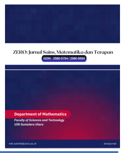
**Zero : Jurnal Sains, Matematika, dan Terapan**
E-ISSN : 2580-5754; P-ISSN : 2580-569X
Volume 9, Number 1, 2025
DOI: 10.30829/zero.v9i1.24064
Page: 66-76



# An Applied Computer Mathematics Approach to Transliteration: YOLOv8-Based Detection of Harah Jawoe Script

[1] Khairan AR
Department of Information Technology, Universitas Islam Negeri Ar-Raniry Banda Aceh, Aceh, Indonesia

[2] Hendri Ahmadian
Department of Information Technology, Universitas Islam Negeri Ar-Raniry Banda Aceh, Aceh, Indonesia

[3] Aesha Durratul Nasihah
Department of Physics Engineering, Universitas Islam Negeri Ar-Raniry Banda Aceh, Aceh, Indonesia

[4] Abd Mujahid Hamdan
Department of Physics Engineering, Universitas Islam Negeri Ar-Raniry Banda Aceh, Aceh, Indonesia

**Article Info**

***Article history:***

Accepted, 28 May 2025

***Keywords:***

Harah Jawoe;
Object Detection;
Transliteration;
YOLOv8

**ABSTRACT**

The preservation and accessibility of traditional scripts such as Harah Jawoe—a regional Jawi variant used in Acehnese manuscripts, pose significant challenges due to the lack of automated transliteration tools and the structural complexity inherent in word-level handwriting. This study directly addresses these limitations by proposing an AI-based transliteration approach that focuses on word-level recognition through the YOLOv8 object detection architecture. A dataset comprising 9,000 augmented images from the Hikayat Aceh manuscript was utilized to train and evaluate four YOLOv8 variants (small, medium, normal, and big). The primary contribution of this research is the integration of advanced object detection techniques with the transliteration of low-resource scripts, enabling robust and accurate word-level recognition. Experimental results indicate that the YOLOv8-big model delivers the most consistent and reliable performance, achieving a peak mAP50–95 of 72.4% and an accuracy of 99.95%. These outcomes underscore the potential of modern AI models to advance the digital preservation and accessibility of Southeast Asian manuscript heritage.



***Corresponding Author:***

Khairan AR,
Department of Information Technology,
Universitas Islam Negeri Ar-Raniry Banda Aceh, Aceh, Indonesia
Email: khairan.ar@ar-raniry.ac.id

## 1. INTRODUCTION

Transliteration plays a vital role in preserving and interpreting ancient manuscripts by converting obsolete scripts into modern writing systems while retaining their phonetic fidelity [1]. This task becomes particularly challenging for under-documented regional scripts such as Harah Jawoe, an Acehnese adaptation of Jawi, due to their hybrid orthographies, linguistic diversity, and significant contextual variation [2], [3], [4], [5], [6], [7]. Declining public literacy and the lack of formal instruction have further endangered these traditional scripts. Jawi and Harah Jawoe encapsulate historical and cultural narratives, such as those found in the Hikayat Aceh manuscript, making effective transliteration crucial for cultural heritage preservation [8], [9], [10], [11], [12], [13]. In this research, annotated images from the Hikayat Aceh manuscript were used to develop a machine learning transliteration dataset, connecting traditional philological knowledge with advanced computational approaches [14], [15], [16]. While prior computational studies primarily addressed isolated Jawi character recognition [17], [18], transliteration at the word level remains a significant challenge due to the positional and morphological variability of Arabic-derived scripts. Techniques such as Freeman Chain Code, Support Vector Machines (SVM) [19], Local Binary Patterns (LBP) [20], and especially Convolutional Neural Networks (CNNs) [21] have demonstrated strong performance, with CNNs achieving high accuracy even under challenging imaging conditions.

Despite these advances, no prior research has proposed an object detection-based approach, such as YOLOv8, specifically targeting word-level segmentation and transliteration for Harah Jawoe or similar scripts. Existing solutions largely overlook the need to detect and transliterate meaningful word clusters, focusing instead on isolated character classification. This represents a critical research void, there is a lack of end-to-end transliteration systems that can robustly handle the segmentation, recognition, and conversion of entire word units from historical manuscripts written in regional scripts [22], [23].

To address this gap, this study introduces a novel transliteration framework based on the advanced YOLOv8 object detection model, recognized for its high speed and accuracy in real-time image analysis [24], [25]. Here, YOLOv8 is adapted to identify and classify entire word segments in Harah Jawoe script, using annotated images from the Hikayat Aceh manuscript. Expert linguistic annotation ensures precise pairing of script images with their Latin transliterations, allowing the model to learn robust visual-word mappings. This integrated approach demonstrates the efficacy of deep learning for word-level script recognition and transliteration, offering a scalable solution for digital philology and the preservation of endangered regional scripts.

In summary, this research fills the gap left by existing transliteration methods by providing a scalable, automated solution for word-level transliteration of Harah Jawoe manuscripts using advanced object detection techniques. This approach not only addresses the segmentation and contextual recognition challenges inherent to regional scripts, but also sets a precedent for the digital preservation of other low-resource, endangered writing systems.

## 2. RESEARCH METHOD

The transliteration workflow from Harah Jawoe to Latin script (Figure 1) consists of sequential stages: dataset collection, pre-processing, splitting, model development, YOLOv8-based implementation, and comprehensive evaluation. Experiments were supported by an HP 14s-fq2 laptop (AMD Ryzen 7 5825U, 16 GB RAM, 512 GB SSD, Windows 11) and accelerated in the cloud using Google Colab Pro+ for GPU-based training. Python and TensorFlow formed the core software stack, enabling efficient model development and analysis across all phases of the study.


**Figure 1.** Workflow of the Harah Jawoe to Latin Script Transliteration Process

### 2.1. Dataset Collection

The process commences with comprehensive dataset collection, wherein images of Harah Jawoe script are systematically gathered from the **Hikayat Aceh** manuscript [17]. The initial dataset comprises 132 high-resolution JPG images (1080 × 1080 pixels), with each image corresponding to one of 20 unique words. To capture the inherent variability in handwriting, each word is represented by 12 to 18 different samples, reflecting the diverse visual forms found in actual manuscript contexts. To further enhance the dataset and support robust model training, additional images were synthesized using an online Jawi text editor (https://www.lexilogos.com/keyboard/jawi.html) and by employing various widely used fonts in Microsoft Word, including Arabic Traditional, Arial, Adobe Arabic, and Courier New. This multi-faceted approach ensures a comprehensive and representative dataset, encompassing the range of character shapes and styles present in Harah Jawoe script, and thereby supports the development of a more generalizable transliteration model.

### 2.2. Pre-processing

The subsequent stage, pre-processing, is essential for transforming the raw data into a form suitable for model development. Initially, sentence-level images are meticulously cropped to extract and isolate individual word segments, as illustrated in Figure 2. This step is critical to ensure that the model can effectively learn word-level features rather than broader sentence structures. Following the cropping process, each image is systematically labeled and annotated with its corresponding Latin transliteration, thereby facilitating supervised learning. To further enhance the diversity and robustness of the dataset, a variety of data augmentation techniques are employed, including rotation, shifting, and scaling. These augmentations not only expand the dataset but also introduce variability that simulates different writing conditions and styles. As a result of these comprehensive pre-processing and augmentation procedures, the final dataset comprises 3,000 high-quality word-level images, providing a solid foundation for effective model training and evaluation.

**Figure 2.** Harah Jawoe script dataset samples. (a) Example of an original sentence-level image containing multiple words in Harah Jawoe script. (b) Example of a cropped word-level image, isolated from the original sentence-level dataset.

### 2.3. Data Splitting

Following pre-processing and augmentation, the dataset undergoes a structured partitioning process to facilitate effective model development and objective performance evaluation, as illustrated in Figure 1. The complete dataset, consisting of 9,000 annotated word-level images, is systematically divided into three distinct subsets: training, validation, and testing datasets.

Specifically, 7,800 images are allocated to the training set, which serves as the foundation for fitting and learning the parameters of the YOLOv8 model. The validation set comprises 600 images and is utilized throughout the training process to fine-tune hyperparameters, prevent overfitting, and monitor model generalization. The remaining 600 images are reserved as the testing set, providing an unbiased and independent means to evaluate the model's final performance.

This carefully designed data split ensures that each subset serves a distinct and complementary role, training data enables the model to learn the complex patterns of Harah Jawoe script, validation data supports iterative optimization and early stopping, and testing data yields a robust assessment of the transliteration system's accuracy and generalizability on previously unseen examples. Such partitioning is crucial for the development of a reliable and generalizable deep learning model.

### 2.4. Model Development

The central component of this research is the implementation of the YOLOv8 object detection architecture, which was selected due to its demonstrated effectiveness in real-time, high-precision detection tasks across various domains. YOLOv8 employs a modular design consisting of three main elements: the backbone, the neck, and the head. The backbone is responsible for extracting hierarchical visual features from input images, enabling the model to capture intricate patterns and structures inherent in the Harah Jawoe script. The neck serves to aggregate and refine the extracted features, facilitating the flow of relevant information to subsequent layers. Finally, the head generates bounding boxes and assigns class labels, thereby producing precise localizations and confident classifications of word-level objects within each image.

To optimize the model's learning capacity and assess its performance under varying training conditions, the YOLOv8 architecture is trained on the annotated dataset using multiple epoch settings (specifically, 30, 50, 70, 90, and 100 epochs). This approach enables a systematic investigation of the learning dynamics, convergence behavior, and the identification of the optimal training duration for achieving robust word-level transliteration. The use of labeled data throughout the training process ensures that the model effectively learns to detect, segment, and transliterate Harah Jawoe script with high accuracy.

### 2.5. Model Implementation

Upon completion of the training phase, the optimized YOLOv8 model is deployed to conduct word-level segmentation and transliteration on previously unseen images of the Harah Jawoe script. In this inference stage, the model systematically analyzes each input image, identifying and localizing word objects by generating bounding boxes around them. For every detected word segment, the system assigns a class label that corresponds to its appropriate Latin script transliteration, as learned from the annotated dataset. This integrated process enables the automated recognition and conversion of Harah Jawoe words into their Latin equivalents, thereby facilitating efficient and accurate transliteration across diverse image samples. The end-to-end workflow not only streamlines the transliteration of complex regional scripts, but also enhances the accessibility and digital preservation of Acehnese manuscript heritage.

### 2.6. Evaluation

The performance of the proposed transliteration system is rigorously evaluated using a set of widely accepted quantitative metrics: recall, precision, F1-score, and accuracy. Each of these metrics is derived from the confusion matrix, which provides a comprehensive summary of the model's predictions in relation to the ground truth annotations (see Equations 1–4) [26], [27], [28], [29]. Collectively, these measures assess the system's ability to accurately detect, recognize, and transliterate Harah Jawoe words within manuscript images.

Recall quantifies the proportion of actual positive instances (correctly identified word segments) that the model successfully detects, as defined by Equation (1):

$$Recall = \frac{TP}{TP+FN} \tag{1}$$

Precision measures the proportion of predicted positive instances that are truly correct, as shown in Equation (2):

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

The F1-score, given in Equation (3), is the harmonic mean of precision and recall, providing a balanced evaluation that accounts for both false positives and false negatives:

$$F1\ score = \frac{2\ x\ Precision\ x\ Recall}{Precision+Recall} \tag{3}$$

Accuracy, as defined in Equation (4), represents the overall proportion of correct predictions (both positive and negative) relative to the total number of cases:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{4}$$

In these equations, TP (true positives) refers to the number of correctly identified Harah Jawoe word segments, TN (true negatives) refers to the number of correctly identified non-word segments, FP (false positives) denotes segments incorrectly identified as words, and FN (false negatives) indicates actual word segments that were missed by the model.

By employing these rigorous evaluation metrics, the study ensures a robust and objective assessment of the transliteration model's performance, facilitating meaningful comparison with existing approaches and providing insight into areas for further improvement.

## 3. RESULT AND ANALYSIS

### 3.1. Data Augmentation Techniques

Subsequently, a rotation process was applied to generate variations in object orientation by rotating the images at various angles. In addition to rotation, several other augmentation techniques were employed in this study, including changing the image background color, adjusting brightness levels (both enhancement and reduction), and modifying the text color within the images. In this phase, the text was transformed into several color variations, such as red and green. Further diversity was achieved by combining multiple augmentation techniques such as rotation and background adjustment to generate a more varied dataset than the original. Examples of the applied augmentation techniques are presented in Figure 3.


(a) Example of an Object Before Augmentation


(b) Rotational Variation Applied to Script


(c) Background Color Variation


(d) Text Color Variation Applied to Script


(e) Brightness Adjustment on Script Object

**Figure 3.** Examples of user input images (post-augmentation) supplied to the YOLOv8 model for detection and transliteration. These images represent the visual diversity—including rotation, background and text color changes, and brightness adjustments—that the model must generalize to during evaluation.

### 3.2. Dataset Expansion and Preparation

The dataset was expanded from an initial 3,000 to 9,000 images using various augmentation techniques to improve model robustness and performance. This expansion aimed to assess the effectiveness of different YOLO model variants, including the small, medium, normal, and big versions. To accommodate these models, the final dataset of 9,000 images was resized into multiple dimensions specifically 80px, 540px, and 1080px according to the model types.

Resizing the images ensured that the model could process different input sizes during training, allowing it to learn from varying image scales and improving its ability to detect and recognize objects with higher accuracy. After resizing, the images were merged into a single, unified dataset to streamline the training process.

### 3.3. Implementation Details and Performance Assessment

The overall workflow for this research is illustrated in Figure 4 (see below). The workflow begins with the augmentation of the original dataset (Figure 3), followed by image resizing and dataset splitting. Each YOLOv8 variant (small, medium, normal, big) was then trained and evaluated on the same standardized workflow to ensure comparability of results [30], [31], [32], [33].

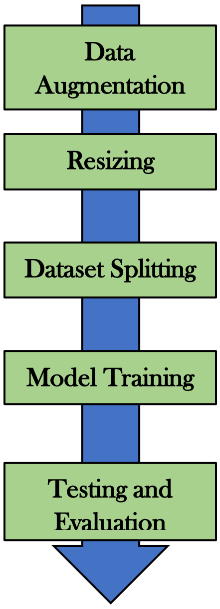



**Figure 4.** Workflow of the Harah Jawoe script recognition system from augmentation to evaluation.

- Step 1: Data Augmentation
  Images were augmented using rotation, background color changes, brightness adjustments, and text color variations (see Figure 3).
- Step 2: Resizing
  All images were resized to 80×80, 540×540, and 1080×1080 pixels, depending on model requirements.
- Step 3: Dataset Splitting
  The dataset was split into training (80%), validation (10%), and testing (10%) sets (see Table 1).
- Step 4: Model Training
  Four YOLOv8 models were trained using the same hyperparameters on Google Colab Pro+ with GPU acceleration.
- Step 5: Testing and Evaluation
  The trained models were tested using the 600-image test set. Performance metrics, including mAP50-95, accuracy, precision, recall, and F1-score, were computed at epochs 30, 50, 70, 90, and 100.

### 3.4. Dataset Splitting

The dataset was then partitioned into three subsets: training, validation, and testing. This split followed a standard ratio of 80% for training, 10% for validation, and 10% for testing. The purpose of this division was to provide the model with sufficient data for training while also allowing for effective validation and testing to evaluate model performance.

Table 1 provides an overview of the dataset distribution after augmentation. It shows that out of the 9,000 images, 7,800 images were designated for training, 600 images for validation, and 600 images for testing. This distribution ensures that the model can be thoroughly evaluated, with the validation and testing subsets offering a means to gauge its generalization and accuracy on unseen data.

**Table 1.** Dataset Distribution for Harah Jawoe Script

| Dataset Subset | Number of Image |
|---|---|
| Training Dataset | 7800 |
| Validation Dataset | 600 |
| Testing Dataset | 600 |
| Total | 9000 |

### 3.5. YOLOv8 Model Performance Across Variants

Table 2 presents the mAP50-95 accuracy results for the Harah Jawoe script testing on each YOLO model variant, including the small, medium, normal, and big models, evaluated at five training epochs: 30, 50, 70, 90, and 100. These accuracy values are calculated using the mean average precision (mAP) metric, specifically focusing on the intersection of the predicted bounding boxes with ground truth annotations at 50-95% overlap thresholds. This table serves as a key performance indicator, showing the evolution of model performance as training progresses.

**Table 2.** mAP50-95 Accuracy Results for Harah Jawoe Script Testing on Each YOLO Model

| Model | Epoch | mAP50-95 |
|---|---|---|
| Small | 30 | 0.744 |
| | 50 | 0.719 |
| | 70 | 0.717 |
| | 90 | 0.704 |
| | 100 | 0.707 |
| Medium | 30 | 0.722 |
| | 50 | 0.713 |
| | 70 | 0.72 |
| | 90 | 0.714 |

| Model | Epoch | mAP50-95 |
|---|---|---|
| | 100 | 0.711 |
| Normal | 30 | 0.723 |
| | 50 | 0.722 |
| | 70 | 0.718 |
| | 90 | 0.704 |
| | 100 | 0.715 |
| Big | 30 | 0.724 |
| | 50 | 0.722 |
| | 70 | 0.723 |
| | 90 | 0.721 |
| | 100 | 0.724 |

The small model achieved its peak mAP50-95 of 0.744 (74.4%) at epoch 30, indicating high early detection performance but declining over subsequent epochs, with scores dropping to 0.719 at epoch 50, 0.717 at epoch 70, and reaching a low of 0.704 at epoch 90 before a slight rebound to 0.707 at epoch 100. These fluctuations suggest overfitting and limited generalization capacity for the small model over time. The medium model followed a similar but more stable pattern, with an initial mAP of 0.722 at epoch 30, declining gradually to 0.713 at epoch 50, 0.72 at epoch 70, 0.714 at epoch 90, and 0.711 at epoch 100. This steady decrease implies moderate reliability, making it preferable for resource-constrained scenarios but still facing challenges with extended training.

The normal model began with a mAP of 0.723 at epoch 30, maintaining stability through epoch 70 but showing a more pronounced drop at epoch 90 (0.704) before a modest improvement to 0.715 at epoch 100. This pattern reflects better consistency than the small or medium models but still reveals issues with long-term generalization.

In contrast, the big model demonstrated superior stability and accuracy across all epochs, starting at 0.724 at epoch 30, with only minor fluctuations: 0.722 at epoch 50, 0.723 at epoch 70, 0.721 at epoch 90, and returning to 0.724 at epoch 100. The consistent performance of the big model underscores its robustness for Harah Jawoe word detection, making it the most suitable for applications demanding accuracy and sustained reliability.

These findings, as detailed in Table 2, highlight the crucial role of model complexity in balancing early detection strength with long-term stability. While smaller models excel in the early stages, only the big model maintains top performance throughout training, confirming its advantage for challenging, high-accuracy transliteration tasks.

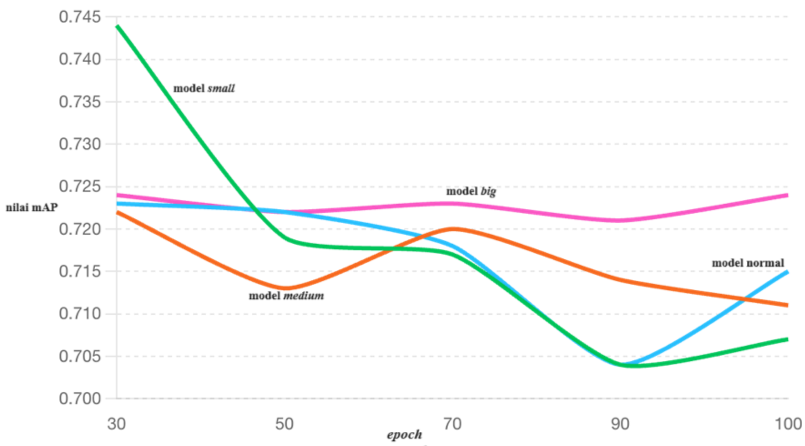



**Figure 5.** mAP50-95 Comparison of YOLO Models on Harah Jawoe Script Recognition: The chart shows accuracy scores across epochs 30–100. The small model achieved the highest mAP, while the big model demonstrated the most consistent performance.

Figure 5 tracks mAP50-95 accuracy for each YOLOv8 variant (small, medium, normal, big) over five training epochs (30, 50, 70, 90, and 100), revealing distinct performance trajectories. The small model peaked at epoch 30 with a mAP of 0.744 (74.4%), but its accuracy declined to 0.704 (70.4%) by epoch 90, only slightly recovering to 0.707 at epoch 100—evidence of overfitting and instability over time. The medium model showed greater stability, starting at 0.722 (72.2%) and ending at 0.711 (71.1%), reflecting a more gradual, consistent decline. The normal model paralleled the medium model, with a mAP of 0.723 (72.3%) at epoch 30 and 0.715 (71.5%) at epoch 100, demonstrating moderate reliability but some volatility in later stages.

The big model stood out for its consistent and robust performance, maintaining a mAP around 0.724 (72.4%) across all epochs, with minimal fluctuation. This stability, visible as a nearly flat line in the graph, underscores the big model's ability to generalize and maintain accuracy through prolonged training—critical for complex recognition tasks like Harah Jawoe script detection.

These patterns highlight the limitations of smaller models, which achieve high early accuracy but struggle with generalization as training continues. In contrast, the big model's architecture supports both high accuracy and sustained stability, making it the most reliable for extended training and challenging datasets. Consequently, final evaluations focus on the small and big models at epoch 30, as shown in Table 3, to compare their respective strengths in object detection metrics under identical conditions.

**Table 3.** Evaluation Metrics for Small and Big Models at 30 Epochs

| Model | *epoch* | *accuracy* | *precision* | *recall* | *F1 score* |
|---|---|---|---|---|---|
| Small | 30 | 0.9988 | 0.9883 | 0.9883 | 0.9883 |
| Big | 30 | 0.9995 | 0.995 | 0.995 | 0.995 |

The *accuracy* metric reflects the proportion of correctly classified instances (both positive and negative) out of the total number of instances. At epoch 30, the small model achieved an exceptionally high accuracy of 0.9988 (99.88%), while the big model performed slightly better with an accuracy of 0.9995 (99.95%). The small model's accuracy is already very high, indicating that the model correctly identifies the majority of instances in the dataset. However, the big model slightly surpasses it, achieving a higher accuracy by detecting more instances correctly. This difference in accuracy, although marginal, suggests that the big model is more precise in correctly classifying the objects in the dataset at this point in training.

### 3.6. Mathematical Formulation and Performance Evaluation

To robustly assess the performance of the YOLOv8 models in Harah Jawoe script detection, several widely accepted quantitative metrics were utilized, each calculated according to the formulas defined in Section 2.6 (Equations 1–4).

For instance, the accuracy metric (see Equation 1) quantifies the overall proportion of correctly classified instances—both true positives and true negatives—over the total predictions. Applying this to the small model at 30 epochs, with 593 true positives, 11,399 true negatives, 7 false positives, and 7 false negatives, yields:

$$\text{Accuracy} = \frac{593 + 11{,}399}{593 + 11{,}399 + 7 + 7} = \frac{11{,}992}{12{,}006} = 0{,}9988\ (99{,}88\%)$$

Similarly, the precision (Equation 2), recall (Equation 3), and F1-score (Equation 4) are derived using the corresponding confusion matrix values, as detailed in Table 3. For the small model at 30 epochs, both precision and recall are:

$$\text{Precision} = \frac{593}{593 + 7} = 0{,}9883\ (98{,}83\%)$$

$$\text{Recall} = \frac{593}{593 + 7} = 0{,}9883\ (98{,}83\%)$$

The F1-score, as the harmonic mean of precision and recall (Equation 4), is:

$$\text{F1} - \text{score} = \frac{2x0{,}9883x0{,}9883}{0{,}9883 + 0{,}9883} = 0{,}9883\ (98{,}83\%)$$

The F1-score's strength lies in its ability to account for imbalances between precision and recall, giving a single robust indicator of overall detection performance.

Taken together, these results paint a clear picture of the small YOLOv8 model's capabilities at 30 epochs. The near-identical values for precision, recall, and F1-score indicate an exceptionally well-calibrated model that is both accurate in its positive predictions and comprehensive in its coverage of true script instances. This balance is critical for script recognition tasks, where both missed detections and false positives can be costly—particularly in cultural heritage contexts where information loss or misclassification may compromise further analysis or preservation.Applying the same calculations to other models and epochs throughout the study ensured consistent, reproducible benchmarking. The confusion matrix serves as a transparent record of model outcomes, and each metric's formula ties directly to practical questions of detection: how often does the model get it right, how much can its positive predictions be trusted, how thoroughly does it find what matters, and how well are these qualities balanced.Ultimately, the comprehensive breakdown of these performance metrics, as detailed in Table 3, not only supports comparative analysis between the small and big YOLOv8 variants but also provides a solid foundation for future improvements and adoption in real-world Harah Jawoe transliteration pipelines.

### 3.7. Analysis

The evaluation results demonstrate the strong performance of both the small and big YOLOv8 models in the task of Harah Jawoe script recognition, with each model offering distinct strengths. At 30 training epochs, the small model achieved an accuracy of 0.9988 (99.88%), while the big model slightly surpassed this with an accuracy of 0.9995 (99.95%). This small difference in accuracy implies that the big model is marginally more reliable in minimizing misclassifications, especially in challenging or complex cases. When examining precision, which measures the proportion of correct positive predictions, the small model scored 0.9883 (98.83%), outperforming the big model's precision of 0.995 (99.5%). Although the big model's precision is slightly lower, it is more conservative in its positive predictions, resulting in fewer false positives—a valuable trait for applications where unnecessary detections must be minimized.

Recall, representing the model's sensitivity to all true positive cases, was also 0.9883 (98.83%) for the small model and 0.995 (99.5%) for the big model. This indicates both models are highly effective at identifying relevant script instances, but the small model's slightly higher recall shows it can detect more true instances, though with a potential for slightly more false alarms. The F1 score, which balances precision and recall, matched the individual precision and recall values for each model: 0.9883 (98.83%) for the small model and 0.995 (99.5%) for the big model, reflecting strong, balanced performance.

Taken together, these findings suggest that the small model, with its higher precision and recall, is exceptionally effective at both detecting script instances and minimizing errors in short-term evaluation. However, the big model's slightly higher overall accuracy, combined with its consistently high precision and recall, highlights its stability and robustness, particularly over extended epochs and diverse datasets. This characteristic is vital for applications demanding sustained reliability and minimal error rates, such as digital script preservation and large-scale manuscript processing.

The exceptional performance of both models stands out in comparison with results from earlier research. Numerous studies have shown that object detection accuracy is strongly influenced by dataset size, training duration, and annotation quality [16]; [15]; [17]. For instance, [10] achieved 91% accuracy in Arabic character recognition with CNNs. [12] found that increasing training data led to detection accuracy of up to 92% for Sundanese scripts. These results reinforce that both the dataset's scale and the depth of model training play critical roles in detection outcomes, though even with such improvements, their best metrics remain below those achieved by the YOLOv8 models in this study.

Recent studies confirm the effectiveness of YOLO architectures for image detection, with YOLOv8 achieving up to 92% accuracy in fundus image classification [19], and targeted augmentation strategies—such as varying background color—further boosting accuracy to 97–98% [13]. These improvements closely align with the present study's approach and outcomes. The Harah Jawoe script, characterized by significant visual and morphological complexity, presents an even greater challenge; thus, the high accuracy, precision, recall, and F1-scores achieved here underscore the advantages of combining state-of-the-art YOLOv8 models with robust data augmentation. This not only outperforms conventional CNN-based methods, as shown in earlier research [16], but also demonstrates the critical importance of model architecture and data strategy in complex script detection tasks.

Moreover, these results are particularly noteworthy given the study's focus on word-level, rather than isolated character-level, recognition—a more practical and demanding scenario for digital manuscript preservation. In sum, the consistency and robustness of the findings affirm the value of the YOLOv8-based workflow and offer compelling new evidence for its effectiveness in deep learning-based recognition and transliteration of culturally unique, endangered scripts.

## 4. CONCLUSION

This study confirms that YOLOv8 models, particularly the big variant deliver robust and accurate recognition of the Harah Jawoe script, exceeding the stability and detection accuracy of conventional CNN-based methods. While the small model demonstrated the highest initial mAP, only the big model maintained consistent, reliable performance across training epochs, supporting the practical application of YOLOv8 for transliteration and digital preservation of Acehnese manuscripts.

Nevertheless, important limitations remain. The dataset, despite augmentation, largely consists of synthetic images and does not yet capture the diversity, noise, or calligraphic variation found in historical handwritten manuscripts. As a result, the model's generalizability to real-world archival materials is not fully validated, and annotation quality or limited sample diversity may restrict recognition capability.

Future research should expand the dataset with authentic handwritten and historical manuscript images, rigorously test models under real-world archival conditions, and employ adaptive or hybrid deep learning techniques (such as CRNNs or transformers) for greater robustness. Collaborations with cultural heritage organizations and local libraries will be essential—not only for large-scale deployment, but also to ensure ethical stewardship and respect for cultural sensitivities associated with digitizing sacred or traditional texts.

Ultimately, this work offers a scalable pathway for empowering educators, archivists, and local communities to revitalize and preserve endangered scripts through digital technology, while emphasizing the need for ongoing ethical consideration and dialogue with stakeholders. As such, it represents a significant step toward sustainable, culturally sensitive digital philology and heritage preservation.

## 5. REFERENCES


[1] Yusra, M. Fikry, S. Syahfrida, M. Irsyad, and S. Sanjaya, "Aplikasi Android untuk Alih Aksara Latin ke Arab Melayu dengan Pendekatan Berbasis Aturan," *Jurnal CoreIT*, vol. 6, no. 2, pp. 117–121, Dec. 2020.

[2] V. Venkataraghavan, S. Sivapatham, and A. Kar, "Wav2Lip Bridges Communication Gap: Automating Lip Sync and Language Translation for Indian Languages," *IEEE Access*, 2025, doi: 10.1109/ACCESS.2025.3562883.

[3] G. Shimi, C. J. Mahibha, and D. Thenmozhi, "An Empirical Analysis of Language Detection in Dravidian Languages," *Indian J Sci Technol*, vol. 17, no. 15, pp. 1515–1526, Apr. 2024, doi: 10.17485/IJST/v17i15.765.

[4] X. Ma and Y. Qi, "Design and Practice of Virtual Experimental Scenes Integrating Computer Vision and Image Processing Technologies," *Scalable Computing: Practice and Experience*, vol. 25, no. 6, Oct. 2024, doi: 10.12694/scpe.v25i6.3361.

[5] A. R. Widiarti, R. Pulungan, A. Harjoko, Marsono, and S. Hartati, "A proposed model for Javanese manuscript images transliteration," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Oct. 2018. doi: 10.1088/1742-6596/1098/1/012014.

[6] Z. Arifin Nugraha Kantor Bahasa Provinsi Bengkulu, J. Zainul Arifin, K. Timur Indah, K. Singaran Pati, and K. Bengkulu, "Manuscripts and Philological Studies in Indonesia," 2023. [Online]. Available: http://ejournal.undip.ac.id/index.php/culturalistics52

[7] P. Coluzzi, "Jawi, an endangered orthography in the Malaysian linguistic landscape," *Int J Multiling*, pp. 1–17, 2020, doi: 10.1080/14790718.2020.1784178.

[8] S. Nurrahma Kuswati, "Kegiatan Digitalisasi Naskah Kuno Sebagai Upaya Diseminasi Informasi," *LIBRIA*, vol. 13, no. 1, pp. 107–129, Jun. 2021.

[9] R. Abdul Ghani, M. Shanudin Zakaria, and K. Omar, "Jawi-Malay Transliteration," in *2009 International Conference on Electrical Engineering and Informatics*, Selangor, Malaysia: UKM, Aug. 2009, pp. 5–7.

[10] A. S. Bania and B. Akob, "Preserving the Jawi Script in Aceh: Assessing Literacy, Cultural Heritage, and Modern Paradigm Challenges," *Studies in English Language and Education*, vol. 12, no. 1, pp. 457–470, 2025, doi: 10.24815/siele.v12i1.36629.

[11] M. Mahyeddin Mohd Salleh, wani Aziz Nurdalila, S. Gunardi, H. Mas Nooraini Haji Mohiddin, and dan Chairill Anuar Abdul Ghani, "Analisis Penyenaraian Haiwan Menerusi Kitab Kasyf Al-Litham 'An As-Ilah Al-Anam Karya Syeikh Zainul Abidin Bin Muhammad Al-Fatani Menurut Perspektif Halal Dan Haram An Analytical Study on the Classification of Animals in the Book Kashf al-Litham 'an As'ilah al-Anam by Sheikh Zainul Abidin bin Muhammad al-Fatani from the Halal and Haram Perspective," vol. 30, no. 2, 2025, doi: 10.33102/jfatwa.vol.30no2.671.

[12] I. Afrianto, R. Riyanda, and S. Atin, "Implementasi Algoritma Freeman Chain Code Dan Support Vector Machine (SVM) Pada Identifikasi Aksara Arab Melayu," 2018.

[13] Hermansyah, "Manuskrip Aceh: Revitalisa Sikearifan Masyarakat Aceh Di Era Global," *Wacana Etnik, Jurnal Ilmu Sosial dan Humaniora*, vol. 5, pp. 25–36, Apr. 2015.

[14] D. Siahaan, N. P. Sutramiani, N. Suciati, I. N. Duija, and I. W. A. S. Darma, "DeepLontar dataset for handwritten Balinese character detection and syllable recognition on Lontar manuscript," *Sci Data*, vol. 9, no. 1, Dec. 2022, doi: 10.1038/s41597-022-01867-5.

[15] Y. Madhani *et al.*, "Aksharantar: Open Indic-language Transliteration datasets and models for the Next Billion Users." [Online]. Available: https://opensource.org/licenses/MIT

[16] C. Simmons, R. Diehl Martinez, and D. Jurafsky, "SumTablets : A Transliteration Dataset of Sumerian Tablets," 2024.

[17] K. F. H. Van Langen, *Atjehsche Taal.* Woordenboek, 1889. [Online]. Available: http://books.google.com

[18] B. Yanto, B. Herawan Hayadi, and U. Pasir Pengaraian, "Indentifikasi Pola Aksara Arab Melayu Dengan Jaringan Syaraf Tiruan Convolutional Neural Network (CNN)," *JSAI : Journal Scientific and Applied Informatics*, vol. 3, no. 3, pp. 106–114, 2020, doi: 10.36085.

[19] I. Akil and I. Chaidir, "Deteksi Karakter Huruf Arab Dengan Menggunakan Convolutional Neural Network," *Inti Nusa Mandiri*, vol. 15, no. 2, pp. 183–188, Feb. 2021, doi: 10.33480/inti.v15i2.XXXX.

[20] C. A. Sari, W. S. Sari, P. Mega, and A. Wijayanti, "Pengaruh Linear Binary Pattern (Lbp) Dalam Pengenalan Citra Aksara Jawa Berbasis Optical Character Recognition (OCR)," *2 st Proceeding STEKOM*, vol. 2021, 2022.

[21] S. N. Rahmawati, E. W. Hidayat, and H. Mubarok, "Implementasi Deep Learning Pada Pengenalan Aksara Sunda Menggunakan Metode Convolutional Neural Network," *INSERT: Information System and Emerging Technology Journal*, vol. 2, no. 1, 2021.

[22] M. W. A. Kesiman and I. M. D. Maysanjaya, "A model for posttransliteration suggestion for balinese palm leaf manuscript with text generation and lstm model," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Mar. 2021. doi: 10.1088/1742-6596/1810/1/012011.

[23] M. Farid Naufal, J. Siswantoro, and J. T. Soebroto, "International Journal On Informatics Visualization journal homepage : www.joiv.org/index.php/joiv International Journal On Informatics Visualization

Transliterating Javanese Script Images to Roman Script Using Convolutional Neural Network with Transfer Learning." [Online]. Available: www.joiv.org/index.php/joiv
[24] A. Sani and S. Rahmadinni, "Deteksi Gestur Tangan Berbasis Pengolahan Citra," *Jurnal Rekayasa Elektrika*, vol. 18, no. 2, Jul. 2022, doi: 10.17529/jre.v18i2.25147.
[25] D. Marcella and S. Devella, "Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network Dengan Arsitektur VGG-19," *Jurnal Algoritme*, vol. 3, no. 1, pp. 60–70, 2022.
[26] N. K. Chau, W. J. Kim, C. H. Lee, K. J. Chae, G. Y. Jin, and S. Choi, "Quantitative computed tomography imaging classification of cement dust-exposed patients-based Kolmogorov-Arnold networks," *Artif Intell Med*, vol. 167, Sep. 2025, doi: 10.1016/j.artmed.2025.103166.
[27] M. M. Hemal and S. Saha, "Explainable deep learning-based meta-classifier approach for multi-label classification of retinal diseases," *Array*, vol. 26, Jul. 2025, doi: 10.1016/j.array.2025.100402.
[28] A. Mukherjee, V. Hassija, V. Chamola, and K. K. Gupta, "A Detailed Comparative Analysis of Automatic Neural Metrics for Machine Translation: BLEURT & BERTScore," *IEEE Open Journal of the Computer Society*, 2025, doi: 10.1109/OJCS.2025.3560333.
[29] N. A. AL Ajmi and M. Shoaib, "Optimization Strategies in Quantum Machine Learning: A Performance Analysis," *Applied Sciences (Switzerland)*, vol. 15, no. 8, Apr. 2025, doi: 10.3390/app15084493.
[30] R. A. Lashaki, Z. Raeisi, N. Razavi, M. Goodarzi, and H. Najafzadeh, "Optimized classification of dental implants using convolutional neural networks and pre-trained models with preprocessed data," *BMC Oral Health*, vol. 25, no. 1, Dec. 2025, doi: 10.1186/s12903-025-05704-0.
[31] D. K. Venkatesh, D. Rivoir, M. Pfeiffer, F. Kolbinger, and S. Speidel, "Data Augmentation for Surgical Scene Segmentation with Anatomy-Aware Diffusion Models," in *Proceedings - 2025 IEEE Winter Conference on Applications of Computer Vision, WACV 2025*, Institute of Electrical and Electronics Engineers Inc., 2025, pp. 2280–2290. doi: 10.1109/WACV61041.2025.00228.
[32] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/j.gltp.2022.04.020.
[33] T. Ozcan and H. Gungor, "Baby Cry Classification Using Structure-Tuned Artificial Neural Networks with Data Augmentation and MFCC Features," *Applied Sciences (Switzerland)*, vol. 15, no. 5, Mar. 2025, doi: 10.3390/app15052648.