PRELIMINARY PREPRINT VERSION – DO NOT CITE
This is not the version of record. The final version will appear in the Proceedings of IJCAI-ECAI 2026.

Interactive System for Reducing Error Propagation in Multi-Stage Ancient Egyptian Text Analysis

Maksim Golyadkin[1,2] , **Innokentiy Humonen**[1,2] and **Ilya Makarov**[1,2]
[1]AXXX, Moscow, Russia
[2]Trusted AI Research Center, RAS, Moscow, Russia
{golyadkin, humonen, makarov}@axxx.tech

Abstract

We introduce a web tool that puts the full image-to-text pipeline for Ancient Egyptian hieroglyphs inside a single annotation workspace. Instead of only producing a final transcription, our system exposes editable intermediate results so users can validate and correct the pipeline step by step. User edits are stored as image-aligned annotations, which supports both text analysis and dataset creation. Quantitative results indicate improved efficiency and output quality relative to a manual baseline. A demonstration video is available at Google Drive.

1 Introduction

Hieroglyphic inscriptions are a primary record of language, religion, and daily life in Ancient Egypt. Yet much of this material is available primarily as images (photographs, facsimiles, scanned editions), and the lack of reliable digitization makes it hard to apply even basic text-based tooling such as search, indexing, and structured reuse. Consequently, turning image-based sources into machine-readable representations remains a major bottleneck, keeping analysis slow and making outputs harder to standardize and reproduce.

Recent deep learning advances suggest substantial potential for assisting Egyptological analysis. Prior work has explored text-domain modeling for tasks such as transliteration and translation [Wiesenbach and Riezler, 2019; De Cao *et al.*, 2024; Ranathunga *et al.*, 2023], showing that neural language models can learn mappings between established Egyptological representations and produce plausible outputs from textual inputs. In practice, however, the limiting factor is often not the text-domain model but the availability of digitized text. Manually converting image-based sources into consistent machine-readable representations remains slow and expertise-intensive, creating a bottleneck between image sources and downstream language processing. Image-to-text pipelines have appeared more recently [Kim *et al.*, 2022; Li *et al.*, 2023; Wei *et al.*, 2024; Golyadkin *et al.*, 2025a], but they lack an integrated interface, so users cannot easily fix errors before they propagate downstream.

In this work, we tackle an adoption barrier by introducing pyThoth, a web-based system that embeds an image-to-text pipeline into one interactive workspace for Egyptological work. Instead of only showing final predictions, pyThoth surfaces editable intermediate representations so users can verify and correct results as they go. These edits are stored as image-aligned annotations, enabling human-in-the-loop refinement while simultaneously generating reusable labeled data. By letting users fix mistakes early, the design limits downstream error propagation and keeps the system usable without ML expertise. In an experiment with experts and students, pyThoth led to faster task completion and higher-quality outputs than a manual workflow.

2 Related Work

Manual interpretation of a hieroglyphic inscription is a multi-step workflow. Linguists first identify individual signs, then produce a transliteration, and finally translate the text into a modern language. The process is complicated by uncertainty and the need for many revisions. Sign classification is often the main bottleneck due to the existence of a large number of signs represented with Gardiner codes (e.g., D36, G1) [Franken and van Gemert, 2013], which makes lookup unavoidable. Although digital resources exist, the tools are not coordinated with one another. General-purpose annotation tools such as Label Studio [Tkachenko *et al.*, 2020 2025] support image annotation but lack hieroglyph-specific helpers, while domain-specific tools such as JSesh [Rosmord,] support these but are not designed for image annotation. Several machine-learning systems target hieroglyphic text analysis [De Cao *et al.*, 2024; Miyagawa, 2025; Barucci *et al.*, 2021; Guidi *et al.*, 2023; Fuentes-Ferrer *et al.*, 2025; Golyadkin *et al.*, 2025c], yet most operate in text-only settings that depend on already-digitized inputs. Few image-to-text pipelines [Golyadkin *et al.*, 2025a; Elnabawy *et al.*, 2018; Sobhy *et al.*, 2023] exist, and they are research prototypes without an interface. As a result, there is still no available tool that supports image-based hieroglyphic analysis in routine practice.

pyThoth addresses this gap by unifying image annotation and hieroglyphic text analysis in a single workspace. It provides a hieroglyph look-up table and a visualization tool. Moreover, it adds DL suggestions that can be iteratively improved. The goal is both practical utilization for research and education and help with digitization, because users may switch tools less, and the output of their work may be exported as digitized text.

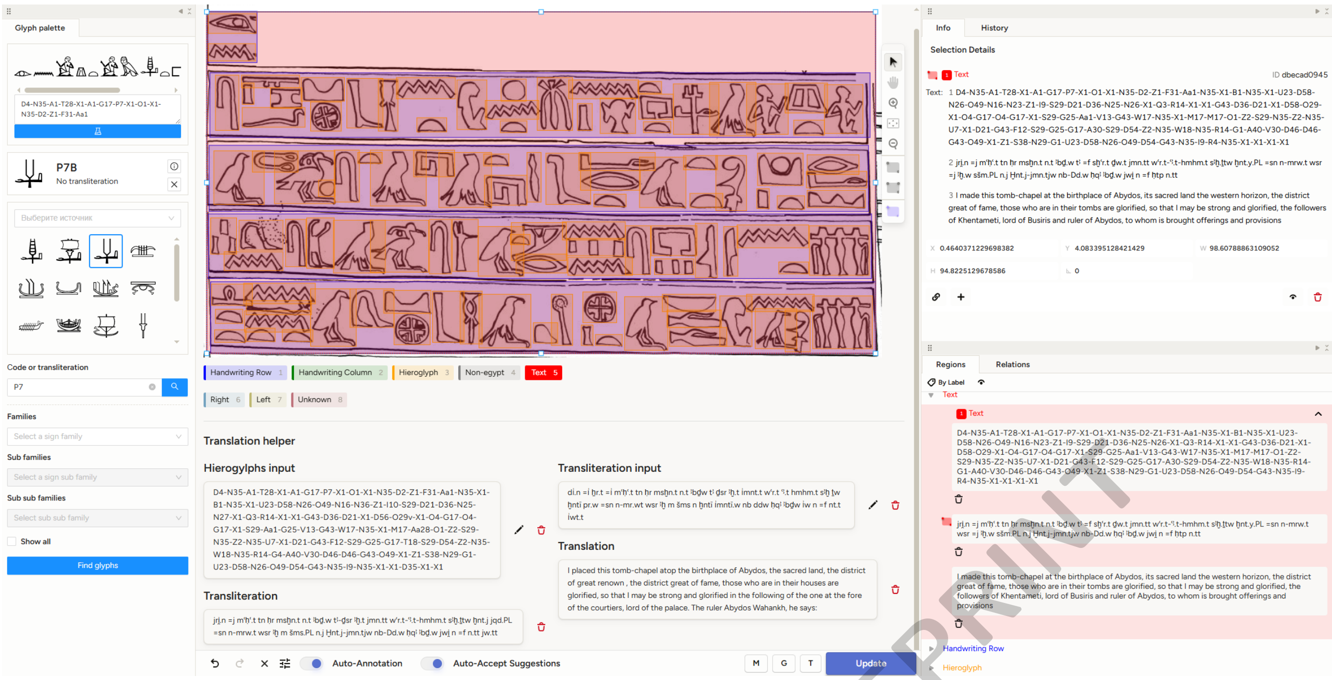

Figure 1: The pyThoth workspace, built on Label Studio and extended with three Egyptology-specific tools: a *Hieroglyph Palette* for browsing and entering Gardiner codes, a *Hieroglyph Renderer* for previewing code sequences, and a *Translation Helper* that returns on-demand transliteration and translation suggestions. All codes, transliteration, and translation fields remain editable and linked to the image regions, so corrections can be fed back into the pipeline at any stage.

3 pyThoth Overview

We present pyThoth, a web system integrating a deep-learning pipeline into an interactive workspace for multi-step image-to-text analysis of hieroglyphic texts. The main design goal is to minimize constant tool switching and to support fast but still reasonable automatic text analysis which can be exported as structured outputs, so they can be used as annotated data for following iterations of training.

Our system builds on Label Studio, which provides a strong interface for text- and image-level annotation. Inside one workspace, users go from an image to Gardiner codes, then to a transliteration, and finally to an English translation, refining each stage along the way. All intermediate representations remain editable, enabling users to correct model predictions that are then reused as inputs to subsequent stages of the DL pipeline. The pipeline is connected to Label Studio via the *ML Backend*, which provides a communication protocol between the frontend and backend.

Proposed system is available as a hosted web application and can be accessed freely from any standard browser at https://ls-v2.pythoth.com/.

System Interface. pyThoth provides an OCR-style interface where users draw bounding boxes and edit corresponding text fields (codes, transliteration, translation), connecting text and image domains. We extend Label Studio with a *Hieroglyph Palette* for fast Gardiner-code lookup, a *Hieroglyph Renderer* for in-place code visualization, and a *Translation Helper* suggesting transliteration and English translation for codes and transliterations of arbitrary length and source.

ML Backend. The backend is built around a DL pipeline [Golyadkin *et al.*, 2025a] for Egyptian text analysis. It produces predictions at several stages: sign bounding boxes with Gardiner codes, their reading order, and then candidate transliteration and translation using Transformer models. Because end-to-end outputs are still imperfect [Golyadkin *et al.*, 2025b], exposing intermediate representations lets users correct them before errors propagate. Moreover, this step-by-step setup is close to the normal Egyptological workflow, so the system can be introduced without asking users to work in a completely different way.

Technical Details. Generating initial suggestions requires about 8 GB RAM and takes roughly 2 minutes per image on CPU or about 25 seconds on a consumer GPU. The system is available as a hosted web application and runs in a standard browser without special local hardware.

4 User Study

To see how pyThoth performs in real Egyptological practice, we ran a user study against the manual workflow as baseline. We track both the speed and the quality of the produced outputs, and we record how participants edit the model's draft suggestions.

Setup. We evaluated pyThoth in a within-subject study with $N{=}8$ participants (6 undergraduate Egyptology students and 2 PhD Egyptologists). The task used 4 cropped images (3-4 lines each) taken from facsimile-style drawings of Middle Egyptian stelae, selected to be comparable in difficulty and mutually independent. Per image, participants de-

Subtask	**Time spent**				**User edits**	
	Manual (min)	**pyThoth (min)**	**Saved (min)**	**Saved (%)**	**Edit distance NLD (%)**	**Plain summary**
Recognition	39	34	5	12.8	8.6	Few edits
Transliteration	17	11	6	35.3	16.0	Some edits
Translation	9	6	3	33.3	23.6	More edits
Total	**61**	**51**	**10**	**16.4**	–	–
Mean	**22**	**17**	**5**	**22.7**	**16.0**	Some edits

Table 1: Per-subtask breakdown of how much time pyThoth saves and how much editing the participant has to do on top of the DL pipeline draft. Time spent is reported in minutes for the manual baseline and for pyThoth, alongside the absolute and relative time saved per subtask. Editing effort is reported as normalized Levenshtein distance (NLD, in percent, lower means the participant kept more of the model's draft) between the pipeline's initial prediction and the participant's output, with a plain-language summary in the rightmost column for quick reading. Across all three subtasks pyThoth is faster while edits stay modest, suggesting that the time gain is not eaten by a correction burden.

Setting	**Recogn.**	**Translit.**	**Transl.**
Human	13.4	42.5	44.1
DL pipeline	9.1	59.4	52.0
pyThoth	**8.9**	**59.6**	**52.3**

Table 2: Quality of the three pipeline outputs (recognition, transliteration, translation) under three working conditions. Recognition is scored with character error rate (CER, lower is better), and the two text outputs with chrF++ (higher is better). *Human* denotes the manual workflow without our tool, *DL pipeline* denotes the uncorrected automatic prediction, and *pyThoth* denotes the prediction after the participant edited the intermediate codes and transliteration in our interface. pyThoth is best on all three outputs, with the clearest gains on recognition and translation, which we attribute to fixing errors at the stage where they appear rather than letting them propagate downstream.

livered three outputs in order: the Gardiner-code sequence following the natural reading order, the transliteration, and the English translation. Each participant completed half of the images with their baseline manual workflow (their usual sign lists/dictionaries and optional tools such as PDFs/web resources or JSesh) and the other half with pyThoth. Assignment was counterbalanced, and external resources were allowed in both settings. Participants watched a short video tutorial, submitted outputs via a web form, and recorded completion time per image. We report efficiency as task completion time, and quality as sign-level CER for codes and chrF++ for transliteration and translation. User effort with pyThoth is quantified as the normalized Levenshtein distance between the model's initial suggestion and the participant's answer.

Results. Table 1 reports per-subtask times and the editing effort that pyThoth users had to spend on top of the DL drafts. On total time the pyThoth condition wins (51 minutes against 61 for the manual baseline, a 16% reduction), in line with the formative evaluation reported in [Humonen *et al.*, 2026]. At the same time, the average edit distance over the three subtasks stayed around 16% NLD, suggesting that time savings were not offset by a substantial correction burden.

Regarding quality evaluation, Table 2 compares three settings: *Human* (manual workflow), the uncorrected *DL pipeline*, and *pyThoth* with intermediate user refinements. pyThoth achieves the strongest overall scores, particularly for recognition and translation, highlighting the value of interactive refinement over either human-only work or fully automatic predictions.

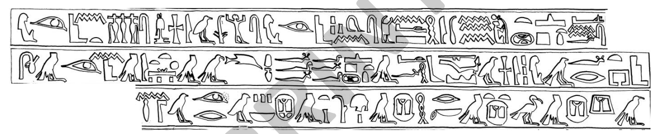

Figure 2: Example of facsimile-style drawing used in the user study.

Overall, pyThoth improves efficiency: participants finished faster while making only modest edits to the pipeline drafts. By surfacing editable intermediate representations, the system lets users correct errors at the appropriate stage and prevents them from propagating downstream, which helps explain why the final outputs outperform both the manual workflow and the end-to-end DL pipeline.

5 Conclusion

We presented pyThoth, an interactive system for image-based Ancient Egyptian text analysis. The tool enables Egyptologists to work more efficiently and produce higher-quality outputs. Next, we plan to improve pyThoth in two directions: training stronger models with more data, and redesigning the interface. Beyond improving the current system, we see pyThoth as a step toward a broader research environment for historical text analysis. By keeping evidence, intermediate representations, and expert corrections connected, it can produce reusable structured data and make scholarly collaboration more transparent. Although pyThoth focuses on Ancient Egyptian, the same design pattern could be useful for other low-resource historical scripts.

Acknowledgments

The work was supported by a grant, provided by the Ministry of Economic Development of the Russian Federation (agreement dated June 20, 2025 No. 139-15-2025-011, identifier 000000C313925P4G0002).

References

[Barucci *et al.*, 2021] Andrea Barucci, Costanza Cucci, Massimiliano Franci, Marco Loschiavo, and Fabrizio Argenti. A deep learning approach to ancient egyptian hieroglyphs classification. *IEEE Access*, 9:123438–123447, 2021.

[De Cao *et al.*, 2024] Mattia De Cao, Nicola De Cao, Angelo Colonna, and Alessandro Lenci. Deep learning meets egyptology: a hieroglyphic transformer for translating ancient egyptian. In *Proceedings of the 1st Workshop on Machine Learning for Ancient Languages (ML4AL 2024)*, pages 71–86, 2024.

[Elnabawy *et al.*, 2018] Reham Elnabawy, Rimon Elias, and Mohammed Salem. Image based hieroglyphic character recognition. In *2018 14th International Conference on Signal-Image Technology and Internet-Based Systems (SITIS)*, pages 32–39, 2018.

[Franken and van Gemert, 2013] Morris Franken and Jan C. van Gemert. Automatic egyptian hieroglyph recognition by retrieving images as texts. In *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, page 765–768, New York, NY, USA, 2013. Association for Computing Machinery.

[Fuentes-Ferrer *et al.*, 2025] Raúl Fuentes-Ferrer, Jaime Duque-Domingo, and Pedro Javier Herrera. Recognition of egyptian hieroglyphic texts through focused generic segmentation and cross-validation voting. *Applied Soft Computing*, 171:112793, 2025.

[Golyadkin *et al.*, 2025a] Maksim Golyadkin, Innokentiy Humonen, Yanis Plevokas, Ekaterina Bureeva, Ekaterina Alexandrova, and Ilya Makarov. Automatic interpretation of ancient egyptian texts for education and research. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Posters*, SIGGRAPH Posters '25, New York, NY, USA, 2025. Association for Computing Machinery.

[Golyadkin *et al.*, 2025b] Maksim Golyadkin, Innokentiy Humonen, Valeria Rubanova, Danil Kalin, Ianis Plevokas, Dmitry Nikolotov, Aleksandr Utkov, Nikita Sidelnikov, Petr Ivanov, Ekaterina Bureeva, Ekaterina Alexandrova, and Ilya Makarov. Mummy: Multimodal dataset supporting vlm-based egyptology research assistant. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 12875–12881, 2025.

[Golyadkin *et al.*, 2025c] Maksim Golyadkin, Valeria Rubanova, Aleksandr Utkov, Dmitry Nikolotov, and Ilya Makarov. Meh: A multi-style dataset and toolkit for advancing egyptian hieroglyph recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 24488–24496, 2025.

[Guidi *et al.*, 2023] Tommaso Guidi, Lorenzo Python, Matteo Forasassi, Costanza Cucci, Massimiliano Franci, Fabrizio Argenti, and Andrea Barucci. Egyptian hieroglyphs segmentation with convolutional neural networks. *Algorithms*, 16(2):79, 2023.

[Humonen *et al.*, 2026] Innokentiy Humonen, Maksim Golyadkin, Valeria Rubanova, Ekaterina Alexandrova, Alexandre A Loktionov, and Ilya Makarov. Human-in-the-loop Egyptology: A system for ancient Egyptian text study. In *Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems*, CHI EA '26, pages 1–5, New York, NY, USA, 2026. Association for Computing Machinery.

[Kim *et al.*, 2022] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. OCR-free Document Understanding Transformer. In *European Conference on Computer Vision*, pages 498–517. Springer, 2022.

[Li *et al.*, 2023] Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. Trocr: Transformer-based optical character recognition with pre-trained models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):13094–13102, Jun. 2023.

[Miyagawa, 2025] So Miyagawa. RAG-enhanced neural machine translation of Ancient Egyptian text: A case study of THOTH AI. In Mika Hämäläinen, Emily Öhman, Yuri Bizzoni, So Miyagawa, and Khalid Alnajjar, editors, *Proceedings of the 5th International Conference on Natural Language Processing for Digital Humanities*, pages 33–40, Albuquerque, USA, May 2025. Association for Computational Linguistics.

[Ranathunga *et al.*, 2023] Surangika Ranathunga, En-Shiun Annie Lee, Marjana Prifti Skenduli, Ravi Shekhar, Mehreen Alam, and Rishemjit Kaur. Neural machine translation for low-resource languages: A survey. *ACM Computing Surveys*, 55(11):229:1–229:37, 2023.

[Rosmord,] Rosmord. Jsesh: An open-source hieroglyphic text editor. https://github.com/rosmord/jsesh. Accessed: March 27, 2025.

[Sobhy *et al.*, 2023] Asmaa Sobhy, Mahmoud Helmy, Michael Khalil, Sarah Elmasry, Youtham Boules, and Nermin Negied. An ai based automatic translator for ancient hieroglyphic language—from scanned images to english text. *IEEE Access*, 11:38796–38804, 2023.

[Tkachenko *et al.*, 2020 2025] Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. Label Studio: Data labeling software, 2020-2025. Open source software available from https://github.com/HumanSignal/label-studio.

[Wei *et al.*, 2024] Haoran Wei, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, Chunrui Han, and Xiangyu Zhang. General ocr theory: Towards ocr-2.0 via a unified end-to-end model, 2024.

[Wiesenbach and Riezler, 2019] Philipp Wiesenbach and Stefan Riezler. Multi-task modeling of phonographic languages: Translating middle egyptian hieroglyphs. *Proceedings of the International Workshop on Spoken Language Translation*, 2019.