JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8217

PERFORMANCE COMPARISON OF EASYOCR AND PADDLEOCR IN YOLO-BASED INDONESIAN LICENSE PLATE RECOGNITION

Riki Afriansyah[1*]; Muhammad Setya Pratama[1]; Vanessa Angelika[1]

Software Engineering Technology[1]
Politeknik Manufaktur Negeri Bangka Belitung, Sungailiat, Bangka Belitung[1]
www.polman-babel.ac.id[1]
riki.afriansyah@polman-babel.ac.id[*]

(*) Corresponding Author
(Responsible for the Quality of Paper Content)

Abstract— *Vehicle license plate recognition plays an important role in intelligent transportation systems for automated traffic monitoring and management, yet real world deployment remains challenging due to variations in illumination, image quality, and plate condition. This study designs and evaluates an artificial intelligence based license plate recognition system combining object detection and character recognition. The detection model was trained on the publicly available Roboflow "plate-indonesia-uakjc" dataset (v1, CC BY 4.0), comprising 661 training, 108 validation, and 92 test images of Indonesian plates; the 92 image test set combines 44 field collected images from Bangka Belitung Province and 48 images from other Indonesian provinces. The proposed pipeline detects license plates using YOLOv8 Nano, then crops and recognizes characters using EasyOCR and PaddleOCR. Detection performance was evaluated using precision, recall, and mean average precision (mAP), while OCR performance was assessed using plate level accuracy, character level accuracy, and Character Error Rate (CER). YOLOv8 achieved strong detection performance, with precision of 0.9975, recall of 1.0000, mAP@0.5 of 0.9950, and mAP@0.5:0.95 of 0.8328. For character recognition, EasyOCR outperformed PaddleOCR on the full 92 image test set, achieving 78.26% plate accuracy (CER: 3.26%) versus 72.83% (CER: 7.03%) for PaddleOCR, with character accuracies of 96.74% and 92.97%, respectively. Error analysis shows that most errors arise from confusion between "Q" and "O" and from border artifacts in bounding box crops of older plates with white frames. These findings demonstrate that YOLOv8 combined with EasyOCR provides an effective, reproducible approach for Indonesian license plate recognition under real world conditions.*

Keywords: *Character Recognition, EasyOCR, License Plate Recognition, Object Detection, PaddleOCR.*

Intisari— *Identifikasi plat nomor kendaraan merupakan unsur penting dalam sistem transportasi cerdas untuk mendukung pengawasan dan pengelolaan lalu lintas otomatis, namun implementasinya di lingkungan nyata masih menghadapi tantangan berupa variasi iluminasi, kualitas citra, dan karakteristik fisik plat nomor. Penelitian ini merancang dan mengevaluasi sistem identifikasi plat nomor berbasis kecerdasan buatan melalui deteksi objek dan pengenalan karakter. Model deteksi dilatih menggunakan dataset publik Roboflow "plate-indonesia-uakjc" (v1, CC BY 4.0) terdiri atas 661 citra latih, 108 citra validasi, dan 92 citra uji; test set mencakup 44 citra lapangan dari Provinsi Bangka Belitung dan 48 citra plat nomor dari provinsi lain di Indonesia. Pendekatan ini mencakup deteksi plat nomor menggunakan YOLOv8 Nano, dilanjutkan pemotongan citra dan pengenalan karakter menggunakan EasyOCR dan PaddleOCR. Performa deteksi dievaluasi menggunakan precision, recall, dan mean average precision (mAP), sedangkan performa pengenalan karakter diukur menggunakan plate recognition accuracy, character recognition accuracy, dan Character Error Rate (CER). Hasil penelitian menunjukkan model YOLOv8 mencapai performa deteksi tinggi dengan precision 0,9975, recall 1,0000, mAP@0.5 sebesar 0,9950, dan mAP@0.5:0.95 sebesar 0,8328. Pada tahap pengenalan karakter, EasyOCR menunjukkan kinerja lebih baik dibandingkan PaddleOCR pada 92 citra uji, dengan plate recognition accuracy 78,26% (CER: 3,26%) berbanding 72,83% (CER: 7,03%) untuk*

PaddleOCR, serta character recognition accuracy masing-masing 96,74% dan 92,97%. Analisis kesalahan menunjukkan kesalahan pengenalan terutama disebabkan oleh kemiripan karakter, khususnya antara huruf "Q" dan "O", serta artefak bingkai plat nomor lama akibat pemotongan bounding box. Kombinasi YOLOv8 dan EasyOCR terbukti efektif dan dapat direproduksi untuk identifikasi plat nomor kendaraan Indonesia pada kondisi lapangan.

Kata Kunci*: Pengenalan Karakter, EasyOCR, Pengenalan Plat Nomor, Deteksi Objek, PaddleOCR.*

INTRODUCTION

The rapid proliferation of motorized vehicles in Indonesia, driven by sustained economic growth, has intensified the demand for efficient Automatic Number Plate Recognition systems to support traffic management, law enforcement, and parking security [1], [2]. These systems encounter significant hurdles including inconsistent illumination, varied plate configurations, and diverse environmental factors, which elevate the risk of erroneous detections in real world deployments [3] .

Traditional methods relying on edge detection and template matching have exhibited limitations in handling such variabilities, prompting a shift toward deep learning paradigms like YOLOv8 integrated with OCR engines for enhanced robustness in license plate detection and character recognition [4], [5]. YOLOv8 achieves this enhancement through refined architectural optimizations, deeper feature extraction layers, and efficient transfer learning strategies that enable superior precision and real time adaptability across diverse surveillance scenarios [6].

EasyOCR and PaddleOCR are widely used open source frameworks for optical character recognition in scene text recognition tasks. EasyOCR adopts a Convolutional Recurrent Neural Network (CRNN) architecture combined with Connectionist Temporal Classification (CTC) decoding to perform character recognition. Meanwhile, PaddleOCR employs the PP OCR pipeline, which integrates a Differentiable Binarization (DB) text detector with CRNN or attention based recognition models to achieve robust text detection and recognition [14].

Indonesian vehicle license plates typically use bold sans serif fonts with relatively consistent stroke widths on a dark background. These visual characteristics influence OCR recognition performance, as CRNN based models rely on sequential feature extraction that may be sensitive to variations in stroke thickness, illumination conditions, and background noise. Therefore, evaluating OCR performance on Indonesian license plates requires careful consideration of these visual characteristics. These features affect OCR performance variably across different architectures. EasyOCR, based on CRNN sequence modeling, generally excels on license plates with even stroke widths and regular character spacing. Conversely, PaddleOCR's detection recognition pipeline proves more vulnerable to background noise and bounding box errors, potentially impairing recognition accuracy.

Their efficacy has been substantiated in diverse ALPR contexts, such as Egyptian and Bangladeshi license plate recognition, where integration with YOLO variants yields robust performance amid regional script complexities and suboptimal imaging conditions [8][9]. This integration underscores the necessity for rigorous performance benchmarking of EasyOCR and PaddleOCR within YOLOv8 frameworks specifically tailored to Indonesian license plates, addressing challenges like regional font variations and real world imaging distortions[9], [10], [11].

EasyOCR is a deep learning–based optical character recognition library that utilizes convolutional neural networks and recurrent neural networks to perform multilingual text recognition with minimal configuration [12]. Optical character recognition encompasses the automated extraction of textual information from images or scanned documents, integrating text detection to localize regions of interest and recognition to decode character sequences, often enhanced by deep learning frameworks such as CNNs and RNNs for contextual understanding [13], [14].

The research design employs a hybrid pipeline integrating YOLOv8 for precise license plate localization with EasyOCR and PaddleOCR for character extraction, benchmarked on the publicly available Roboflow "plate-indonesia-uakjc" dataset (Indonesian plates) and real world Indonesian field images captured in Bangka Belitung Province under variable illumination and occlusion conditions [15], [16]. YOLO's principles hinge on a one stage detection framework that regressively predicts bounding boxes and class probabilities directly from full images in a single forward pass, leveraging convolutional feature pyramids for real time multi scale object localization [17], [18], [19]. YOLOv8 employs an anchor free object detection paradigm by directly regressing bounding box coordinates.

VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8217

This innovation simplifies the architecture and boosts inference efficiency, especially for objects like license plates with diverse aspect ratios.

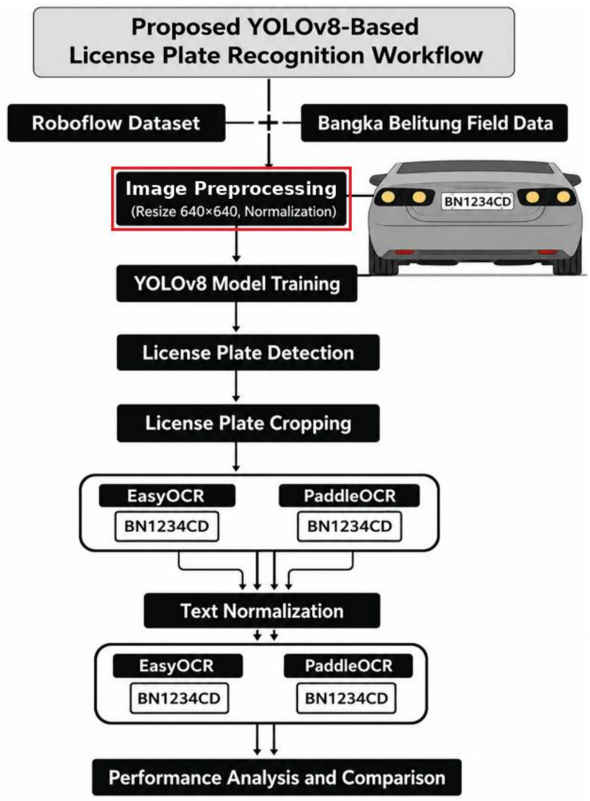
In contrast to earlier YOLO versions that rely on predefined anchor boxes, YOLOv8 adopts an anchor free detection mechanism that directly predicts object centers and bounding boxes. This architecture improves detection efficiency, simplifies model configuration, and maintains high localization accuracy for license plate detection before the OCR stage [20], [21]. Subsequently, EasyOCR and PaddleOCR engines extract textual content from the cropped plate regions, delivering high accuracy character segmentation and recognition across varied fonts and orientations [4], [7], [14] Consequently, this YOLO EasyOCR PaddleOCR pipeline builds upon established deep learning paradigms that enhance license plate detection in complex scenarios through advanced architectures and OCR integration [2], [22]

Several end to end license plate recognition systems have been proposed in previous studies. Approaches such as LPRNet based recognition frameworks and pipelines integrating traditional OCR engines such as Tesseract provide alternative solutions for automatic license plate recognition without requiring explicit character segmentation. These approaches highlight the importance of evaluating different OCR techniques under real world conditions. Furthermore, post processing techniques assisted by language models have been suggested to boost recognition accuracy by exploiting the format specific constraints of license plates. These techniques aid in disambiguating visually confusable characters and are planned for exploration in future research

Unlike previous studies that typically use a single OCR engine for license plate recognition, this research compares two state of the art frameworks EasyOCR and PaddleOCR within a YOLOv8 based detection pipeline. The detection model was trained on a Roboflow dataset, while recognition was evaluated on real world images of Indonesian license plates from Bangka Belitung Province, featuring variations in illumination, plate condition, and viewing angles.

In contrast to prior Indonesian ALPR studies, which typically commit to a single OCR engine and report aggregate accuracy on a single regional source, the novelty of this work lies in four specific aspects taken together. First, EasyOCR and PaddleOCR are benchmarked head to head within an identical YOLOv8 based detection pipeline under standardized preprocessing, cropping, and post OCR normalization, eliminating the engine versus pipeline confound that complicates comparisons across previous studies. Second, the test set is deliberately stratified into 44 field collected images from Bangka Belitung Province and 48 plates from other Indonesian provinces, which permits direct quantification of how well a Roboflow trained detector transfers between regional plate source an evaluation rarely reported for Indonesian plates. Third, error categories are tied to the OCR architectures themselves (CRNN+CTC versus DB detector+CRNN) rather than treated as generic OCR failures, yielding actionable diagnoses such as PaddleOCR's systematic loss of leading single letter regional codes. Fourth, the analysis culminates in four complementary, low cost remedies bounding box shrinking, regex based plate format post processing, an n-gram language model rescoring step, and CLAHE based contrast normalization that intervene at distinct pipeline stages and can be combined without retraining. These findings together enhance ALPR robustness in practical deployments and provide a reproducible reference point for future Indonesian plate studies.

MATERIALS AND METHODS



Source : (Research Results,2026)

Figure 1. Workflow of the proposed YOLO based Indonesian license plate recognition system using Roboflow data and Bangka Belitung field data for EasyOCR and PaddleOCR comparison.

Data Collection

This research employs two distinct datasets: a publicly available dataset for training the object detection model and a real world dataset for assessing optical character recognition performance.

The dataset used for training was sourced from the Roboflow Universe repository, specifically the "plate-indonesia-uakjc" dataset (v1, accessed January 2026), released under the CC BY 4.0 license for research purposes. The dataset focuses exclusively on Indonesian vehicle license plates and includes annotated bounding boxes provided in YOLO format. Annotations were verified prior to training to ensure consistency in bounding box quality and class labeling.

The dataset contains annotated vehicle images with license plate bounding boxes and consists of 661 training images, 108 validation images, and 92 test images. To enable robust geographic evaluation, the test set deliberately combines two sources: 44 field collected images from Bangka Belitung Province (regional code "BN") and 48 images of Indonesian license plates from outside Bangka Belitung covering other regional codes (including H, B, AB, AD, K, AA, E, G, DH, R, and S). All images were resized to 640 × 640 pixels to conform to the YOLOv8 model's input specifications. The training and validation splits were drawn exclusively from the Roboflow source, while the test split additionally incorporates the 44 field collected Bangka Belitung images to permit evaluation under authentic operational conditions.

The 44 field collected images were captured in Bangka Belitung Province under authentic operational conditions, exhibiting variations in illumination, viewing angles, and plate physical state, including older plates with white borders. These images were excluded from YOLO training and used solely for detection and recognition evaluation. The remaining 48 test images, drawn from the Roboflow split, benchmark generalization to plates from other Indonesian provinces.

Model Training

The license plate detection model was trained employing the YOLOv8 Nano architecture, chosen for its lightweight structure and efficient inference performance. Initialization utilized pretrained weights from the COCO dataset to capitalize on transfer learning benefits. Training was conducted using the Ultralytics YOLOv8 framework (version 8.0.x) in a Google Colab environment equipped with an NVIDIA Tesla T4 GPU (16 GB GDDR6), CUDA 12.1, and cuDNN 8.9. The Python runtime used PyTorch 2.1, with EasyOCR 1.7.1 and PaddleOCR 2.7.0 employed in the subsequent recognition stage. A fixed random seed of 42 was set across PyTorch, NumPy, and Python's random module to ensure reproducibility. Key hyperparameters included:

1. Input resolution: 640x640 pixels
2. Batch size: 16
3. Epochs: 100
4. Optimizer: AdamW
5. Learning rate: 0.001
6. Weight decay: 0.0005

Model robustness was enhanced through YOLOv8's built in augmentation (mosaic, HSV color jitter, random scaling, and perspective transformations), combined with a cosine learning rate scheduler decaying from 0.001 toward 0.01 of its initial value, and early stopping with a patience of 50 epochs based on validation mAP@0.5. No layers were frozen; end to end fine tuning from COCO pretrained weights was applied, since the single class task (license plate) benefits from adapting lower level features, and class weighting was not needed. During preprocessing, each training image was further augmented into three variants using random rotations (−5° to +5°), brightness adjustments (−25% to +25%), and Gaussian blur (max sigma 1 pixel) to mimic real world camera tilts, lighting variation, and image noise.

All images were resized to 640×640 pixels and pixel normalized prior to input, stabilizing training and improving feature extraction.

License Plate Detection

Following training, the YOLOv8 model was employed for license plate detection on both the Roboflow test dataset and the Bangka Belitung dataset. The detection procedure produced bounding boxes delineating the locations of license plates within vehicle images. These detected regions were subsequently cropped from the original images and supplied as input to the OCR phase.

OCR Recognition

The cropped license plate images underwent optical character recognition using both EasyOCR and PaddleOCR to identify plate characters. These OCR frameworks were assessed under uniform experimental conditions to enable an impartial comparison. The resulting OCR outputs were normalized by converting all characters to uppercase and excluding non alphanumeric

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8217

symbols, in line with Indonesian license plate conventions.

Evaluation Methods

The performance of the OCR systems was assessed using plate level accuracy and Character Error Rate metrics. Plate level accuracy is defined as the proportion of detected license plate crops for which the normalized OCR output is an exact, case insensitive character by character match against the ground truth plate string. The metric is computed per detected crop produced by the YOLOv8 detector, not per source image. Because the test set is curated to contain one license plate per image and the detector achieved recall = 1.0000 on that set, the per-crop and per-image counts coincide in this study; however, in the general case where multiple plates may be detected per image, each detected crop is paired with its closest ground truth plate by bounding box IoU before the exact match comparison is performed.

$$PlateAccuracy = \frac{N_{correct}}{N_{total}} \quad (1)$$

Where

$N_{correct}$ = number of correctly recognized license plates

N_{total} = total number of evaluated license plates.

Character accuracy is defined as the proportion of correctly recognized characters relative to the total number of characters:

$$Character\ Accuracy = \frac{Number\ of\ correctly\ recognized\ characters}{Total\ number\ of\ characters} \text{ x } 100\% \quad (2)$$

CER is the Levenshtein edit distance between the normalized OCR prediction and the ground truth plate string, divided by the length of the ground truth string (Equation 3). Character level accuracy is reported as 1 − CER. When the prediction is longer or shorter than the ground truth, the Levenshtein distance accounts for the resulting insertions or deletions, so per-character accuracy correctly penalizes both substitutions and length mismatches.

$$CER = \frac{LevenshteinDistance(predicted, groundtruth)}{length(groundtruth)} \quad (3)$$

Equivalently, CER can be expressed as:

$$CER = (S + D + I) / N \quad (4)$$

where S, D, and I denote substitution, deletion, and insertion errors, and N is the total number of ground truth characters. The statistical reliability of the reported recognition accuracy was evaluated by computing a 95% confidence interval according to the binomial proportion formula.

$$CI = p \pm z \times \sqrt{\frac{p(1-p)}{n}} \quad (5)$$

Where

p = plate recognition accuracy

n = number of evaluated samples

z =1.96 for a 95% confidence level.

RESULTS AND DISCUSSION

The first stage focuses on an extended model training configuration and evaluation process, encompassing model training configuration, model evaluation setup, and model evaluation results.

Model Training Configuration

The license plate detection model utilized the YOLOv8 Nano architecture and was trained on a dataset sourced from the Roboflow platform. This dataset consists of annotated vehicle images featuring bounding boxes around license plates, split into 661 training images, 108 validation images, and 92 test images.

Training spanned 100 epochs, each comprising a full iteration over the training dataset. Input images were standardized at a resolution of 640 x 640 pixels, a conventional setting for YOLO based object detectors, which balances the retention of intricate details (especially for small targets like license plates) with computational efficiency. A batch size of 16 was employed to promote stable gradient updates and efficient memory usage. The validation set facilitated performance monitoring and overfitting prevention during training, while the test set was reserved for final model evaluation.

Model Evaluation Setup

Following training, the model was assessed on the test dataset to evaluate its capability for detecting license plates in novel images. The validation dataset served exclusively during training to track performance and avert overfitting.

VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8217

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

An Intersection over Union threshold of 0.5 was employed to judge whether a predicted bounding box adequately overlapped with the ground truth annotation. A detection qualifies as correct if the IoU between the prediction and ground truth surpasses this threshold.

For inference, predictions exceeding a confidence score of 0.25 were kept to exclude low confidence detections, though this applies only to filtering and visualization. The mean Average Precision (mAP) metric, by contrast, is calculated over the entire precision recall curve across diverse confidence thresholds, in line with conventional object detection evaluation practices.

Model efficacy was quantified via standard object detection metrics: Precision, Recall, mAP@0.5, and mAP@0.5:0.95. Precision denotes the share of accurate detections among all predictions, while Recall gauges the model's success in identifying all objects of interest in the dataset.

Model Evaluation Results

The experimental results indicate that the proposed YOLOv8 based license plate detection model delivers robust performance on the test dataset. Elevated precision and recall metrics underscore the model's proficiency in accurately identifying license plates, while curtailing both false positives and false negatives. These outcomes affirm the model's suitability for seamless integration with Optical Character Recognition modules in comprehensive end to end automatic license plate recognition systems.

Table 1. Performance Evaluation of the YOLOv8 License Plate Detection Model

No	Evaluation Metric	Score
1	Precision	0.9975
2	Recall	1.0000
3	mAP@0.5	0.9950
4	mAP@0.5:0.95	0.8328

Source : (Research Results,2026)

The findings demonstrate that, on the held out test set of 92 Indonesian license plate images (44 field collected from Bangka Belitung plus 48 from other Indonesian provinces), the proposed model attains exceptional detection accuracy. A precision of 0.9975 signifies that the vast majority of detected license plates represent true positives, thereby minimizing the false positive rate. A recall of 1.0000 indicates that every license plate in the test split was successfully localized by the model at the standard IoU = 0.5 evaluation threshold. The mean average precision (mAP) at an IoU threshold of 0.5, recorded at 0.9950, reflects precise localization and classification. Moreover, the mAP@0.5:0.95 value of 0.8328, averaging across progressively stringent IoU thresholds, affirms sustained efficacy under rigorous evaluation criteria. Collectively, these metrics all reported on the test split, not the validation split substantiate the trained YOLOv8 model’s viability as a dependable precursor to the OCR recognition phase within the automated license plate recognition framework.

License Plate Detection

The subsequent phase of this research entails identifying license plate regions within vehicle images prior to Optical Character Recognition application. The license plate dataset employed for OCR assessment is the same 92 image test set used for detection evaluation, comprising 44 field collected images from Bangka Belitung Province and 48 images of Indonesian plates from other provinces. Per the Indonesian vehicle registration framework, plates from Bangka Belitung Province incorporate the regional code “BN”, trailed by four numerals and two to three alphabetic characters as the serial identifier, while plates from other provinces use distinct regional codes (e.g., H for Semarang, B for Jakarta, AB for Yogyakarta, AD for Surakarta, K for Pati, AA for Magelang, among others).

License plate detection was executed via the YOLOv8 model trained beforehand. This model analyzes vehicle images to localize license plates spatially. The detection yields bounding boxes demarcating license plate positions, which are subsequently cropped from the source images to furnish inputs for the ensuing OCR phase.

Source : (Research Results,2026)
Figure 2. YOLO Bounding Box Detection and Image Cropping Process

Following the detection phase, license plate regions are extracted from the original images using bounding boxes generated by the YOLO model. These cropped images are subsequently analyzed

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)
VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8217

via EasyOCR and PaddleOCR for optical character recognition of the inscribed characters.

The OCR output consists of recognized text strings with associated per-character confidence scores. Characters exceeding a predefined confidence threshold are retained, while unnecessary symbols and whitespace are removed. The remaining characters are then concatenated to form a standardized license plate string. This string is subsequently compared with the ground truth license plate numbers for performance evaluation. OCR evaluation was conducted per detected license plate crop produced by the YOLO model, where each cropped plate region was matched with its corresponding ground truth text.

Algorithm 1. OCR Text Extraction and Normalization

Input : Cropped license plate image
Output : Normalized license plate text

1. Perform OCR on the cropped license plate image
2. Retrieve recognized text and confidence scores
3. Filter characters with confidence above a predefined threshold
4. Convert characters to uppercase
5. Concatenate characters into a single string
6. Remove whitespace and non alphanumeric characters
7. Validate the text using the Indonesian license plate format pattern
8. **If** format is invalid, return empty result
9. Correct common OCR misrecognitions
10. Reconstruct and return the normalized license plate text

Prior to text normalization, the cropped license plate images are subjected to a sequence of preprocessing procedures designed to enhance OCR recognition accuracy. Concretely, each crop is (a) converted from RGB to single channel grayscale; (b) contrast normalized using Contrast Limited Adaptive Histogram Equalization (CLAHE) with a clip limit of 2.0 and an 8×8 tile grid, which equalizes local lighting on Indonesian plates that are typically bright characters on a dark background; and (c) denoised with a 3×3 Gaussian kernel to suppress sensor noise while preserving stroke edges. Adaptive binarization and explicit affine rectification were evaluated but not retained in the final pipeline because the YOLOv8 cropped regions are already tightly axis aligned and both engines accept grayscale input directly. Each variant was passed unaltered to EasyOCR and PaddleOCR so that any performance gap observed in Section 3 reflects the recognizer rather than asymmetric preprocessing.

The next step is text normalization to conform to the standard Indonesian license plate format. This process includes converting all characters to uppercase, removing non alphanumeric characters, and correcting common misrecognitions, such as "O" interpreted as "0" or "S" as "5". This normalization step is crucial for improving overall recognition accuracy.

Algorithm 2. License Plate Pattern Validation
Input : OCR text string
Output : Validated license plate components

1. Define license plate structure:
   - Region prefix (1–2 letters)
   - Numeric identifier (3–4 characters)
   - Serial suffix (1–3 letters)
2. Match OCR text with defined pattern
3. **If** match is found, extract components
4. **Otherwise**, return invalid

Once the OCR results are obtained, the user inputs the Ground Truth license plate data for evaluation. This Ground Truth data is a reference for calculating the accuracy of character recognition. The evaluation is conducted using the Levenshtein Distance method, which measures the edit distance between the OCR output and the Ground Truth text.

Algorithm 3. Character Error Correction

Input : Numeric part of license plate
Output : Corrected numeric sequence

1. Define character correction mapping
2. **For** each character in numeric sequence:
   - Replace if character exists in mapping
   - **Otherwise**, keep original character
3. **Return** corrected numeric sequence

Algorithm 4. Levenshtein Distance Calculation

Input : Ground Truth string, OCR result string
Output : Edit distance

1. **If** one string is empty, return length of the other
2. Initialize distance matrix
3. Compute insertion, deletion, and substitution costs
4. Select minimum cost for each operation
5. **Return** final edit distance

VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8217

Algorithm 5. OCR Performance Evaluation

Input : Ground Truth text, OCR result text
Output : Character Accuracy, Character Error Rate (CER)
1. Compute Levenshtein Distance
2. Calculate Character Accuracy
3. Calculate Character Error Rate (CER)
4. **Return** evaluation metrics

The experimental results indicate that EasyOCR achieves higher recognition accuracy than PaddleOCR when evaluated on the full 92 image Indonesian test set (comprising 44 field images from Bangka Belitung Province and 48 images from other Indonesian provinces). This performance difference can be attributed to EasyOCR's more robust capability to handle font variations and diverse lighting conditions. The experimental findings indicate that EasyOCR achieves higher recognition accuracy than PaddleOCR.

Table 2. OCR Performance Comparison

OCR Method	Total Samples	Correct Plates	Plate Accuracy (%)	Character Accuracy (%)	95% CI (Plate Accuracy)
EasyOCR	92	72	78.26	96.74	69.8 – 86.7
PaddleOCR	92	67	72.83	92.97	64.0 – 82.0

Source : (Research Results,2026)

Table 2 illustrates the statistical comparison of EasyOCR and PaddleOCR performance on the Indonesian license plate test set. EasyOCR attained a plate level recognition accuracy of 78.26%, compared to PaddleOCR's 72.83%. The calculated 95% confidence intervals provide statistical support for the observed performance difference and indicate that EasyOCR demonstrates more consistent recognition accuracy across the evaluated samples. To probe geographic generalization, accuracy was additionally stratified by the test set source: on the 44 field collected Bangka Belitung images, EasyOCR achieved 77.27% plate accuracy and PaddleOCR 72.73%; on the 48 images of Indonesian plates from outside Bangka Belitung, EasyOCR achieved 79.17% and PaddleOCR 72.92%. The narrow gap between the two subsets (within roughly two percentage points for both engines) indicates that the pipeline generalizes consistently across regional plate variations and is not overly tuned to a single province. The architectural rationale for EasyOCR's edge is interpretable: EasyOCR's CRNN with CTC decoding processes the plate as a single horizontal sequence, which aligns well with the bold, evenly spaced sans-serif characters that dominate Indonesian plates, whereas PaddleOCR's DB detector plus CRNN pipeline introduces an extra text localization step inside the already cropped plate region that is occasionally destabilized by border artifacts on older plates an effect visible in the higher rate of inserted or missing characters in Table 2 alongside the higher overall CER. The statistical comparison supported by confidence interval analysis further reinforces the observed performance difference between EasyOCR and PaddleOCR.

Table 3. Detailed Character Error Rate (CER) Analysis – Error Case Comparison (EasyOCR vs PaddleOCR)

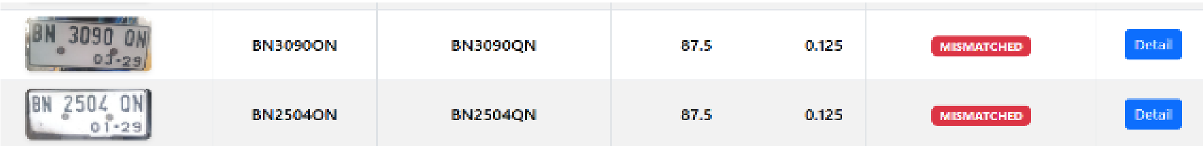
No	Region	Ground Truth	EasyOCR Pred	Easy CER	Easy Error Type	Paddle Pred	Paddle CER	Paddle Error Type
1	Non-BN	AB4622FO	AB4622FEL	0.2500	Insertion	AB4622F	0.1250	Deletion
2	Non-BN	AB4848DH	AB4848DH	0.0000	Correct	AB4848DHY	0.1250	Insertion
3	Non-BN	B5957BGQ	B5957BGQ	0.0000	Correct	B5957BGO	0.1250	Q→O Substitution
4	Non-BN	B6423GLC	B6423CLC	0.1250	Substitution	B6423GLC	0.0000	Correct
5	BN	BN2078QH	BN2078QH	0.0000	Correct	BN2078OHB	0.2500	Insertion
6	BN	BN2151QD	BN2151OD	0.1250	Q→O Substitution	BN2151OD	0.1250	Q→O Substitution
7	BN	BN250LQN	BN250LON	0.1250	Q→O Substitution	BN2504ON	0.2500	Substitution
8	BN	BN2545BA	BN2525BA	0.1250	Substitution	BN2545BA	0.0000	Correct
9	BN	BN2545CD	BN2545CD	0.0000	Correct	BN2545CDZ	0.1250	Insertion
10	BN	BN2732BF	BN2732BFI	0.1250	Border Insertion	BN2732BF	0.0000	Correct
11	BN	BN3090QN	BN3090ON	0.1250	Q→O Substitution	BN3090ON	0.1250	Q→O Substitution
12	BN	BN3882QW	IBN3882QW	0.1250	Border Insertion	BN3882QW	0.0000	Correct
13	BN	BN4057PR	BR2057PR	0.2500	Substitution	BN4057PR	0.0000	Correct
14	BN	BN4181BI	BN4181BI	0.0000	Correct	BN1818I	0.2500	Deletion
15	BN	BN5283AC	BN5283AC	0.0000	Correct	BN5283ACC	0.1250	Insertion
16	BN	BN5313DD	BH5313DD	0.1250	Substitution	BN5313DD	0.0000	Correct
17	BN	BN6321QW	BN6321QH	0.1250	Substitution	BN6321OW	0.1250	Q→O Substitution
18	BN	BN6489CA	BN6489CA	0.0000	Correct	BN6489CAH	0.1250	Insertion
19	BN	BN6594QP	BN659LOP	0.2500	Substitution	BN6594OP	0.1250	Q→O Substitution
20	BN	BN6751EG	BN6751EG	0.0000	Correct	EG6751BN	0.5000	Substitution

VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8217

No	Region	Ground Truth	EasyOCR Pred	Easy CER	Easy Error Type	Paddle Pred	Paddle CER	Paddle Error Type
21	BN	BN6828EF	BN6828EF	0.0000	Correct	H0O	1.0000	Catastrophic
22	Non-BN	DH3183KT	H3183KT	0.1250	Leading Code Deletion	DH3183KT	0.0000	Correct
23	Non-BN	G4327BV	G4327BVL	0.1429	Insertion	G4327BV	0.0000	Correct
24	Non-BN	H2019WB	H2019HB	0.1429	Substitution	-	0.0000	-
25	Non-BN	H2088QB	H2088OB	0.1429	Q→O Substitution	H2088OB	0.1429	Q→O Substitution
26	Non-BN	H2169QB	H2169QB	0.0000	Correct	2169Q	0.2857	Deletion
27	Non-BN	H2180LK	H2180LK	0.0000	Correct	2180LK	0.1429	Leading Code Deletion
28	Non-BN	H2698WI	H2698W	0.1429	Deletion	H2698WI	0.0000	Correct
29	Non-BN	H3282KK	H3282KK	0.0000	Correct	S3282KK	0.1429	Substitution
30	Non-BN	H3527HC	H3527HC	0.0000	Correct	3527HC	0.1429	Leading Code Deletion
31	Non-BN	H5293LK	H5293LK	0.0000	Correct	5293LK	0.1429	Leading Code Deletion
32	Non-BN	H5606CI	H5606CD	0.1429	Substitution	H5606CI	0.0000	Correct
33	Non-BN	H5824JK	HH5824JK	0.1429	Insertion	H5824JK	0.0000	Correct
34	Non-BN	H6379DK	H6379DK	0.0000	Correct	6379DK	0.1429	Leading Code Deletion
35	Non-BN	H6912QK	H6912OK	0.1429	Q→O Substitution	H6912QK	0.0000	Correct
36	Non-BN	K4997XC	K4997XC	0.0000	Correct	XC4997	0.5714	Deletion
37	Non-BN	S4073JBO	S4073JBO	0.0000	Correct	S4073JB	0.1250	Deletion

Source: (Research Results, 2026)

In Table 3, the "-" entry indicates a single case (H2019WB) for which the PaddleOCR pipeline did not produce an output and is therefore reported as a catastrophic recognition failure. Several recurring recognition errors appeared during testing. One common issue is the misclassification of "Q" as "O". Historically, the letter "Q" was used as a regional identifier for certain areas in Bangka Regency.



Source : (Research Results, 2026)
Figure 3. Misclassification Between Characters 'O' and 'Q' in OCR Results

Another issue involves older license plates with black backgrounds and white borders. During the bounding box cropping process, the plate border is sometimes included, causing the OCR system to misinterpret it as "I".

Source : (Research Results,2026)
Figure 4. Recognition Errors Caused by License Plate Border

These recognition errors are mainly caused by visual similarity between characters and the inclusion of license plate borders within the cropped detection regions, which introduces additional noise into the OCR process. A detailed tally of the 20 EasyOCR errors on the test set reveals three dominant error categories: (i) Q→O substitutions on Bangka plates with serial letter Q (e.g., BN2151QD→BN2151OD, BN3090QN→BN3090ON, H2088QB→H2088OB), accounting for roughly 25% of EasyOCR failures; (ii) border driven character insertions or deletions (e.g., BN3882QW→IBN3882QW, BN2732BF→BN2732BFI, H5824JK→HH5824JK), where the white frame of older plates was partly retained inside the crop; and (iii) within string character confusions on visually similar pairs such as W/H, N/H, and G/C (e.g., H2019WB→H2019HB, BN5313DD→BH5313DD, B6423GLC→B6423CLC). PaddleOCR exhibits the same Q→O pattern but with a notably higher frequency, and adds two further failure modes: (iv) systematic dropping of the leading single letter regional code on non-Bangka plates (e.g., H5293LK→5293LK, H6379DK→6379DK, H3527HC→3527HC), which appears tied to its DB text detection step missing the narrow first character; and (v) catastrophic re-readings on glare affected plates (e.g., BN6828EF→H0O, B5641AC→BN0727YAM).

To address these limitations, several concrete improvements are proposed and form part of the planned follow up work. First, bounding box refinement either via an additional shrink margin (e.g., 3–5% inward) before cropping or via a lightweight border detection step directly targets the border insertion errors that dominate category (ii). Second, plate format aware post processing using a regular expression for Indonesian plates (regional prefix + 1–4 digits + 1–3 letters) can disambiguate Q vs. O at positions where letters are required and 0 vs. O at positions where digits are required, which would have corrected the majority of category (i) errors at zero added inference cost. Third, a small character level language model rescoring step (e.g., an n-gram model over Indonesian plate serials) can be appended to the OCR output to suppress unlikely sequences such as the catastrophic mis-readings in category (v). Fourth, light targeted preprocessing CLAHE based contrast normalization and morphological border removal is expected to mitigate glare and frame artifacts before recognition. These solutions are deliberately complementary: they intervene at the cropping, recognition, and post processing stages respectively, so they can be combined without retraining the underlying YOLOv8 detector.

CONCLUSION

The findings of this study demonstrate that the proposed YOLO based license plate detection model achieves strong performance on the 92 image Indonesian test split (44 field collected Bangka Belitung images plus 48 plates from other Indonesian provinces), with a test set precision of 0.9975, recall of 1.0000, and mAP@0.5 of 0.9950, indicating accurate localization of license plates. The test set mAP@0.5:0.95 score of 0.8328 further confirms the robustness of the model across multiple intersection over union thresholds, suggesting its suitability for real world deployment in outdoor environments. All four detection metrics are reported on the test split; the validation split was used exclusively for monitoring during training and is not the source of any number reported in this conclusion. It is important to note that this test partition comprises only 92 images; a sample size below 100 images is relatively limited, and this constrains the extent to which the reported performance can be generalized to large scale, real world operational deployment. Future work should therefore validate the model on a substantially larger and more diverse test set before drawing firm conclusions about its operational reliability.

During the optical character recognition stage, a comparative evaluation of EasyOCR and PaddleOCR on the full 92 image Indonesian test set (44 Bangka Belitung field images and 48 plates from other Indonesian provinces) indicates that EasyOCR achieves higher recognition accuracy. EasyOCR obtained a plate recognition accuracy of 78.26%, a character recognition accuracy of 96.74%, and a Character Error Rate (CER) of 3.26%, compared to 72.83%, 92.97%, and 7.03% CER for PaddleOCR, respectively. These results indicate that EasyOCR performs more consistently under variations in lighting conditions, font styles, and image quality commonly observed in field acquired license plate images. Error analysis shows that most recognition errors arise from visual similarity between characters, particularly confusion between “O” and “Q”, as well as interference caused by white borders on older license plates. In some cases, these borders are partially included in the detected bounding boxes, introducing noise that affects the OCR recognition process.

To address these limitations, several improvements can be considered. First, bounding box refinement techniques can be applied to ensure that cropped regions focus only on license plate characters. Second, image preprocessing methods, such as contrast enhancement and noise reduction, may improve character visibility before OCR

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8217

processing. Third, rule based text validation based on Indonesian license plate format patterns can help correct ambiguous characters and reduce misclassification errors. Overall, the integration of YOLOv8 based detection and EasyOCR recognition provides an effective framework for automatic license plate recognition across Indonesian regional plates, demonstrated on both Bangka Belitung field data and plates from multiple other provinces. Future work may explore more diverse datasets, advanced OCR architectures, and adaptive preprocessing techniques to further improve recognition accuracy in complex real world environments.

REFERENCE

[1] R. Muhammad Ridwan, M. A. Musyaffa, K. Azis, S. Sumaryo, and A. Zamhuri, "Application of OCR Technology for Vehicle License Plate Detection and Yolo V8 for Car Counting," *Jurnal Indonesia Sosial Teknologi*, vol. 5, no. 9, pp. 3369–3385, Sep. 2024, doi: 10.59141/jist.v5i9.1378.

[2] C. Dewi, S. A. Philemon, G. Dai, and A. P. S. Chen, "Image Enhancement Technique Utilizing YOLO Model for Automatic Number Plate Recognition," *International Journal of Transport Development and Integration*, vol. 9, no. 1, pp. 59–68, Mar. 2025, doi: 10.18280/ijtdi.090106.

[3] J. T. Wijaya and A. M. Arymurthy, "Indonesian License Plate Detection and Recognition System using Gaussian YOLOv7," *Jurnal Ilmu Komputer dan Informasi*, vol. 18, no. 2, pp. 141–153, Jun. 2025, doi: 10.21609/jiki.v18i2.1320.

[4] W. Gata, D. Riana, M. Haris, M. I. Prasetiyowati, and D. P. Metalica, "Automated Indonesian Plate Recognition: YOLOv8 Detection and TensorFlow-CNN Character Classification," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 9, no. 3, pp. 544–553, Jun. 2025, doi: 10.29207/resti.v9i3.6310.

[5] A. Amin, R. Mumtaz, M. J. Bashir, and S. M. H. Zaidi, "Next-Generation License Plate Detection and Recognition System using YOLOv8," in *Proc. 2023 IEEE 20th Int. Conf. Smart Communities: Improving Quality of Life using AI, Robotics and IoT (HONET)*, Boca Raton, FL, USA, pp. 179–184, Dec. 2023, doi: 10.1109/HONET59747.2023.10374756.

[6] R. Khanna, S. K. Sahoo, S. Maggu, S. Kalonia, and A. K. Kaushik, "Enhanced Surveillance: Developing a Computer Vision-based CCTV Footage Analysis Tool for Improved Vehicle Number Plate Recognition," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 12, no. 4, pp. 4622–4628, Apr. 2024, doi: 10.22214/ijraset.2024.60807.

[7] M. Paul, K. A. Rafidha Rehiman, and M. K. Sabu, "Automated License Plate Recognition Using YOLOv8 and EasyOCR," *in Fifth Congress on Intelligent Systems (CIS 2024), Lecture Notes in Networks and Systems, vol. 1278, S. Kumar, E. A. Mary Anita, J. H. Kim, and A. Nagar, Eds. Singapore: Springer*, 2025, pp. 301–312. doi: 10.1007/978-981-96-2703-5_20.

[8] A. Istiak and W. Feng, "Enhancing Bangladeshi License Plate Recognition: A YOLOv8 Approach with Roboflow Integration for Accuracy and Speed Optimization," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 12, no. 5, pp. 1686–1699, May 2024, doi: 10.22214/ijraset.2024.61520.

[9] A. Sarhan *et al.*, "Egyptian car plate recognition based on YOLOv8, Easy-OCR, and CNN," *Journal of Electrical Systems and Information Technology*, vol. 11, no. 1, p. 32, Aug. 2024, doi: 10.1186/s43067-024-00156-y.

[10] T. A. Shyaa and A. A. Hashim, "Superior Use of YOLOv8 to Enhance Car License Plates Detection Speed and Accuracy," *Revue d'Intelligence Artificielle*, vol. 38, no. 1, pp. 139–145, Feb. 2024, doi: 10.18280/ria.380114.

[11] R. A. Tavares, "Comparison of Image Preprocessing Techniques for Vehicle License Plate Recognition Using OCR: Performance and Accuracy Evaluation," arXiv:2410.13622, Oct. 2024.

[12] J. Sainui, C. Thepporn, and P. Chusuwan, "Thai License Plate Recognition using SSD MobileNet and EasyOCR," in *Proceedings of the 2024 6th International Conference on Image Processing and Machine Vision*, New York, NY, USA: ACM, Jan. 2024, pp. 36–41. doi: 10.1145/3645259.3645266.

[13] L. Qiao *et al.*, "DavarOCR: A Toolbox for OCR and Multi-Modal Document Understanding," in *Proceedings of the 30th ACM International Conference on Multimedia*, New York, NY, USA: ACM, Oct. 2022, pp. 7355–7358. doi: 10.1145/3503161.3548547.

[14] S. Chopra, "Leveraging YOLO Algorithm and PaddleOCR in Machine Learning Applications," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 12, no. 4, pp. 4576–4584, Apr. 2024, doi: 10.22214/ijraset.2024.60850.

[15] H. Moussaoui *et al.*, “Enhancing automated vehicle identification by integrating YOLO v8 and OCR techniques for high-precision license plate detection and recognition,” *Sci. Rep.*, vol. 14, no. 1, p. 14389, Jun. 2024, doi: 10.1038/s41598-024-65272-1.

[16] A. K. Al Mhdawi, N. Nnamoko, S. M. Raafat, M. K. S. Al-Mhdawi, and A. J. Humaidi, “Connecting Vision and Emissions: A Behavioural AI Approach to Carbon Estimation in Road Design,” arXiv:2506.18924, Jun. 2025.

[17] Z. Zhao, H. Kang, B. Wang, and C. He, “DocLayout-YOLO: Enhancing Document Layout Analysis through Diverse Synthetic Data and Global-to-Local Adaptive Perception,” arXiv:2410.12628, Oct. 2024.

[18] F. Li, H. Yan, and L. Shi, “Multi-scale coupled attention for visual object detection,” *Sci. Rep.*, vol. 14, no. 1, p. 11191, May 2024, doi: 10.1038/s41598-024-60897-8.

[19] C. Liu, Y. Li, J. Gu, Y. Lou, and T. Shen, “ENHANCING PEDESTRIAN TARGET RECOGNITION IN OPEN COMMUNITY MULTI-SCENE SPACES USING THE YOLO-STP NETWORK,” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. X-1/W1-2023, pp. 319–328, Dec. 2023, doi: 10.5194/isprs-annals-X-1-W1-2023-319-2023.

[20] M. A. M. Ali, “Advancing Crowd Object Detection: A Review of YOLO, CNN, and Vision Transformers Hybrid Approach,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 12, no. 6, pp. 1240–1268, Jun. 2024, doi: 10.22214/ijraset.2024.63293.

[21] H. Moussaoui, N. El Akkad, and M. Benslimane, “License plate text recognition using deep learning, NLP, and image processing techniques,” *Statistics, Optimization & Information Computing*, vol. 12, no. 3, pp. 685–696, Feb. 2024, doi: 10.19139/soic-2310-5070-1966.

[22] X. Wang, M. Wang, H. Guo, J. Li, X. Wang, and Y. Zhang, “License plate recognition system for complex scenarios based on improved YOLOv5s and LPRNet,” *Sci. Rep.*, vol. 15, no. 1, p. 34741, Oct. 2025, doi: 10.1038/s41598-025-18311-4.

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri