

A Hierarchical Dataset and Deep Learning Evaluation for Ancient Arabic Manuscript Segmentation

Aya Bouchantouf^{ID*}, Nidal Lamghari^{ID}

LIPIM Laboratory, ENSA Khouribga, Sultan Moulay Slimane University, Khouribga, Morocco

Abstract The complexity associated with the recognition of ancient Arabic manuscripts lies in the various styles used in calligraphy, the use of ligatures, and the extent of degradation in the documents. This investigation focuses on the implementation of hierarchical segmentation at the line, word, pseudo-word, and character levels. For this purpose, a domain-specific dataset was developed using a variety of sources and was manually annotated to incorporate the various styles and configurations used in the documents. Normalization, denoising, and enhancement techniques were used for preprocessing the dataset. Three architectures of deep learning models, YOLOv8, Faster R-CNN, and Mask R-CNN, were used to test the dataset developed in this case. From the results obtained from this experiment, it was evident that YOLOv8 was effective at the line level, while Mask R-CNN was effective at the character level. However, Faster R-CNN was competitive at the word level, but cursive scripts presented a problem for this model. This investigation's contribution lies in the development of a hierarchical dataset and a comprehensive benchmarking test, which together contribute to the development of sustainable OCR technology and the preservation of Arabic manuscript heritage. To provide a more comprehensive evaluation, we additionally benchmark two segmentation-specific baselines (U-Net and DeepLabV3+) at the character level, and report YOLOv8 metrics as mean $\pm$ standard deviation over three random seeds. A striking finding is that architecture choice can matter more than dataset size at fine granularity levels: Faster R-CNN achieves 70.2 AP50 on pseudo-words where YOLOv8 reaches only 30.9 ± 23.1 , due to input resolution constraints. We position this work as a dataset-and-benchmark contribution: the primary deliverable is a publicly released, hierarchically annotated corpus (CC BY 4.0), accompanied by a reproducible evaluation protocol.

Keywords YOLOv8, Faster R-CNN, Mask R-CNN, OCR, hierarchical segmentation, dataset, U-Net, DeepLabV3+, benchmark.

DOI: 10.19139/soic-2310-5070-4121

1. Introduction

Handwritten Arabic recognition has gained increasing attention because of its importance in digitizing cultural heritage documents, enabling document retrieval, and advancing text analysis. Arabic manuscripts constitute invaluable resources in history, science, and literature, yet they pose significant challenges for automatic recognition systems [1, 8].

The primary challenges are its intricate script: cursive writing, changing calligraphic styles, diacritics, and overlapped shapes [4, 29]. Moreover, ancient documents generally lack clearness with faded ink and deteriorating papers and non-standardized layouts and hence segmentation and recognition are further complicated [14, 17].

A number of resources has been introduced to Arabic handwriting recognition tasks like IFN/ENIT [2], DBAHCL [3], KHATT [5, 13], and Muharaf [15, 24] lately. The latter ones are predominant with respect to using modern handwriting samples. Datasets devoted to ancient or historical Arabic scripts are still scarce and limited in scope and annotation richness [14, 18].

*Correspondence to: Aya Bouchantouf (Email: aya.bouchantouf@usms.ac.ma). LIPIM Laboratory, ENSA Khouribga, Sultan Moulay Slimane University, Khouribga, Morocco.

Arabic handwriting recognition has benefited greatly from recent breakthroughs in deep learning and especially from convolutional and recurrent neural networks [7, 10]. Higher-level architectures like CNN-RNN hybrids [9] and Transformers [11] are promising but need vast annotated sets. Filling this gap between resources at hand and state-of-the-art techniques is key to creating reliable OCR tools for Arabic manuscripts.

The present paper makes three contributions:

- We present here a hierarchical set of annotated ancient Arabic texts at line & word & pseudo-word & character levels obtained by trusted resources like Gallica (BNF) [19], BNRN [22], and Qatar Digital Library [20]. The dataset is publicly released under CC BY 4.0 on Zenodo with a persistent DOI.
- We compare three recent models (YOLOv8 vs Faster R-CNN vs Mask R-CNN) for segmentation at various levels. We additionally benchmark two segmentation-specific baselines (U-Net and DeepLabV3+) at the character level, the only level with polygon annotations.
- We provide remarks regarding the benefits and drawbacks of each framework and determine areas for future research like transformer-based architecture and hybrid deep learning models [11, 12]. YOLOv8 results are reported as mean $\pm$ std over three random seeds to assess statistical stability.

We emphasise that the contribution of this paper is a *dataset and reproducible benchmark*, not a new network architecture. The challenge in ancient Arabic manuscript analysis is not primarily the absence of capable models but the absence of an openly available, multi-level, quality-controlled corpus on which competing models can be compared under identical conditions. This paper is organized as follows: Section 2 presents a review of related works on Arabic handwriting recognition and existing datasets. Section 3 describes the dataset construction and preprocessing steps. Section 4 introduces the proposed models and experimental setup. Section 5 presents the experimental results, and Section 6 concludes the paper with directions for future work.

2. Related works

The field of Arabic handwritten text recognition has undergone substantial transformation with the advent of deep learning. Existing surveys [1, 8, 12] document the shift from traditional methodologies toward CNN-, RNN-, and more recently Transformer-based architectures. Initial approaches were largely built upon Hidden Markov Models (HMMs) [4] or handcrafted feature techniques like PHOG descriptors combined with classifiers [7]; however, these methods proved inadequate when confronted with document degradation and stylistic inconsistencies.

The availability of annotated datasets has been instrumental in advancing this field. Resources such as IFN/ENIT [2], DBAHCL [3], and KHATT [5, 13] enable recognition at word and line levels. Additional contributions include MOJ-DB [17], the Arabic Text Line Dataset [18], and specialized collections targeting ancient manuscripts [14, 16]. Despite this diversity, the majority of these datasets are oriented toward modern handwriting, leaving historical documents comparatively underrepresented in current research.

The introduction of deep architectures—spanning CNNs, RNNs, and hybrid Transformer-based systems—has yielded notable gains in recognition accuracy [9, 10, 11]. Segmentation-free frameworks such as KHATT [13] further illustrate the promise of end-to-end recognition pipelines. Despite these advances, ancient manuscripts continue to pose considerable difficulties owing to their calligraphic diversity, physical degradation, irregular spacing, and inconsistent diacritical marks [34]. Notably, our prior study [34] (Bouchantouf and Lamghari) examined the structural intricacies of Arabic historical manuscripts, underscoring the necessity for purpose-built datasets and adapted modeling strategies.

Recent advances in historical line segmentation have demonstrated strong line-level IoU performance, even under few-shot learning conditions—as evidenced by methods such as FEST, UNet++ combined with topology-aware loss functions, FCN-derived line masks, and seam-carving integrated with semantic segmentation. Bounding-box-based detection, however, remains comparatively underutilized despite its potential merits. Additionally, UTLS demonstrates competitive unsupervised performance on Arabic document pages. For densely packed text regions, instance-aware refinements to Mask R-CNN could offer performance benefits, though this direction has received limited attention. Meanwhile, segmentation-free and Transformer-based recognition

approaches (e.g., HATFormer) continue to deliver robust results, reinforcing the observed correlation between line segmentation quality and downstream Character Error Rate (CER) outcomes.

Overall, prior studies highlight three major limitations. First, the availability of annotated datasets for ancient Arabic handwriting remains limited [14, 15, 16]. Second, hierarchical segmentation approaches (line, word, and character levels) have been insufficiently explored. Third, there is a lack of Transformer-based or hybrid architectures capable of effectively handling stylistic variability and document degradation [11, 12].

These limitations motivate our work. We introduce a new hierarchical dataset for ancient Arabic manuscripts and establish baseline results using YOLOv8, Faster R-CNN, and Mask R-CNN for multi-level segmentation.

3. Methodology

The system under consideration consists of the following main elements: the acquisition of the dataset in order to have diversified handwriting samples, preprocessing of the data in order to increase its quality, annotation to have valid information, and the choice of the model that will suit the deep learning model for the recognition of ancient Arabic handwritings.

3.1. Dataset Collection

Historical Arabic manuscripts were sourced from multiple digital repositories and archival institutions, namely Gallica (BNF) [19], the National Library of the Kingdom of Morocco (BNRM) [22], the Qatar Digital Library [20], and the King Abdulaziz Public Library [23]. These repositories encompass a diverse range of manuscript types—religious, scientific, and literary—rendered in various calligraphic traditions, including Naskh, Maghribi, and Kufic styles. Beyond these archival sources, publicly available scholarly datasets were incorporated, such as MuHARaf [15, 24], MOJ-DB [17], along with specialized datasets specifically dedicated to ancient handwriting [14, 16, 18]. This integration further diversifies our corpus by contributing additional samples of degraded and historically significant Arabic script.

3.2. Preprocessing

Preprocessing represents an important stage in the process of analysis during processing of outdated manuscripts. The objective of the preprocessing stage is to standardize the data, minimize visual deterioration, and improve readability of the data at the next stage of segmentation. An automated pre-processing pipeline has been created using Python, and it includes the following stages:

3.2.1. Normalization

The first step in the pre-processing stage is all about normalizing the images. This ensures that the input data is consistent for the image segmentation and recognition models. All the images are resized to a standard size, 1024 x 1024 pixels. This ensures that the dimensions of the images are standardized. The manuscripts, irrespective of their original sizes, are treated equally. The next step in the pre-processing stage is the conversion of the images into grayscale. This helps in reducing the complexity of the data while keeping the important characteristics of the writing strokes. The images are converted into grayscale, ensuring that the models are steered towards the textual structure and not any other color-related differences.

3.2.2. Cleaning

After that, the cleaning process is implemented to improve the readability of the image. At the same time, the image should be free from the presence of noise. In this case, the noise reduction is carried out using the fastNlMeansDenoising algorithm. This algorithm helps to improve the preservation of the writing details. In other cases, other algorithms such as the Gaussian algorithm, medial algorithm, and bilateral algorithm do not improve the preservation of the writing details as much as the non-local denoising algorithm does. After that, the image is subjected to the local histogram equalization to improve the image contrast. This is particularly important to account for the changes that occur in the lighting of the image. As a result, the image becomes darker, thus

becoming easier for analysis. Finally, the image is subjected to global standardization using the CV2.normalize function to improve the distinction of the writing.

3.2.3. *Improvement*

The last stage of the pre-treatment phase lays on the improvement of the characters definition in the writing so it can be made even more readable and usable for segmentation algorithms. A Lightweight sharpening filter is applied to improve the strokes of the writing while eliminating unwanted visual artifacts. This is helpful in the enhancement of characters definition and ligatures, which is a crucial requirement for segmentation at the lines, words, and character levels. The final result of the pre-processing is the sharpness and contrast of the images.

3.3. *Segmentation and Manual Annotation*

3.3.1. *Definition of hierarchical levels*

Arabic is written cursively, and letters within a word connect only when their shapes permit. We define the four levels precisely:

- **Line:** a maximal horizontal run of text forming one baseline unit of the page.
- **Word:** a lexical unit delimited by inter-word white space.
- **Pseudo-word (sub-word / PAW, *Piece of Arabic Word*):** a maximal connected component of ink within a word, bounded by non-connecting letters. A word contains one or more pseudo-words.
- **Character:** a single Arabic grapheme in its positional form (isolated, initial, medial, or final).

Segmentation is a fundamental step in the preparation of our dataset. It aims to break down Arabic manuscripts into different usable units (lines, words, pseudo-words, characters) in order to allow the training of models adapted to each level of analysis. In our work, segmentation was carried out according to four hierarchical levels.

3.3.2. *Manual annotation*

The creation of a dataset dedicated to ancient Arabic manuscripts relies on manual annotation, which is both crucial and unavoidable. Due to the complexity of these documents, which have a great deal of calligraphic variability, frequent ligatures and deterioration related to time, it is impossible to be satisfied with automatic annotations to get reliable data. Human intervention makes it possible to define with precision the contours of lines, words, pseudo-words and characters, thus guaranteeing the quality of the references used for supervised learning of recognition and segmentation.

3.3.3. *Annotation Protocol and Quality Control*

To complement the manual annotation process described above, a structured annotation protocol was established to ensure the reliability and consistency of the dataset. All annotations were performed by the first author with experience in Arabic manuscript analysis. All annotations were subsequently reviewed and verified by the second author, an expert in Arabic handwriting analysis. Although relying on a single annotator may introduce a certain degree of subjectivity, several precautions were taken to guarantee consistency and accuracy throughout the dataset construction.

Before entering the annotation phase, a well-defined and detailed set of guidelines was developed to standardize this phase. For example, in case of text line annotation, each text line was marked using the smallest region possible, even when there was overlap in text, thus maintaining the flow of text. The same goes for the segmentation of words and pseudo-words, where segmentation was done using spacing and morphology rather than visible gaps. This was all done considering the features of the Arabic script. Also, special care was taken for connected characters, in which the character segments remained the same semantically. The annotation of degraded regions was conservative in nature, where consistency was more valued than over-segmentation. To further improve the quality of the annotations, a multi-stage verification was developed. Initially, annotation was performed, followed by a review phase in which all the annotations were reviewed and refined. Furthermore, a validation phase was conducted on all the annotations, specifically on those that were ambiguous, to ensure a high level of precision and coherence. To evaluate consistency, a portion of the dataset was re-annotated after a certain period of time, and

then the results were compared in order to evaluate intra-annotator consistency. Although inter-annotator agreement metrics such as Cohen's kappa could not be computed due to the single-annotator setting, the repeated verification and validation strategy provides strong evidence of the reliability of the annotation process for supervised learning applications. Overall, this protocol ensures reproducibility and significantly reduces annotation bias despite the single-annotator setting.

3.3.4. *Choosing the Annotation Tool*

To carry out this annotation, we opted for the Roboflow platform, a tool widely recognized in the computer vision community for the creation and production of management of annotated datasets. Roboflow offers an intuitive interface that makes it easy to quickly manipulate and annotate images, which is especially useful for a large volume of ancient manuscripts. The platform also allows for direct export annotations in several standard formats such as YOLO, COCO and Pascal VOC domain, which ensures out-of-the-box compatibility with leading deep frameworks learning. In addition, Roboflow centralizes project management and enables collaboration between multiple annotators, while integrating data augmentation functions useful for enriching the dataset and improving the generalization of the models.

3.3.5. *Nature of Annotation Projects*

To capture all the levels of segmentation needed for analysis manuscripts, several annotation projects have been created on the Roboflow platform. Each project corresponds to a specific hierarchical level, ranging from the entire page to the individual typeface. For the line segmentation, we used detection objects with bounding boxes, because the rows have a relatively large structure that lends itself to this approach. For the segmentation of words, pseudo-words and characters, the complexity of the forms and the cursiveness of the Arabic script make it possible to use the segmentation instance with polygons is essential. This method makes it possible to accurately capture the irregular contours and complex ligatures that the bounding boxes cannot represent correctly.

Each project has been designed to support both polygonal and polygonal annotations. encompassing boxes, thus guaranteeing maximum flexibility for use with different models. The annotated data can thus be used by YOLOv8 to object detection, or by Mask R-CNN and Faster R-CNN for segmentation precise. This methodical approach makes it possible to build a rich, accurate dataset compatible with modern architectures for the recognition and segmentation of Arabic handwriting.

3.3.6. *Annotated corpus*

The annotated corpus has two main types of images that allow one to deal with both original manuscripts and artificially degraded ones. One type of images is original manuscripts, gathered from digital and archive services such as Gallica, Qatar Digital Library, National Library of the Kingdom of Morocco, etc. These images represent real conditions of historical documents, including their stylistic variety and degradation over time. These images are a basis for learning a model in a supervised manner. The second type of images is manuscripts that are artificially noisy, simulating Common alterations when scanning or dealing with old documents. The main degradations applied are:

- Gaussian noise to mimic the disturbances associated with digitization,
- artificial shadows to reproduce irregular lighting conditions,
- different types of blur (Gaussian, median, medium) to represent the loss of sharpness.

This dual approach makes it possible to diversify the dataset and improve the robustness of the models in the face of unexpected variations in manuscripts.

3.3.7. *Annotated Segmentation Levels*

The corpus was segmented according to four levels of hierarchy in order to Provide fine-grained annotations adapted to each type of model. The four levels are described in detail in what follows.

Line segmentation: Each of the lines is segmented to identify the sections that exhibit continuity of textual statements. This step is important because it enables the structuring of the page, which is a precursor to word

recognition, thus allowing the identification of lines even if there are overlaps and smudges, addresses the issue of varying distances between lines.

Word segmentation: Each word is segmented to enable proper word recognition. This phase is essential as it enables the addressing of word recognition, addresses the issue of word spacing, and allows the proper word-line correlation.

Segmentation into pseudo-words: Pseudo-words are sets of characters that are specific to the Arabic script of writing. This stage is significant because it enables the proper word recognition, addresses the issue of word spacing, and allows the proper correlation of words with lines.

Segmentation in characters: Segmentation of words into segments of characters is the most important and complex step, because it enables the proper word recognition, addresses the issue of cursivity, and allows proper annotations.

The annotated corpus thus benefits from the richness of the original manuscripts as well as the diversity of noisy data.

3.4. Final constitution of the dataset

The dataset formation is the final result regarding a series of consecutive operations: the acquisition of ancient Arabic manuscripts, their digitalization or recovery from digital sources, the addition of artificial noises, the multi-level annotation of the data, and increasing it at the end of the process. This section will describe the characteristics of the data used for training the segmentation models.

3.4.1. Format and structure

The dataset has been organized in a hierarchical manner in order to facilitate access to the different annotation levels:

- **Pages:** Complete images of the manuscripts.
- **Lines:** Textual segments extracted from the pages, annotated by bounding boxes (object detection).
- **Words and pseudo-words:** Finer textual units, annotated by polygons (instance segmentation).
- **Characters:** Basic elements, also annotated by polygons to manage the cursivity and ligatures of the Arabic script.

Each directory contains the images and their corresponding annotations, exported in standardised formats (COCO, YOLO, Pascal VOC), guaranteeing compatibility with several deep learning frameworks (YOLOv8, Mask R-CNN, U-Net, etc.). This hierarchical organization allows the dataset to be used for both object tasks instance detection that segmentation.

3.4.2. Annotation and Data Volume

All the images in the dataset have been annotated using the Roboflow platform. Based on the segmentation level, bounding boxes as well as polygon annotations have been utilized to effectively describe the images. To guarantee the reliability of the annotations, a systematic proofreading process was carried out in order to avoid any potential errors in the annotations.

To make the dataset robust and similar to real-life scenarios, artificial degradation along with data augmentation techniques have been utilized. In this context, artificial degradation techniques include the addition of gaussian noise, shadows, motion blur, as well as gaussian blur. On the other hand, data augmentation techniques have been carried out using the Roboflow platform and custom Python scripts. The techniques used in this context include rotation at various angles, brightness as well as contrast changes, horizontal/vertical flipping, random zooming and cropping.

The final dataset comprises a balanced number of original images as well as artificially degraded images. Moreover, the annotations are available in multiple standard formats, making it easier to use the dataset in conjunction with various deep learning frameworks. A comprehensive summary of the dataset at all the segmentation levels is provided in Table 1.

The distribution of the dataset across the training, validation, and the test set is provided in Table 2.

Table 1. Detailed dataset statistics per segmentation level (after data curation, see Section 3).

Segmentation Level	Number of Images	Number of Annotations
Lines	233	3,979
Words	785	9,558
Pseudo-words	166	2,560
Characters	283	5,953
Total	—	22,050

Table 2. Dataset split across training, validation, and test sets (page-level split after data curation). Numbers show images (annotations).

Segmentation Level	Train	Validation	Test
Lines	149 (2,534)	43 (738)	41 (707)
Words	530 (6,413)	114 (1,324)	141 (1,821)
Pseudo-words	85 (1,379)	48 (651)	33 (530)
Characters	207 (4,169)	46 (1,070)	30 (714)

3.4.3. Dataset Size

The proposed dataset is well-suited for training and testing segmentation models for ancient Arabic manuscripts. It is composed of a diverse collection of documents obtained from various sources. It is divided into training, validation, and test datasets with a ratio of 70%, 15%, and 15%, respectively. This distribution is performed at the page level to prevent data leakage between sets (see data curation section below).

3.4.4. Dataset curation and quality control

The HAAM dataset follows a cascaded annotation pipeline: lines are annotated on full manuscript pages (1024×1024 pixels), while words, pseudo-words, and characters are annotated on individual line-strip images extracted from those pages (typical dimensions: $\sim 600 \times 40$ pixels). During quality inspection, we discovered that the original Roboflow export had mixed full manuscript pages into the line-strip annotation levels. At the pseudo-word level, 19 out of 104 training images were full pages, carrying 84% of all training annotations. All images with aspect ratio < 3 were removed from the word, pseudo-word, and character levels. The phantom root category inserted by Roboflow (carrying no annotations) was also removed. Additionally, the train/validation/test split was revised to operate at the *page level* rather than the image level, ensuring that no page’s content appears in more than one split. Tables 1 and 2 report the final statistics after cleaning.

Annotation format. All four levels provide bounding box annotations in COCO format. The character level additionally includes polygon segmentation masks, enabling instance and semantic segmentation evaluation. The line, word, and pseudo-word levels provide bounding boxes only; consequently, Mask R-CNN, U-Net, and DeepLabV3+ are evaluated only at the character level.

In the proposed dataset, four levels of segmentation are implemented: line, word, pseudo-word, and character. As shown in Table 1, the cleaned dataset contains 3,979 line-level, 9,558 word-level, 2,560 pseudo-word-level, and 5,953 character-level annotations, for a total of 22,050 annotated instances across 154 unique manuscript pages. Additionally, the number of characters is 29 with 242 different shape variations. This is because the writing of the Arabic alphabet is complex and the shape of the characters changes according to their position in the word and the writing style.

The lower counts compared to the original Roboflow export (which reported 39,351 instances) are due to the removal of contaminating full-page images from the line-strip levels and the page-level re-splitting described above. In comparison with other well-known Arabic handwritten datasets like KHATT and IFN/ENIT datasets, the proposed dataset has more diversity in terms of hierarchy and multiple levels of annotation in the images. In addition, the annotation hierarchy ranges from the line level down to the character level. This diversity in the

hierarchy of the proposed dataset makes it more suitable for addressing the segmentation problems in the complex and degraded images of the historical documents.

3.4.5. Dataset characteristics

The dataset is characterized by several important features that make the dataset particularly relevant and challenging for the analysis of ancient Arabic handwriting. Firstly, the dataset is marked by a high level of diversity, given that the manuscripts contained in the dataset come from different geographical areas and time periods. Secondly, the dataset shows complexity, which is caused by several factors, including natural degradation factors such as noise resulting from scanning, ink smudges, folds, and holes. Finally, the dataset is recognized by a high level of granularity, because the annotations are carried out on four hierarchical levels.

3.5. Models Evaluated

After the dataset is built and manually labeled, we selected three models of hierarchical segmentation architectures that employ deep learning: YOLOv8, Faster R-CNN, and Mask R-CNN. Each of the models was categorized into a specific segmentation level according to the structural richness of the units of the manuscripts.

3.5.1. YOLOv8

YOLOv8 is a single-stage detector from Ultralytics that predicts bounding boxes and class information in one pass. The detection head of the YOLOv8 model is designed to optimize the following loss function:

$$\mathcal{L}_{YOLO} = \lambda_{cls} \mathcal{L}_{cls} + \lambda_{obj} \mathcal{L}_{obj} + \lambda_{box} \mathcal{L}_{box}, \quad (1)$$

where $\mathcal{L}_{cls}$ denotes classification loss, $\mathcal{L}_{obj}$ denotes objectness loss, and $\mathcal{L}_{box}$ denotes bounding box loss. YOLOv8 achieves a good balance between accuracy and speed, which makes it applicable for line detection. However, it does not work well for cursive characters.



Figure 1. General YOLOv8 pipeline for line and word-level segmentation.

Figure 1 depicts the pipeline of the proposed approach for Arabic manuscript line segmentation using the YOLOv8 model. The pipeline starts with the high-resolution scanned image of the Arabic manuscript. Then, line detection is performed using the YOLOv8 model. The model predicts the bounding box of the line region with confidence. The region of each line is accurately predicted by the YOLOv8 model. Finally, the detected lines are saved as individual images.

3.5.2. Faster R-CNN

Faster R-CNN, according to Ren et al., is a model that makes use of a Region Proposal Network (RPN) and has candidate regions refined with the aid of classification and regression heads. The loss function is defined as follows:

$$\mathcal{L}_{FRCNN} = \mathcal{L}_{cls}(p, p^*) + \lambda \mathcal{L}_{reg}(t, t^*), \quad (2)$$

Here, p and p^* represent the predicted and actual class, while t and t^* represent the predicted and actual coordinates. The model is robust for handling complex handwritten data like words and pseudo-words.

Figure 2 illustrates the Faster R-CNN framework employed for detecting Arabic words. The pipeline initiates with a CNN backbone (ResNet or VGG) responsible for feature extraction, followed by a Region Proposal Network (RPN) tasked with generating candidate regions. These regions are then aligned through ROI Pooling to enable subsequent classification and bounding box refinement. The resulting output consists of localized word detections accompanied by their respective confidence scores. The bottom portion of the figure displays sample detections applied to Arabic manuscripts, demonstrating the model's capacity to differentiate between adjacent words even under conditions of overlap or textual degradation.

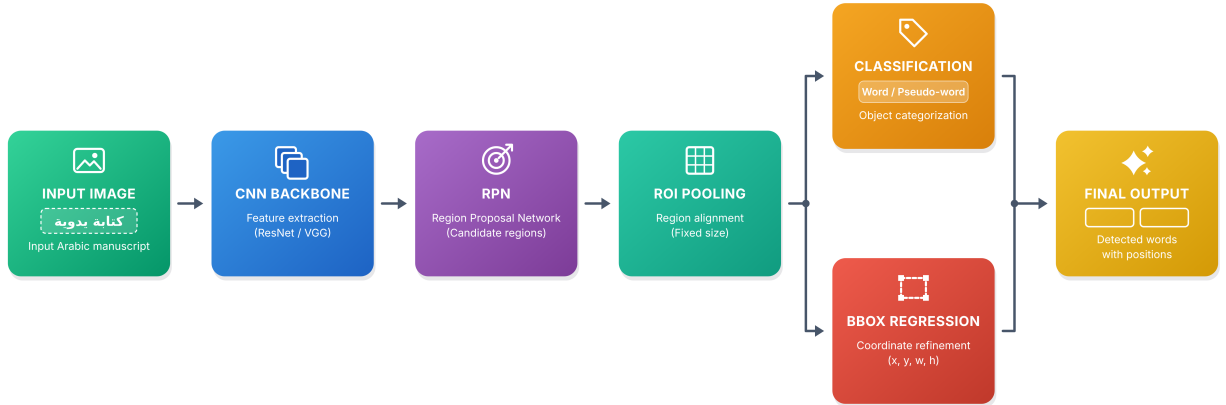


Figure 2. Faster R-CNN pipeline for word and pseudo-word segmentation.

3.5.3. Mask R-CNN

Mask R-CNN is similar to the Faster R-CNN algorithm, but it has the addition of the instance segmentation branch. In addition to the bounding box regression and object classification loss, the mask loss is also added:

$$\mathcal{L}_{MaskRCNN} = \mathcal{L}_{FRCNN} + \lambda_{mask} \mathcal{L}_{mask}, \quad (3)$$

where $\mathcal{L}_{mask}$ is the binary cross-entropy loss. The addition of the instance segmentation branch in the Mask R-CNN algorithm helps it to achieve accurate instance segmentation, which makes it particularly suitable for fine-grained segmentation tasks such as pseudo-word and character segmentation in Arabic manuscripts.

In this research, polygon-based annotations were used for fine-grained segmentation tasks. These annotations were converted into binary masks, which were used to train the segmentation branch of the Mask R-CNN algorithm. This helps the algorithm to accurately segment the manuscripts because the use of bounding boxes alone would not have captured the complexity of the manuscripts.

However, the use of the instance segmentation branch in the Mask R-CNN algorithm requires higher computational resources compared to the object detection algorithms.

Overall, the hierarchical integration of YOLOv8, Faster R-CNN, and Mask R-CNN reflects the progressive increase in segmentation difficulty, from relatively structured line detection to highly complex character-level segmentation in Arabic manuscripts.

3.5.4. Segmentation baselines: U-Net and DeepLabV3+

To contextualise the detection models, we add two semantic segmentation baselines evaluated at the character level only (the sole level with polygon annotations):

- **U-Net:** a symmetric encoder–decoder with skip connections (ResNet-34 backbone, pretrained on ImageNet).
- **DeepLabV3+:** an atrous-convolution encoder with ASPP module and lightweight decoder (ResNet-34 backbone, pretrained on ImageNet).

Training: 50 epochs, batch size 8, Adam optimizer with learning rate 10^{-3} . Input images resized to 128×512 pixels to preserve the aspect ratio of line-strip images. Output classes: 30 (29 characters + background). Best model selected by validation mIoU.

3.6. Training models by segmentation level

3.6.1. Line Segmentation

To train a YOLOv8 model for row detection, a dataset was created using YOLO detection format annotations. The YOLOv8 model was found to be highly accurate, and this can be attributed to the high structural regularity found in the layout of text lines in Arabic manuscripts from antiquity. The reason for this high accuracy can be attributed to the fact that a text line has a rectangular layout, making it suitable for object detection using bounding boxes. Although a certain level of detection inaccuracies was found, it was due to irregularities in the positioning of text lines or their curves.

3.6.2. Word segmentation

For word detection, two different models were implemented. The first model, based on the YOLOv8 model, was able to obtain a moderate level of detection accuracy even though the model was trained on a reliable dataset. This is mainly because the shape of the words is quite varied in the Arabic writing system, where there is no repetition or specific pattern that the words might follow. The second model was based on the Mask R-CNN model; however, the model was unable to obtain satisfactory results. The failure of the model is not because the manuscript forms are complicated, but the model fails to segment the word structures properly.

3.6.3. Pseudo-word segmentation

For pseudo-word detection, two detection models were tested. The faster R-CNN model was able to achieve efficient detection speeds, but its accuracy was poor when faced with the significant variability of the data in the ancient Arabic manuscripts, especially in the context of shape distortion and writing irregularities. The Mask R-CNN model was able to achieve improved accuracy in the delineation of pseudo-word boundaries due to its ability to perform segmentation and accurate localization of object boundaries. However, this accuracy is attained at the expense of increased complexity and training time.

3.6.4. Character segmentation

In the case of character-level segmentation, we looked at using Mask R-CNN, as it's the only model capable of dealing effectively with the cursiveness and ligature complexity of the Arabic script. While it performed well in certain cases, such as when the characters were loosely connected, it struggled when dealing with characters that are closely connected or have undergone a certain level of weaving. This can be explained by the complexity associated with Arabic handwriting, where shared strokes and unclear boundaries between characters make it difficult to achieve a good character segmentation task.

Summary and comparison The following table summarizes the models applied to each level of segmentation, Their advantages and limitations:

Table 3. Comparison of models applied at different segmentation levels.

Level	Model	Results	Limitations
Lines	YOLOv8	Very high accuracy	Simple structural patterns
Words	YOLOv8, Mask R-CNN	Moderate accuracy	High variability, lack of repetitive structure
Pseudo-words	Fast-CNN, Mask R-CNN	Acceptable results	Sensitive to handwritten variations
Characters	Mask R-CNN	Partial results	Difficulty due to Arabic cursiveness

4. Experiments and Results

4.1. Experimental Setup

All the experiments are carried out on Google Colab with an NVIDIA T4 GPU (16 GB VRAM), PyTorch 2.11, CUDA 12.8, Detectron2 0.6, and Ultralytics 8.x. Data was segregated into training with 70%, validation with 15%, and testing with 15% sets at the page level to prevent data leakage (see Section 3). The evaluation parameters were mean Average Precision (mAP), Intersection over Union (IoU), Precision, and Recall, as is the norm in Arabic handwriting recognition [1, 8, 10]. YOLOv8 was trained with three random seeds (0, 1, 2) at each level; we report the mean and standard deviation. Faster/Mask R-CNN and semantic segmentation models are reported for a single run each. Training parameters: YOLOv8 — 50 epochs, batch 8, imgsz 1024; Faster/Mask R-CNN — 30 effective epochs, batch 2, LR 2.5×10^{-3} ; U-Net/DeepLabV3+ — 50 epochs, batch 8, Adam, LR 10^{-3} .

4.2. Line Segmentation Results

The results obtained with YOLOv8 for line segmentation show a remarkable performance. Figure 3 illustrates the evolution of losses and metrics during training as well as an example of correct segmentation in figure 4.

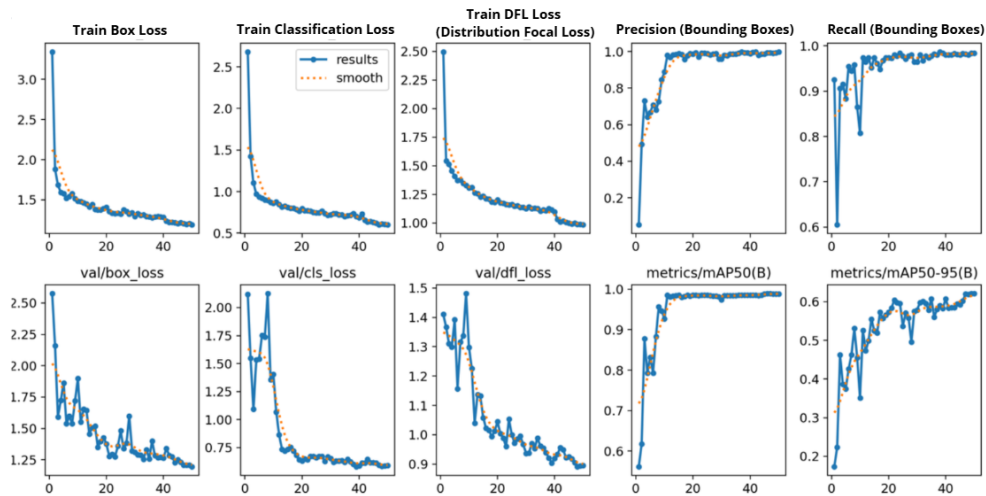


Figure 3. Line segmentation model metrics using YOLOv8

- **Losses:** the *train/box_loss*, *train/cls_loss* and *train/dfl_loss* curves decrease steadily over the epochs, showing a stable and progressive learning process. The *train/box_loss* reflects the model's ability to localize objects,

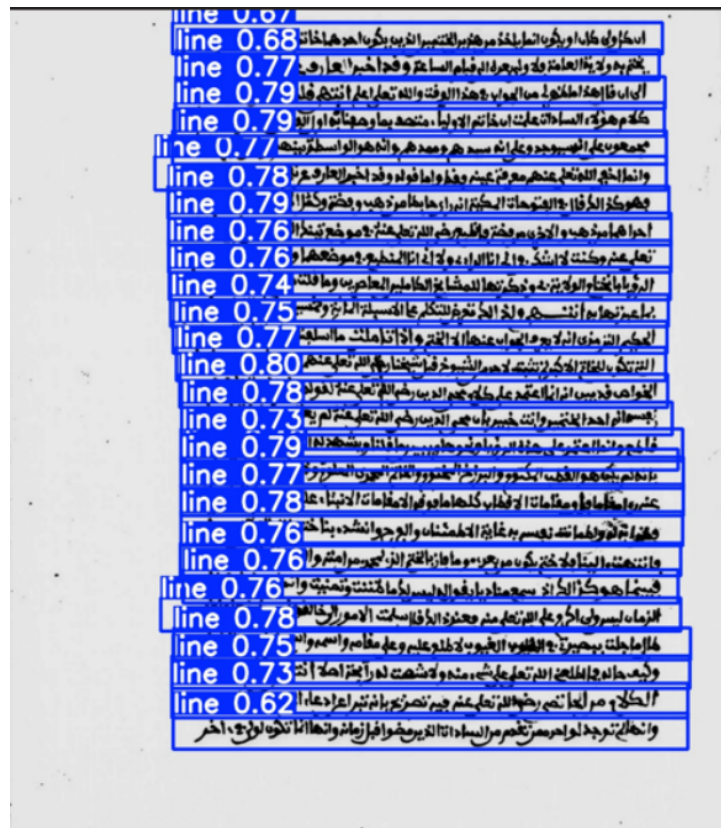


Figure 4. Example of line segmentation with YOLOv8

train/cls_loss measures its accuracy in classifying objects, and *train/dfl_loss* (Distribution Focal Loss) refines bounding box positions. The validation curves (*val/box_loss*, *val/cls_loss*, *val/dfl_loss*) follow a similar trend, indicating good generalization and no strong signs of overfitting.

- **Precision:** it very quickly reaches a value close to 0.98–1.00, This indicates that the model generates very few false detections.
- **Recall:** it converges to 0.95–1.00 after about ten epochs, confirming the ability of the network to detect almost all the lines present.
- **mAP@50 (mean Average Precision at IoU = 0.5):** the metric converges to 0.99, This demonstrates excellent line detection quality at standard thresholds.
- **mAP@50–95 (average over several thresholds of IoU):** it reaches about 0.6, reflecting a decrease in performance when stricter recovery criteria are applied. This difference can be attributed to calligraphic variations and the presence of slightly distorted lines in ancient manuscripts.

In brief, YOLOv8 proves very useful for line segmentation, forming an excellent foundation for further segmentation tasks like word, pseudo-word, and character segmentation.

It is clear that the lines have been detected and bounded accurately, with perfect alignment of their boundaries. Such an output is a reflection of the robustness of the model, especially considering the variation in the writing style and density of the text. Although there are some cases of overlap of lines which are very close to each other, it does not hamper the overall output.

4.3. Word Segmentation Results

Faster R-CNN achieved similar accuracy in word segmentation but was not stable in handling overlying or adjacent scripts, as in the case of the Arabic script [4, 29]. The main reason for this limitation is the difficulty in accurately generating region proposals in densely written or connected areas of text. To overcome this limitation in future work, it is recommended to use attention-based mechanisms or CNN-Transformer models, which are believed to improve the accuracy in word segmentation. YOLOv8 achieved better inference speed with competitive accuracy compared to other region-based models.

As illustrated in Figure 5, Faster R-CNN exhibits limitations in word segmentation caused by complex Arabic ligatures. By comparison, Figure 6 demonstrates that YOLOv8 achieves better word detection; nevertheless, segmentation errors remain due to the inherent complexity of Arabic handwriting.

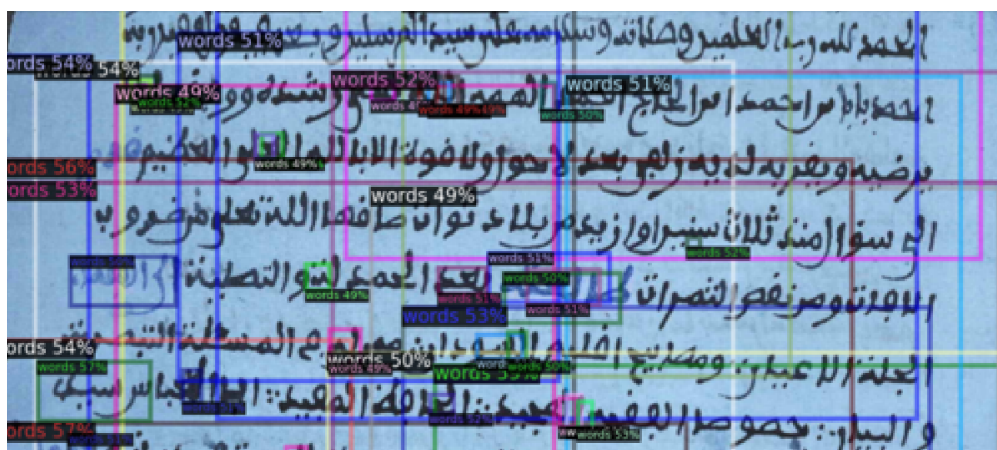


Figure 5. Example of word segmentation with Faster R-CNN (visible errors in ligatures).

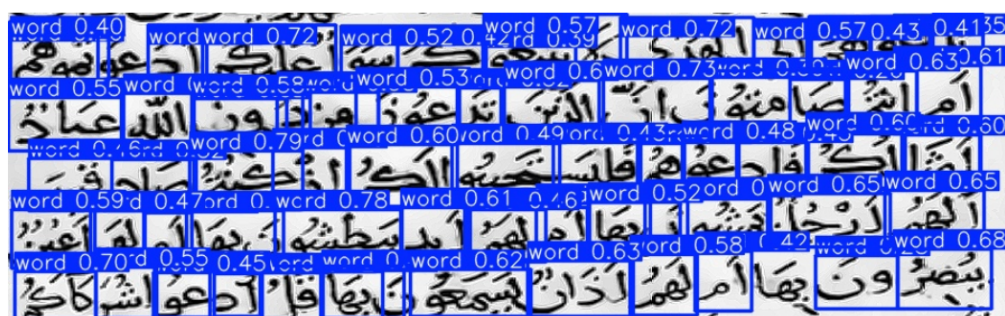


Figure 6. Example of word segmentation with YOLOv8 (better detection but still inaccurate).

The model has achieved better accuracy in word contour detection compared to Faster R-CNN model. However, the accuracy achieved in word segmentation is not sufficient as it is partially truncated or merged, which indicates that word segmentation in ancient Arabic manuscripts is a challenge.

4.4. Pseudo-word Segmentation Results

Mask R-CNN was used for pseudo-words, achieving an average IoU of 0.55. These results prove that this task is more challenging due to the presence of cursiveness and calligraphy.



Figure 7. Example of segmentation into pseudo-words with Mask R-CNN.

This example has demonstrated the use of Mask R-CNN for pseudo-word segmentation. The results obtained are average. Some results are correct and well aligned to the expected pseudo-words, while others are approximate or close to other segments. This proves that the model is challenged by the complexity of Arabic handwriting.

4.5. Character Segmentation Results

Segmenting individual characters using Mask R-CNN presents interesting results that reflect the difficulty of segmenting individual characters in ancient Arabic manuscripts. The model is able to achieve an accuracy of about 0.82 and a recall of about 0.78. This means that the majority of the characters are correctly detected while still allowing some segments. The mAP@50 is about 0.80, which again proves the potential of the model to correctly segment the most easily separable individual characters. However, the mAP@50–95 is just about 0.42.

Figure 8 is an example of character-level segmentation, reflecting the difficulty of segmenting cursive Arabic handwriting.



Figure 8. Example of character segmentation

5. Comparative Analysis and Discussion

Table 4 demonstrates the performance of various models for the different levels of segmentation. There is a significant dependence between the performance of the model and the complexity of the target units.

Table 4. Performance of segmentation models across different hierarchical levels of Arabic manuscripts. YOLOv8: mean $\pm$ std over 3 seeds. Best AP50 per level in **bold**. Faster R-CNN evaluated at lines, words, pseudo-words; Mask R-CNN at characters only (polygon annotations required).

Level	Model	Precision	Recall	mAP@50	mAP@50–95
Line	YOLOv8	.98 $\pm$.01	.97 $\pm$.01	97.7$\pm$0.2	58.0 $\pm$ 1.1
	Faster R-CNN	—	—	98.0	57.6
Word	YOLOv8	.68 $\pm$.01	.76 $\pm$.01	75.4$\pm$1.7	38.8 $\pm$ 0.8
	Faster R-CNN	—	—	73.3	32.3
Pseudo-word	YOLOv8	.31 $\pm$.17	.42 $\pm$.16	30.9 $\pm$ 23.1	10.7 $\pm$ 8.3
	Faster R-CNN	—	—	70.2	27.6
Character	YOLOv8	.46 $\pm$.04	.37 $\pm$.03	39.4$\pm$0.7	18.9 $\pm$ 0.2
	Mask R-CNN	—	—	33.7	15.9
<i>Semantic segmentation (characters only):</i>					
Character	U-Net	mIoU = 0.076			
Character	DeepLabV3+	mIoU = 0.089			

For the line level, YOLOv8 has almost perfect performance with $\text{mAP@50} = 97.7 \pm 0.2$ over three seeds. Faster R-CNN achieves comparable performance ($\text{mAP@50} = 98.0$), confirming that line detection is effectively solved by both architectures. For the word level, YOLOv8 achieves $\text{mAP@50} = 75.4 \pm 1.7$ while Faster R-CNN reaches 73.3. The higher recall (0.76) than precision (0.68) for YOLOv8 suggests that over-segmentation is more prevalent than under-segmentation. The pseudo-word level reveals a striking architecture sensitivity. YOLOv8 achieves only 30.9 ± 23.1 mAP@50 , with one of three seeds failing to converge entirely ($\text{mAP@50} = 4.3$ vs. ~ 44 for the two convergent seeds). Faster R-CNN, by contrast, reaches 70.2 mAP@50 on the same data. We attribute this gap to YOLOv8's fixed square input (1024×1024), which severely distorts the highly elongated line-strip images ($\sim 600 \times 40$ pixels) used at this level. This result demonstrates that architecture choice can matter more than dataset size at fine granularity levels. For the character level, YOLOv8 achieves $\text{mAP@50} = 39.4 \pm 0.7$ and Mask R-CNN achieves box AP50 = 33.7 and mask AP50 = 32.1. The small drop from box to mask metrics (-1.6 AP50) indicates that the primary difficulty lies in detection rather than shape delineation. U-Net (mIoU = 0.076) and DeepLabV3+ (mIoU = 0.089) confirm that semantic segmentation of 29 character classes on 207 training images remains an open problem. Overall, it is evident from the results that the performance of segmentation models degrades with increasing granularity. No single architecture dominates across all levels: YOLOv8 outperforms Faster R-CNN on words and characters but fails on pseudo-words. This complementarity validates the purpose of a multi-architecture benchmark.

The constancy of the obtained results at various levels of segmentation confirms the reliability of the suggested method and emphasizes the significance of using the presented data set to test segmentation models in harsh environmental conditions.

5.1. Error Analysis

In the case of **line detection**, mistakes occur infrequently: false mergers of nearby lines into one bounding box, and false non-detections of short lines close to page borders. For **word detection**, over-segmentation (splitting words by internal gaps), and under-segmentation (merging consecutive short words) constitute the major sources of error. In terms of **pseudo-word detection**, in addition to architecture-specific convergence problems, it is sometimes unclear visually whether the boundary of pseudo-words should be detected by human annotators. In the **character detection** task, a class imbalance with many classes (29 classes, ~ 7 images per class) leads to very different IoUs of 0.10 and 0.01 for common and rare characters.

Effectiveness of dataset cleansing. With the elimination of page images with contaminants, YOLOv8 AP50 (seed 0) became different in the following way: words rose from 68.3 to 74.8 (+6.5 points); pseudo-words decreased from 19.6 to 4.3 in seed 0 (as seen, the performance of the former was overstated due to training using entire pages that had 84% of labels, yet seeds 1-2 have ~ 44); characters stayed the same. These results demonstrate the importance of careful data curation in hierarchical annotation pipelines.

Limitations. We acknowledge several limitations: (i) annotations were performed by a single annotator and verified by the second author; while this two-stage process reduces errors, formal inter-annotator agreement metrics (e.g., Cohen’s kappa) could not be computed since both annotators did not annotate independently. (ii) The dataset does not include metadata on historical periods or calligraphic style classification. (iii) Transformer-based architectures (DETR, LayoutLMv3) were not evaluated due to computational constraints. (iv) Ablation studies and cross-dataset validation are left for future work.

6. Conclusion and Future Work

This paper presents a method for the analysis of ancient Arabic cursive manuscripts based on dataset construction, pre-processing, annotation, and segmentation using deep learning architectures. A detailed analysis of the experimental results is provided for the segmentation of ancient Arabic manuscripts at different hierarchical levels, highlighting the strengths and limitations of each evaluated model. The observed performance confirms that the proposed dataset constitutes a valuable contribution to research on historical Arabic handwriting. However, the results also reveal persistent challenges related to character complexity, ligatures, and calligraphic variability. State-of-the-art approaches, such as the integration of Transformer-based models, CRNN architectures, or recent segmentation frameworks, offer promising directions for improving existing systems. These findings pave the way for multiple future improvements at both the dataset and model levels, and contribute toward the development of more accurate and comprehensive OCR systems for ancient Arabic manuscripts.

However, despite the promising nature of the outcomes, there are various issues that exist. Some of these issues arise from the fact that the nature of handwritten Arabic script is complex, due to reasons like the presence of stylistic variations, ligatures, and even document degradation, as well as the shortage of labeled data. In light of these issues, there have been more studies on deep learning techniques, along with other approaches.

Subsequent research efforts should prioritize the following directions:

- Expanding annotated dataset resources to encompass a broader temporal and stylistic range of historical Arabic manuscripts.
- Designing hybrid architectures that combine the processing speed of YOLO-based approaches with the precision offered by region-based and segmentation-focused techniques.
- Exploring Transformer architectures and CNN–Transformer hybrid models [11, 12] to achieve more precise fine-grained recognition.
- Leveraging data augmentation strategies and semi-supervised learning approaches to alleviate the burden of manual annotation.

Overall, the findings support the potential of deep learning networks in identifying ancient Arabic writing and underscore the need for deeper databases and hybrid networks in overcoming current challenges.

Acknowledgements

The authors gratefully acknowledge the libraries and institutions that granted access to historical Arabic manuscripts. The dataset annotations were performed by the authors. The hierarchical dataset of ancient Arabic manuscripts introduced in this study is publicly available on Zenodo under CC BY 4.0: <https://doi.org/10.5281/zenodo.21627021>. The complete reproduction package (training and evaluation code, preprocessing scripts, and step-by-step README) is available at: <https://github.com/A2001ya/HAAM-benchmark>.

This research was funded by the National Center for Scientific and Technical Research (CNRST), Morocco, under the PhD-Associate Scholarship (PASS) Program.

REFERENCES

1. N. Alrobah and S. Albahli, *Arabic Handwritten Recognition Using Deep Learning: A Survey*, Arabian Journal for Science and Engineering, 2021. Available: <https://link.springer.com/article/10.1007/s13369-021-06363-3>
2. M. Pechwitz et al., *IFN/ENIT-database of handwritten Arabic words*, ResearchGate, 2002. Available: https://www.researchgate.net/publication/228904501_IFNENIT-database_of_handwritten_Arabic_words
3. N. Lamghari and S. Raghay, *DBAHCL: Database for Arabic Handwritten Characters and Ligatures*, Cadi Ayyad University, May 2017. Available: https://www.researchgate.net/publication/317204941_DBAHCL_database_for_Arabic_handwritten_characters_and_ligatures
4. F. Biadsy, J. El-Sana, and N. Habash, *Online Arabic Handwriting Recognition Using Hidden Markov Models*, Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition (IWFHR), 2006, pp. 200–205.
5. S. A. Mahmoud et al., *KHATT: Arabic Offline Handwritten Text Database*, KFUPM, 2012. Available: <https://pure.kfupm.edu.sa/en/publications/khatt-arabic-offline-handwritten-text-database>
6. A. Al-Shoshan et al., *Arabic Handwriting Database for Text Recognition*, ScienceDirect, 2013. Available: <https://www.sciencedirect.com/science/article/pii/S221201731300385X>
7. M. Dabhal, N. Aboutabit, and N. Lamghari, *Arabic Handwritten Character Recognition Based on PHOG Descriptor*, ICMDS 2021, Beni Mellal, Morocco, pp. 28–30.
8. N. Alrobah and S. Albahli, *Arabic Handwritten Recognition Using Deep Learning: A Survey*, Arabian Journal for Science and Engineering, vol. 46, no. 4, pp. 3577–3595, 2021. Available: <https://doi.org/10.1007/s13369-021-06363-3>
9. M. Elsayed, A. Alnaggar, M. Abdeen, A. Wahdan, and W. Gomaa, *Arabic Handwritten Text Recognition Using Advanced CNN–RNN Architecture*, Signal, Image and Video Processing, vol. 18, no. 3, pp. 355–372, 2024. Available: <https://doi.org/10.1007/s11760-024-02057-0>
10. S. A. Mahmoud et al., *Machine Learning Approach for Arabic Handwritten Recognition*, Applied Sciences, vol. 14, no. 19, p. 9020, 2024. Available: <https://doi.org/10.3390/app14199020>
11. C. Boufekar et al., *Bridging the Gap: Fusing CNNs and Transformers to Decode the Elegance of Handwritten Arabic Script*, Scientific Reports, vol. 15, p. 12045, 2025. Available: <https://doi.org/10.1038/s41598-025-12045-z>
12. S. Rakesh et al., *Handwritten Text Recognition Using Deep Learning Techniques: A Survey*, MATEC Web of Conferences, vol. 343, p. 01126, 2024. Available: <https://doi.org/10.1051/mateconf/202439201126>
13. S. Aabed and A. Khairaldin, *An End-to-End, Segmentation-Free, Arabic Handwritten Recognition Model on KHATT*, arXiv preprint, 2024. Available: <https://arxiv.org/abs/2406.15329>
14. R. Najam and S. Faizullah, *A Scarce Dataset for Ancient Arabic Handwritten Text Recognition*, Data in Brief, vol. 56, p. 110813, 2024. Available: <https://doi.org/10.1016/j.dib.2024.110813>
15. M. Saeed et al., *Muharaf: Manuscripts of Handwritten Arabic Dataset for Cursive Text Recognition*, NeurIPS 2024, Datasets and Benchmarks Track. Available: https://proceedings.neurips.cc/paper_files/paper/2024/hash/6b8cb6b291045e217c3ff3f854f2fd0f-Abstract-Datasets_and_Benchmarks_Track.html
16. British Library Labs, *Ground Truth Transcriptions for Training OCR of Historical Arabic Handwritten Texts*. Available: <https://doi.org/10.23636/1135>
17. A. Zoizou et al., *MOJ-DB: A New Database of Arabic Historical Handwriting and a Novel Approach for Sub-words Extraction*, ScienceDirect, 2022. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0167865522001520>
18. H. Bouchal and A. Belaid, *Arabic Handwritten Text Line Dataset*, arXiv preprint, Dec. 2023. Available: <https://arxiv.org/abs/2312.07573>
19. Bibliothèque nationale de France (BNF), *Gallica: Arabic Manuscripts of Morocco*. Available: <https://gallica.bnf.fr>
20. Qatar Foundation, *Qatar Digital Library*. Available: <https://www.qdl.qa/en>
21. M. Saeed et al., *Zenodo Datasets Repository*. Available: https://data.niaid.nih.gov/resources?id=zenodo_11492214
22. Bibliothèque Nationale du Royaume du Maroc, *BNRM Digital Catalog*. Available: <http://opac.bnrm.ma>
23. King Abdulaziz Public Library, *Digital Arabic Manuscripts Collection*. Available: <https://www.kapl.org.sa>
24. *MuHARaf and VML-HD: Academic Datasets of Arabic Manuscripts*.
25. Crowley Imaging, *Can We Get a Resolution? The Mystery of the Right DPI/PPI*, Blog post. Available: <https://www.thecrowleycompany.com/blog/can-we-get-a-resolution/>
26. Library of Congress, *When We Say We Want Resolution: How Many DPI/PPI Is Too Much?*, The Signal Blog. Available: <https://blogs.loc.gov/thesignal/2013/11/when-we-say-we-want-resolution-how-many-dpipppi-is-too-much/>
27. N. Lamghari, M. E. H. Charaf, and S. Raghay, *Hybrid Feature Vector for the Recognition of Arabic Handwritten Characters Using Feed Forward Neural Network*, IJCSIS, vol. 14, no. 12, pp. 31–40, 2022. Available: <https://link.springer.com/article/10.1007/s13369-017-2969-1>
28. Library of Congress, *Guidelines for Electronic Preservation of Visual Materials*. Available: <https://www.loc.gov/preservation/resources/rt/>
29. K. S. Younis, *Arabic Handwritten Character Recognition Based on Deep Convolutional Neural Networks*, Jordanian Journal of Computers and Information Technology, vol. 3, no. 3, pp. 186–198, 2017. Available: <https://www.jjcit.org/papers/vol3no3/vol3no3.pdf>

30. PyPI, *pdf2image Library*. Available: <https://pypi.org/project/pdf2image/>
31. Stack Overflow, *Convert PDF to PNG with PyMuPDF (fitz)*. Available: <https://stackoverflow.com/questions/16718669/convert-pdf-to-png-image-in-python>
32. ScanHouse, *Understanding DPI: Guide to Quality Document and Photo Scanning*. Available: <https://www.scanhouse.com/understanding-dpi/>
33. Roboflow, *Roboflow Annotate: Label Images Faster Than Ever*. Available: <https://roboflow.com/annotate>
34. A. Bouchantouf and N. Lamghari, *Deep Learning Methods for Ancient Arabic Handwritten Script Recognition: A Review of Challenges and Approaches*, Informatica, vol. 49, no. 28, 2025. Available: <https://www.informatica.si/index.php/informatica/article/view/8920>
35. S. Ren, K. He, R. Girshick, and J. Sun, *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, Advances in Neural Information Processing Systems, vol. 28, 2015.
36. K. He, G. Gkioxari, P. Dollár, and R. Girshick, *Mask R-CNN*, in Proc. ICCV, 2017, pp. 2961–2969.
37. O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, in Proc. MICCAI, 2015, pp. 234–241.
38. L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*, in Proc. ECCV, 2018, pp. 801–818.
39. H. Moussaoui, N. El Akkad, and M. Benslimane, *License Plate Text Recognition Using Deep Learning, NLP, and Image Processing Techniques*, Statistics, Optimization & Information Computing, vol. 12, no. 3, pp. 685–696, 2024. <https://doi.org/10.19139/soic-2310-5070-1966>
40. H. Moussaoui, N. El Akkad, and M. Benslimane, *A Hybrid Skin Lesions Segmentation Approach Based on Image Processing Methods*, Statistics, Optimization & Information Computing, vol. 11, no. 1, pp. 95–105, 2023. <https://doi.org/10.19139/soic-2310-5070-1549>
41. A. A. Sheikh, M. S. Azmi, W. A. K. Abuain, and M. A. Aziz, *Segmentation Techniques for Arabic Handwritten: A Review*, Statistics, Optimization & Information Computing, vol. 14, 2025. <https://doi.org/10.19139/soic-2310-5070-2466>
42. A. Bouchantouf and N. Lamghari, *Hierarchical Segmentation of Ancient Arabic Manuscripts: Dataset Construction and Deep Learning Evaluation*, In: Advances in Machine Intelligence and Computer Science Applications, N. Aboutabit and I. Hafidi (Eds.), Springer Nature Switzerland, Cham, 2026. Available: https://doi.org/10.1007/978-3-032-19963-8_11