Indonesian Journal of Data and Science

Volume 6 Issue 1
ISSN 2715-9936
https://doi.org/10.56705/ijodas.v6i1.239

Research Article

Optimizing Javanese Numeral Recognition Using YOLOv8 Technology: An Approach for Digital Preservation of Cultural Heritage

Lukman Syafie [1,*], Huzain Azis [2], Fadhilah Tangguh Admojo [3]

[1] Universiti Kuala Lumpur, 50250 Kuala Lumpur, Malaysia, lukman.syafie@s.unikl.edu.my
[2] Universiti Kuala Lumpur, 50250 Kuala Lumpur, Malaysia, huzain.azis@s.unik.edu.my
[3] Universiti Kuala Lumpur, 50250 Kuala Lumpur, Malaysia, fadhila.tangguh@s.unikl.edu.my

Correspondence should be addressed to Lukman Syafie; lukman.syafie@s.unikl.edu.my

Received 10 December 2024; Accepted 20 March 2025; Published 31 March 2025

Abstract:

Introduction: The preservation of Javanese script as part of Indonesia's cultural heritage is increasingly urgent in the digital era, especially due to declining literacy among younger generations. This study aims to explore the effectiveness of YOLOv8, an advanced object detection algorithm, for recognizing handwritten Javanese numerals to support efforts in cultural digitization and education. **Methods**: A dataset of 2,790 handwritten Javanese numerals (0–9) was collected from 93 respondents. Each numeral was manually annotated using bounding boxes via the MakeSense.ai platform. The YOLOv8 model was trained using 80% of the data and validated on the remaining 20%. Training was performed in the PyTorch framework with data augmentation techniques to increase robustness. Model performance was evaluated using precision, recall, F1-score, and mean Average Precision (mAP), along with visualization through confidence curves and confusion matrices. **Results**: The model achieved a high validation precision of 88.3%, recall of 89.1%, and mAP of 0.88 at IoU 0.90. F1-score peaked at a confidence threshold of 0.89, while certain numerals like 'six' and 'nine' achieved near-perfect detection. Visualizations confirmed the model's ability to accurately classify and localize characters in both training and unseen data. Minor misclassifications occurred between visually similar numerals. **Conclusions**: YOLOv8 demonstrates high effectiveness in recognizing handwritten Javanese numerals and holds significant potential for digital heritage preservation. Future work should focus on expanding the dataset, improving generalization under varied conditions, and integrating this model into educational tools and augmented reality applications for interactive learning.

Keywords: Educational Technology, Handwriting Recognition, Javanese Script, Object Detection, YOLOv8.

1. Introduction

Handwritten Javanese Script Recognition, a traditional writing system used on the island of Java, Indonesia, has become an important topic in the study of language and cultural preservation. Javanese Script not only functions as a means of communication but also plays a crucial role in traditional rituals and arts. However, in today's digital era, the utilization and understanding of Javanese Script tend to decline, especially among the younger generation. This creates an urgent need to integrate digital technology into the preservation and teaching of Javanese Script, particularly through the automatic recognition of numerals in handwritten Javanese Script—a topic that remains underexplored in handwriting recognition research.

In a continuous effort to integrate advanced technology in cultural preservation and digitization, the use of object recognition algorithms—specifically YOLOv8—has demonstrated significant progress across various application domains [1]–[3]. Related studies, such as that described by Zhang et al. (2023) in "YOLOv8-SC: Target Detection Algorithm of Live Detection System for Tensioning Clamps in High-Voltage Transmission Lines," have shown improvements in object detection precision in complex environments by modifying YOLOv8 network components, thereby proving the algorithm's flexibility and effectiveness in different technical settings [4]. Additionally, Ghania Zidani (2023) combined YOLOv8 with the DeepSort algorithm to enhance pedestrian tracking in "Optimizing Pedestrian Tracking for Robust Perception with YOLOv8 and DeepSort," illustrating the algorithm's capability to improve perception in computer vision applications [5].

This study focuses on using YOLOv8 to recognize Javanese Script numerals—a writing system that is not only visually intricate but also culturally significant. As discussed in related research, YOLOv8 has been successfully adapted in various scenarios to detect and classify objects with high accuracy [6], [7]. In this context, YOLOv8 is employed to address the challenges of recognizing the diverse and complex handwritten characters of Javanese Script, similar to its application in the study by Thakur et al. (2023) in "YOLOv8-Based Helmet and Vest Detection System for Safety Assessment," which adapted YOLOv8 to detect helmets and safety vests, demonstrating the algorithm's capability in industrial settings [8].

In this study, YOLOv8 is adapted to learn and recognize Javanese Script numerals from a dataset of handwritten images, creating opportunities not only for a deeper understanding of the application of object recognition technology in digital preservation but also for broadening its use in education and research on language and culture. The results of this study are expected to provide a significant contribution to the existing literature by linking advanced object recognition technology with efforts in profound cultural preservation.

2. Method:

In this study, we adopted an experimental research design to test the ability of the YOLOv8 algorithm to recognize Javanese numerals from a dataset of handwritten samples. This approach involves systematic data collection, labeling, algorithm training, and performance evaluation using predefined metrics.

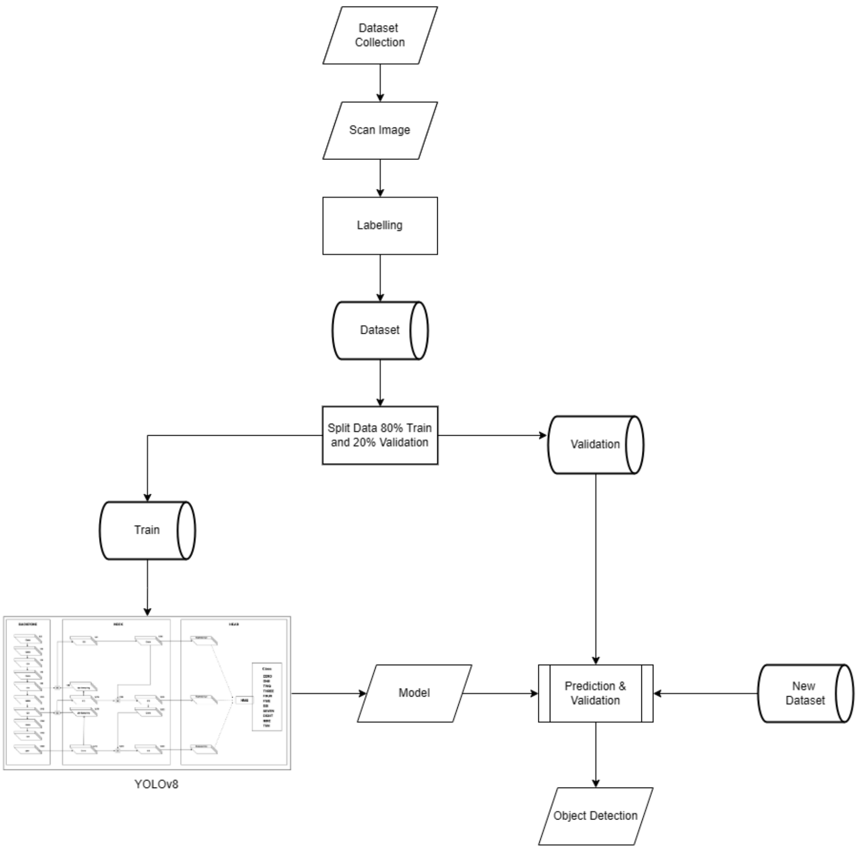


Figure 1. Research Design

Data Collection

The data collection process began with the distribution of a sheet of paper to 93 respondents, consisting of students, university students, and educators from various educational institutions in Java. Each respondent was asked to write the numbers 0 to 9 in Javanese Script three times, each in a separate column on the sheet. This approach aimed to capture sufficient variations in handwriting for a more comprehensive analysis. The sheets were then collected and scanned into a high-resolution digital format to ensure image quality suitable for further processing.

Figure 2. Sample of Dataset

Image Pre-processing

After the scanning process, the images were uploaded to the platform https://www.makesense.ai/, an online annotation tool that allows users to label objects in images. On this platform, bounding boxes were manually drawn around each numeral. Each bounding box was then associated with the corresponding class label (0–9), as shown in **Table 1**. The bounding box coordinates ($\boldsymbol{x_{min}}, \boldsymbol{y_{min}}, \boldsymbol{x_{max}}, \boldsymbol{y_{max}}$) were stored in .txt format according to the YOLO standard, where each box is defined by:

$$box = \left(\frac{x_{min} + x_{max}}{2}, \frac{y_{min} + y_{max}}{2}, x_{max} - x_{min}, y_{max} - y_{min}\right) \quad (1)$$

These coordinates were then adjusted to the dimensions of the input images so that YOLOv8 could accurately detect and recognize the numerals. The results of the annotation process can be seen in **Table 1**.

Table 1. Annotation Results

No	Class	Bounding box
1	Zero	0.860383 0.198001 0.062232 0.045492
2	Zero	0.862492 0.266239 0.049574 0.049221
…	…	..
10229	Nine	0.856268 0.708305 0.056500 0.043750
10230	Nine	0.858913 0.769028 0.057611 0.046013

Split Data

The annotated data were then divided into two subsets: 80% for training and 20% for model validation [9]. This split was performed randomly to ensure that the model generalizes well to unseen data, thereby representing the true variability in handwriting.

Implementation Algorithm

YOLOv8, like its predecessors in the YOLO (You Only Look Once) series [10], [11], [12], is an advanced deep learning algorithm designed for object detection tasks. It involves the real-time identification and classification of objects in images [11], [3]. Below is a visualization of how YOLOv8 works, as shown in **Figure 3**:

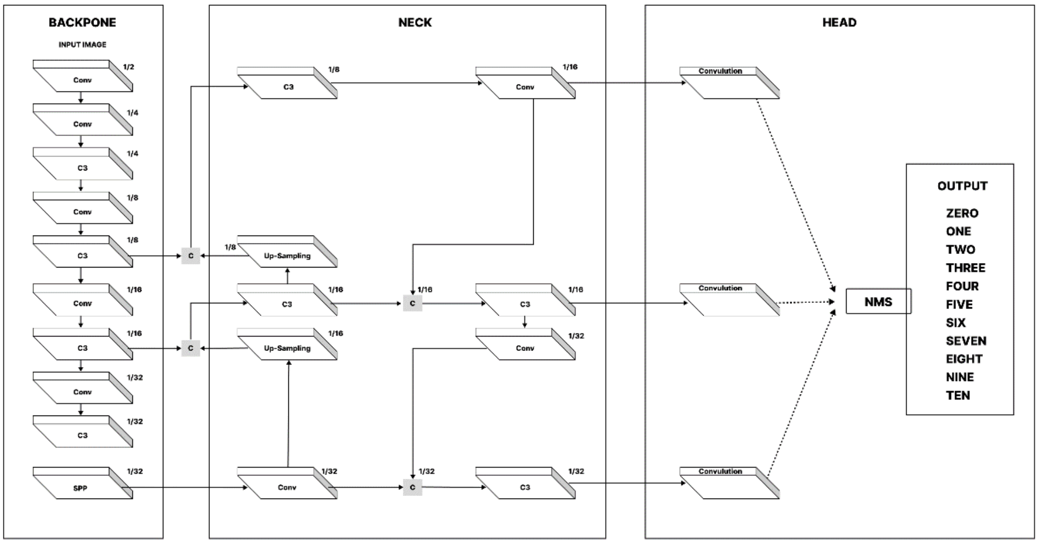


Figure 3. Workflow YOLOv8

a. Input Image: YOLOv8 takes the entire image as input.

b. Grid Division: The image is divided into an S×S grid. Each grid cell is responsible for detecting objects whose centers fall within that cell.

c. Bounding Box Prediction: Each cell predicts several bounding boxes. Each bounding box prediction includes:
   - Coordinates (center, width, height),
   - Confidence score (reflecting the accuracy of the bounding box),
   - Class probabilities (the likelihood that the object belongs to a particular class).

d. Feature Extraction: YOLOv8 employs a deep convolutional neural network to extract features from the image. This network processes the image through multiple layers of convolution, pooling layers, and fully connected layers to detect features.

e. Anchor Boxes: Predefined anchor boxes are used to improve the accuracy of bounding box predictions. These boxes are set dimensions that correspond to the typical sizes of objects in the training set.

f. Prediction Across Scales: YOLOv8 makes predictions at three different scales, enabling it to effectively detect small, medium, and large objects. This is achieved through a technique known as the Feature Pyramid Network (FPN), which extracts high-level semantic information from various depths within the network.

g. Non-max Suppression: Among the multiple predicted boxes for each object, many are redundant. Non-maximum suppression is used to eliminate overlapping boxes based on their confidence scores and a predefined Intersection over Union (IoU) threshold. This step ensures that each detected object is represented by a single bounding box.

h. Output: The final output consists of bounding boxes around the detected objects, along with their class labels and confidence scores.

The YOLOv8 model was chosen for its effectiveness in real-time object detection with high precision [13], [14]. Initial settings, such as learning rate, number of epochs, and batch size, were adjusted based on the characteristics of the Javanese Script dataset. We used the PyTorch framework to implement and train the model, incorporating data augmentation techniques such as rotation, scaling, and translation to enhance the model's robustness against variations in the data.

Data Analysis Method

The model's performance is measured based on Precision, Recall, and mean Average Precision (mAP) [15]–[18]. Precision measures the proportion of positive identifications that are correct, while Recall measures the proportion of actual positives that are correctly identified [19]–[21]. mAP is the average of the Average Precision (AP) for each class, computed with:

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

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$Average\ Precision = \int_0^1 P(R)dR \tag{4}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{5}$$

Where,

TP = true positive (when the algorithm correctly detects a numeral with a bounding box);

FP = false positive (when the algorithm predicts a bounding box in a location without a numeral);

FN = false negative (when the target numeral is not detected);

AP = a value between 0 and 1 used to summarize the different precision values obtained during recall. The AP for each class is calculated based on the area under the Precision-Recall (PR) curve generated from the model's predictions on the validation data;

N = the number of classes.

Testing

After the training and validation phases, the model is evaluated using new data, including photos and videos of Javanese handwritten text collected from various external sources, to assess its real-world performance [22]–[24]. This testing phase includes an analysis of the model's ability to recognize and detect numerals under uncontrolled conditions, which is a crucial test of the model's practical applicability in the field [25].

3. Results and Discussion

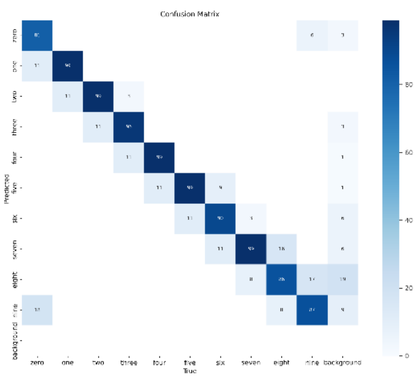
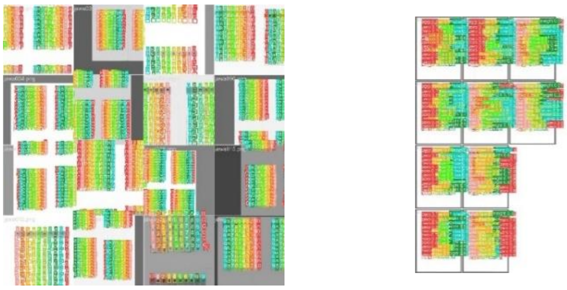


Figure 4. Confusion Matrix

The conducted research has provided significant insights into YOLOv8’s ability to recognize Javanese numerals with high accuracy. The confusion matrix demonstrates excellent performance, with most numerals being correctly identified—digits such as 'zero', 'six', and 'nine' achieving high recognition rates. However, challenges remain, such as the misclassification between similar-looking digits; for example, 'one' is sometimes mistakenly recognized as 'zero'. Overall, the confusion matrix indicates outstanding performance for most classes, with only a few misclassifications—particularly for 'eight' and 'nine'—as reflected by the relatively low number of false positives and false negatives for most classes.



(a) (b)

Figure 5. (a) Training (b) Validation

Figure 5 displays the results of the training and validation process for detecting Javanese numerals using the YOLOv8 algorithm. Each image shows Javanese numerals that have been identified and labeled by the model, with different colors representing each numeral class from 0 to 9. Colored areas indicate the model's predictions for the precise location and class of the numerals, while red indicates incorrect or imprecise predictions made by the model. This visualization is crucial for visually assessing the model's real-time accuracy in recognizing and positioning Javanese numerals, as well as for providing direct insights into the model’s ability to generalize from training data to validation data.

Table 2. Validation Results

Class	Images	Instances	Box(P)	R	mAP90	mAP50-90)
All	10	1100	0.883	0.891	0.88	0.477
Zero	10	110	0.882	0.813	0.812	0.351
One	10	110	0.894	0.9	0.896	0.453
Two	10	110	0.878	0.9	0.873	0.513
Three	10	110	0.894	0.9	0.848	0.504
Four	10	110	0.872	0.9	0.877	0.489
Five	10	110	0.896	0.865	0.894	0.442
Six	10	110	0.9	0.901	0.952	0.488
Seven	10	110	0.9	0.899	0.894	0.505
Eight	10	110	0.855	0.999	0.833	0.488
Nine	10	110	0.864	0.926	0.869	0.503

Table 2 shows that the model validation achieved a high mean Average Precision (mAP) of 0.88 at an IoU threshold of 0.90, indicating the model’s strong capability in accurately recognizing and classifying Javanese numerals.

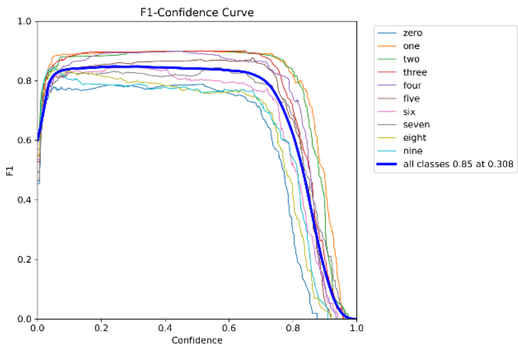


Figure 6. F1-Confidence Curve

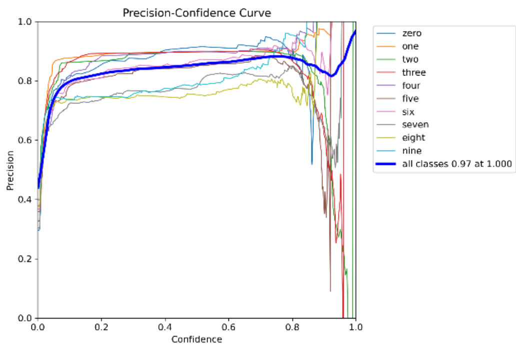


Figure 7. Precision-Confidence Curve

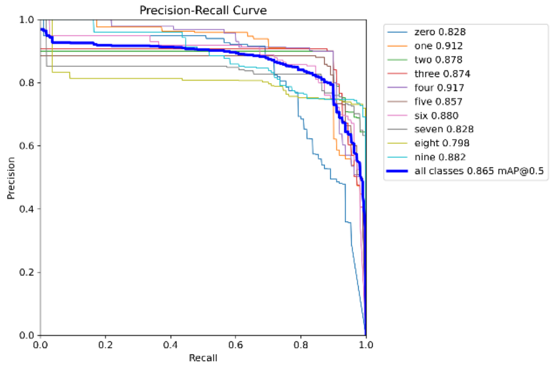


Figure 8. Precision-Recall Curve

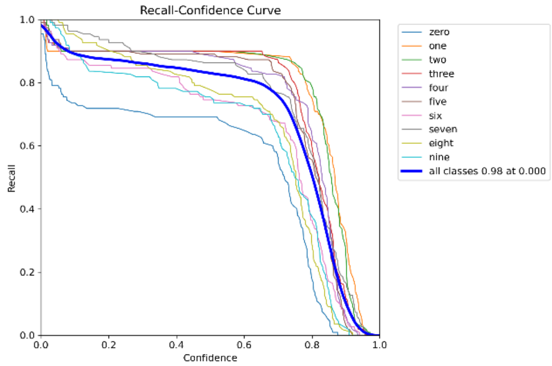


Figure 9. Recall-Confidence Curve

Figures 6 and **7** show that at a confidence threshold of 0.228, the overall F1 score reaches 0.89, and at a confidence threshold of 0.966, all classes achieve perfect precision. **Figure 8** presents consistent results with a mean Average Precision (mAP) of 0.880 at an IoU threshold of 0.5, indicating the model's effectiveness in recognizing Javanese numerals against diverse backgrounds. **Figure 9** illustrates that the overall recall reaches 0.97 at a very low confidence threshold, demonstrating the model's high sensitivity in accurately detecting numerals.

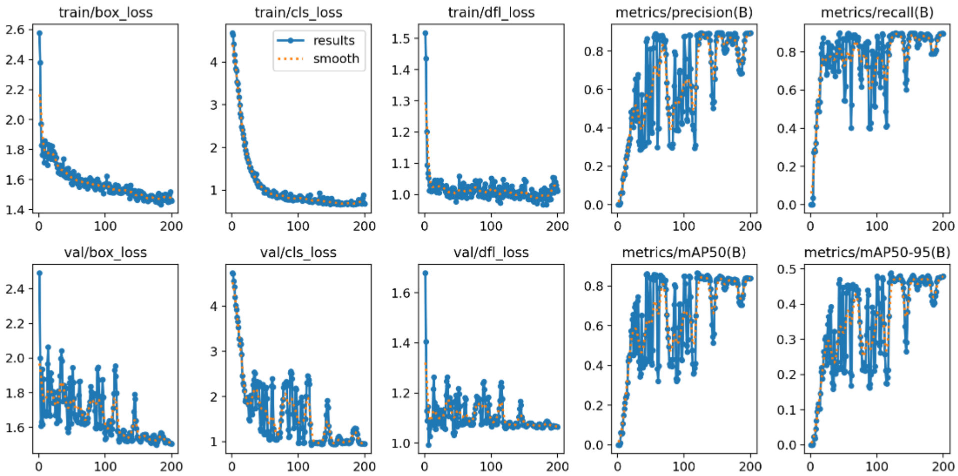


Figure 10. Results

Figure 10 presents a series of graphs documenting the training and validation process of the YOLOv8 object detection model used to recognize Javanese numerals. Each graph illustrates a different aspect of the model's performance during the training and validation iterations:

a. Train/Box Loss and Val/Box Loss: These graphs show the decrease in loss for bounding box predictions during both training and validation. The loss value drops sharply at the beginning of training and then stabilizes at a lower level, indicating that the model is progressively improving in aligning the bounding boxes with the target objects.
b. Train/Cls Loss and Val/Cls Loss: These graphs depict the classification loss during training and validation. Similar to the box loss, the classification loss shows a significant decrease at the start of training, reflecting the model's enhanced ability to accurately classify Javanese numerals.
c. Train/Dfl Loss and Val/Dfl Loss: These graphs represent the loss from the prediction of directional feature location, which is crucial for pinpointing more specific feature locations on the objects. The loss remains relatively stable and low, demonstrating that the model has effectively learned to predict feature locations.
d. Metrics/Precision(B) and Metrics/Recall(B): This graph displays the fluctuations in precision and recall throughout the training process, indicating both consistency and variability in the model’s ability to detect and recognize objects accurately. High precision at certain points signals accurate detections, while the variability in recall indicates changes in the frequency with which the model detects all relevant objects.

e. Metrics/mAP50(B) and Metrics/mAP50-95(B): These graphs show the mean Average Precision at an IoU threshold of 0.50 and across an IoU range of 0.50 to 0.95. Both metrics stabilize after the initial training phase, with mAP50 demonstrating better performance compared to the broader range. This suggests that the model is more effective at detections with a lower tolerance for positional errors.

Discussion

A deeper analysis of the results indicates that YOLOv8 can serve as a highly effective tool for handwritten character recognition, particularly for Javanese Script. This is crucial given the need to preserve local languages and cultures through effective digitization. The model shows great potential for practical applications, especially in educational systems to support the learning of Javanese Script or in the development of digital archival systems that can automatically classify and index documents based on Javanese numeral characters.

Compared to previous studies, this research demonstrates significant advancements in both accuracy and the speed of visual data processing. This suggests that with the appropriate technology, the challenges of interpreting and processing traditional languages or symbols can be minimized.

However, several limitations were identified in this study. For instance, variations in data quality and the limited quantity of data used for training and validation could affect the model's generalization in real-world scenarios. Therefore, it is recommended that future research incorporate larger and more diverse datasets and test the model under various lighting conditions and backgrounds to verify its robustness.

Furthermore, further exploration of parameters and fine-tuning techniques may lead to additional improvements in model performance. Implementing more advanced data augmentation techniques and utilizing synthetic datasets might also help overcome the current data limitations. This approach would contribute to optimizing the model for broader applications, including mobile or online platforms that require rapid and accurate responses.

4. Conclusion

This study tests the ability of YOLOv8 technology to recognize Javanese numerals from handwritten text, with results demonstrating a very high level of accuracy. The confusion matrix and analyses through various curves—F1-Confidence, Precision-Confidence, and Precision-Recall—confirm that YOLOv8 can accurately identify and differentiate numerals with high precision. The high confidence level in the model's predictions underscores its reliability in practical applications. However, challenges remain in recognizing numerals with similar shapes, indicating that despite the model's effectiveness, further adjustments and optimization are necessary.

This research makes an important contribution to the existing literature by implementing and evaluating a state-of-the-art object recognition model in the context of preserving language and culture, particularly Javanese Script. The success of YOLOv8 in this study paves the way for the development of more advanced document digitization tools that can facilitate both preservation and education. For future research, it is recommended to develop larger and more diverse datasets—not only to enhance the model's generalization capabilities but also to explore its effectiveness under a wider range of conditions. Additionally, future studies could investigate integrating this model with other machine learning systems or augmented reality technology to create interactive methods for learning Javanese Script.

References:

[1] G. G. Casas, "Automatic Detection and Counting of Stacked Eucalypt Timber Using the YOLOv8 Model," *Forests*, vol. 14, no. 12, 2023, doi: 10.3390/f14122369.

[2] X. Zhang, "Detection of Leg Diseases in Broiler Chickens Based on Improved YOLOv8 X-Ray Images," *IEEE Access*, vol. 12, pp. 47385–47401, 2024, doi: 10.1109/ACCESS.2024.3382193.

[3] V. R. del Castillo, "Adapting YOLOv8 as a Vision-Based Animal Detection System to Facilitate Herding," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 14001. pp. 603–610, 2023, doi: 10.1007/978-3-031-40725-3_51.

[4] W. Zhang, Z. Jia, C. Feng, and M. Chen, "YOLOv8-SC:Target Detection Algorithm of Live Detection

System for Tensioning Clamps in High-Voltage Transmission Lines," in *2023 8th International Conference on Control, Robotics and Cybernetics (CRC)*, Dec. 2024, pp. 177–181, doi: 10.1109/CRC60659.2023.10488490.

[5] G. Zidani, "Optimizing Pedestrian Tracking for Robust Perception with YOLOv8 and Deep sort," *Appl. Comput. Sci.*, vol. 20, no. 1, pp. 72–84, 2024, doi: 10.35784/acs-2024-05.

[6] J. Terven, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4. pp. 1680–1716, 2023, doi: 10.3390/make5040083.

[7] E. Casas, "YOLOv5 vs. YOLOv8: Performance Benchmarking in Wildfire and Smoke Detection Scenarios," *J. Image Graph.*, vol. 12, no. 2, pp. 127–136, 2024, doi: 10.18178/joig.12.2.127-136.

[8] D. Thakur, P. Pal, A. Jadhav, N. Kable, B. V, and S. Deshpande, "YOLOv8- Based Helmet and Vest Detection System for Safety Assessment," in *2023 International Conference on Network, Multimedia and Information Technology (NMITCON)*, Sep. 2023, pp. 1–10, doi: 10.1109/NMITCON58196.2023.10275958.

[9] V. R. Joseph and A. Vakayil, "SPlit: An Optimal Method for Data Splitting," *Technometrics*, vol. 64, no. 2, pp. 166–176, Apr. 2022, doi: 10.1080/00401706.2021.1921037.

[10] B. Gašparović, "Evaluating YOLOV5, YOLOV6, YOLOV7, and YOLOV8 in Underwater Environment: Is There Real Improvement?," *2023 8th Int. Conf. Smart Sustain. Technol. Split. 2023*, 2023, doi: 10.23919/SpliTech58164.2023.10193505.

[11] C. Dewi, D. Manongga, Hendry, E. Mailoa, and K. D. Hartomo, "Deep Learning and YOLOv8 Utilized in an Accurate Face Mask Detection System," *Big Data Cogn. Comput.*, vol. 8, no. 1, p. 9, Jan. 2024, doi: 10.3390/bdcc8010009.

[12] Z. Afrin, "YOLOv8 Based Object Detection for Self-driving Cars," *2023 26th International Conference on Computer and Information Technology, ICCIT 2023*. 2023, doi: 10.1109/ICCIT60459.2023.10441381.

[13] F. Solimani *et al.*, "Optimizing tomato plant phenotyping detection: Boosting YOLOv8 architecture to tackle data complexity," *Comput. Electron. Agric.*, vol. 218, p. 108728, Mar. 2024, doi: 10.1016/j.compag.2024.108728.

[14] J. C. Camacho, "Mask R-CNN and YOLOv8 Comparison to Perform Tomato Maturity Recognition Task," *Commun. Comput. Inf. Sci.*, vol. 1885, pp. 382–396, 2023, doi: 10.1007/978-3-031-45438-7_26.

[15] G. Liu, J. C. Nouaze, P. L. Touko Mbouembe, and J. H. Kim, "YOLO-Tomato: A Robust Algorithm for Tomato Detection Based on YOLOv3," *Sensors*, vol. 20, no. 7, p. 2145, Apr. 2020, doi: 10.3390/s20072145.

[16] I. A. Bakhri and H. P. Sidik, "Realtime Recognition of Handwritten Lontara Makassar Characters Using Yolov5 Algorithm," in *2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Dec. 2022, pp. 309–313, doi: 10.1109/ICITISEE57756.2022.10057924.

[17] C. Dewi, "Deep Learning and YOLOv8 Utilized in an Accurate Face Mask Detection System," *Big Data Cogn. Comput.*, vol. 8, no. 1, 2024, doi: 10.3390/bdcc8010009.

[18] P. Zhang, W. Hou, D. Wu, B. Ge, L. Zhang, and H. Li, "Real-Time Detection of Small Targets for Video Surveillance Based on MS-YOLOv5," in *2023 6th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, May 2023, pp. 690–694, doi: 10.1109/ICAIBD57115.2023.10206275.

[19] M. R. A. Priyadi, "Comparison of YOLOv8 and EfficientDet4 Algorithms in Detecting the Ripeness of Oil Palm Fruit Bunch," *10th Int. Conf. ICT Smart Soc. ICISS 2023 - Proceeding*, 2023, doi: 10.1109/ICISS59129.2023.10291928.

[20] F. Dang, D. Chen, Y. Lu, and Z. Li, "YOLOWeeds: A novel benchmark of YOLO object detectors for multi-class weed detection in cotton production systems," *Comput. Electron. Agric.*, vol. 205, p. 107655, Feb. 2023, doi: 10.1016/j.compag.2023.107655.

[21] M. Sportelli *et al.*, “Evaluation of YOLO Object Detectors for Weed Detection in Different Turfgrass Scenarios,” *Appl. Sci.*, vol. 13, no. 14, p. 8502, Jul. 2023, doi: 10.3390/app13148502.

[22] Y. Gong, “Handwritten Chinese Character Recognition in Ancient Books Based on Improved YOLOv7,” 2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology, AINIT 2024. pp. 1526–1532, 2024, doi: 10.1109/AINIT61980.2024.10581730.

[23] Q. Sun, “Recognition Method for Handwritten Steel Billet Identification Number Based on Yolo Deep Convolutional Neural Network,” Proceedings of the 32nd Chinese Control and Decision Conference, CCDC 2020. pp. 5642–5646, 2020, doi: 10.1109/CCDC49329.2020.9164720.

[24] M. I. S. Zabala, “Handwritten Text Recognition Algorithm,” IFMBE Proceedings, vol. 106. pp. 61–79, 2024, doi: 10.1007/978-3-031-61960-1_7.

[25] J. Sarmah, “Performance Analysis of Deep CNN, YOLO, and LeNet for Handwritten Digit Classification,” Lecture Notes in Networks and Systems, vol. 844. pp. 215–227, 2024, doi: 10.1007/978-981-99-8479-4_16.