

Robust Balinese Script Detection for Transliteration Using YOLOv8

Hans Juwiantho^{1*}, Jonathan Eka Saputra²

^{1,2}Informatics Department, Petra Christian University, Surabaya, East Java, Indonesia

E-mail: ^{1*}hans.juwiantho@gmail.com, ²c14210120@john.petra.ac.id

(Received: 24 Feb 2026, revised: 27 Mar 2026, accepted: 29 Mar 2026)

Abstract

YOLOv8-based detection system was developed to support automatic transliteration of Balinese script by addressing challenges related to small character size, visual similarity, and varying image quality. The study evaluates detection performance using a Balinese script dataset consisting of four dataset variations, Original, Grayscale, Augmented, and book-excluded dataset, designed to analyze the impact of preprocessing and data variation on model accuracy. The Original and Grayscale datasets demonstrating superior performance by achieving mAP@50 values above 0.93. Hyperparameter tuning experiments involving optimizer selection, epoch variation, and model scale indicated that the YOLOv8x variant with the auto optimizer and a fine-tuning strategy delivered the highest performance, reaching an mAP@50 of 0.9405 while maintaining balanced precision and recall. Confusion matrix analysis showed that most detection errors occurred between visually similar characters. Robustness evaluation further revealed that detection performance remains reliable when image clarity moderate, rotation does not exceed $\pm 3^\circ$, shear remains within $\pm 2^\circ$, and character bounding boxes occupy approximately at least 1% of the total image dimension. These findings contribute the importance of dataset preparation, model configuration, and robustness constraints to achieve accurate Balinese script detection for real-world transliteration applications.

Keywords: Balinese Script Detection, YOLOv8, Hyperparameter Tuning, Robustness Evaluation.

I. INTRODUCTION

Balinese script (*Aksara Bali*) is a traditional writing system that holds significant cultural value as part of Indonesia's heritage and regional identity, and is widely used in religious manuscripts, traditional literature, temple inscriptions, and public signage across Bali [1]. However, increasing modernization and declining proficiency in Balinese script have resulted in reduced usage in daily communication contexts. The government has carried out preservation efforts through various initiatives, ranging from curriculum implementation [2] to regulating the use of Balinese script in public facilities, as stipulated in Governor Regulation No. 80 of 2018 and Circular Letter No. 3172 of 2019 [3].

Previous studies have applied the Finite State Machine (FSM) method for Balinese script transliteration, both from Balinese script to Latin and vice versa. FSM was used to manage character transitions and syllable segmentation based on predefined states and UNICODE mapping with the Noto Serif Balinese font [4], [5]. The method offers advantages such as simplicity, predictable response, and low computational cost. Experimental results showed promising performance, with transliteration accuracy reaching 97.67%. However,

limitations were reported in handling certain transliteration cases, indicating constraints of rule-based approaches.

Subsequently, machine learning and deep learning approaches were introduced to improve recognition performance. The implementation of convolutional neural network achieved an accuracy of 81.3% [6] and 89.93% [7], while an ensemble deep convolutional neural network approach reached 93.6% accuracy [8]. Research has also shown that Faster R-CNN has been applied for Balinese character detection with high mean average precision [9]. An LSTM-based system, AKSALont, was developed for transliteration of Balinese palm-leaf manuscripts, although it still reported a relatively high character error rate of 19.78% [10]. Beyond Balinese script, studies on other Indonesian scripts such as Sundanese [11] and Bima script [12] have demonstrated that deep learning can effectively support the recognition of regional writing systems. However, most existing research is still limited to isolated character recognition rather than full word transliteration.

From a linguistic perspective, Balinese script originates from the ancient Brahmi script of India and has evolved into a complex writing system with multiple classifications [13]. These classifications include *Aksara Wyanjana* (or *Wreastra*),



This study proposes a YOLOv8-based system for transliteration of Balinese script into Latin characters. The system aims to perform character detection and segmentation as a foundation for word-level transliteration. The model is trained and evaluated using datasets that include variations in blur, rotation, and character size to assess robustness under realistic conditions. By emphasizing both accuracy and real-time performance, this research seeks to demonstrate the

The inclusion of book-based images was intended to supplement rarely used characters that were underrepresented in the other data sources, thereby improving dataset balance. Samples data from the dataset are shown in Figure 1.



All collected images were processed through a pre-processing stage to ensure data consistency and improve image quality. Image annotation was then performed using the Roboflow platform, where each Balinese script character was manually labeled using bounding boxes and assigned to one of the 75 predefined classes. Initially, a list of class names was defined based on the scope of the study and the availability of

Balinese script characters within the dataset, corresponding to the 75 predefined character classes. Each Balinese script character appearing in the images was manually annotated according to its respective label, focusing on accurate character-level localization to support detection and segmentation tasks. Example of annotation process can be seen at Figure 2.



Figure 2. Annotation Process

After annotation, the processed images were organized into a structured dataset and divided into training and validation sets using an 80:20 ratio. Additional pre-processing was conducted using Roboflow, including automatic image orientation and resizing all images to a resolution of 640×640 pixels. To evaluate the impact of color information on model performance, two dataset variants were generated: (1) a Grayscale Dataset, in which images were resized and converted to grayscale, and (2) an Original Dataset, in which images were resized without grayscale conversion. These datasets were used to analyze the effect of color information on character detection and segmentation performance.

B. YOLOv8 Model Training and Hyperparameter Tuning

The YOLOv8 training workflow utilizes the annotated and pre-processed dataset described in Dataset Preparation. After the dataset was finalized and divided into training and validation subsets, the YOLOv8 model was initialized and configured with the required training parameters, including the number of training epochs and optimizer selection, which play a crucial role in determining model convergence and performance.

The selection of pre-trained YOLOv8 models was motivated by the size of the dataset and the high visual complexity of the Balinese script images. Larger model variants, namely YOLOv8-Large (yolov8l.pt) and YOLOv8-Extra Large (yolov8x.pt), were employed in this study due to their higher parameter capacity and deeper network architecture compared to smaller variants. These models are designed to handle large-scale and complex datasets more effectively, enabling the learning of richer feature representations. Given the substantial variation in character shapes and sizes, the presence of noise, and diverse lighting conditions in the dataset, the use of larger pre-trained models is expected to improve detection accuracy and robustness in Balinese script character analysis.

Hyperparameter tuning was conducted to identify the optimal training configuration. The number of epochs was varied between 50 and 100 to analyze the effect of training duration on model learning behavior. Training with 50 epochs

provides insight into short-term learning efficiency, while increasing the number of epochs to 100 allows the model to capture more complex character-level patterns. This variation was designed to evaluate whether longer training improves accuracy without introducing overfitting, thereby achieving a balance between model performance and training efficiency.

In addition to epoch variation, optimizer selection was explored as part of the tuning process. Two optimizers were evaluated in this study: Adam and AdamW. Adam combines momentum and adaptive learning rates, enabling faster convergence during training. AdamW extends Adam by incorporating weight decay regularization, which helps reduce overfitting and improves generalization. By comparing these optimizers, the study aims to identify the most effective optimization strategy for Balinese script character detection and segmentation.

During training, the model iteratively learns character-level visual patterns from the training dataset based on bounding box annotations and class labels. Optimization is performed across multiple epochs to minimize detection loss and improve localization and classification accuracy. Upon completion of training, the resulting model is saved and evaluated using the validation dataset. Model performance is assessed using quantitative metrics to compare different combinations of model scale, epochs, and optimizers. The results of these experiments are summarized in a comparative evaluation table, which is discussed further in the Results and Discussion section.

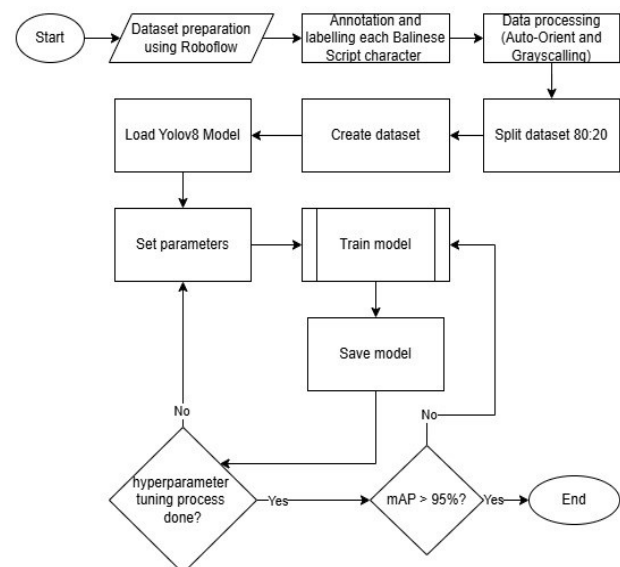


Figure 3. Flowchart YOLOv8 Training

After the training process, the best-performing model weight was selected based on validation performance. This model was then integrated into the inference pipeline for Balinese script detection and recognition. During inference, input images are processed by the trained YOLOv8 model to detect individual Balinese script characters. The output of this process consists of bounding box coordinates and corresponding character class labels, which form the basis for

subsequent transliteration. Flowchart of YOLOv8 training can be seen at Figure 3.

C. Model Evaluation Metrics

To evaluate the performance of the YOLOv8 model on Balinese script transliteration system, Mean Average Precision (mAP) is used. mAP is widely adopted for assessing detection accuracy in object detection tasks. Mean Average Precision measures the model ability to correctly localize and classify objects by calculating the average precision across all character classes at specific Intersection over Union (IoU) thresholds. mAP at an IoU threshold of 0.5 (mAP@50) was used to assess detection accuracy.

In addition to mAP, precision and recall metrics were also considered to analyze the balance between false positives and false negatives in character detection. Precision reflects the proportion of correctly detected Balinese characters among all detections, while recall indicates the proportion of correctly detected characters relative to the total number of ground truth characters.

These evaluation metrics were used to compare different training configurations, including variations in the number of training epochs, optimizer selection, and dataset pre-processing strategies. The evaluation results form the basis for selecting the best-performing model configuration for subsequent transliteration and implementation stages.

III. RESULTS AND DISCUSSION

A. Dataset Variation and Baseline Performance

Training was conducted using two additional dataset variations, namely augmented datasets and datasets that excluded images sourced from books. The four types of datasets are shown in Table 1.

Tabel 1. Dataset List

Dataset Name	Number of Images	Explanation
Original Dataset	520	Original Dataset
Grayscale Dataset	520	Original Dataset converted to Grayscale
Augmented Dataset	1107	Increase the number of Original Dataset train data to 881 and valid to 226
Pure Dataset	266	Original Dataset without using images from books

Datasets evaluated in this study demonstrated varied performance. Among them, the Original Dataset and Grayscale Dataset exhibited significantly better performance, achieving mAP@50 values exceeding 0.85 and outperforming the other datasets using 100 training epochs and the auto optimizer. This improved performance may be attributed to the preservation of essential structural features in the Original dataset and the reduction of color-related noise in the Grayscale dataset, which helps the model focus on

character shapes and stroke patterns that are critical for accurate Balinese script detection.

From the results of the dataset testing that has been carried out in Table 2, the most suitable datasets for the YOLOv8 method are the Original Dataset and Grayscale Dataset with mAP@50 results that are much higher than the Augmented Dataset and Pure Dataset.

Table 2. Performance From Different Datasets

Varian	Datasets	mAP@50
yolov8l.pt	Original Dataset	0.931439
yolov8l.pt	Grayscale Dataset	0.933464
yolov8l.pt	Augmented Dataset	0.736829
yolov8l.pt	Pure Dataset	0.808216
yolov8x.pt	Original Dataset	0.939836
yolov8x.pt	Grayscale Dataset	0.926144
yolov8x.pt	Augmented Dataset	0.747810
yolov8x.pt	Pure Dataset	0.834936

B. Hyperparameter Tuning Analysis

Through a series of hyperparameter tuning experiments, the performance of YOLOv8 was evaluated by comparing different optimizers (Adam and AdamW), epoch settings, model variants, and dataset configurations (Table 3). The best result from the hyperparameter tuning experiments was achieved using the YOLOv8l variant trained for 100 epochs with the Adam optimizer on the Original Dataset, yielding an mAP@50 score of 0.9355. However, despite this improvement over other tuned configurations, the result did not surpass the performance of a previous training setup that used the YOLOv8x variant with the auto optimizer, which achieved a higher mAP@50 score. This indicates that larger model variants with automated optimization strategies may provide superior performance compared to manual hyperparameter tuning on smaller variants.

Table 3. Performance From Hyperparameter Tuning

		Adam		AdamW	
		50 epochs	100 epochs	50 epochs	100 epochs
Original Dataset	Large	0.910	0.935	0.903	0.931
	XLarge	0.908	0.932	0.894	0.930
Grayscale Dataset	Large	0.915	0.923	0.897	0.932
	XLarge	0.910	0.897	0.885	0.899

After several training experiments, the mAP@50 score had not yet reached the target value of 0.95. Therefore, an additional training phase was conducted using the fine-tuning function provided by YOLOv8 to perform weight transfer from the previously trained model. This approach allows the model to continue learning from pre-trained weights rather than starting from scratch. The additional training utilized the Original Dataset, which consistently produced the highest average mAP@50 among all evaluated datasets. Furthermore, the experiment employed the YOLOv8x variant with 100 epochs and the “auto” optimizer, following the best-performing configuration identified in earlier experiments. In



this experiment, the highest mAP@50 achieved was 0.9405. In addition to mAP@50, further evaluation was conducted using accuracy, precision, recall, and F1-score to provide a more comprehensive assessment of the model’s detection performance. The detailed results are presented in Table 4 and Figure 4.

Table 4. Performance Metrics of YOLOv8 Hyperparameter Tuning Experiments.

Model	Precision	Recall	F1 – Score	Accuracy	mAP@50
A	0.8945	0.9085	0.8981	89.46%	0.9314
B	0.8732	0.8507	0.8404	87.32%	0.9104
C	0.8952	0.8883	0.8859	89.53%	0.9355
D	0.8506	0.8492	0.8365	85.07%	0.9039
E	0.8690	0.8994	0.8779	86.90%	0.9318
F	0.9004	0.9039	0.8917	90.05%	0.9398
G	0.8688	0.8496	0.8458	86.88%	0.9082
H	0.8733	0.8887	0.8680	87.33%	0.9325
I	0.8410	0.8651	0.8370	84.10%	0.8944
J	0.8887	0.9011	0.8903	88.88%	0.9307
K	0.9079	0.9023	0.9008	90.79%	0.9334
L	0.8186	0.9012	0.8498	81.86%	0.9159
M	0.8813	0.8754	0.8655	88.14%	0.9235
N	0.8774	0.8286	0.8328	87.74%	0.8970
O	0.8894	0.8721	0.8673	88.95%	0.9325
P	0.8875	0.8656	0.8560	88.75%	0.9261
Q	0.8698	0.8387	0.8406	86.99%	0.9106
R	0.8550	0.8352	0.8236	85.51%	0.8976
S	0.8360	0.8597	0.8409	83.61%	0.8850
T	0.8431	0.8642	0.8385	84.32%	0.8998
U	0.8466	0.6166	0.6885	84.66%	0.7368
V	0.7847	0.7533	0.7336	78.48%	0.8082
W	0.8394	0.6509	0.7091	83.95%	0.7478
X	0.8783	0.7517	0.7735	87.83%	0.8349
Y	0.9133	0.8968	0.8994	91.34%	0.9405

Among the evaluated configurations, Model Y (fine-tuning) achieved the highest performance with mAP@50 score of 0.9405, accompanied by a precision of 0.9133 and recall of 0.8968, resulting in an F1-score of 0.8994. This indicates a well-balanced detection capability with both high precision and recall. In contrast, several configurations such as Model U and Model W that using augmented dataset, demonstrated significantly lower recall values, which negatively affected their overall mAP performance. These results suggest that balanced optimization between precision and recall plays a crucial role in achieving superior transliteration detection performance.

Although Model Y achieved the highest mAP@50, Model F (the highest mAP from Table 2) demonstrated competitive performance with an mAP@50 of 0.9398, only slightly lower than the best configuration. Notably, Model F achieved a higher recall (0.9039) compared to the best-performing model, indicating a stronger ability to detect relevant objects. However, its slightly lower precision suggests a marginal increase in false positive detections. In the context of Balinese

script transliteration, higher recall may be preferable to minimize missed character detections, even at the cost of a slight increase in false positives.

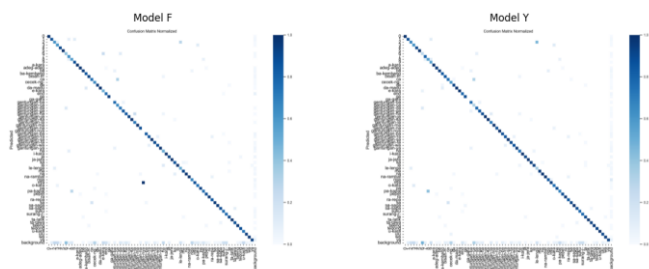


Figure 4. Confusion Matrix of Model F and Y

Model F exhibits a strong diagonal pattern in the confusion matrix, indicating that most classes are correctly classified. However, several notable misclassifications (false positives) are observed in specific classes, particularly among visually similar characters. Further analysis indicates that these errors primarily occur between characters such as “nya” and “gantungan-nya,” which share similar stroke patterns and structural features. As illustrated in Figure 5, the primary distinction between these characters lies in their positional usage and subtle morphological differences, which may not be sufficiently captured by the YOLOv8 detection model that relies mainly on spatial and visual features. Additionally, variations in character size, cropping boundaries, and limited training samples for certain classes may further contribute to the increased confusion between these visually similar characters. Consequently, although Model F demonstrates strong classification capability, Models Y achieve better overall performance by providing more consistent detection across multiple classes and reducing misclassification among structurally similar characters.

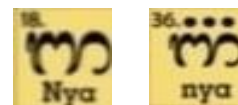


Figure 5. Characters “nya” and “gantungan-nya”

C. Robustness Evaluation

To identify the limitations of the Balinese script character detection system, this study conducted robustness evaluations under two main conditions: image quality degradation and geometric variations. Image quality testing focused on blur-related factors, including sharpness (Laplacian variance), clarity (Laplacian x contrast), and edge detail (Sobel operator). In addition, geometric robustness was evaluated through controlled variations in image tilt, shear, and character size relative to the overall image.

Image sharpness, determined by the variance of the Laplacian operator, reflects the level of focus within an image. A sharpness value below 50 indicates a blurred image, values between 50 and 150 represent moderate focus, and values above 150 indicate a highly sharp image. Clarity serves as a general indicator of overall image quality. Images with clarity

values below 1000 are typically considered low quality, values between 1000 and 3000 indicate moderate, and values above 3000 represent clear and well-defined images. Resolution values below 50 generally indicate low detail, values between 50 and 150 represent moderate detail, and values above 150 correspond to high-detail images.

Table 5. Robustness Evaluation

Sharpness		Clarity		Resolution		mAP@50
Value	Level	Value	Level	Value	Level	
135	High	7156	High	844	High	0.6767
51	Moderate	2642	Moderate	719	High	0.6167
29	Low	1470	Moderate	643	High	0.5622
18	Low	871	Low	578	High	0.5012
11	Low	548	Low	519	High	0.4426
8	Low	387	Low	475	High	0.3877

Based on the results presented in Table 5, the first image, which has sharpness, clarity, and resolution values of 135, 7156, and 844 respectively (all categorized as high), achieved mAP@50 score of 0.6767. This confirms that superior image quality contributes positively to detection performance. In contrast, images with low sharpness and low clarity resulted mAP@50 values below 0.50, indicating a significant degradation in detection capability. These findings suggest that sharpness and clarity play a more critical role than resolution in determining detection reliability. A combination of low sharpness and moderate clarity represents the minimum threshold required to maintain detection performance above 50%, with mAP@50 of 0.5622. Therefore, this condition can be considered the lowest acceptable image quality level at which the system is still able to generate more correct detections than incorrect ones.

To evaluate the impact of geometric variations on detection performance, image tilt testing and shear variation were conducted using the imgaug library. Image tilt testing was performed by rotating the test images into several predefined angle ranges and shear variation. Shear variation testing was applied by introducing controlled shear transformations. This approach was applied to simulate rotated images within the test dataset, ensuring that the evaluation included variations in image orientation. Each test image was rotated and incrementally from 1° to 10°, followed by detection performance evaluation. The selection of small and gradual rotation angles was intended to simulate natural variations in handwriting and real-world image capture conditions. Such geometric transformations reflect practical scenarios where images are not perfectly aligned during acquisition.

Table 6. Image Tilt Testing

Rotation Angle (Degrees)	mAP@50
-1°, 1°	0.7102
-2°, 2°	0.6090
-3°, 3°	0.5320
-4°, 4°	0.4166
-5°, 5°	0.3444

-10°, 10°	0.2394
-----------	--------

Table 6 illustrates the relationship between rotation angle and the resulting mAP@50 performance. As a reference, mAP@50 value above 0.50 is considered the minimum threshold indicating that the system produces more correct detections than incorrect ones, thus reflecting acceptable detection reliability. The results demonstrate a clear declining trend in mAP@50 as the rotation angle increases. When the object is rotated within the range of -1° to 1°, the system achieves mAP@50 of 0.7102, indicating very strong detection performance. Similarly, at -2° to 2°, the mAP@50 remains above the threshold at 0.6090. However, as the rotation range increases to -3° to 3°, the mAP@50 decreases to 0.5320. Although this value is still above 0.50, the decline suggests that additional geometric variation begins to challenge the detection model. Once the rotation reaches or exceeds -4° to 4°, the mAP@50 drops below the acceptable threshold, with a value of 0.4166, and decreases further at ±5° (0.3444) and ±10° (0.2394). These findings indicate that the detection system has limited robustness against larger rotational variations. This behavior suggests that the detection is optimized for small geometric deviations. As rotation angle increases, the visual features become less aligned with the learned representations, reducing detection accuracy. Therefore, the system performs optimal when rotation does not exceed ±3°.

Table 7. Shear Variation Testing

Shear Variation	mAP@50
-1°, 1°	0.7013
-2°, 2°	0.5605
-3°, 3°	0.4303
-4°, 4°	0.3551
-5°, 5°	0.2935
-10°, 10°	0.1620

Table 7 illustrates the relationship between shear transformation and mAP@50 performance. The results show that detection accuracy remains above the 0.50 threshold only within the shear range of ±1° (mAP@50 = 0.7013) and slightly above the threshold at ±2° (mAP@50 = 0.5605). However, when shear increases beyond ±2°, performance declines significantly, dropping to 0.4303 at ±3° and decreasing further as the shear magnitude becomes larger. At extreme shear levels of ±10°, mAP@50 falls drastically to 0.1620, indicating substantial detection failure. These findings suggest that the system performs optimal under limited geometric distortion, and shear transformations should ideally be restricted to within ±2° to maintain acceptable detection accuracy.

Based on experiments conducted by cropping the original images into smaller segments, it was found that enlarging Balinese script characters through image cropping significantly improved the YOLO model's detection performance. A total of six test images contained relatively small character sizes compared to the overall image

dimensions. The comparison results from the experiments conducted are presented in Table 8.

Table 8. Image Cropping Testing

Number of Crops	Final Image size	Width Percen tage	Height Percen tage	Confidence
4	561x147	2.3%	9.1%	0.75
3	467x254	4.1%	7.9%	0.74
2	960x832	2.6%	3.0%	0.79
2	960x841	2.6%	3.1%	0.80
4	789x250	2.5%	7.3%	0.73
4	332x109	9.8%	2.2%	0.68
3	886x803	4.7%	4.4%	0.89

Table 8 presents the results of the image cropping experiments conducted using various image sizes and numbers of crops, along with the resulting cropped outputs and the percentage of bounding box dimensions relative to overall image width and height. Based on Table 8, larger bounding box proportions generally produce more representative cropped images. For example, images with an initial size of 1706×960 cropped to 561×147 and 332×109 show bounding box proportions exceeding 9% in either width or height. This suggests that scaling in a single dimension (weight or height) does not necessarily guarantee improved detection performance. In contrast, images with more balanced bounding box proportions across both width and height achieved more stable and higher confidence scores, with the best result reaching 0.89.

The model performs optimal when the character bounding box occupies a sufficient and proportionally balanced area of the image, with a minimum threshold of approximately 1% of the total image dimension. Cropping strategies that enlarge small characters while maintaining proportional consistency improve detection confidence in the YOLO-based transliteration system.

The results show that the system can maintain detection performance above 0.5 mAP@50 under specific minimum quality conditions. Detection remains reliable when image sharpness reaches at least 29 (low) with a clarity value of 1470 (moderate). In terms of geometric variations, the system performs well when image tilt does not exceed $\pm 3^\circ$ and shear transformation stays within the range of -2° to 2° . Additionally, the character bounding box must have a minimum 1% the total image dimension with balanced bounding box proportions across both width and height.

IV. CONCLUSION

This study evaluated the performance of the YOLOv8 model for Balinese script character detection across different dataset variations, hyperparameter configurations, and robustness conditions. The results demonstrate that dataset selection significantly affects detection accuracy, where the Original and Grayscale Datasets consistently outperformed the

Augmented and Pure Datasets. Hyperparameter tuning showed that although manual optimization improved performance, the YOLOv8x variant with the auto optimizer and fine-tuning strategy achieved the best overall result, reaching mAP@50 of 0.9405 with well-balanced precision and recall. Confusion matrix analysis further revealed that most misclassifications occurred between visually similar characters, indicating that structural similarity remains a primary challenge in transliteration detection.

Robustness evaluation identified the operational limits of the system under image quality degradation and geometric variations. Detection performance remained reliable when moderate clarity, rotation did not exceed $\pm 3^\circ$, shear transformation remained within $\pm 2^\circ$, and character bounding boxes occupied at least approximately 1% of the total image dimension with balanced proportions. Performance declined significantly beyond these thresholds. These findings confirm that maintaining image quality, controlled geometric variation, and sufficient character size is essential to ensure stable and accurate detection in real-world applications.

REFERENCES

- [1] K. R. Hardiyanti, I. P. Purnami, and I. B. P. M. Aryana, "Meningkatkan Kemampuan Menulis Aksara Bali Menggunakan Aplikasi Transliterasi Aksara Bali Kelas XI MIA 1," *Jurnal Pendidikan Bahasa Bali Undiksha*, vol. 9, no. 1, pp. 1–10, Mar. 2022, doi: 10.23887/jpbb.v9i1.35934.
- [2] Governor, "Peraturan Daerah (Perda) Provinsi Bali Nomor 1 Tahun 2018 tentang Bahasa, Aksara, dan Sastra Bali." Accessed: Feb. 10, 2026. [Online]. Available: <https://peraturan.bpk.go.id/Details/85710/perda-prov-bali-no-1-tahun-2018>
- [3] Governor, "Bali Governor Regulation number 80 of 2018 About Protection and Use Of Bali, Aksara, And Literature as well as the Implementation of the Bali language," <https://jdih.baliprov.go.id/dokumen-hukum/b89a7884-30d0-4826-900b-9b53ecb112c3>. Accessed: Feb. 10, 2026. [Online]. Available: <https://jdih.baliprov.go.id/dokumen-hukum/b89a7884-30d0-4826-900b-9b53ecb112c3>
- [4] D. A. Putu *et al.*, "Pengembangan Metode Pemilahan Suku Kata Pada Transliterasi Teks Latin Ke Aksara Bali Berbasis Finite State Machine Dengan Font Noto Serif Balinese," *Jurnal Ilmu Komputer Indonesia (JIK)*, vol. 6, no. 2, 2021, doi: <https://doi.org/10.23887/jik.v6i2.3659>.
- [5] C. Oka Birawidya, G. Indrawan, and I. A. Gunadi, "Pengembangan Transliterasi Teks Aksara Bali Ke Latin Menggunakan Finite State Machine," *Jurnal Ilmu Komputer Indonesia (JIK)*, vol. 7, no. 1, 2022, doi: <https://doi.org/10.23887/jik.v7i1.3851>.
- [6] I. G. N. A. C. P. I Gusti Ngurah, I. K. A. A. Putra, I. B. G. Dwidasmara, M. Widiartha, N. A. S. ER, and I. P. G. H. Suputra, "Implementasi Metode Convolutional Neural Network Pada Pengenalan Aksara Bali Berbasis Game Edukasi," *SINTECH (Science and Information*

- Technology) Journal*, vol. 6, no. 1, pp. 1–15, Apr. 2023, doi: 10.31598/sintechjournal.v6i1.1298.
- [7] K. A. S. Wijaya and W. Widyastuti, “Recognition of Balinese letters with convolutional neural network,” 2022, p. 030009. doi: 10.1063/5.0103942.
- [8] D. A. S. Dewi, D. M. S. Arsa, G. A. A. Putri, and N. L. P. L. S. Setiawati, “Ensembling Deep Convolutional Neural Networks For Balinese Handwritten Character Recognition,” *ASEAN Engineering Journal*, vol. 13, no. 3, pp. 133–139, Aug. 2023, doi: 10.11113/aej.v13.19582.
- [9] A. A. Pratama, M. D. Sulistiyo, and A. F. Ihsan, “Balinese Script Handwriting Recognition Using Faster R-CNN,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 6, pp. 1268–1275, Nov. 2023, doi: 10.29207/resti.v7i6.5176.
- [10] M. W. A. Kesiman and K. T. Dermawan, “AKSALont: Automatic transliteration application for Balinese palm leaf manuscripts with LSTM Model,” *Jurnal Teknologi dan Sistem Komputer*, vol. 9, no. 3, pp. 142–149, Jul. 2021, doi: 10.14710/jtsiskom.2021.13969.
- [11] Muhammad Husni Wahid and Erik Iman Heri Ujianto, “Efficient Pattern Recognition of Sundanese Script Variants Using CNN,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 8, no. 6, pp. 808–818, Dec. 2024, doi: 10.29207/resti.v8i6.6122.
- [12] A. A. Handoko, M. A. Rosid, and U. Indahyanti, “Implementasi Convolutional Neural Network (CNN) Untuk Pengenalan Tulisan Tangan Aksara Bima,” *SMATIKA JURNAL*, vol. 14, no. 01, pp. 96–110, Jul. 2024, doi: 10.32664/smatika.v14i01.1196.
- [13] I. M. D. R. Mudiarta *et al.*, “Balinese character recognition on mobile application based on tesseract open source OCR engine,” *J. Phys. Conf. Ser.*, vol. 1516, no. 1, p. 012017, Apr. 2020, doi: 10.1088/1742-6596/1516/1/012017.
- [14] Ilham Fikriansyah, “Aksara Bali Lengkap Beserta Arti dan Aturan Menulisnya.” Accessed: Feb. 10, 2026. [Online]. Available: <https://www.detik.com/bali/budaya/d-6366936/aksara-bali-lengkap-beserta-arti-dan-aturan-menulisnya>
- [15] S. Hudaa, “Transliterasi, Serapan, dan Padanan Kata: Upaya Pemutakhiran Istilah dalam Bahasa Indonesia,” *SeBaSa*, vol. 2, no. 1, pp. 1–6, May 2019, doi: 10.29408/sbs.v2i1.1346.
- [16] I. K. Paramarta, A. M. Mbete, I. B. P. Yadnya, and A. A. P. Putra, “Transliteration From Latin Into Balinese Script (Aksarabali) Using Computerized Program Of Bali Simbar,” *e-Journal of Linguistics*, 2015, [Online]. Available: <https://ojs.unud.ac.id/index.php/eol/article/view/11972>
- [17] R. Sapkota, D. Ahmed, and M. Karkee, “Comparing YOLOv8 and Mask R-CNN for instance segmentation in complex orchard environments,” Oct. 2025, doi: 10.1016/j.aiia.2024.07.001.
- [18] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, “A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS,” *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, pp. 1680–1716, Nov. 2023, doi: 10.3390/make5040083.
- [19] T. M. Al-Hasan, V. Bonnefille, and F. Bensaali, “Enhanced YOLOv8-Based System for Automatic Number Plate Recognition,” *Technologies (Basel)*, vol. 12, no. 9, p. 164, Sep. 2024, doi: 10.3390/technologies12090164.
- [20] L. Gumirang, A. Pramono, and G. Triyono, “Integration of Yolov8 and OCR As E-KTP Data Extraction and Validation Solution for Digital Administration Automation,” *Eduvest - Journal of Universal Studies*, vol. 5, pp. 13954–13965, Feb. 2025, doi: 10.59188/eduvest.v5i11.52365.
- [21] R. Laroca, L. A. Zanlorensi, G. R. Gonçalves, E. Todt, W. R. Schwartz, and D. Menotti, “An efficient and layout-independent automatic license plate recognition system based on the YOLO detector,” *IET Intelligent Transport Systems*, vol. 15, no. 4, pp. 483–503, Apr. 2021, doi: 10.1049/itr2.12030.
- [22] N. P. Sutramiani, N. Suciati, and D. Siahaan, “MAT-AGCA: Multi Augmentation Technique on small dataset for Balinese character recognition using Convolutional Neural Network,” *ICT Express*, vol. 7, no. 4, pp. 521–529, Dec. 2021, doi: 10.1016/j.icte.2021.04.005.
- [23] Z. Liu, R. Song, K. Li, and Y. Li, “From Detection to Understanding: A Systematic Survey of Deep Learning for Scene Text Processing,” *Applied Sciences*, vol. 15, no. 17, p. 9247, Aug. 2025, doi: 10.3390/app15179247.
- [24] R. Harizi, R. Walha, and F. Drira, “Towards complex scene text reading with selective region proposal and two-stage deep reinforcement learning,” *Appl. Soft Comput.*, vol. 170, p. 112701, Feb. 2025, doi: 10.1016/j.asoc.2025.112701.

