EUROGRAPHICS Workshop on Graphics and Cultural Heritage (2024)
M. Corsini, D. Ferdani, A. Kuijper, and H. Kutlu (Editors)

Unsupervised Detection and Localization of Egyptian Hieroglyphs

P. Lion[1], E. Trunz[1] and R. Klein[1]

[1]University of Bonn, Germany

Abstract

The extensive variability in hieroglyph forms, coupled with erosion, fading, damage, and lighting effects, makes hieroglyphic script highly complex and difficult to segment. This complexity, along with the scarcity of labeled data, poses challenges for traditional supervised learning methods. In this paper, we present a novel unsupervised approach for detecting and localizing Egyptian hieroglyphs in images. Our method employs classical computer vision algorithms to generate pseudo-labels, which are then used to train a Faster R-CNN model. Augmented by post-processing techniques, our approach achieves detection results comparable to that of previous supervised methods for hieroglyph segmentation. Evaluated on unseen backgrounds, it demonstrates significant potential for advancing research in Egyptian culture and history.

CCS Concepts

• ***Computing methodologies*** → ***Object detection; Computer vision;*** • ***Applied computing*** → ***Arts and humanities;*** *Archaeology;*

1. Introduction

Accurate detection and localization of hieroglyphs in Egyptian texts can greatly enrich the study of Egyptian culture and history. Automated methods can significantly aid this research by providing tools to efficiently analyze large amounts of hieroglyphic data, but deep neural networks, which are commonly used for such tasks, require large labeled datasets for effective training. Unfortunately, there are not enough labeled images of Egyptian hieroglyphs, necessitating unsupervised methods that can work without annotated data. Recent unsupervised and self-supervised neural approaches have shown promising results through transfer learning, where models are pre-trained on large datasets and then fine-tuned for specific tasks. However, these methods struggle with hieroglyphic text images, which differ significantly from typical image datasets.

To address this problem, we present an unsupervised method for detection and localization of hieroglyphs in images. We first apply image analysis algorithms like Hough transform and Canny edge detection to generate bounding boxes, which serve as pseudo-labels for training a Faster R-CNN model. Post-processing further enhances the accuracy and robustness of hieroglyph localization. We evaluate our approach on the Unas dataset [FvG13], the only publicly available dataset we could find, and show that it outperforms recent unsupervised method for hieroglyph detection [EES18]. Although our focus was on vertically arranged hieroglyphs on stone, we tested our resulting model on different unseen backgrounds. The results are comparable to a supervised method [GPF*23] where these backgrounds were included in training. We will publish the code online to support future research: https://github.com/tenosel/Unsupervised_hieroglyph_localization.

In summary, the key contributions of our work are:

- An unsupervised approach to the detection and localization of Egyptian hieroglyphs in images, effectively addressing the scarcity of labeled data for Egyptian hieroglyphs.
- Significantly improved localization accuracy and robustness by combining classical computer vision methods with a Faster R-CNN and post-processing techniques, achieving results comparable to supervised methods for hieroglyph segmentation.
- Robust inference across various backgrounds, demonstrating potential for generalization.

2. Background and related work

Egyptian hieroglyphs are linguistic signs used in ancient Egypt, consisting of 763 ideograms categorized into 26 groups, each identified by a Gardiner code [Gar57]. Hieroglyphs can be arranged vertically or horizontally, often divided by lines into rows or columns, and may appear in cartouches that enclose multiple hieroglyphs to denote names of rulers or minor gods. Hieroglyphs were inscribed on diverse materials (backgrounds) such as papyrus or wood, and often carved into stone walls of temples, tombs and pyramids, typically using limestone or sandstone. This variety reflects the significant and multifaceted role of hieroglyphic writing in ancient Egyptian culture for religious, administrative and decorative purposes.

Hieroglyph detection and localization. Several studies use classical computer vision techniques to detect and localize hieroglyphs [FvG13, DDHVC17, EES18]. The closest work to ours is by

DOI: 10.2312/gch.20241259

Elnabawy et al. [EES18]. Their method converts images to edge maps, slices them using the Hough transform, applies Connected Component Labeling to obtain foreground objects, and extracts bounding boxes. Small images are removed in post-processing. We use a similar initial pipeline for generating pseudo-labels but improve results significantly by training a deep neural network and extending post-processing, as detailed in Section 4.

Apart from these unsupervised methods there are several approaches that involve neural networks and labeled data for hieroglyph segmentation [GPF*23] and classification [BCF*21]. Sobhy et al. [SHK*23] use R-CNN for localization and a Siamese network enhanced with a language model for classification. Guidi et al. [GPF*23] focus on segmenting hieroglyphs across varied backgrounds using Mask R-CNN. Although we lacked a similar dataset, our method yields comparable results on unseen backgrounds.

Unsupervised object detection and localization. In addition to methods for hieroglyph analysis, other unsupervised object detection techniques include vision transformers for foreground detection [SPV*21] and segmentation [KMR*23], saliency detection using attention mechanisms [GSPK20], and co-saliency detection to find common objects across images [FLL*22]. However, these approaches are often designed for typical image types found in large datasets and may not perform well on hieroglyphic texts without specific labeled fine-tuning.

3. Glyph detection

The goal of this work is to localize as many hieroglyphs as possible in a given image using only unsupervised methods. For network training, input images must be 224x224 pixels. Since most hieroglyphic scripts are on larger surfaces and the corresponding images are usually much larger than 224x224 pixels, we divide the input image into overlapping 224x224 pixel tiles, with a 50-pixel step size for training and a 150-pixel step size for testing.

The main steps of our approach are as follows: 1) Connected Component Labeling creates bounding boxes for the foreground objects in each 224x224 tile. 2) These boxes serve as pseudo-labels for training of a pre-trained Faster R-CNN. 3) Inference on unseen data is performed on 224x224 images. Predictions are reassembled into the full image, and global boxes undergo post-processing to produce the final bounding boxes of detected and localized hieroglyphs. 4) Optionally, these final boxes can be used as new pseudo-labels for further refinement. Each step is detailed below.

3.1. Automatic pseudo-labeling

To create bounding boxes, we exploit two properties of hieroglyphs: they clearly stand out from the background, allowing separation into foreground (hieroglyphs) and background, and they do not overlap, so touching objects usually belong to the same hieroglyph. However, factors like photography, decay, and vandalism introduce noise. Connected Component Labeling (CCL) [RP66] is well-suited for these properties. First, images are binarized using Otsu's method [O*75], which automatically determines the threshold. CCL then segments the image, and bounding boxes are extracted from the segmentation masks. However, lighting or other factors may cause glyphs to be split into multiple boxes. To address this, we enhance image details using the Canny edge detector [Can86] and combine the edges with the original image. Since hieroglyphs are often close together, the separating vertical lines can worsen the results. Therefore, we remove vertical lines using the Hough transform before applying edge detection and binarization. Not all objects detected by CCL are accepted. We filter out noise by setting thresholds for bounding box dimensions (width and height between 5 and 175 pixels) and area (between 100 and 45,000 pixels). Figure 1 illustrates this pseudo-label generation process.

Figure 1: *Generation of pseudo-labels. From left to right: Input image; Vertical lines removed; Edges enhanced; Otsu's binarization; CCL applied; Blue boxes are kept as pseudo-labels, red ones are removed.*

3.2. Glyph detection training

The bounding boxes from the 224x224 tiles in the previous step serve as pseudo-labels to train a Faster R-CNN [RHGS15] model. We used the implementation from Detectron2 [WKM*19] with ResNet50 as the backbone and DINO weights [CTM*21] as pre-trained weights. Training involved stochastic gradient descent with a 0.02 learning rate, running for 20,000 iterations with a batch size of 8. To enhance robustness, we applied data augmentation techniques like gamma correction, contrast change, and sharpness/blur adjustment during training.

3.3. Post-processing

For inference on large test images, the images are divided into overlapping 224x224 tiles with a 150-pixel step size. Detected bounding boxes from the tiles are then combined to reconstruct the full image. Post-processing involves three main steps: cleaning up predictions by merging boxes that likely represent the same hieroglyph, isolating individual characters in boxes containing multiple hieroglyphs, and identifying cartouches to recognize the hieroglyphs inside.

Removing redundant boxes. After merging the image tiles, overlapping prediction boxes may detect the same object. To reduce redundancy, we use the Intersection over Union (IoU) measure. If two boxes have an IoU over 0.9, they are merged, with new coordinates set by the smallest and largest x and y values. Additionally, we merge boxes when one is entirely or mostly within another. To handle cases where IoU alone isn't sufficient, we calculate the overlap ratio relative to the smaller box, removing the inner box if the overlap exceeds 0.6. Hieroglyphs are assumed not to overlap, except for cartouches, which are addressed separately in the third post-processing step.

Separating multiple hieroglyphs. The second post-processing step identifies bounding boxes containing multiple hieroglyphs and separates them. Hieroglyphs are often close and may be recognized

as a single object. To address this, boxes are processed iteratively. In the first iteration, CCL is applied to detect single objects. Boxes with only one object are stored as *Group 1*, while others go to *Group 2*. Next, *Group 2* boxes are compared to *Group 1* using the Structural Similarity Index (SSIM). If a match is found, the box is added to *Group 1*. This process is repeated, reducing *Group 2* items. Remaining *Group 2* boxes likely contain multiple glyphs. For these, CCL is applied after trinarization using k-means clustering, which separates foreground, background, and unclear regions. This preserves details missed by Otsu's method to handle cases where glyphs may be split due to damage.

The separated hieroglyphs are then processed with vertical erosion and horizontal dilation. Erosion removes pixels lengthwise based on vertical neighbors, while dilation fills gaps horizontally, which helps merge connected parts of the same hieroglyph. These processes are designed for vertically arranged hieroglyphs, minimizing issues with side-by-side placements. Finally, CCL is applied to extract individual bounding boxes, which are added to *Group 1*. All boxes are cleaned again using IoU and overlap thresholding at 0.9 to merge highly overlapping boxes.

Recognition of cartouches. To identify cartouches, we consider any box with an area at least 95% of the largest box as a cartouche. We then use CCL to extract hieroglyphs within it. Since hieroglyphs are close to the cartouche edges, edge detection and k-means clustering are ineffective. Instead, we binarize the image with Otsu's method to remove cartouche edges.

After binarization, we create and smooth histograms of foreground pixels by row and column. Cartouche images typically have many foreground pixels at the edges, which we use to find the boundaries between the cartouche and its hieroglyphs. We identify these boundaries by locating the second and penultimate low points in the histograms, adjust for the cartouche's bottom bar, and convert the outer areas to background. The modified image is then trinarized using k-means clustering, followed by erosion, dilation and CCL. Only boxes with a height up to 70% of the cartouche height are kept to avoid misidentifying borders as hieroglyphs.

3.4. Second iteration

The bounding boxes from post-processing can serve as new pseudo-labels for an additional 10,000 training iterations from the last checkpoint. In our experiments, this often improved precision but sometimes reduced recall, as shown in Table 1. An example of detection after second iteration is illustrated in Figure 3.

4. Experiments

Data. We used the Unas dataset [FvG13] with labeled bounding boxes for performance evaluation. This dataset includes 10 images from the Unas pyramid, each with dimensions varying between 1071x1544 and 1160x1683, featuring 4210 characters across 172 hieroglyphic classes. While small compared to the 763 total hieroglyphic classes, it is useful for evaluating our method.

Evaluation. To test our approach on each large image separately, we used the remaining nine images as an unlabeled training set and generated pseudo-labels as described in Section 3.1. We compared our method with Elnabawy et al.'s approach [EES18] on the same images. Table 1 presents the precision and recall results for each image. In addition to the quantitative evaluation on the labeled dataset, we investigated whether our pseudo-supervised model, trained on one background type, could detect hieroglyphs on different backgrounds. Inference on images from Guidi et al. [GPF*23] showed that our method performs comparably, and even better on damaged hieroglyphs, as seen in Figure 2. Figure 4 highlights detection on another unseen background.

Figure 2: *Detection on a difficult image taken from [GPF*23]. Left: Segmentation results from [GPF*23]; Middle: Input; Right: Our detection with the same model as in Figure 4. The model of [GPF*23] struggles to detect hieroglyphs and instead detects some artifacts, whereas our approach detects almost everything correctly.*

Limitations. Our method, while label-free, depends on thresholds that may need adjustment based on data. Misdetections can occur, such as detecting closely spaced hieroglyphs as one or vice versa, and similar issues affect cartouche processing.

5. Conclusions

We presented an unsupervised approach for detecting Egyptian hieroglyphs by combining classical computer vision techniques with a Faster R-CNN model and post-processing enhancements. Our method achieves detection performance comparable to supervised methods and shows robustness on unseen backgrounds, advancing automated hieroglyph recognition. This approach could significantly impact Egyptology and historical research. Our work focuses on hieroglyph detection and localization, with future research planned for unsupervised classification and extension to other ancient scripts, like Mayan hieroglyphs.

Acknowledgements. This work was funded by Federal Ministry of Education and Research within the project BNTrAinee (funding code 16DHBK1022).

References

[BCF*21] Barucci A., Cucci C., Franci M., Loschiavo M., Argenti F.: A deep learning approach to ancient egyptian hieroglyphs classification. *IEEE Access 9* (2021), 123438–123447. 2

[Can86] Canny J.: A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8*, 6 (1986), 679–698. 2

[CTM*21] Caron M., Touvron H., Misra I., Jégou H., Mairal J., Bojanowski P., Joulin A.: Emerging properties in self-supervised vision transformers, 2021. arXiv:2104.14294. 2

Table 1: *Precision and Recall for each large image of the Unas dataset [FvG13]. Top row: Elnabawy et al. [EES18]; Second row: Our pseudo-labels on 224x224 tiles; Third row: Our detection after initial training and post-processing; Bottom row: Our detection after a second iteration of training and post-processing. Best results are in bold.*

	3	5	7	9	20	21	22	23	39	41	avg
[EES18]	0.34; 0.39	0.76; 0.73	0.35; 0.39	0.39; 0.44	0.47; 0.53	0.46; 0.62	0.65; 0.65	0.37; 0.47	0.38; 0.50	0.58; 0.57	0.48; 0.53
ours pseudo	0.60; 0.52	0.67; 0.67	0.59; 0.55	0.55; 0.51	0.72; 0.67	0.54; 0.61	0.67; 0.57	0.51; 0.48	0.54; 0.64	0.49; 0.54	0.59; 0.58
ours 1st	0.81; 0.70	0.83; **0.84**	0.81; 0.77	0.75; 0.79	0.86; 0.80	0.76; 0.73	0.71; 0.73	0.79; 0.77	0.88; **0.89**	0.88; **0.87**	0.81; 0.79
ours 2nd	**0.87**; **0.78**	**0.89**; 0.81	**0.88**; **0.80**	**0.89**; **0.82**	**0.91**; **0.83**	**0.78**; **0.79**	**0.85**; **0.84**	**0.85**; **0.84**	**0.91**; **0.89**	**0.93**; 0.84	**0.88**; **0.82**

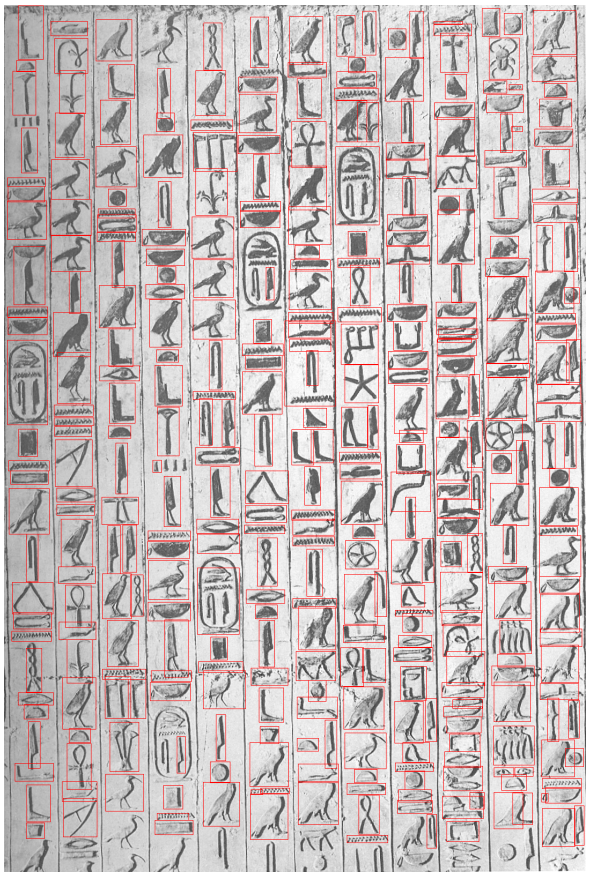

Figure 3: *Results of detection steps on the image Nr. 41 from Unas database after second iteration of training and post-processing*

[DDHVC17] DUQUE-DOMINGO J., HERRERA P. J., VALERO E., CERRADA C.: Deciphering egyptian hieroglyphs: Towards a new strategy for navigation in museums. *Sensors 17*, 3 (2017). 1

[EES18] ELNABAWY R., ELIAS R., SALEM M.: Image based hieroglyphic character recognition. In *2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)* (2018), pp. 32–39. 1, 2, 3, 4

[FLL*22] FAN D.-P., LI T., LIN Z., JI G.-P., ZHANG D., CHENG M.-M., FU H., SHEN J.: Re-thinking co-salient object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 44*, 8 (2022), 4339–4354. 2

[FvG13] FRANKEN M., VAN GEMERT J.: Automatic egyptian hieroglyph recognition by retrieving images as texts. *MM 2013 - Proceedings of the 2013 ACM Multimedia Conference* (10 2013). 1, 3, 4

[Gar57] GARDINER A.: *Egyptian Grammar*. Griffith Institute, Ashmolean Museum, Oxford, 1957. 1

[GPF*23] GUIDI T., PYTHON L., FORASASSI M., CUCCI C., FRANCI M., ARGENTI F., BARUCCI A.: Egyptian hieroglyphs segmentation with convolutional neural networks. *Algorithms 16*, 2 (2023). 1, 2, 3

[GSPK20] GUPTA A. K., SEAL A., PRASAD M., KHANNA P.: Salient object detection techniques in computer vision—a survey. *Entropy 22*, 10 (2020). 2

[KMR*23] KIRILLOV A., MINTUN E., RAVI N., MAO H., ROLLAND C., GUSTAFSON L., XIAO T., WHITEHEAD S., BERG A. C., LO W.-Y., DOLLÁR P., GIRSHICK R.: Segment anything, 2023. arXiv:2304.02643. 2

[O*75] OTSU N., ET AL.: A threshold selection method from gray-level histograms. *Automatica 11*, 285-296 (1975), 23–27. 2

[RHGS15] REN S., HE K., GIRSHICK R., SUN J.: Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems 28* (2015). 2

[RP66] ROSENFELD A., PFALTZ J. L.: Sequential operations in digital picture processing. *Journal of the ACM 13*, 4 (1966), 471–494. 2

[SHK*23] SOBHY A., HELMY M., KHALIL M., ELMASRY S., BOULES Y., NEGIED N.: An ai based automatic translator for ancient hieroglyphic language—from scanned images to english text. *IEEE Access 11* (2023), 38796–38804. 2

[SPV*21] SIMÉONI O., PUY G., VO H. V., ROBURIN S., GIDARIS S., BURSUC A., PÉREZ P., MARLET R., PONCE J.: Localizing objects with self-supervised transformers and no labels. *CoRR abs/2109.14279* (2021). arXiv:2109.14279. 2

[WKM*19] WU Y., KIRILLOV A., MASSA F., LO W.-Y., GIRSHICK R.: Detectron2, 2019. URL: https://github.com/facebookresearch/detectron2. 2

Figure 4: *Example inference with the model after second iteration of training and post-processing, which were trained on Unas dataset with our pseudo-labels.*