics & Key Findings	Limitations / Observations
Rabus (2019)	Church Slavonic manuscripts	Transkribus HTR+ (Neural Networks + Transfer Learning)	CER: as low as 3–4%. Demonstrates effectiveness of combined models	Limited to Church Slavonic. Relies heavily on CER. Potentially overfits formal scripts
Arkhipov et al. (2021)	Bilingual Evenki–Russian	Transkribus HTR+ vs. PyLaia	CER: as low as 2.9%. PyLaia excels at diacritics; HTR+ better for standard letters	Complex bilingual data structure. Manual ground-truth annotation is labour-intensive
Abdallah et al. (2020)	Kazakh and Russian handwriting	Attention-gated CNN + BGRU	CER: 4.5%, WER: 19.2%. Outperforms Bluche and Puigcerver models	Overfitting challenges. Lacks pre-trained Cyrillic resources. Needs broader datasets
Kalken & Jantayev (2022)	General Cyrillic OCR	Survey of convolutional, recurrent, and attention-based OCR	Summarises CER ranging ~2% to 10%. Showcases HKR data with ~4.13% CER (best)	Strong coverage but lacks deep focus on Transformers. Inconsistent reporting of WER/SER
Cristea et al. (2023)	Old Cyrillic Romanian documents	YOLO + CNN pipeline; linear design	Character precision up to 95.65% (grayscale preprocessing). Modular but linear system	Reliance on manual annotation (202,042 objects). No advanced Transformers or BERT used
Pinche & Stokes (2023)	Historical docs (multi-script)	ATR overview: eScriptorium, Kraken, Transkribus	Highlights advancements from early Optophone to modern	Limited dataset-specific outcomes. Encourages more

			CNN/RNN. CER ~2–10% in examples	multilingual data, lacks detail on novel architectures

Table 1: Comparative overview of key studies

Findings

Research on handwritten text recognition has primarily concentrated on Church Slavonic, Russian, and Kazakh scripts, with little attention given to Jewish historical records in Cyrillic. This highlights a significant gap, as no models have been specifically trained on such records. Additionally, much of the prior work has relied on conventional models like CNNs and RNNs, with only limited exploration of modern architectures such as Transformers or hybrid CNN-RNN-Transformer approaches, limiting potential performance improvements. Though there are some good foundational models for Cyrillic handwriting that could be taken as a basis and then expanded.

Many datasets used in these studies are small, highly specialized, and lack multilingual or mixed-script content, which reduces model generalization capabilities. Moreover, evaluation metrics are often limited to Character Error Rate (CER) and Word Error Rate (WER), with less emphasis on alternatives like Named Entity Recognition (NER) accuracy or Sequence Error Rate (SER). Document processing pipelines are typically linear, offering minimal adaptive features or feedback mechanisms, restricting their ability to handle diverse document types.

Insights from the reviewed literature show that transfer learning improves recognition accuracy when applied to diverse datasets. Attention-based models have proven effective for visually complex and irregular scripts. Comprehensive data augmentation strategies mitigate overfitting and enhance model robustness across varying handwriting styles. Training models with data from multiple sources strengthens their capacity to handle diverse handwriting styles and complex layouts, particularly in mixed-language settings. Combining CNNs for feature extraction with RNNs for sequence modeling has produced strong results. Platforms like Transkribus can expedite early evaluations by providing foundational model-building tools.

Opportunities

KETUVIM has several promising opportunities based on these insights. Developing a specialized dataset of Jewish records in Cyrillic through partnerships with archives and leveraging crowdsourcing for annotations could address data scarcity.

Semi-supervised learning could further reduce manual annotation efforts. Implementing a hybrid model combining CNNs, RNNs, and Transformers could enhance recognition accuracy. Exploring models optimized for low-resource historical scripts would further boost system performance.

Modern preprocessing tools like RotNet for document deskewing and DnCNN for noise reduction could be integrated, while adaptive layout analysis could manage mixed-language document formats. Expanding evaluation metrics to include CER, WER, SER, NER accuracy, and contextual language model assessments would enable a more comprehensive performance review. Designing a feedback-driven, adaptive pipeline with real-time user corrections could improve both initial and long-term system performance. Active learning techniques could support continuous model refinement.

Lastly, KETUVIM could combine open-source HTR tools like Kraken or Transkribus with custom-built models, fostering flexibility and performance gains. Collaboration with historical data repositories would ensure broader data coverage and model validation. By addressing these gaps while applying key learnings, KETUVIM can fill a crucial void in the field of historical document recognition, enabling automated transcription of Jewish records in Cyrillic and advancing genealogical research.

Design

Project overview

Successful project design requires balancing technological capabilities with user requirements while addressing domain-specific challenges. The following sections describe the project's domain, present real-world evidence of user needs, and outline the key design decisions that inform the selected features.

Domain

As discussed in the Introduction, the domain of this project exists at the intersection of historical document transcription, genealogical research, and cultural heritage preservation. The following points outline essential contextual factors that must be taken into consideration when designing the project:

- **Complex Orthographic Conventions**
  Historical Cyrillic documents often contain archaic letterforms and inconsistent spelling rules. In a preliminary analysis of 50 sample pages from the beginning of the 20th century, over 30 distinct letter variants were observed for commonly used characters. Consequently, the project requires advanced

handwriting recognition techniques that can accommodate significant variability.

- **Irregular Handwriting Styles**
  Even within a single registry book, multiple scribes may have recorded entries with differing levels of legibility. Informal feedback from genealogists forums and family history groups suggests that inconsistent handwriting is widely perceived as one of the main obstacles when interpreting archival birth and marriage records.
- **Inconsistent Administrative Formats**
  Archives from different regions (e.g., Lublin vs. Białystok) followed diverse templates. Layout analysis reveals frequent mismatches between expected column structures and actual record layouts, underscoring the need for adaptive and flexible processing methods.
- **Ink Fading and Document Degradation**
  The upheavals of the 20th century, including wars, revolutions, and shifting borders, led to widespread displacement and poor storage of archival materials. These disruptions often caused ink fading and physical degradation, significantly affecting legibility and OCR performance. To address this, preprocessing steps such as noise reduction, contrast enhancement, and deskewing are critical components of the pipeline.
- **Mixed-Language Records**
  Numerous Jewish registries in the Russian Empire contain a mix of Cyrillic for official entries and Hebrew or Yiddish for marginal notes. Observations from family history forums and online communities indicate a common need for partial translation or at least detection of non-Cyrillic scripts, even though the primary focus of the transcription is Cyrillic.

Users

In the Introduction, key user groups were identified—amateur family historians, professional genealogists and historians, and community organizations and cultural preservationists—each with distinct needs influencing our project's design. Table 2 provides a summary of these user types and their specific requirements, supported by relevant evidence:

User type	**User needs**	**Evidence**
Amateur family historians	Obtain clear and accurate transcriptions of historical	Amateur researchers often face language barriers and rely on transcriptions and

	records without needing Cyrillic or Russian expertise	translations to interpret documents [16]
	Access searchable transcriptions for easier genealogical research	Searchable databases and transcriptions reduce the complexity of navigating vast historical records [17]
Professional genealogists and historians	Perform detailed archival research with accurate, context-aware transcriptions	Professionals depend on context-rich transcriptions to interpret historical records accurately and avoid misinterpretation [18]
	Resolve ambiguous historical entries with reliable transcriptions	Reliable transcriptions help resolve unclear or damaged entries in original documents, which is essential for high-standard historical research [19]
Community organisations and cultural preservationists	Create accurate transcriptions to support digital archiving and preservation efforts	Transcription is a foundational step in document preservation and digital heritage projects [20]
	Generate machine-readable records for easier integration into public archives	Crowdsourced and machine-readable transcription projects support open access and archival integration [21]

Table 2: Users and their needs

Justification of features

Based on the project's domain and identified user needs, the following features have been designed to address key transcription challenges specific to historical Jewish records in Cyrillic:

- **Advanced handwritten text recognition (HTR) model:** The system will address irregular handwriting styles and complex orthographic conventions found in historical records. A hybrid model combining convolutional neural networks (CNNs), recurrent neural networks (RNNs), and Transformers will process visual, sequential, and contextual aspects of handwritten texts. Rabus demonstrated the effectiveness of such hybrid models for historical Cyrillic manuscripts, achieving low character error rates [4].
- **Specialised preprocessing pipeline:** Enhancing text readability is critical due to document aging, ink fading, and paper degradation. Preprocessing steps such as noise reduction, contrast enhancement, and document deskewing will improve model accuracy. Arkhipov et al. [2021] underscored the importance of preprocessing in achieving high OCR accuracy in Evenki-Russian bilingual manuscripts .
- **Language and script filtering:** Since mixed-language records are common, the system will detect and ignore non-Cyrillic scripts while focusing exclusively on relevant content, reducing transcription errors.
- **Data augmentation and transfer learning:** Given the scarcity of annotated historical records, data augmentation techniques such as synthetic ink fading, handwriting variation simulation, and transfer learning from existing Cyrillic models will enhance model performance. Abdallah et al. [2020] explored transfer learning’s impact on improving Cyrillic OCR, noting significant error reductions.
- **Context-aware error correction and post-processing:** Ensuring contextual accuracy is essential for transcription reliability. Named Entity Recognition (NER) and language models will detect transcription errors and resolve contextual inconsistencies. Cristea et al. [2023] demonstrated the potential of contextual models in improving the linguistic coherence of Old Cyrillic texts .
- **Evaluation metrics beyond accuracy:** Comprehensive performance measurement will include Character Error Rate (CER), Word Error Rate (WER), Sequence Error Rate (SER), and NER-based semantic accuracy. Kalken and Jantayev [2022] advocated for expanded OCR metrics to better assess real-world transcription performance .
- **User feedback and active learning loop:** Long-term improvement will be achieved through an active learning system that integrates user feedback. Genealogists and historians will provide corrections, continuously refining the model through iterative retraining.
- **User-friendly interface for document management:** The system will feature a web-based interface allowing users to upload historical records, process

transcriptions, and store results in a searchable, structured database. This interface will simplify genealogical and archival research, enhancing accessibility and user engagement.

Project structure

The KETUVIM project follows a modular structure designed to address the unique challenges of transcribing historical Jewish records in Cyrillic using advanced AI-based handwriting recognition (HTR). The project is divided into five key modules:

- Data collection;
- Preprocessing and enhancement;
- Model development and training;
- Post-processing and error correction;
- User interface and feedback integration.

Data collection

Data collection is foundational, ensuring diverse and representative samples of historical records. The project sources birth, death, marriage, and other civil records from the national archives of regions that once formed part of the Russian Empire and are now located in various Eastern European countries. Metadata such as dates, record types, and archival references are documented to support contextual understanding and downstream tasks like search and retrieval.

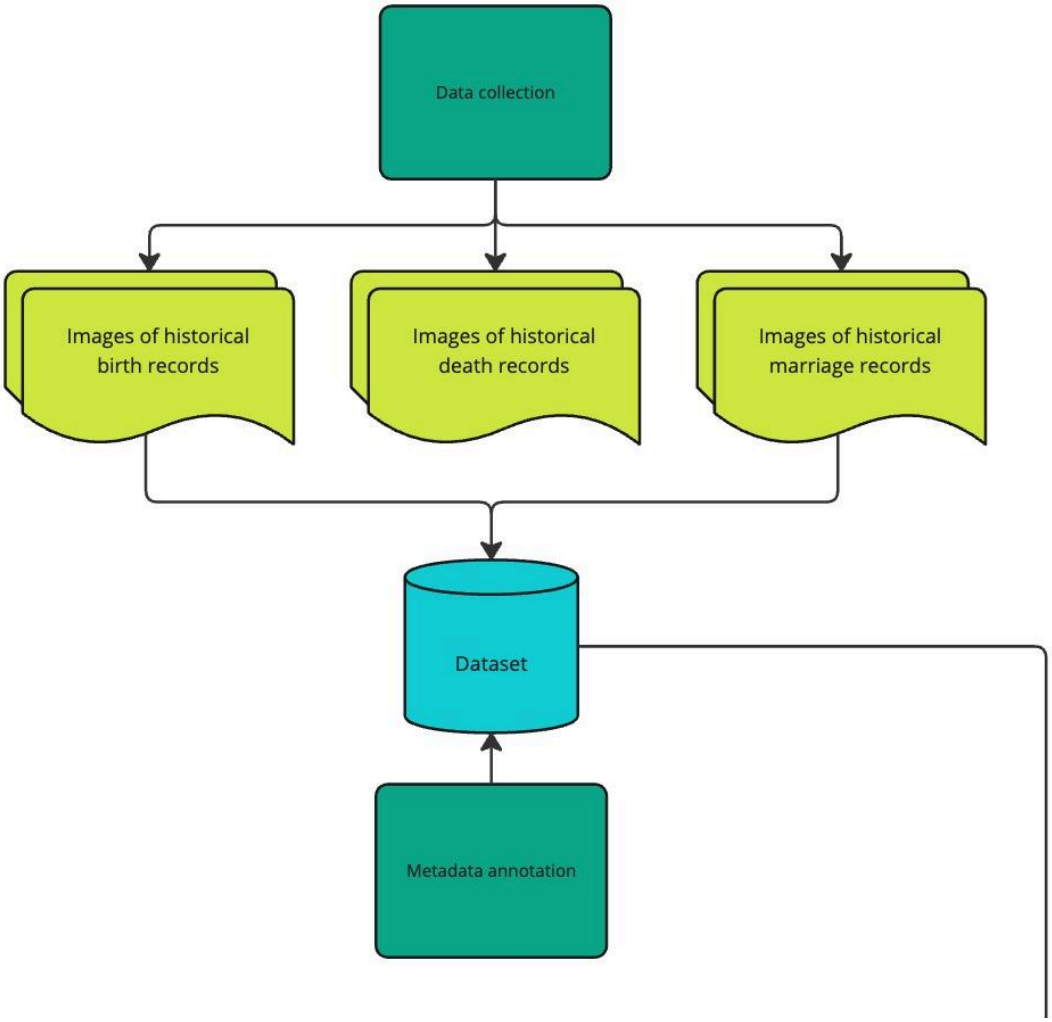


Figure 1: Data collection overview

Preprocessing and enhancement

Enhancing text readability is essential due to document aging, ink fading, and paper degradation. The preprocessing pipeline will include the following key steps:

- **Data cleaning:** An automated script is used to remove duplicate and low-quality images. Duplicate detection relies on hashing each image, while low-quality scans will be filtered based on grayscale variance to exclude those with insufficient contrast.
- **Image standardisation:** All images are converted to grayscale to ensure uniformity and simplify downstream processing.

- **Directory traversal and organisation:** The script recursively processes input folders, saving cleaned images into a mirrored directory structure to preserve archival provenance.

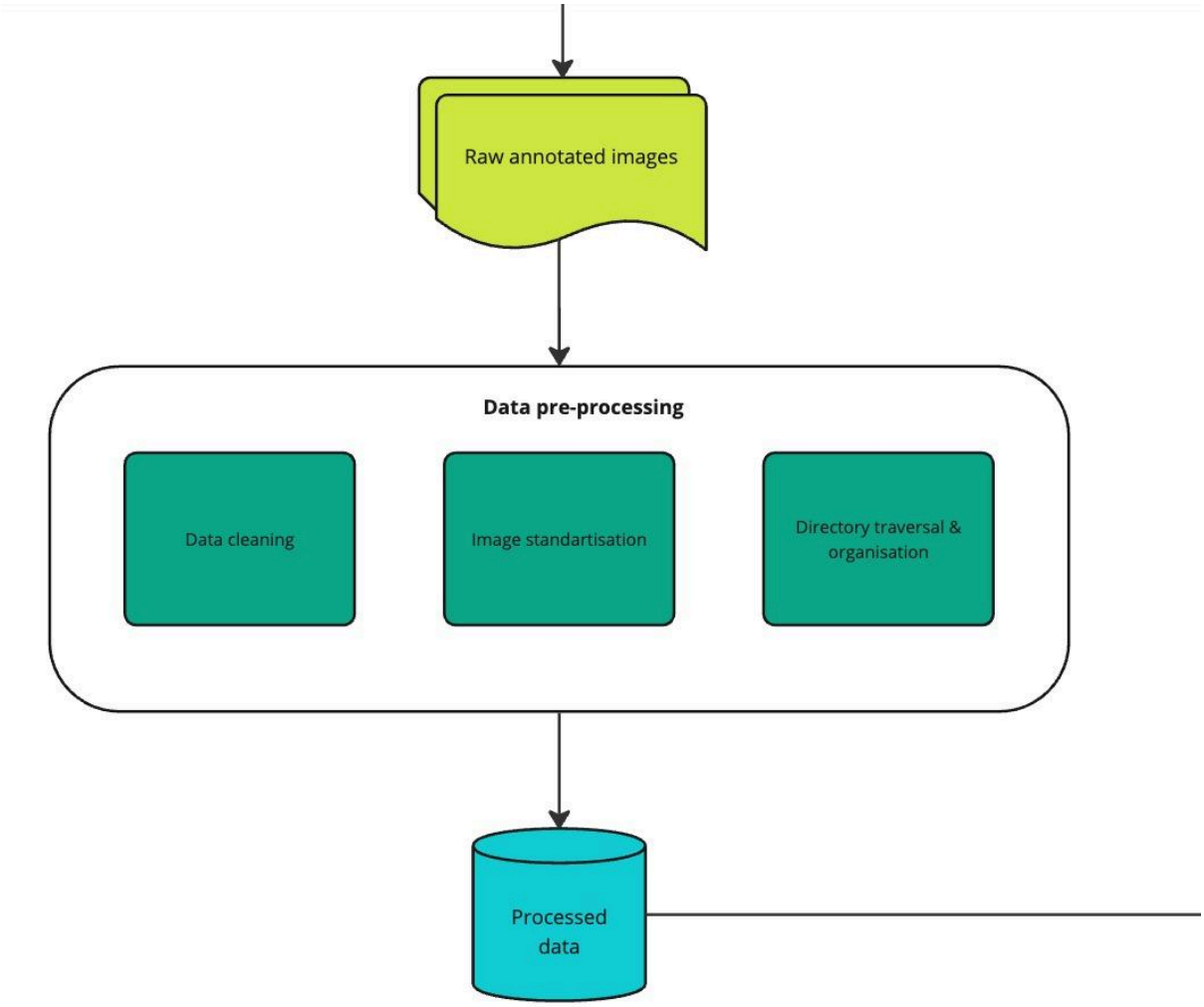


Figure 2: Preprocessing and enhancement workflow

Model development and training pipeline

The system will follow a three-model architecture, with each component dedicated to a specific stage in the transcription pipeline: **segmentation**, **recognition**, and **post-processing**.

- **Segmentation model:**
  The first stage involves fine-tuning a pre-trained model for identifying handwritten text regions within scanned archival pages. This model follows a Convolutional Recurrent Neural Network (CRNN) architecture, combining convolutional layers for spatial feature extraction with bidirectional recurrent layers to model sequence-like dependencies in the page layout. Fine-tuning is performed using manually corrected ground truth data from historical Jewish civil records to adapt the model to domain-specific structures and handwriting variability.
- **Recognition model:**
  Once text lines are segmented, a custom recognition model is trained from scratch using the Kraken OCR engine. This model also employs a CRNN structure, incorporating Bidirectional LSTM layers and Connectionist Temporal Classification (CTC) loss, which allows it to align variable-length input images with transcriptions without the need for pre-segmented characters. This configuration is particularly well suited for transcribing irregular, handwritten Cyrillic text with inconsistent spacing and orthography.
- **Correction model:**
  The final model in the pipeline focuses on post-processing and contextual correction. It leverages Transformer-based language architectures to handle higher-level linguistic tasks, including sequence-level coherence, Named Entity Recognition (NER), and orthographic normalisation, refining the raw output of the recogniser into more accurate, semantically coherent transcriptions.

To optimise performance and ensure generalisability across document types and handwriting styles, hyperparameter tuning will be carried out for each model. This will include systematic adjustment of learning rates, batch sizes, training durations, and other architecture-specific parameters.

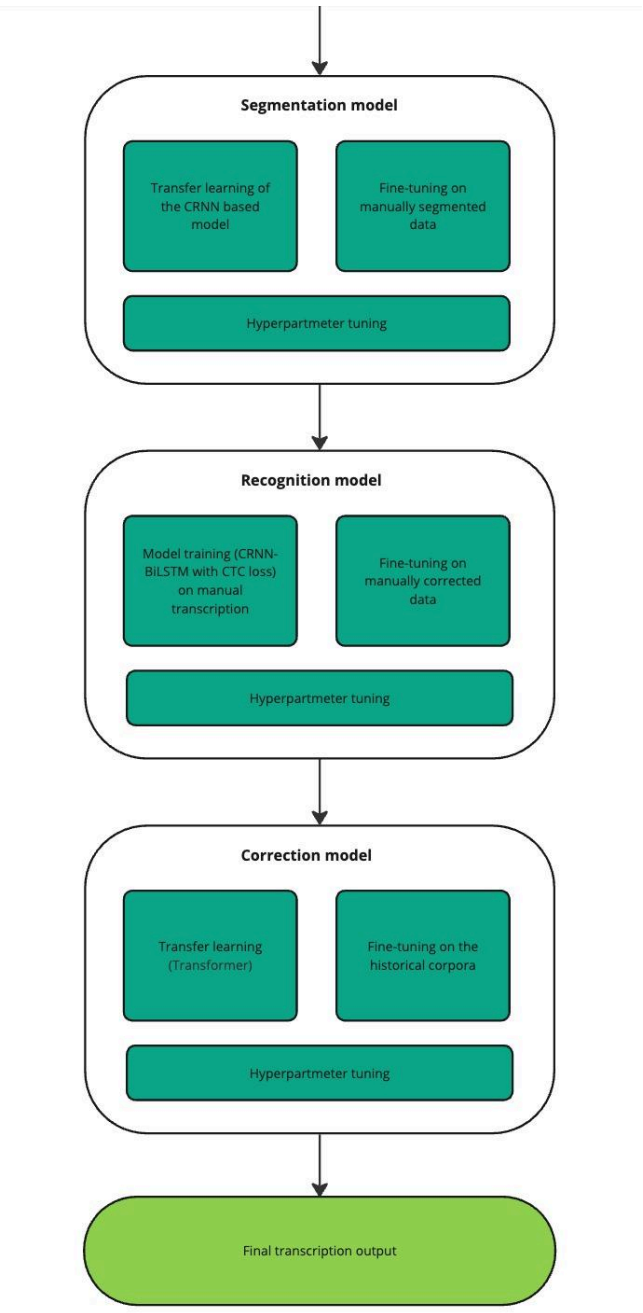


Figure 3: Model development and training pipeline

User interface

The project involves two distinct interfaces, each supporting a different stage of the transcription workflow. The **eScriptorium interface** is used for document segmentation and initial transcription. Users upload historical document images and can run two models:

- Segmentation model identifies and isolates handwritten text regions.
- Recognition model transcribes the segmented lines of text.

The output of this process—a raw transcription—can be then transferred to the second environment.

The **web-based interface** focuses on contextual refinement. It allows users to upload the same document image (for visual reference) and paste the corresponding transcribed text. Users can run a transformer-based model that refines the output through contextual correction and entity recognition.

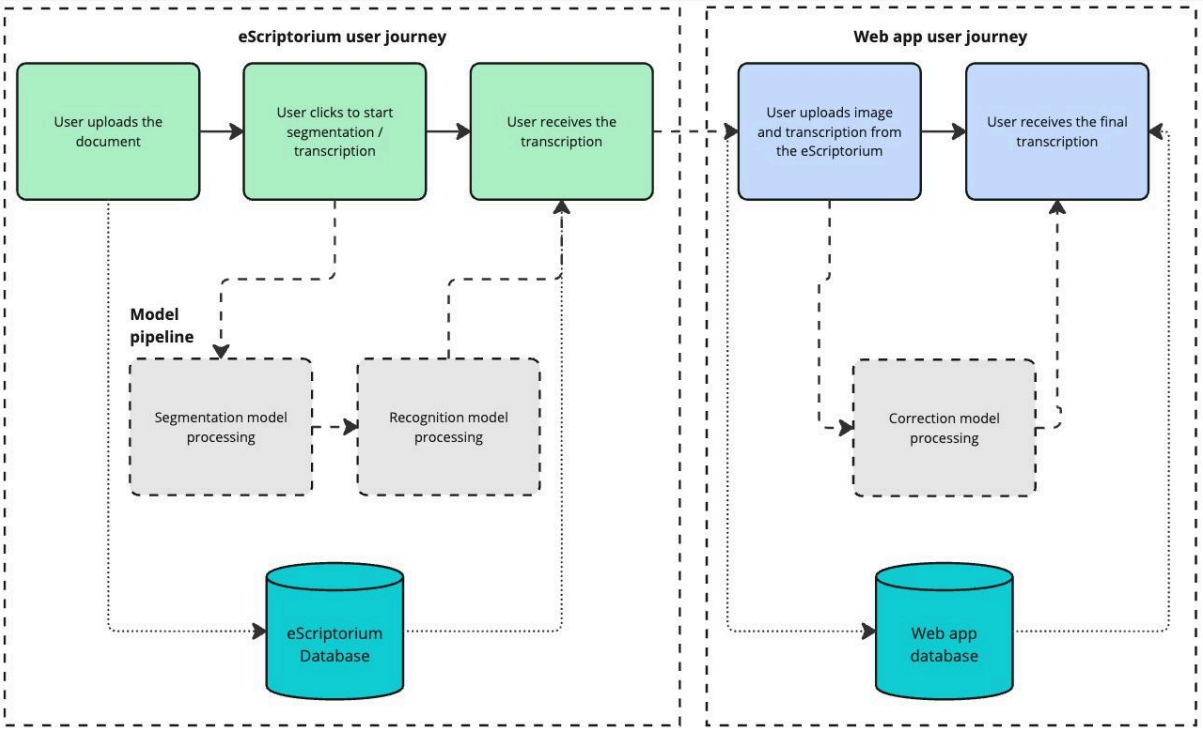


Figure 4: Ketuvim service blueprint

Technologies and methods

Based on the project's design and structure, the following technologies and methodologies will be applied to address the complexities of historical Cyrillic handwriting transcription effectively.

Technologies

The project uses **Python** for backend development, preprocessing scripts, and AI model integration. **eScriptorium** serves as the primary environment for managing training data, correcting segmentations, and helping to train both the segmentation and recognition models. Custom handwritten text recognition is handled via **Kraken OCR**.

The web application, built with **Flask**, integrates the **Transformers** model. It provides a lightweight interface developed in **HTML**, **CSS**, and **JavaScript**, enabling users to upload documents through a drag-and-drop workflow and retrieve corrected results. All correction history is stored in a local **SQLite** database for future reference and potential model improvement.

Supporting libraries include **OpenCV** for image preprocessing tasks such as grayscale conversion, duplicate detection, and low-quality filtering; **NumPy** and **Pandas** for efficient data handling.

Methodologies

The project adopts a modular architecture comprising four key parts: preprocessing, segmentation, recognition, and correction:

1. Preprocessing is handled with routines for grayscale conversion, duplicate removal, and low-quality image filtering to ensure clean training inputs
2. A pre-trained CRNN-based model is fine-tuned via transfer learning to detect handwritten text regions, while a custom HTR model is trained from scratch using Kraken on manually transcribed Cyrillic data.
3. The recognition model follows a CRNN architecture with Connectionist Temporal Classification (CTC) loss, which enables accurate transcription of unsegmented handwritten sequences.
4. A Transformer-based language model is fine-tuned for contextual correction and Named Entity Recognition (NER), enhancing the semantic quality of the final output.

Project plan

The project timeline, including tasks, durations, and deadlines, is presented in **Figure 5**.

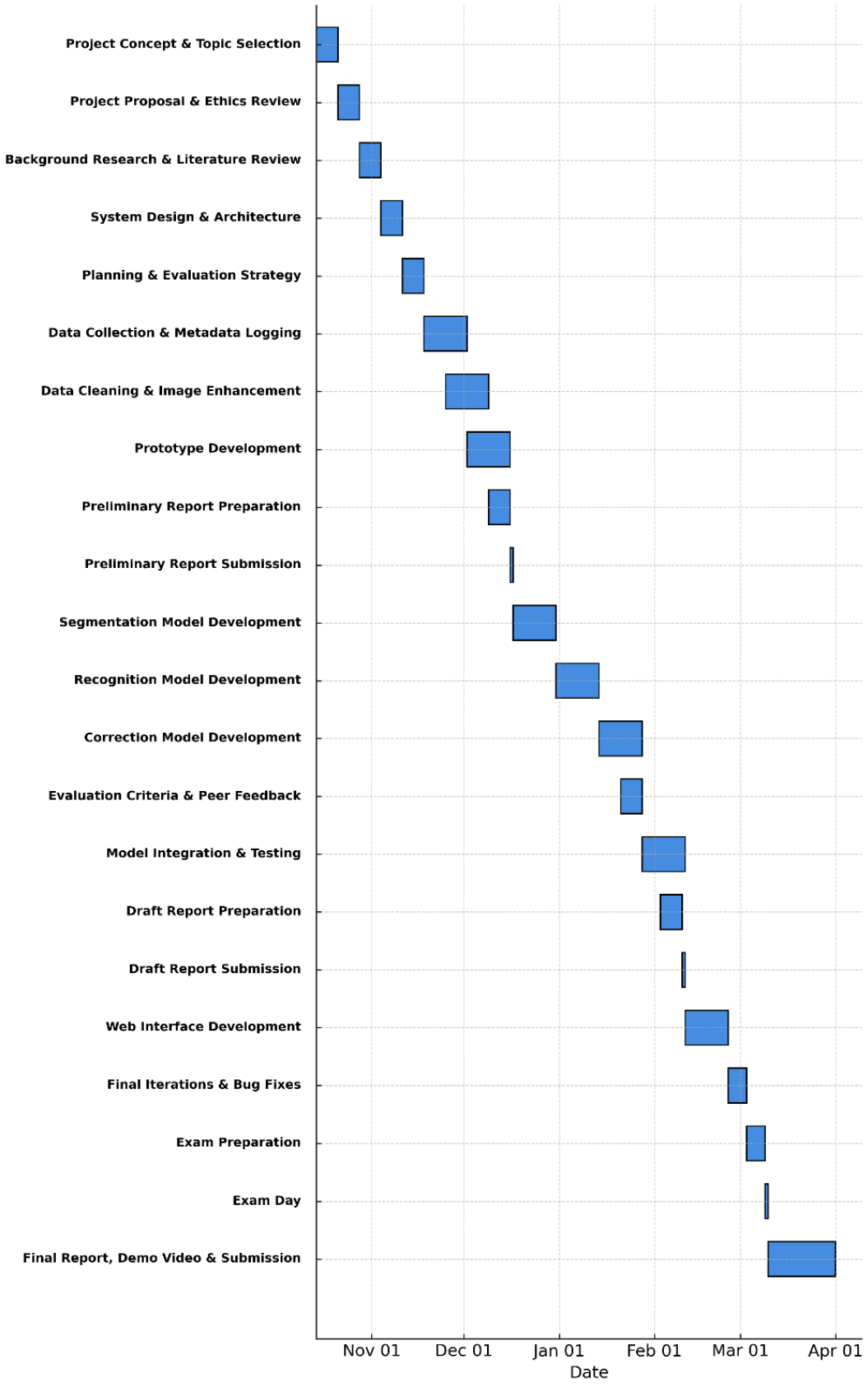


Figure 5: Project timeline

Testing and evaluation

The KETUVIM project uses a comprehensive strategy for testing and evaluation. Each model within the pipeline is assessed independently using a held-out validation dataset, compiled from manually annotated archival pages that reflect a broad spectrum of handwriting styles, document layouts, and scan qualities. The dataset is stratified to include both typical use cases and edge conditions, such as multilingual records, damaged pages, and faded ink.

Beyond model-level validation, user testing is conducted with target audiences to evaluate real-world applicability. Participants interact with the web interface to upload documents, review automated transcriptions, and submit corrections.

Model-specific evaluation employs tailored metrics: the segmentation model is assessed via line detection accuracy, calculated as the proportion of correctly identified baselines relative to ground truth. The recognition model is evaluated using Character Error Rate (CER) and Word Error Rate (WER). The correction model is tested using Sequence Error Rate (SER) Finally, usability is formally assessed through the System Usability Scale (SUS), completed during structured user testing sessions.

Component	Metric
Segmentation model	Line Detection Accuracy
Recognition model	Character Accuracy
	Character Error Rate (CER)
	Word Error Rate (WER)
Correction model	Sequence Error Rate (SER)
Full system	System Usability Scale (SUS) score

Table 3: Evaluation strategy

Implementation

The implementation of the prototype for recognising historical Cyrillic handwriting is structured into several key stages. First, I describe the collection and preparation of our archival data, detailing how we curated a diverse dataset from Polish state archives and standardised the images to ensure consistency. Following this, I outline the segmentation process, where an initial automatic segmentation model was employed and subsequently refined through manual corrections to establish reliable ground truth. Finally, I discuss the training of our recognition model, which was developed using a robust neural network architecture tailored for handwritten text. The subsequent sections elaborate on each of these stages, detailing the methodologies, algorithms, and tools that were employed to achieve our objectives.

Data collection

For the project, I have assembled a diverse dataset of historical civil records from various Polish state archives. The primary source is the website Szukaj w Archiwach [22], the official state collection of Polish archival records, which permits the reuse of its high-quality digital copies under specific conditions. These conditions require that any reuse must include proper attribution (archive name, fonds, reference code, scan link, and access time) and adherence to guidelines regarding sensitive materials.

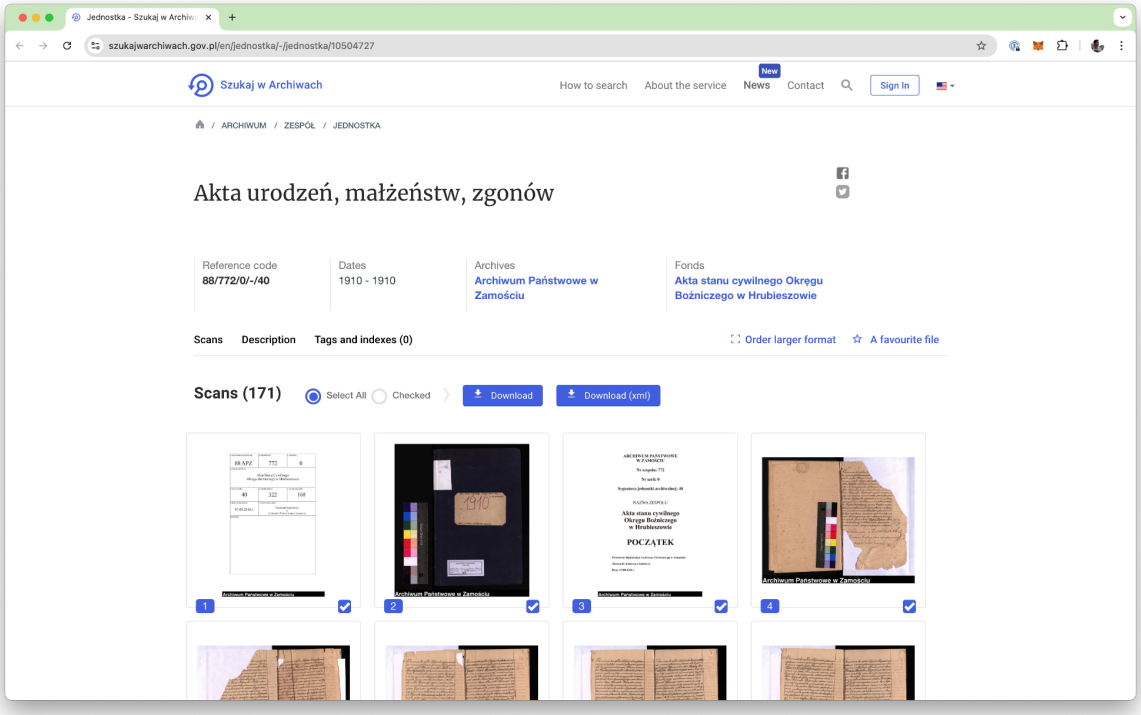


Figure 6: Szukaj w Archiwach (Search the Archives) website

I selected three key collections for my research:

1. **Białystok Death Records for 1900 [23]:** Examples of handwritten death records with a distinctive regional layout, illustrating typical handwriting and document structures of the period.

2. **Hrubieszów Civil Records (Births, Marriages, Deaths) for 1910 [24]:** A diverse collection covering births, marriages, and deaths, selected for its varied administrative layouts and multiple handwriting styles.

3. **Lublin Birth Records for 1890–1891 [25]:** Birth records offering additional insights into the document layouts and handwriting conventions specific to this area at the end of the 19th century.

By gathering records from multiple regions (Białystok, Hrubieszów, Lublin) and across different decades, we ensure that the training corpus encapsulates a variety of handwriting styles, page layouts, and document structures. This diversity is crucial for fine-tuning a robust HTR model capable of handling the idiosyncrasies of historical Cyrillic handwriting. It also helps to mitigate overfitting to a single style or layout, thereby improving the model's performance on unseen documents.

Data preparation

After assembling the raw dataset of archival scans from various Polish state archives, the next crucial step was data preparation. The raw scans included not only the historical handwritten records but also modern technical pages that the archive had incorporated for catalogue purposes. In order to create a robust training dataset focused solely on the historical documents, we first manually removed these modern technical pages. This ensured that the training data would be homogeneous in terms of content and layout, thereby minimising noise and potential biases during model training.

Next, I implemented an automated cleaning script using Python and OpenCV to systematically prepare the dataset for training the Handwritten Text Recognition (HTR) model. The script carries out three key functions:

- **Duplicate detection:** Identifies and removes duplicate images by computing a unique hash for each image file. Any duplicates, determined by matching hashes, are excluded from the dataset.

- **Quality filtering:** Filters out low-quality images by assessing the variance of their grayscale representation. Images exhibiting low variance typically suffer from fading or poor contrast—issues common in older or inadequately scanned documents—and are therefore discarded to maintain the quality of the dataset.

- **Recursive directory processing:** Traverses the raw data directory structure, converting suitable images into grayscale for simplified processing, and saves them to a new directory that mirrors the original folder hierarchy.

This approach ensures a consistent, standardised dataset, composed solely of high-quality images suitable for effective HTR model training. Figure 7 shows an example of the image after automated cleaning.

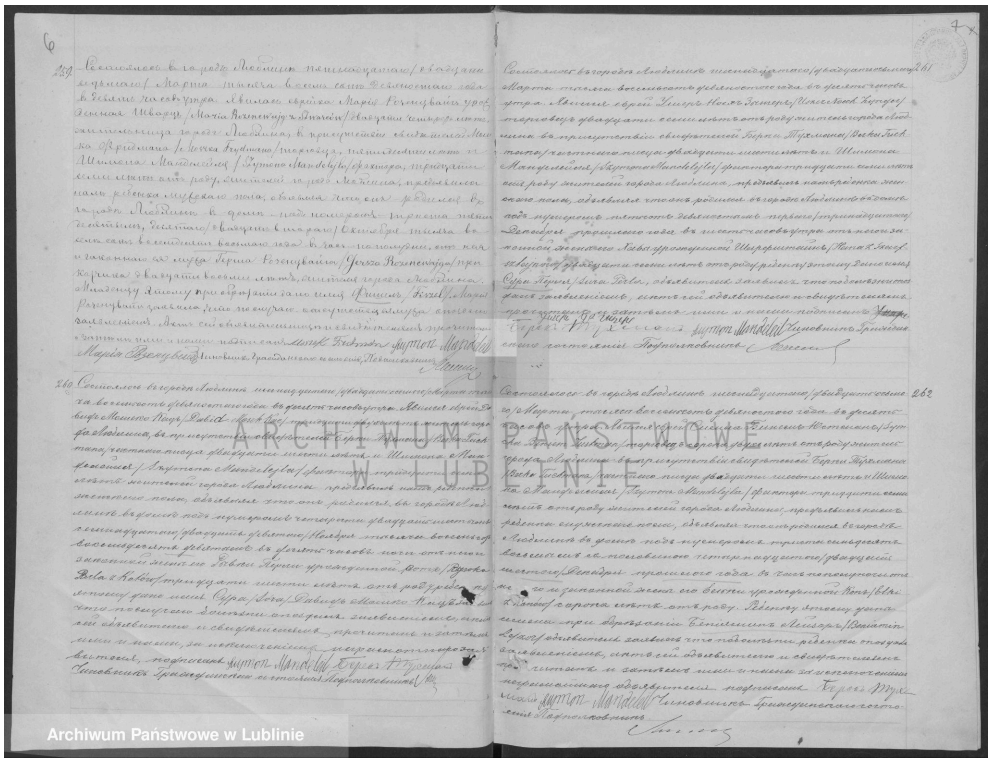


Figure 7: Example of the processed image

Local setup for model training

After extensive research into various options for Handwritten Text Recognition (HTR), including Transkribus and several other platforms, I decided to proceed with eScriptorium [26]. Its advantages are clear: it is open source and free to use, it supports loading, training, and downloading custom models, and it provides an intuitive user interface that accelerates the training process.

Initially, I attempted to deploy eScriptorium locally by cloning its repository from the public GitLab and utilising Docker Compose to build and run the containerised services. This approach involved configuring key components such as the web interface, database, and Celery workers to ensure seamless communication. However, I encountered significant challenges when training models within the Docker environment on Apple Silicon hardware, particularly due to compatibility issues with GPU acceleration. These obstacles made it impractical to proceed with the Docker-based setup.

Consequently, I opted for a full local installation of eScriptorium [27], following the comprehensive instructions provided in the project's documentation. This method allowed me to bypass the limitations associated with Docker on Apple Silicon. I optimised the training process by configuring Kraken parameters in the .env file. These adjustments included setting KRAKEN_TRAINING_LOAD_THREADS and KRAKEN_TRAINING_BATCH_SIZE to 8 for efficient data loading and processing, enabling 16-bit mixed precision with KRAKEN_TRAINING_PRECISION=16-mixed to reduce memory usage, and specifying KRAKEN_TRAINING_DEVICE=mps to utilise the Metal Performance Shaders backend for GPU acceleration on Apple M2 chips. These configurations significantly enhanced training performance by leveraging the capabilities of the Apple M2 GPU.

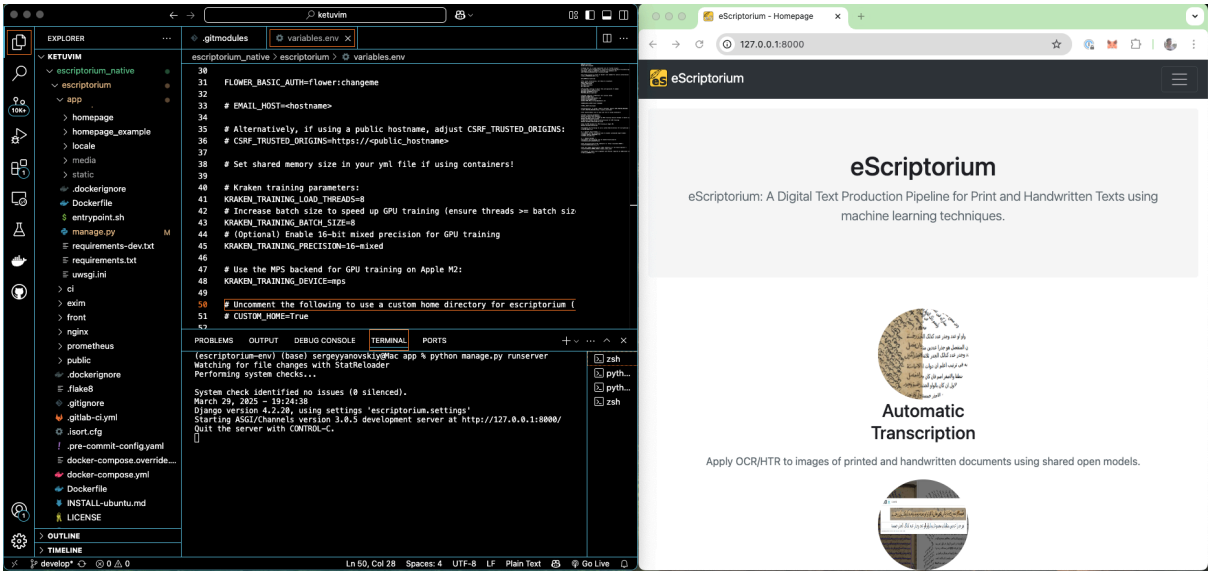


Figure 8: eScriptorium environment (full install)

Segmentation model training

To achieve the overall goals for robust handwritten text recognition, I began by focusing on the segmentation model. I created a new project in eScriptorium and uploaded a corpus of processed images of historical civil records.

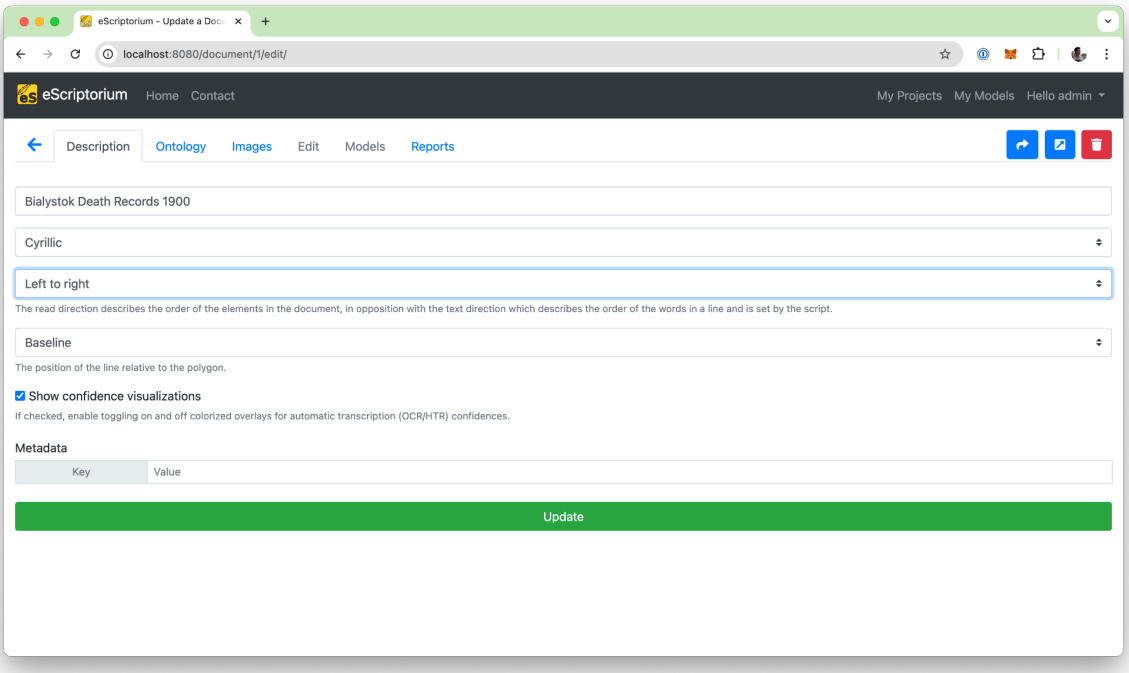


Figure 9: Creating document in eScriptorium

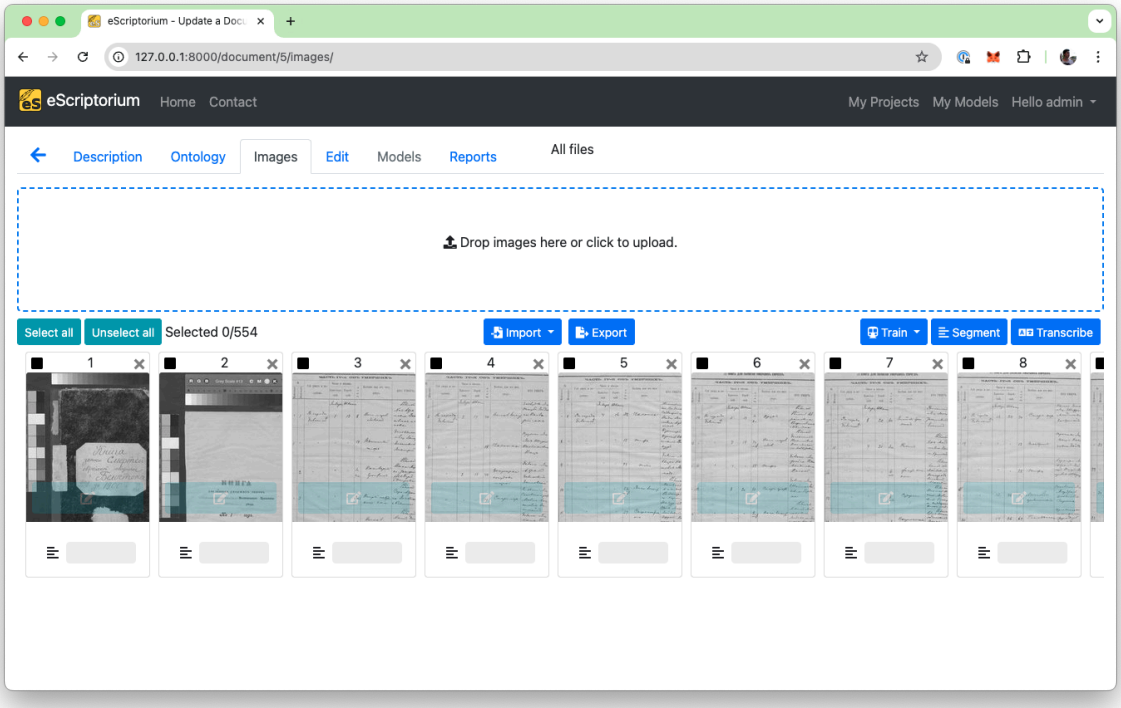


Figure 10: Uploaded images on eScriptorium

As a first step, I utilised the standard blla segmentation model [28] to automatically delineate text areas on the pages. The blla model is based on a Convolutional Recurrent Neural Network (CRNN) architecture; its convolutional layers extract spatial features from the images while its recurrent layers (typically using Bidirectional LSTMs) capture the sequential structure of text. This design enables the model to detect regions of interest even in documents with complex layouts, although it is primarily tuned to standard formats.

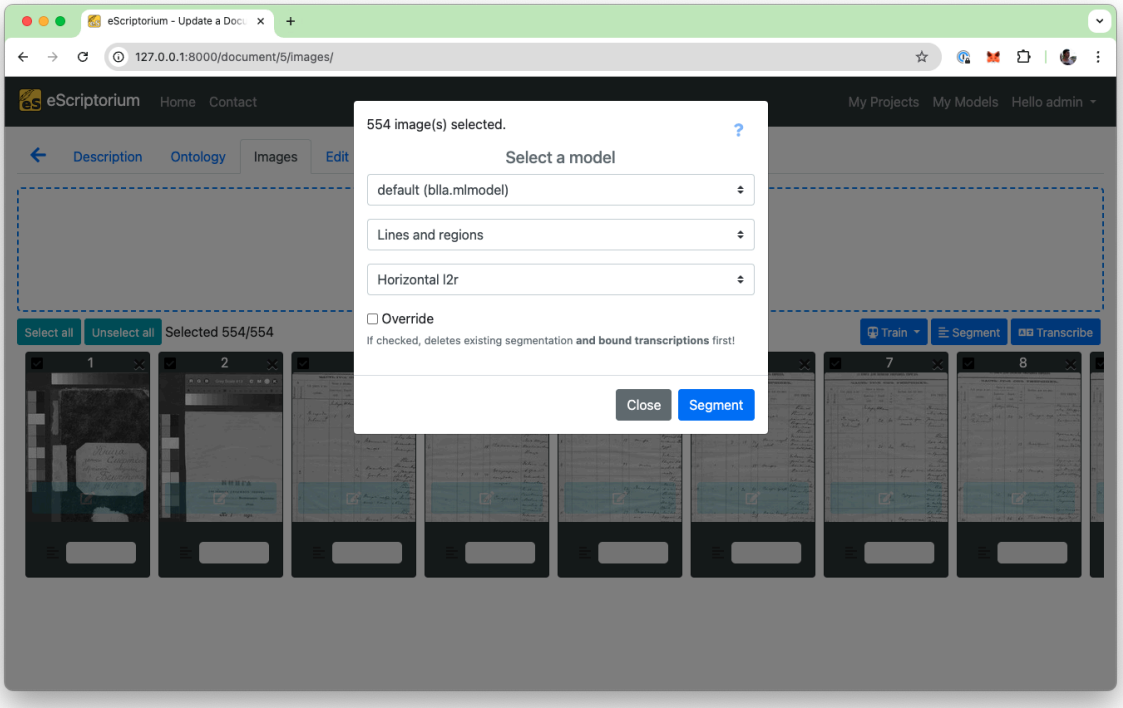


Figure 11: Segmentation on eScriptorium

For the initial pass, I segmented all available 554 pages using the blla model. I then reviewed the segmentation output through the eScriptorium user interface and manually corrected the boundaries of the detected text regions. These manual corrections ensured that the annotations precisely captured the handwritten areas, thereby creating a high-quality training dataset.

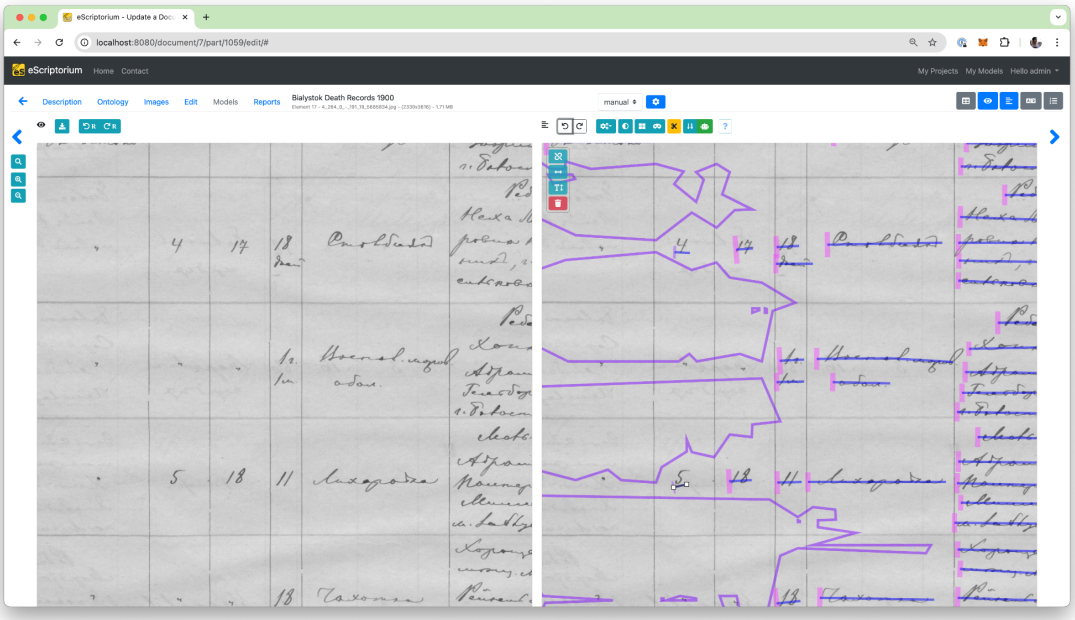


Figure 12: Process of the segmentation correction

Following the correction phase, I used this dataset to fine-tune a custom segmentation model – the ketuvim_segmenter – which is based on the blla architecture, as suggested in the eScriptorium documentation.

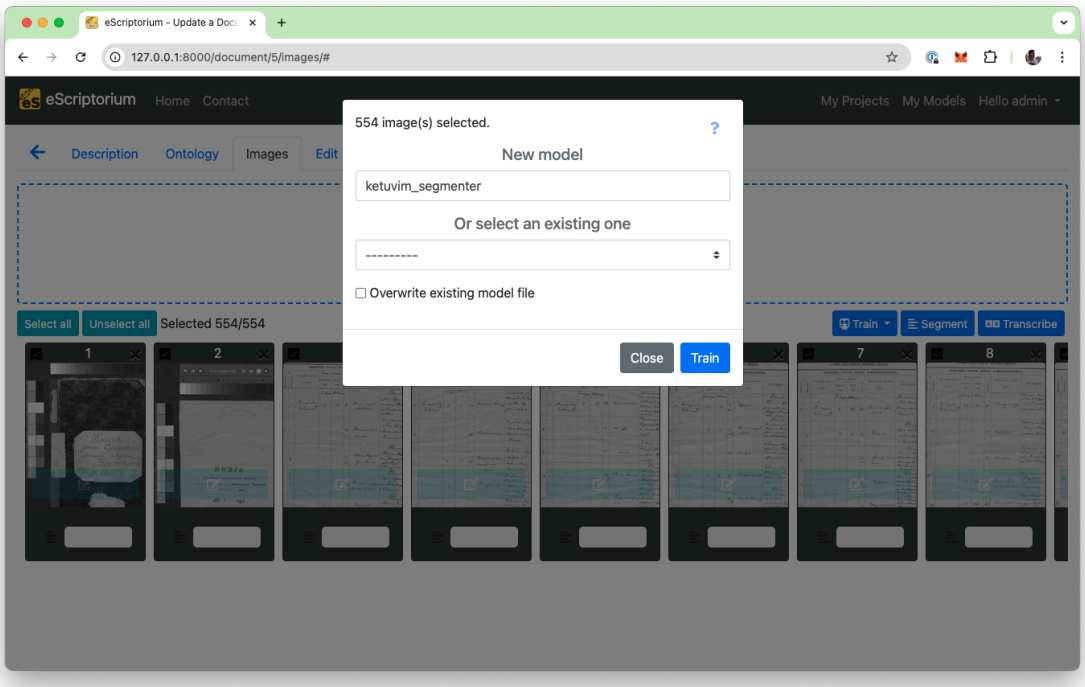


Figure 13: Training of the segmentation model

The training process was run for 99 epochs, and the model was evaluated using eScriptorium's standard assessment procedure (which compares the predicted segmentation against a validation set to calculate an accuracy metric). The final accuracy achieved was 49.6%, indicating considerable scope for improvement, which could be attained by incorporating more images from varied datasets, and better manual segmentation of text areas.

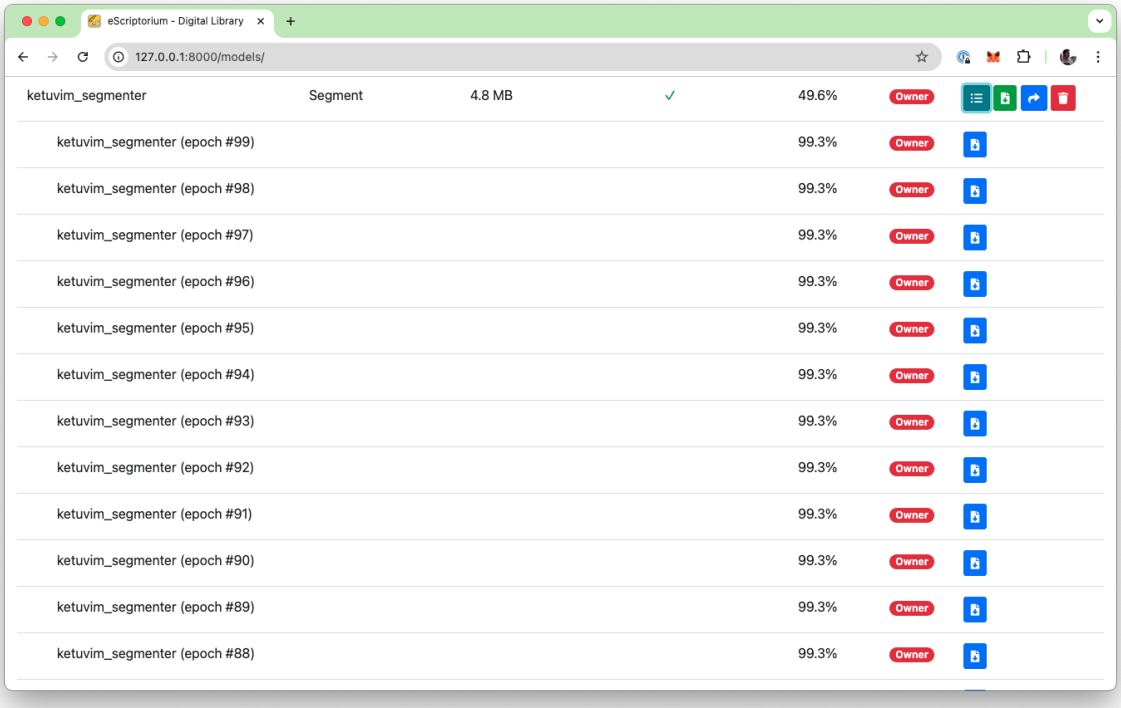


Figure 14: ketuvim_segmenter training process

Recognition model training

After completing the segmentation phase, I carried out extensive research to determine whether any pre-existing Cyrillic recognition models were available that could be applied to our dataset. I discovered several models on Transkribus, a platform well-known for historical handwriting recognition. However, due to the terms and conditions imposed by Transkribus, these models could not be exported for use within our eScriptorium environment. I therefore decided to reserve these models for future comparison.

In the meantime, I proceeded with manual transcription directly inside e-Scriptorium based on the segmented layouts created by the ketuvim_segmenter model.

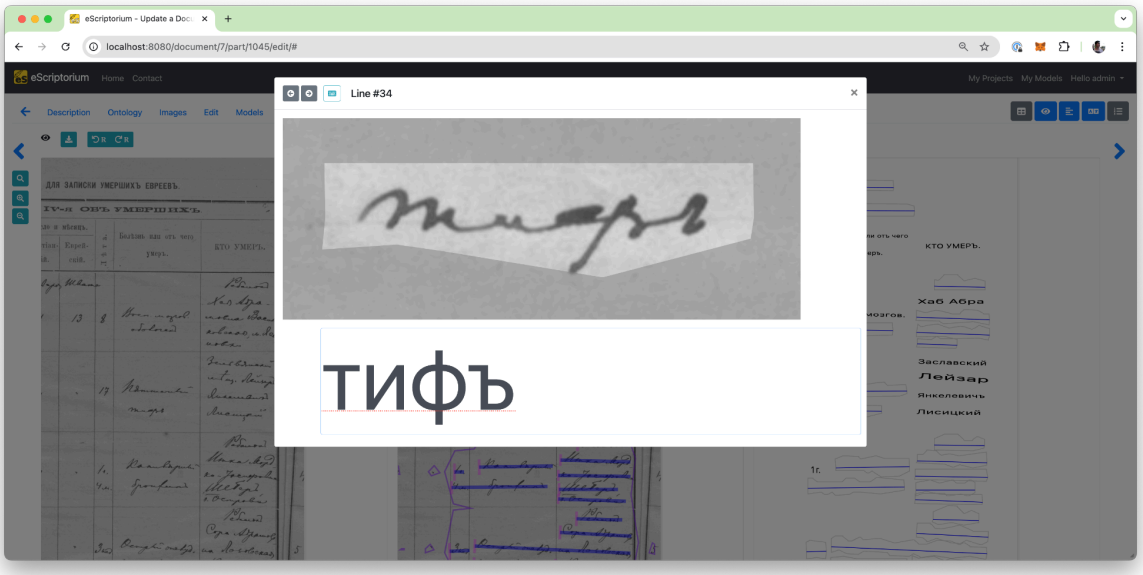


Figure 15: Manual text recognition process

Since each page often comprised both typed and handwritten text, I endeavoured to transcribe as much as possible from each page. This process was challenging in instances of particularly difficult handwriting styles, but it allowed me to generate a ground truth dataset.

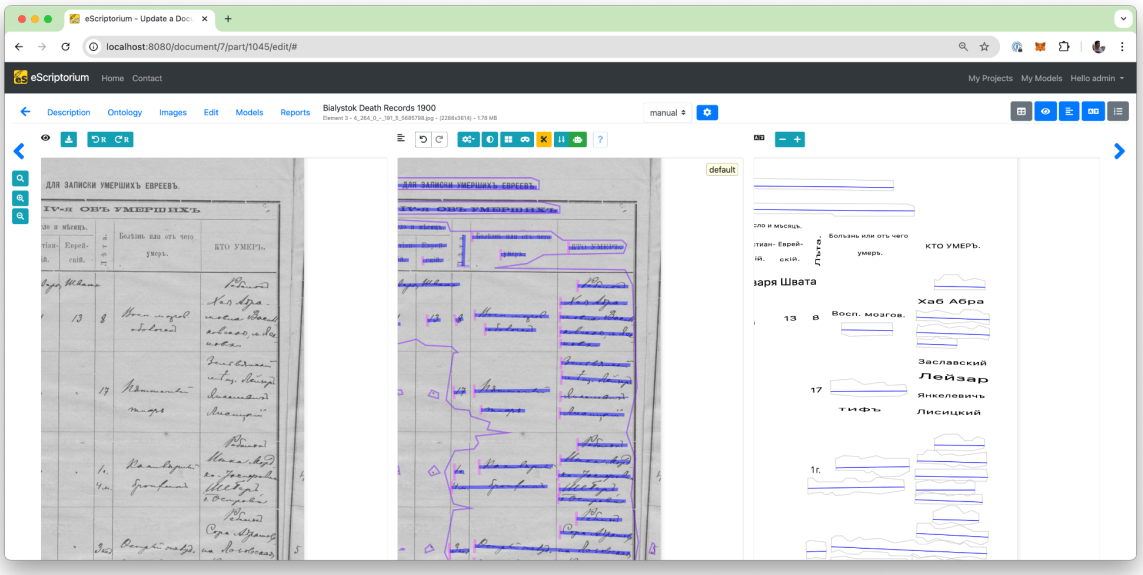


Figure 16: Manual text recognition of one of the pages

The main technology used for ketuvim_segmenter is Kraken. Kraken is a versatile, open-source Optical Character Recognition (OCR) system designed specifically for historical and non-Latin scripts. It offers fully trainable layout analysis, reading order, and text recognition, and supports scripts written in various directions, including right-to-left and vertical orientations. The underlying machine learning framework of Kraken employs a Convolutional Recurrent Neural Network (CRNN) architecture, integrating convolutional layers for robust feature extraction with recurrent layers—typically Bidirectional Long Short-Term Memory (BiLSTM)—to effectively capture sequential dependencies in textual data. This architecture is particularly suited for recognising handwritten texts, employing a Connectionist Temporal Classification (CTC) loss function to align unsegmented input images with corresponding transcriptions. Importantly, the blla model, from which I derived the ketuvim_segmenter through transfer learning, shares this CRNN architecture.

I had two primary options: training Kraken OCR through eScriptorium's graphical interface or directly using Kraken's command-line interface. Initially, I attempted training within eScriptorium, attracted by its user-friendly interface and integrated project management capabilities. However, the results obtained were unsatisfactory. A significant limitation was the inability to fine-tune hyperparameters through the interface, restricting experimentation with different configurations. Consequently, I decided to train the model directly via Kraken's command-line interface, enabling comprehensive hyperparameter tuning, greater flexibility, and significantly better results.

First, I exported all page images and their corresponding XML files from eScriptorium. These XML files included line-level annotations and transcriptions. In line with best practices for unsupervised pretraining of text recognition models [29], I compiled the data into a binary format suitable for efficient processing. To ensure that lines without transcriptions were not excluded—since these can still carry structural or contextual significance—I applied an option that preserves empty lines during compilation.

I then began pretraining the model using a setup optimised for my local machine, fine-tuning it with the following parameters: a batch size of four, a mask width of four characters, a masking probability of 20%, and three negative samples per training instance. Training ran on Apple’s Metal Performance Shaders (MPS) to take advantage of GPU acceleration. I used a small batch size of four to manage limited memory capacity. Short segments of text (four characters wide) were randomly hidden with a probability of 20%. This helps the model learn to predict missing content based on surrounding context. Additionally, for each training example, the model was shown three similar but incorrect alternatives, helping it learn to distinguish correct transcriptions from incorrect ones.

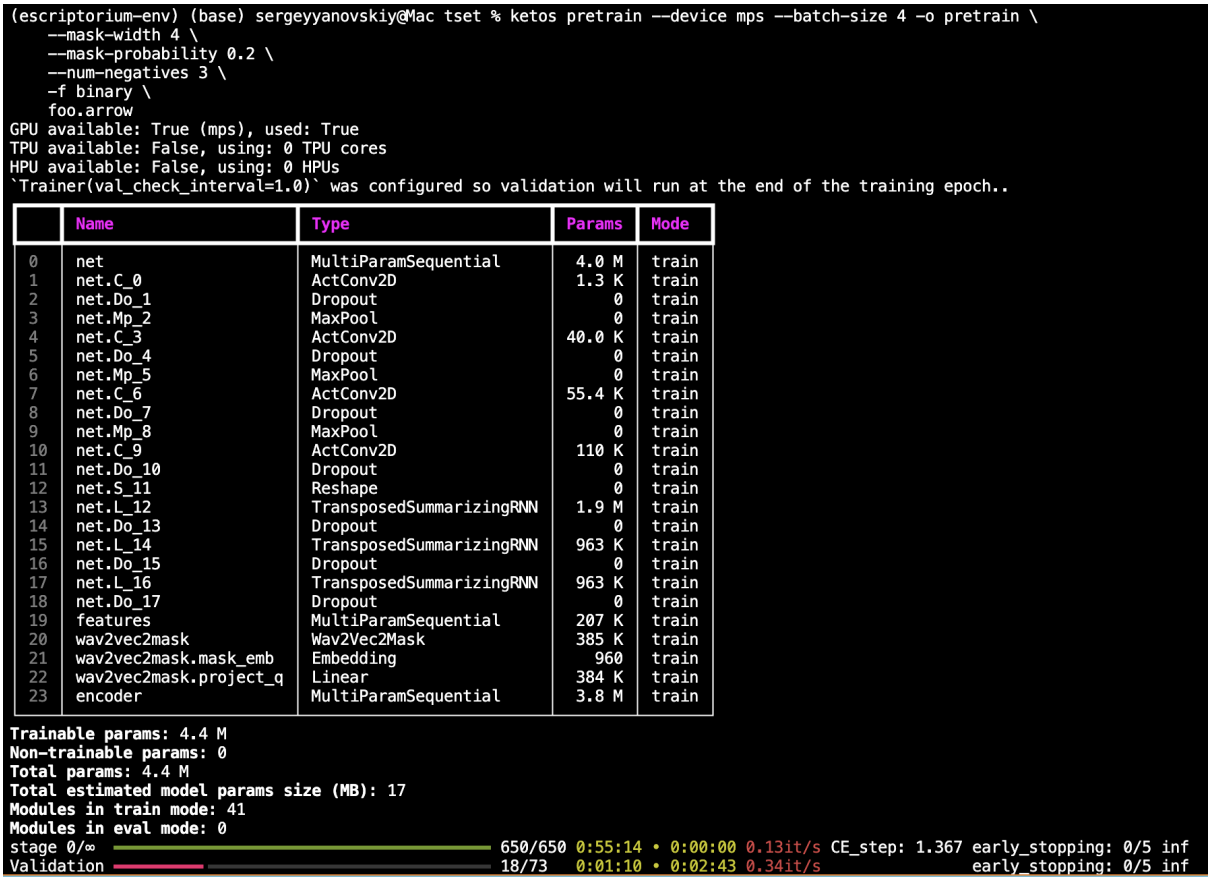
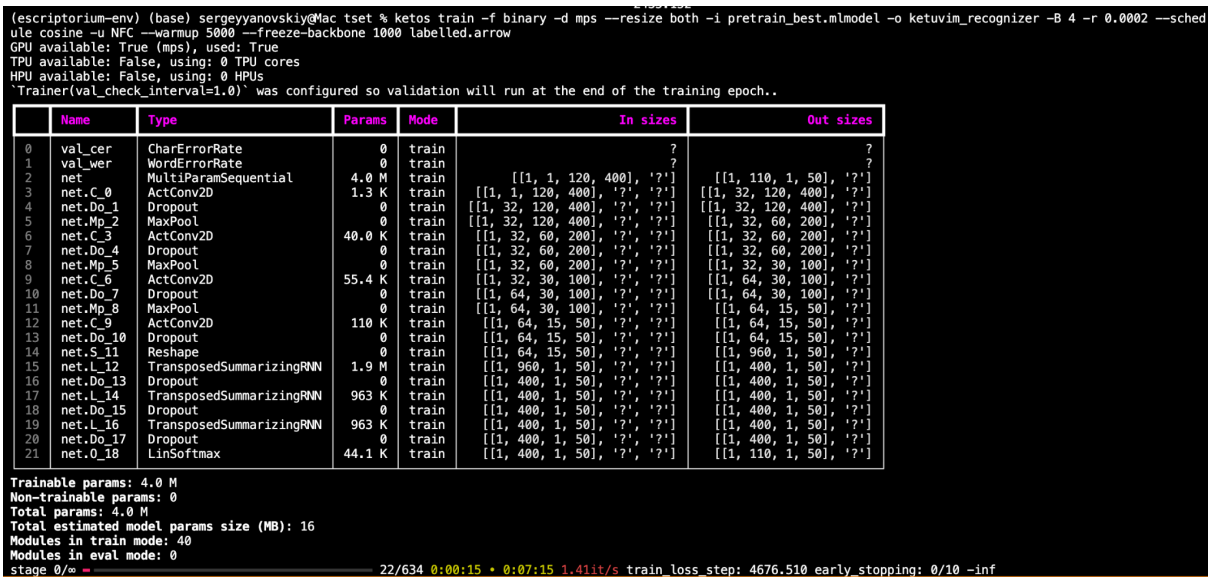


Figure 17: Unsupervised pretraining of the recognition model

After completing the pretraining stage, I fine-tuned the model following established best practices [30]. I used the pretrained weights as a starting point and trained the model further using a binary dataset of labelled lines.

The training process was executed via the terminal to allow finer control over the configuration. I enabled input resizing in both dimensions to accommodate variable image sizes and applied Unicode normalisation (NFC) to ensure consistent character representation—especially important when working with historical scripts that may include multiple encodings. I also implemented a cosine learning rate schedule with a warmup phase to stabilise early training. To conserve resources and retain learned visual features, the backbone of the network was temporarily frozen for the first 1,000 steps.

Key training parameters included a batch size of 4 and a learning rate of 0.0002. This balance was chosen to match local GPU memory constraints, as increasing the batch size typically requires a proportionally higher learning rate (scaled by the square root of the batch size) to maintain stability and convergence.



	Name	Type	Params	Mode	In sizes	Out sizes
0	val_cer	CharErrorRate	0	train	?	?
1	val_wer	WordErrorRate	0	train	?	?
2	net	MultiParamSequential	4.0 M	train	[[1, 1, 120, 400], '?']	[[1, 110, 1, 50], '?']
3	net.C_0	ActConv2D	1.3 K	train	[[1, 1, 120, 400], '?', '?']	[[1, 32, 120, 400], '?']
4	net.Do_1	Dropout	0	train	[[1, 32, 120, 400], '?', '?']	[[1, 32, 120, 400], '?']
5	net.Mp_2	MaxPool	0	train	[[1, 32, 120, 400], '?', '?']	[[1, 32, 60, 200], '?']
6	net.C_3	ActConv2D	40.0 K	train	[[1, 32, 60, 200], '?', '?']	[[1, 32, 60, 200], '?']
7	net.Do_4	Dropout	0	train	[[1, 32, 60, 200], '?', '?']	[[1, 32, 60, 200], '?']
8	net.Mp_5	MaxPool	0	train	[[1, 32, 60, 200], '?', '?']	[[1, 32, 30, 100], '?']
9	net.C_6	ActConv2D	55.4 K	train	[[1, 32, 30, 100], '?', '?']	[[1, 64, 30, 100], '?']
10	net.Do_7	Dropout	0	train	[[1, 64, 30, 100], '?', '?']	[[1, 64, 30, 100], '?']
11	net.Mp_8	MaxPool	0	train	[[1, 64, 30, 100], '?', '?']	[[1, 64, 15, 50], '?']
12	net.C_9	ActConv2D	110 K	train	[[1, 64, 15, 50], '?', '?']	[[1, 64, 15, 50], '?']
13	net.Do_10	Dropout	0	train	[[1, 64, 15, 50], '?', '?']	[[1, 64, 15, 50], '?']
14	net.S_11	Reshape	0	train	[[1, 64, 15, 50], '?', '?']	[[1, 960, 1, 50], '?']
15	net.L_12	TransposedSummarizingRNN	1.9 M	train	[[1, 960, 1, 50], '?', '?']	[[1, 400, 1, 50], '?']
16	net.Do_13	Dropout	0	train	[[1, 400, 1, 50], '?', '?']	[[1, 400, 1, 50], '?']
17	net.L_14	TransposedSummarizingRNN	963 K	train	[[1, 400, 1, 50], '?', '?']	[[1, 400, 1, 50], '?']
18	net.Do_15	Dropout	0	train	[[1, 400, 1, 50], '?', '?']	[[1, 400, 1, 50], '?']
19	net.L_16	TransposedSummarizingRNN	963 K	train	[[1, 400, 1, 50], '?', '?']	[[1, 400, 1, 50], '?']
20	net.Do_17	Dropout	0	train	[[1, 400, 1, 50], '?', '?']	[[1, 400, 1, 50], '?']
21	net.O_18	LinSoftmax	44.1 K	train	[[1, 400, 1, 50], '?', '?']	[[1, 110, 1, 50], '?']

Figure 18: Full training of the recognition model

Correction model training

To refine the outputs of the recognition model—especially where culturally specific names and terms were prone to transcription errors—I trained a correction model ketuvim_nert using a sequence-to-sequence architecture. Instead of treating the task as traditional Named Entity Recognition, the ketuvim_nert model was trained to rewrite imperfect text into its corrected form using a generative approach.

I manually compiled a domain-specific vocabulary by collecting Jewish given names, surnames, place names, and key terms from the JewishGen website [31]. This list was used to inform the construction of training data and to guide evaluation, ensuring that the corrected outputs were historically and linguistically appropriate.

The model was built using the T5-small Transformer architecture [32]. I implemented the training in a Jupyter Notebook using PyTorch, with data loaded via a custom dataset class and tokenised using the T5Tokenizer. The model was trained for 100 epochs with a batch size of 16 per device. A learning rate of 5e-5 was used, following a linear decay schedule to ensure stable convergence. The training process included regular checkpoints, with the model saved every 1,000 steps and logs recorded every 100 steps to monitor progress.

Figure 19: Correction model training

Application development

The project initially sought to implement a web-based pipeline integrating three machine learning components: a segmentation model, a recognition model (both exported from eScriptorium in `.mlmodel` format), and a named entity correction model, (a T5-based transformer). However, significant technical issues arose when attempting to run the `.mlmodel` files within the Python/Flask environment. Tools such as coremltools and Kraken's CLI failed to deliver consistent results, largely due to platform-specific limitations and internal compatibility errors.

In light of these challenges, the focus was revised. Ketuvim_segmenter and ketuvim_recognizer are now demonstrated within eScriptorium, where the models function as intended. The Flask web application, by contrast, centres on the ketuvim_nert model, which integrates smoothly with Python using the Hugging Face transformers library.

The final web interface enables users to upload an image (for context) and submit a corresponding transcription (typically sourced from eScriptorium). The backend applies the correction model and returns a refined version of the text. Users can review and edit the result in a dedicated interface, save their contribution to a local database, and download the corrected text as a `.txt` file. A separate history view allows access to previous submissions.

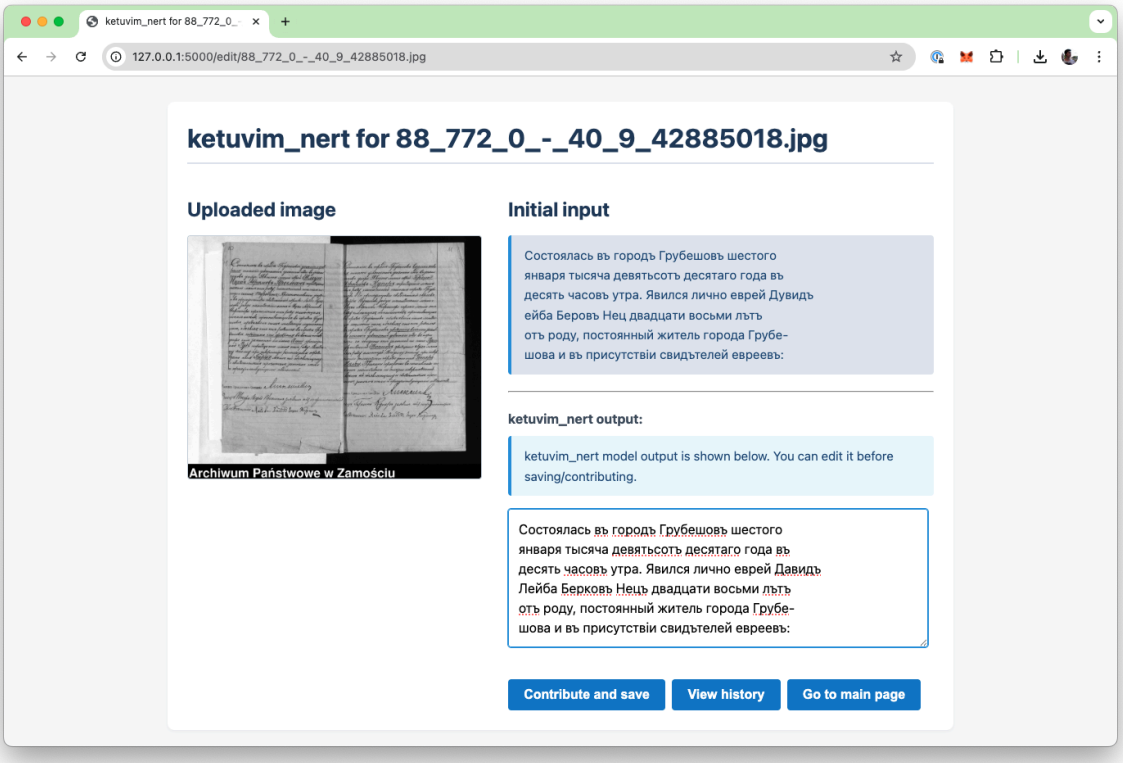


Figure 20: Web application

Evaluation

Model performance

Segmentation model (ketuvim_segmenter)

The segmentation model attained a reported accuracy of 49.6%, based on eScriptorium's standard internal evaluation. It should be noted, however, that such quantitative metrics—typically calculated at the pixel or bounding-box level—may not always correlate with real-world usability. As eScriptorium's own documentation and scholarly literature suggest [33], segmentation performance is better assessed qualitatively. In practice, even models with modest scores may yield outputs that require minimal manual adjustment. This was broadly the case for ketuvim_segmenter, which often provided a plausible baseline segmentation. While far from flawless, it was sufficiently accurate to reduce the labour involved in initial layout delineation. Visual inspection revealed that the model could usually detect text baselines and region boundaries with acceptable precision, even in complex layouts. Nonetheless, it frequently misidentified marginalia, headers, or unusually spaced lines. Manual correction remained necessary, particularly in documents exhibiting extreme handwriting variability, non-standard formats, or degradation due to ink fading. The model's utility was thus not in eliminating human input, but rather in significantly accelerating it—turning a time-consuming task into a more manageable verification step. Given the limited training data and constrained ability to fine-tune hyperparameters within eScriptorium, this accuracy should be seen as a baseline, not a ceiling. Additional annotated examples, especially those including edge cases, would likely yield significant performance improvements. Exploring alternative training environments or integrating custom loss functions could further enhance segmentation fidelity, particularly if pixel-wise confidence maps or IoU (Intersection over Union) scores are incorporated into future evaluations.

Recognition model (ketuvim_recognizer)

The initial performance of the recognition model was notably poor, with a Character Accuracy of 9.1%, equating to a Character Error Rate (CER) of 90.9% and a Word Error Rate (WER) of 100%. These figures confirm that the raw outputs are almost entirely unusable without human intervention. During user testing, most recognised text bore little resemblance to the underlying manuscript, with frequent character substitutions, missed graphemes, and total misreadings of cursive handwriting.

This result reflects both the complexity of the domain—late 19th-century handwritten Cyrillic with archaic orthography—and systemic limitations within the Kraken/eScriptorium ecosystem. In particular, there was no existing pre-trained

model suitable for fine-tuning on Jewish civil records from this period. The reliance on manually produced ground truth, albeit accurate, was insufficient in quantity to train a robust recogniser. Moreover, the inability to perform detailed hyperparameter optimisation or augment the training with synthetic data (e.g. handwriting simulations or language model priors) further hindered performance.

While these figures are discouraging at first glance, they must be interpreted within the broader context of early-stage system prototyping. Many state-of-the-art recognition pipelines exhibit similar behaviour in their initial iterations, especially when applied to underrepresented scripts or orthographic domains. A more scalable and flexible training framework, potentially outside eScriptorium, is therefore essential for progressing beyond this baseline.

Correction model (ketuvim_nert)

In contrast to the recognition component, the correction model demonstrated moderate success. Evaluated on a manually curated set of 5,000 word pairs, it achieved a Sequence Error Rate (SER) of 40.3%. This suggests that, while far from perfect, the model was able to recover a significant portion of the semantic structure and correct errors that were systematic in the raw recognition output.

The approach taken here—treating correction as a sequence-to-sequence mapping problem—was influenced by recent research in neural error correction and NER (Named Entity Recognition). The model benefits from exposure to transliteration patterns, common substitutions, and orthographic normalisation schemes. However, the relatively high SER indicates that post-processing alone cannot compensate for severely degraded recognition inputs. Correction becomes effective only when the underlying output retains at least some structural correspondence to the original text, which was not consistently the case.

Future iterations may benefit from integrating language models that are both historical and domain-specific (e.g. trained on 19th-century rabbinic Russian), or from incorporating transformer-based architectures capable of modelling broader contextual dependencies. The potential to combine this module with translation or modernisation pipelines is also a promising avenue, as many users expressed interest in outputs aligned with modern orthographic standards.

Usability assessment

A System Usability Scale (SUS) survey was conducted with 20 participants, yielding a mean score of 54. This places the interface and workflow within the "Marginal" or "Okay" range, notably below the industry-average benchmark of 68. Such a score

suggests that while the system is functional, it suffers from several usability issues that impede user satisfaction and efficiency.

User interviews and think-aloud protocols during the transcription sessions revealed that the current pipeline imposes a significant cognitive burden. As one participant succinctly put it, *“Having to use one tool for the initial transcription and then another to clean it up felt a bit disjointed.”* This fragmented workflow—segmentation, followed by recognition, then correction—requires users to switch mental modes and interfaces multiple times, introducing friction at each stage.

Moreover, participants frequently expressed frustration with the quality of automatic transcriptions. One noted, *“The way it drew boxes around the lines [segmentation] was actually quite helpful as a starting point, but the text it generated [recognition] was mostly wrong. The final correction step improved it, but it still needed checking.”* Even when corrections were applied, users still had to validate each entry, which diminished the value of automation.

Another recurring theme was the desire for more automation and integration: *“It would be great if it could just process the whole document automatically and let me fix the few remaining mistakes.”* This feedback aligns with contemporary trends in document transcription systems, where models are increasingly expected to handle multi-stage processing with minimal supervision. Additionally, users expressed interest in modernising archaic orthography, and several mentioned that an integrated translation function (e.g. into modern Russian or English) would considerably enhance the tool's utility.

Summary

Component	Metric	Result
ketuvim_segmenter (segmentation model)	Line Detection Accuracy	49.6%
ketuvim_recognizer (recognition model)	Character Accuracy	9.1%
	Character Error Rate (CER)	90.9%
	Word Error Rate (WER)	100%
Ketuvim_nert (correction model)	Sequence Error Rate (SER)	40.3%
KETUVIM	System Usability Scale (SUS) score	54

Table 4: Evaluation summary

The evaluation of the KETUVIM system highlights a functional yet early-stage prototype with distinct strengths and limitations across its three core components: segmentation, recognition, and correction. The segmentation model (ketuvim_segmenter) achieved moderate success, with an accuracy of 49.6%. While this figure may appear modest, it belies the model's practical utility—serving as a useful baseline for visual layout detection and significantly reducing the manual effort required to begin transcription.

The recognition model (ketuvim_recognizer) performed poorly in its current form, producing raw transcriptions with a Character Error Rate of 90.9% and a Word Error Rate of 100%. These outputs, largely unintelligible without intervention, underscore the difficulty of the task and the limitations of working within a constrained training environment without access to pre-trained historical Cyrillic models.

In contrast, the correction model (ketuvim_nert) demonstrated some potential, yielding a Sequence Error Rate of 40.3% on a set of 5,000 word pairs. Though far from perfect, it was able to correct a significant proportion of recogniser errors and restore partial semantic coherence to the output, particularly where recognition retained some structural resemblance to the input.

The usability assessment, incorporating both SUS scoring and user interviews, further contextualised these results. A SUS score of 54—well below the average threshold—reflected user frustration with the multi-stage workflow, inconsistent output quality, and the cognitive effort required to manage fragmented tools. Nevertheless, participants acknowledged the value of segmentation outputs and expressed strong interest in the project's potential, particularly if greater automation and linguistic modernisation were introduced.

Taken together, these findings confirm that while KETUVIM is not yet suitable for unsupervised transcription, it offers a promising proof of concept. Its current iteration demonstrates the feasibility of building a domain-specific pipeline for historical Cyrillic handwriting using open-source tools, and provides a meaningful starting point for further development.

Conclusion

The KETUVIM project has laid the groundwork for a functional transcription pipeline tailored to historical Jewish records written in Cyrillic script. Using open-source tools—primarily eScriptorium and Kraken—the project developed, trained, and evaluated models for segmentation, recognition, and correction within a local environment. This approach was necessitated by the lack of suitable pre-trained

models for the domain and the highly specific nature of the material, which included late 19th-century civil records sourced from Polish state archives. These documents presented considerable challenges: faded ink, inconsistent orthography, and handwritten styles unfamiliar to most OCR systems. Despite these barriers, KETUVIM demonstrated that meaningful progress can be achieved through iterative development and domain-specific adaptation.

Evaluation revealed a mixed picture. The segmentation model (ketuvim_segmenter) achieved 49.6% accuracy and proved helpful in generating an initial layout structure for further correction. The recognition model (ketuvim_recognizer) performed poorly (CER: 90.9%; WER: 100%), reflecting both the complexity of the script and the limitations of the training pipeline. The correction model (ketuvim_nert) performed moderately better (SER: 40.3%) and showed promise in refining raw outputs, though not to the level required for unsupervised use. A System Usability Scale (SUS) survey produced a score of 54, confirming user concerns about the disjointed nature of the multi-stage workflow and the cognitive overhead involved. While these results confirm that the system is not yet suitable for large-scale deployment, they validate the core design and underscore its potential as a modular and adaptable research tool.

Looking forward, there are several avenues for further exploration and improvement. Based on both user feedback and literature review, future work might be focused on:

- **Dataset expansion and diversification**: The training dataset was deliberately curated to represent variation in handwriting, layout, and content types. However, its size and scope remain limited. Expanding the dataset—both in terms of volume and representativeness—is essential for improving generalisation. This includes sourcing additional material from different regions of the former Russian Empire, covering more document types (e.g. census records, community lists, migration logs), and incorporating a broader range of handwriting styles. Such diversity would help mitigate overfitting and improve model robustness across unseen data. In parallel, collaboration with historical archives and genealogical organisations could facilitate access to materials and metadata, and potentially support crowd-sourced annotation efforts.
- **Adoption of more advanced model architectures:** The limitations encountered with CRNN-based recognition suggest that newer architectures may offer significant advantages. Transformer-based models, particularly those capable of end-to-end document understanding, should be explored. The *Party* model (PAge-wise Recognition of Text-y) [34] is especially relevant. It eliminates the need for explicit segmentation by learning to detect and transcribe lines directly, using a Swin Vision Transformer encoder and a decoder inspired by LLaMA. Pretrained across a range of languages and

scripts, including Cyrillic, it offers a compelling alternative to the segmented, line-by-line approach currently in use. Fine-tuning such a model on historical Jewish Cyrillic data could reduce preprocessing requirements, improve transcription accuracy, and bring the system closer to real-world usability.

- **Workflow integration and automation:** One of the most consistent pieces of feedback from participants was the need for a smoother, more integrated workflow. Currently, users must manually initiate segmentation, recognition, and correction in sequence, often switching contexts and interfaces. A single-page application or unified web interface—where the user can upload a document, preview segmentation, trigger recognition, and edit results in-place—would greatly enhance usability. Automation of common steps (e.g. auto-processing on upload, intelligent line merging, or batch correction suggestions) would further reduce friction, particularly for non-specialist users.
- **Historical orthography and language support**: The transcription of historical Cyrillic often results in text that is difficult for modern readers to interpret due to outdated vocabulary, non-standard spelling, and typographic conventions. Incorporating automatic orthographic normalisation—mapping archaic spellings to modern equivalents—would improve accessibility for users unfamiliar with 19th-century Russian. In addition, offering optional post-transcription translation into modern Russian or English could broaden the utility of the system beyond academic researchers to include genealogists and community users. This could be achieved using neural machine translation models trained or fine-tuned on parallel corpora from similar historical contexts.
- **Evaluation framework and metrics extension:** While standard metrics such as CER, WER, and SER offer useful benchmarks, they are not always sufficient to capture the usability of output in historical contexts. Future iterations of KETUVIM should incorporate additional evaluation layers, including semantic accuracy (e.g. how well names and place names are preserved), user-centred measures (e.g. correction time per page), and confidence-based scoring. Visualisation of model uncertainty could also guide users during correction, focusing attention on areas with high predicted error likelihood.
- **Community engagement and sustainability:** To ensure continued development, KETUVIM should be positioned not only as a research prototype but also as an open platform for community collaboration. Publishing annotated datasets, pre-trained models, and scripts for pipeline execution would allow other researchers to build on this work. Involving archivists, genealogists, and domain historians in the feedback loop could guide priorities and create a shared ecosystem of tools, models, and best practices.

In summary, KETUVIM has achieved its primary goal: to design and prototype an open-source pipeline for the transcription of historical Cyrillic documents of Jewish provenance. It has demonstrated the viability of this approach in a domain that remains largely underexplored and technically demanding. While the system is not yet fit for unsupervised or large-scale deployment, it represents a significant step forward. By clearly identifying the technical, linguistic, and usability challenges involved, and by articulating concrete strategies for future improvement, the project provides a roadmap for transforming a promising prototype into a robust, scalable tool for historical and genealogical scholarship.

References

[1] THE JEWISH AGENCY. 2023. Jewish population rises to 15.7 million worldwide in 2023. The Jewish Agency for Israel. https://www.jewishagency.org/jewish-population-rises-to-15-7-million-worldwide-in-2023/

[2] TRACING THE TRIBE. 2023. Tracing the tribe: Jewish genealogy on Facebook. Facebook. https://www.facebook.com/groups/tracingthetribe/?locale=en_GB

[3] INTERNATIONAL ASSOCIATION OF JEWISH GENEALOGICAL SOCIETIES. 2023. Member societies. https://www.iajgs.org/membership/member-societies/

[4] RABUS, A. 2019. *Recognizing handwritten text in Slavic manuscripts: A neural-network approach using Transkribus*. Scripta & e-Scripta 18, 305–320. Bulgarian Academy of Sciences.https://e-scripta.ilit.bas.bg/18

[5] FERGUSON, D. 2024. *Rise in DNA tests being used to claim citizenship of other countries*. The Guardian. https://www.theguardian.com/science/article/2024/aug/18/rise-in-dna-tests-used-to-claim-citizenship-of-other-countries-brexit-eu

[6] POLARIS MARKET RESEARCH. 2023. Genealogy products & services market size, share, future outlook 2023–2032. https://www.polarismarketresearch.com/industry-analysis/genealogy-products-and-services-market

[7] CREDENCE RESEARCH. 2021. Handwriting recognition (HWR) market share & forecast 2027. https://www.credenceresearch.com/report/handwriting-recognition-hwr-market

[8] GROWTH MARKET REPORTS. 2023. Handwriting recognition market global industry analysis. https://growthmarketreports.com/report/handwriting-recognition-market-global-industry-analysis

[9] ARKHIPOV, A., BARINSKAYA, A., AND SHTEFURA, R. 2021. *Using handwritten text recognition on bilingual Evenki-Russian manuscripts of Konstantin Rychkov*. Scripta & e-Scripta 21, 233–244. Bulgarian Academy of Sciences. https://e-scripta.ilit.bas.bg/21

[10] ABDALLAH, A., HAMADA, M., AND NURSEITOV, D. 2020. *Attention-based fully gated CNN-BGRU for Russian handwritten text*. J. Imaging 6, 12, 141. https://doi.org/10.3390/jimaging6120141

[11] KALKEN, M. AND JANTAYEV, R. 2022. *Cyrillic handwritten optical character recognition: A review of various recognition methods*. Proceedings of IYSC, vol. 11, Suleyman Demirel University, Kaskelen, Kazakhstan. https://journals.sdu.edu.kz/index.php/iysw/article/view/800/351

[12] TOIGANBAYEVA, N., KASEM, M., ABDIMANAP, G., BOSTANBEKOV, K., ABDALLAH, A., ALIMOVA, A. AND NURSEITOV, D. 2021. *KOHTD: Kazakh offline handwritten text dataset*. arXiv preprint arXiv:2110.04075. https://doi.org/10.48550/arXiv.2110.04075

[13] LIEPIESHOV, K. AND DOBOSEVYCH, O. 2019. *On recognition of Cyrillic text*. OpenReview. https://openreview.net/forum?id=Ske6GT9c8r

[14] CRISTEA, D., CLEJU, N., REBEJA, P., HAJA, G., COMAN, E., VASILESCU, A., MARINESCU, C., AND DASCĂLU, A. 2023. *Bringing the old writings closer to us: Deep learning and symbolic methods in deciphering old Cyrillic Romanian documents*. Memoirs of the Scientific Sections of the Romanian Academy, Tome XLVI. https://mss.academiaromana-is.ro/mem_sc_st_2023/10_DCristea_final.pdf

[15] PINCHE, A. AND STOKES, P. 2023. *Historical documents and automatic text recognition: Introduction*. Journal of Data Mining and Digital Humanities (2023). https://jdmdh.episciences.org/13247/pdf

[16] DARBY, P. AND CLOUGH, P. 2013. *Investigating the information-seeking behaviour of genealogists and family historians*. Journal of Information Science 39, 1, 73–84. DOI:https://doi.org/10.1177/0165551512469765

[17] HERSHKOVITZ, A. AND HARDOF-JAFFE, S. 2017. *Genealogy as a lifelong learning endeavor*. Leisure/Loisir 41, 4 (October 2017), 535–560. DOI:https://doi.org/10.1080/14927713.2017.1399817

[18] FORCE, D. AND WILES, B. 2021. *Article 18 Historical Research in the Web Era*. Journal of Contemporary Archival Studies Journal of Contemporary Archival Studies 8, 18. https://elischolar.library.yale.edu/cgi/viewcontent.cgi?article=1140&context=jcas

[19] BYU IDAHO. 2024. *Family History and Genealogical Research*. https://books.byui.edu/fhgen_110_textbook_/chapter_6_high_research_standards_in_professional_research

[20] GARRETT-NELSON, L. 2021. Transcribing Documents: There Is More Than Meets The Eye!

https://familytreewebinars.com/wp-content/uploads/2021/07/webinar-free1555013802.pdf

[21] NOLL, A.G. 2012. *Crowdsourcing Transcriptions of Archival Materials*. https://scholarworks.umb.edu/cgi/viewcontent.cgi?article=1062&context=ghc

[22] ARCHIWA PAŃSTWOWE. 2009. Szukaj w Archiwach. https://www.szukajwarchiwach.gov.pl/

[23] ARCHIWUM PAŃSTWOWE W BIAŁYMSTOKU. 2014. Akta zgonów za rok 1900 w Białymstoku. https://www.szukajwarchiwach.gov.pl/en/jednostka/-/jednostka/17474866

[24] ARCHIWUM PAŃSTWOWE W ZAMOŚCIU. 2014. Akta urodzeń, małżeństw, zgonów za rok 1910 w Hrubieszowie. https://www.szukajwarchiwach.gov.pl/jednostka/-/jednostka/10504727

[25] ARCHIWUM PAŃSTWOWE W LUBLINIE. 2013. Księga urodzeń za roki 1890-1891 w Lublinie. https://www.szukajwarchiwach.gov.pl/en/jednostka/-/jednostka/2349526

[26] PETER A. STOKES, BENJAMIN KIESSLING, DANIEL STÖKL BEN EZRA, ROBIN TISSOT, AND EL HASSANE GARGEM. 2020. *The eScriptorium VRE for Manuscript Cultures*. Classics@ Journal. https://classics-at.chs.harvard.edu/classics18-stokes-kiessling-stokl-ben-ezra-tissot-gargem/

[27] TISSOT, R. 2022. eScriptorium full install. GitLab. https://gitlab.com/scripta/escriptorium/-/wikis/full-install

[28] MITTAGESSEN. 2022. blla.mlmodel. GitHub https://github.com/mittagessen/kraken/blob/main/kraken/blla.mlmodel

[29] KIESSLING, B. 2019. Training — kraken documentation. Kraken.re. https://kraken.re/main/ketos.html#unsupervised-recognition-pretraining

[30] CHAUHAN, R. 2023. Train Your Own OCR/HTR Models with Kraken, part 1. The Digital Orientalist. https://digitalorientalist.com/2023/09/26/train-your-own-ocr-htr-models-with-kraken-part-1/

[31] JEWISHGEN. 2015. JewishGen - The Global Home for Jewish Genealogy. Jewishgen.org. https://jewishgen.org/

[32] HUGGING FACE. 2024. google-t5/t5-small. Huggingface.co. https://huggingface.co/google-t5/t5-small

[33] ESCRIPTORIUM. 2019. eScriptorium Documentation. Readthedocs.io. https://escriptorium-tutorial.readthedocs.io/

[34] MITTAGESSEN. 2024. GitHub - mittagessen/party: Page-wise text recognition with lower-supervision line data models. GitHub. https://github.com/mittagessen/party

Appendix

Github repository: https://github.com/kaneepston/ketuvim