Enhanced Automatic License Plate Detection and Recognition using CLAHE and YOLOv11 for Seat Belt Compliance Detection

Sutikno
Department of Informatics, Faculty of Science and Mathematics, Diponegoro University, Indonesia
sutikno@lecturer.undip.ac.id (corresponding author)

Aris Sugiharto
Department of Informatics, Faculty of Science and Mathematics, Diponegoro University, Indonesia
arissugiharto@lecturer.undip.ac.id

Retno Kusumaningrum
Department of Informatics, Faculty of Science and Mathematics, Diponegoro University, Indonesia
retno@live.undip.ac.id

Received: 16 November 2024 | Revised: 23 December 2024 and 30 December 2024 | Accepted: 1 January 2025

 DOI: https://doi.org/10.48084/etasr.9629

ABSTRACT

Traffic accidents caused by seat belt violations remain a severe problem in low-income countries. Identifying the vehicles of these violators is vital for enhancing safety. Therefore, this research develops a vehicle license plate detection and recognition system to support this problem. The proposed system was divided into three subsystems: windshield detection, license plate detection, and character recognition. The windshield detection subsystem used the You Only Look Once (YOLOv11) model. License plate detection combined the determination of the Region Of Interest (ROI) and YOLOv11. Meanwhile, character recognition combined the Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm and YOLOv11. YOLOv11 is the latest version of YOLO, which is faster and more efficient than the previous version, and CLAHE enhances the contrast of the image dataset, improving its quality. The dataset was collected from highways and toll roads in Semarang, Indonesia. The test results for windshield detection showed that the YOLOv11n model produced higher precision and faster detection time than YOLOv11m and YOLOv8m. The test results for license plate detection showed that the proposed method achieved perfect precision and recall. Meanwhile, the test results for character recognition indicated that the proposed method produced higher precision and average precision than YOLOv11n alone. The proposed method can produce precision and average precision for character recognition of 0.922 and 0.931, respectively. This research can potentially be used for automatic and real-time identification of car license plates for violators who do not wear seat belts on the highway.

Keywords-windshield detection; license plate detection; character recognition; CLAHE; YOLOv11

I. INTRODUCTION

Road traffic accidents remain a significant issue, especially in low-income countries, with seat belt violations being a critical cause [1]. Results indicated that only 15.1% of all vehicle passengers in Saudi Arabia were wearing seat belts at the time of the crash [2]. Despite regulations, violations persist even though seat belts effectively reduce the risk of injury. Monitoring is still mostly manual, limiting detection. A proposed solution is using Closed-Circuit Television (CCTV) with computer vision-based technology to automatically detect seat belt violations. There are three subsystems in license plate identification for seat belt violators: windshield detection, seat belt violator detection, and license plate identification. Several studies have used different methods for windshield detection and unbuckled violator detection. Some researchers used manual methods such as edge detection to determine the driver and seat belt area [3-5]. However, over time, Convolutional Neural Network (CNN)-based methods, such as MobileNetV2 [6] and You Only Look Once (YOLO) [7-9], have become dominant due to their higher accuracy and robustness.

The license plate identification subsystem is used to identify license plates of unbelted violators. This subsystem is divided into license plate detection and character recognition. License plate detection is used to determine the location of a license plate, while character recognition is used to extract character image objects on the license plate. Several methods have been widely used for license plate detection. The image processing method is still commonly applied by combining several techniques, including morphological operations, resizing, contour detection, thresholding, and histogram equalization [10, 11]. Additionally, authors in [12] used several image processing methods, namely edge detection, morphology, and masking. Conventional image processing methods often encounter difficulties in achieving accurate and flexible results under real-world conditions, such as varying angles, lighting, and noise. In addition, deep learning-based models have been widely used, including YOLOv4, YOLOv5, YOLOv8, and YOLOv9. Authors in [13] reported license plate detection using YOLOv4 on moving cars. Authors in [14] used YOLOv5, which can handle variations in lighting, image quality, and vehicle orientation. YOLOv8 has also been widely used for license plate detection, and the authors in [15] compared it to YOLO variants such as YOLO-NAS and YOLOv9.

Several methods have also been used in character recognition, including Easy Optical Character Recognition (EasyOCR), Paddle OCR, Tesseract OCR, and YOLOv5. Authors in [16] used EasyOCR for license plate character recognition, focusing on the Smart Checkpoint Management System (SCMS). Authors in [17] proposed Paddle OCR, which was compared with several variants of previous OCR models. The proposed method yielded higher accuracy than others. Authors in [18] proposed the Tesseract OCR method, which was capable of recognizing characters both during day and night. YOLOv8 had a good performance for license plate detection. Authors in [19] compared YOLOv8, YOLOv9, and SSD: Mobinet. The proposed method produced the best average precision, although its speed was inferior to YOLOv9. Meanwhile, authors in [20] compared several YOLO variants. YOLOv8 produced higher accuracy than YOLOv2, YOLOv3, and YOLOv4. The latest version of YOLO, YOLOv11, has emerged, a series of Ultralytics YOLO real-time object detectors with superior accuracy, speed, and efficiency [21]. YOLOv11 offers significant improvements in its architecture and training method, making it a flexible choice for various computer vision tasks [21].

One of the datasets used in this study consists of license plates. The characters on these license plates are often unclear due to poor lighting, vehicle speed, low camera resolution, and other challenges. Therefore, it needs to be corrected before being processed with the model. One way is to increase the contrast using Contrast Limited Adaptive Histogram Equalization (CLAHE) by increasing the accuracy [22-24]. Therefore, in this study, CLAHE and YOLOv11 are integrated for automatic license plate detection and recognition, with a focus on being used to detect unbuckled violators. The proposed system consists of three subsystems: windshield detection, license plate detection, and character recognition.

II. MATERIAL AND METHODS

A. Dataset

We used three datasets: DS_Frame, DS_ROI, and DS_License_Plate. DS_Frame was a collection of video frames recorded on the highway containing vehicle objects such as cars and motorcycles, and this dataset consisted of 2231 frames. This dataset was divided into three parts, including 80% training data, 10% validation data, and 10% testing data. The recording was conducted in two locations, namely the highway and the toll road of Semarang City, Indonesia. Recording was done during the daytime in bright conditions using a camera placed on a pedestrian bridge above the road. An example image from the DS_Frame dataset is shown in Figure 1. The DS_ROI dataset is a Region Of Interest (ROI) of each car's license plate as seen through the windshield. This ROI is where the license plate is most likely to be, under the windshield. An example of this dataset is shown in Figure 2. Meanwhile, the DS_License_Plate dataset collects car license plates from the DS_ROI dataset. An example of this dataset is shown in Figure 3.

Fig. 1. Examples of the DS_Frame dataset.

Fig. 2. Examples of the DS_ROI dataset.

Fig. 3. Examples of the DS_License_Plate dataset.

B. The Proposed Method

The proposed system is part of a car license plate identification system for violators who do not wear seat belts. This study focuses on detecting and identifying car license plates, as shown in Figure 4. The proposed system is divided into three subsystems: windshield detection, license plate detection, and character recognition.

1) Windshield Detection

A windshield can distinguish cars and other objects on the road. Additionally, the windshield can be used to identify unbelted drivers and passengers. The YOLOv11 model is used for this windshield detection. YOLOv11, the latest version from Ultralytics, offers lightweight, faster, and more efficient models than previous versions [21]. This version is built on the YOLOv8 codebase, features architectural tweaks, and incorporates elements from YOLOv9 and YOLOv10 for better performance. YOLOv11 supports multiple computer vision tasks, including object detection, instance segmentation, image classification, pose estimation, and oriented object detection. For object detection, YOLOv11 uses a CNN to extract image features, predicts bounding boxes and class probabilities, and applies Non-Maximum Suppression (NMS) to filter results for accuracy. Furthermore, it is trained on the MS-COCO dataset, covering 80 classes. The architecture is an enhanced version of YOLOv8 with additional integrations and tuning. The architecture includes the backbone, neck, and head [25].

The backbone extracts the features from the input image at multiple scales. YOLOv11 introduces the C3k2 (Cross Stage Partial with kernel size 2) block, a custom Cross Stage Partial (CSP) bottleneck with two smaller convolutions, replacing the Convolutional to Fused (C2f) block in YOLOv8 for better efficiency. CSP networks split feature maps, process parts through a bottleneck, and merge them to reduce computational load and enhance feature representation.

The neck aggregates features from different resolutions for prediction, using upsampling and concatenation. YOLOv11 uses the C3k2 block here for improved speed and performance. It also adds the Convolutional block with Parallel Spatial Attention (C2PSA), a spatial attention mechanism that focuses on key regions to help detect smaller or occluded objects, an enhancement over YOLOv8.

The head generates final predictions, such as bounding boxes and classifications. Similar to the neck, the C3k2 block replaces C2f. The Detect layer remains the same as YOLOv8, but the C3k2 block increases speed and parameter efficiency.

2) License Plate Detection

License plate detection is used to determine the location of license plates on vehicles. This subsystem is divided into ROI determination and license plate detection. ROI determination is used to reduce the area of the license plate in order to increase the possibility of license plate detection. In addition, it can reduce computation time. Car license plates are typically located under the windshield; therefore, the location of the ROI is under the windshield. An illustration of this ROI determination is shown in Figure 5. The ROI bounding box can be determined when the windshield bounding box is formed at the upper left (X_{min}, Y_{min}) and lower right (X_{max}, Y_{max}) coordinates. When the ROI bounding box is formed at the upper left (X_{rmin}, Y_{rmin}) and lower right (X_{rmax}, Y_{rmax}) coordinates, then Y_{rmin} and Y_{rmax} can be calculated successively using (1) and (2).

$$Y_{rmin} = Y_{min} + 1.5 \times H \quad (1)$$

$$Y_{rmax} = Y_{rmin} + 1.5 \times H \quad (2)$$

where H is the height of the windshield bounding box ($Y_{max} - Y_{min}$). After the ROI is determined, the next step is to detect the license plate in that area. The data used are from cars in Indonesia and consists of several background colors: black, white, yellow, red, and green. To detect the license plate, we use YOLOv11 by comparing 5 model variants, namely YOLOv11n, YOLOv11s, YOLOv11m, YOLOv11l, and YOLOv11x.

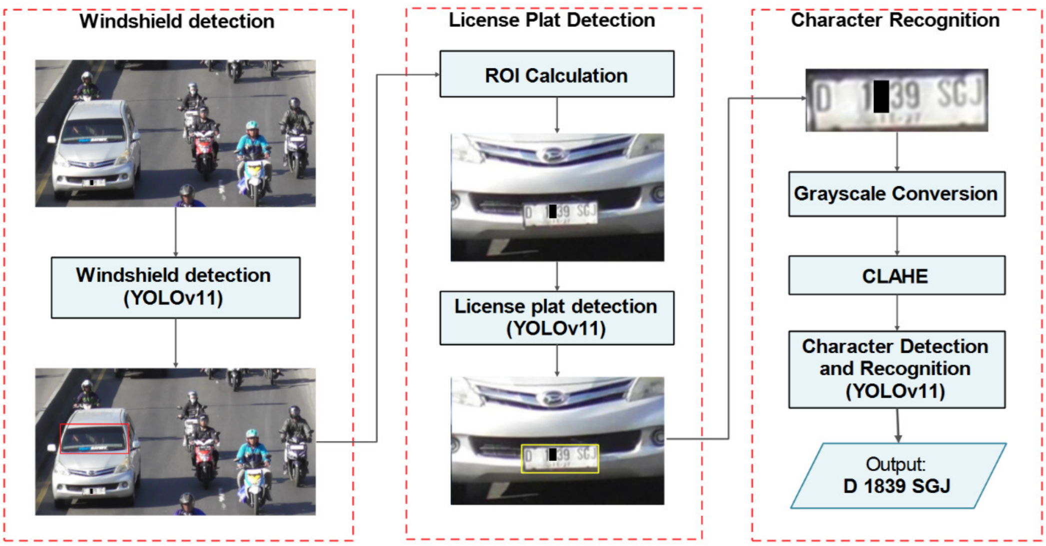


Fig. 4. The proposed system.

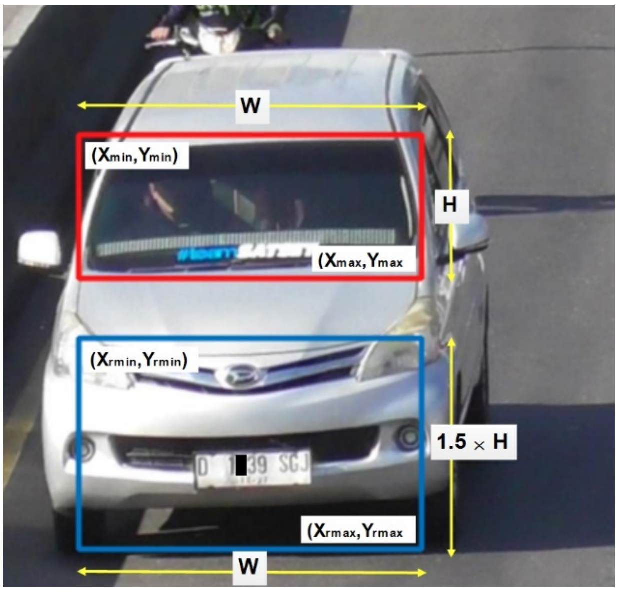
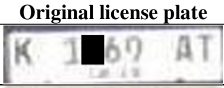
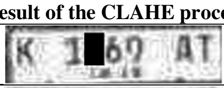
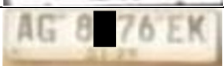
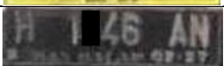


Fig. 5. Illustration of determining ROI (red bounding box is windshield and blue bounding box is ROI object).

3) *Character Recognition*

License plate character recognition extracts letters and numbers from objects on license plates. For this purpose, we have integrated the CLAHE algorithm and the YOLOv11 model. CLAHE is used to increase the contrast of the license plate image after it is converted to a grayscale image, and YOLOv11 is used for character detection and classification. Examples of some results of the CLAHE algorithm for increasing the contrast of license plate images are shown in Table I.

TABLE I. EXAMPLES OF THE CLAHE ALGORITHM APPLICATION

Original license plate	Result of the CLAHE process
	
	
	
	
	

CLAHE is an evolution of AHE (Adaptive Histogram Equalization), which tends to increase noise in relatively homogeneous areas. This contrast enhancement technique adjusts pixel intensity in small image regions. It improves local detail while minimizing artifacts and excessive noise amplification [26]. CLAHE provides a limit value on the histogram in the AHE problem. This limit value is called the clip limit. The CLAHE algorithm is generally divided into three main stages: tile generation, histogram equalization, and bilinear interpolation [27]. Tile generation divides the input image into several parts, each called a tile. Histogram equalization is then performed on each tile using a predetermined clip limit. The clip limit (β_c) is defined as in (3) [28]:

$$\beta_c = \frac{M_t}{N_g}\left(1 + \frac{\alpha_c}{100}(s_{max} - 1)\right) \tag{3}$$

where M_t is the area of the tile, N_g is the grayscale value (256), α_c is the clip factor (ranging from 0 to 100), and s_{max} is the slope limit of the transformation function. Histogram equalization is divided into five stages: histogram calculation, excess calculation, excess distribution, excess redistribution, and scaling and mapping using the Cumulative Distribution Function (CDF). The histogram is calculated as a set of bins for each tile. The histogram bin values above the clip limit are accumulated and distributed to the other bins. The CDF is then calculated using (4) for each histogram [28].

$$f_{i,j}(n) = \frac{(N_g - 1)}{M_t} \cdot \sum_{k=0}^{n} h_{i,j}(k) \tag{4}$$

where $h_{i,j}(k)$ are the histograms of the region (i,j) with $n = 0,1,2,\ldots,N_g - 1$. The CDF values of each tile are scaled and mapped using the pixel values of the input image.

C. *Performance Evaluation*

Precision (P), Recall (R), and Average Precision (AP) are used to measure the performance of the model. P and R are calculated using (5) and (6), while AP is calculated using (7) [29]. TP_{dec} is the correct detection of the ground-truth bounding box, FP_{dec} is the incorrect detection of an existing or absent object, and FN_{dec} is the ground truth undetected. $P_i(R_{n+1})$ is calculated using (8), where $P(\tilde{R})$ is the precision measured at recall $\tilde{R}$.

$$P = \frac{TP_{dec}}{TP_{dec} + FP_{dec}} \tag{5}$$

$$R = \frac{TP_{dec}}{TP_{dec} + FN_{dec}} \tag{6}$$

$$AP = \sum_n (R_{n+1} - R_n) P_i(R_{n+1}) \tag{7}$$

$$P_i(R_{n+1}) = \max_{\tilde{R}:\tilde{R} \geq R_{n+1}} P(\tilde{R}) \tag{8}$$

III. EXPERIMENTS AND RESULTS

Testing was conducted on each subsystem: windshield detection, license plate detection, and character recognition. Windshield detection was tested using the DS_Frame dataset, license plate detection was tested using the DS_ROI dataset, and test character recognition was tested using the DS_License_Plate dataset. The model was trained and tested on the NVIDIA A100. All tests used an image size of 640×640 pixels, a batch size of 16, and an epoch of 100.

A. *Windshield Detection*

Two variants of the YOLOv11 model were used for windshield detection, namely YOLOv11n and YOLOv11m. In addition, the YOLOv8m model was also tested in a previous study [30]. The results of this test are presented in Table II. From this table, it can be seen that the three models produce the same R and AP, but YOLOv11n is superior in terms of P and detection time. The P, R, and AP values of the YOLOv11n model are 0.879, 0.958, and 0.961, respectively.

Meanwhile, the detection time per image of YOLOv11n is 10.1 ms. Therefore, it can be concluded that the proposed method is more suitable for windshield detection due to its precision and computation time. An example of the windshield detection results on the video frame is shown in Figure 6.

TABLE II. WINDSHIELD DETECTION TEST RESULTS

Model	P	R	AP	Detection time (ms)
YOLOv11n	0.879	0.958	0.961	10.1
YOLOv11m	0.869	0.958	0.951	10.4
YOLOv8m [30]	0.869	0.958	0.951	11.0

Fig. 6. Examples of windshield detection on video frame image.

B. License Plate Detection

The model used for license plate detection is YOLOv11. Testing was carried out with several model variants, namely YOLOv11n, YOLOv11s, YOLOv11m, YOLOv11l, and YOLOv11x. The results of this test are presented in Table III. All models produce the same R and AP, 1 and 0.995, respectively. However, the highest P value was obtained with the YOLOv11n model. In addition, YOLOv11n requires the fastest time to detect license plates on each image, which is 8.6 ms.

The proposed method was compared with previous studies and several variants of YOLOv8, YOLOv9, and YOLOv10. Previous studies used YOLOv5 [14, 31, 32], YOLOv8s [15], and YOLOv8x [33]. The results of the comparison are shown in Table IV. From this table, it can be seen that the proposed method produces the highest P, R, and AP. The P values of the YOLOv5s, YOLOv8n, and YOLOv8m models also produce the highest values, and the AP values of all models are almost the same.

Meanwhile, the proposed method requires faster detection time when compared to all variants of the YOLOv8 and YOLOv9 models. The detection time of the YOLOv10 model variant is faster than the proposed method. YOLOv10n with 5.3 ms has the fastest detection time. Therefore, it can be concluded that the proposed model is more suitable for license plate detection with high P, R, and AP. An example of this license plate detection is shown in Figure 7.

TABLE III. TEST RESULTS OF LICENSE PLATE DETECTION

Model	P	R	AP	Detection time (ms)
YOLOv11n	1.000	1.000	0.995	8.6
YOLOv11s	0.985	1.000	0.995	9.7
YOLOv11m	0.999	1.000	0.995	10.3
YOLOv11l	0.997	1.000	0.995	10.6
YOLOv11x	0.989	1.000	0.995	11.6

TABLE IV. COMPARISON OF THE PROPOSED METHOD WITH YOLOV8, YOLOV9, AND YOLOV10 FOR LICENSE PLATE DETECTION

Model	P	R	AP	Detection time (ms)
YOLOv5s [14, 31, 32]	1.000	0.995	0.995	11.3
YOLOv8n	1.000	0.995	0.995	8.9
YOLOv8s [15]	0.999	0.995	0.995	9.0
YOLOv8m	1.000	1.000	0.995	9.3
YOLOv8l	0.990	1.000	0.995	10.1
YOLOv8x [33]	0.994	1.000	0.995	10.2
YOLOv9t	0.999	0.995	0.995	10.4
YOLOv9s	0.994	0.995	0.995	10.2
YOLOv9m	0.999	1.000	0.995	10.7
YOLOv9c	0.989	1.000	0.995	11.5
YOLOv9e	0.994	1.000	0.995	13.3
YOLOv10n	0.989	0.980	0.994	5.3
YOLOv10s	0.979	0.985	0.994	6.8
YOLOv10m	0.980	0.984	0.994	7.5
YOLOv10l	0.995	0.999	0.995	7.8
YOLOv10x	0.989	0.980	0.994	5.7
YOLOv11n (proposed)	1.000	1.000	0.995	8.6

Fig. 7. Examples of license plate detection.

C. Character Recognition

The character recognition test used two scenarios, namely, using only the YOLOv11n model and a combination of CLAHE and YOLOv11n (proposed method). Both tests are conducted to determine the effect of adding the CLAHE algorithm on character recognition. The results of this test are presented in Table V.

TABLE V. CHARACTER RECOGNITION TEST RESULTS

Class	***P***		***R***		***AP***	
	YOLOv11	**YOLOv11+CLAHE**	**YOLOv11**	**YOLOv11+CLAHE**	**YOLOv11**	**YOLOv11+CLAHE**
0	0.947	0.996	0.900	0.900	0.929	0.945
1	0.964	0.971	0.965	0.936	0.967	0.964
2	0.956	0.961	0.986	0.986	0.966	0.966
3	0.960	0.965	0.933	0.933	0.985	0.983
4	1.000	0.994	0.958	0.944	0.989	0.983
5	0.974	0.979	1.000	0.990	0.995	0.995
6	0.967	0.995	0.981	0.981	0.979	0.986
7	0.971	0.977	0.982	0.982	0.979	0.981
8	0.949	0.942	0.936	0.938	0.947	0.937
9	0.933	0.938	1.000	0.978	0.965	0.942
A	0.940	0.984	0.984	0.982	0.979	0.974
B	0.913	0.991	0.961	0.961	0.981	0.984
C	0.864	0.879	0.794	0.857	0.908	0.913
D	0.759	1.000	0.933	0.834	0.940	0.944
E	1.000	0.982	0.931	1.000	0.984	0.995
F	0.954	0.981	1.000	1.000	0.995	0.995
G	0.825	0.920	0.967	0.933	0.939	0.947
H	0.958	0.977	0.961	0.910	0.973	0.975
I	0.852	1.000	0.720	0.676	0.832	0.856
J	1.000	1.000	0.631	0.792	0.820	0.833
K	0.941	0.963	0.963	0.957	0.971	0.967
L	0.841	0.959	0.700	0.700	0.805	0.847
M	0.923	0.881	0.667	0.778	0.858	0.832
N	0.878	0.939	0.857	0.734	0.923	0.917
O	0.527	0.541	0.765	0.824	0.663	0.683
P	0.934	0.937	1.000	1.000	0.995	0.995
Q	1.000	1.000	0.396	0.000	0.541	0.560
R	0.920	1.000	1.000	0.998	0.995	0.995
S	0.903	0.907	1.000	1.000	0.974	0.970
T	0.798	0.832	0.929	0.929	0.904	0.927
U	0.994	1.000	1.000	0.974	0.995	0.995
V	0.886	0.933	1.000	1.000	0.995	0.993
W	0.313	0.996	1.000	0.900	0.746	0.945
X	0.830	0.971	1.000	0.936	0.995	0.964
Y	0.784	0.961	0.875	0.986	0.909	0.966
Z	0.929	0.965	1.000	0.933	0.995	0.983
All	0.891	0.922	0.908	0.876	0.925	0.931

The P, R, and AP values for all classes for the proposed method are 0.992, 0.876, and 0.931, respectively. The proposed method has higher P and AP values, although the R value is lower than the YOLOv11n model. The increase in the AP value is 0.6%. Therefore, it can be concluded that adding CLAHE in the preprocessing process can increase AP and is more suitable for character recognition on license plates. An example of the results of this character recognition is shown in Figure 8.

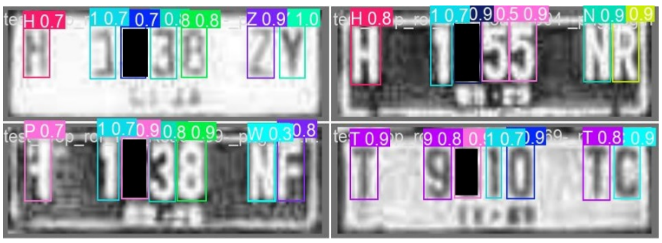


Fig. 8. Examples of license plate character recognition.

IV. CONCLUSION

In this study, a system for the detection and identification of car license plates was developed by combining the Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm and the You Only Look Once (YOLOv11) model. The proposed system consisted of three main subsystems: windshield detection, license plate detection, and character recognition. The proposed method achieved an average precision of 0.961 for windshield detection and 0.995 for license plate detection. For license plate character recognition, the proposed method achieved precision, recall, and average precision of 0.922, 0.876, and 0.931, respectively.

Tests showed that the YOLOv11n model had the best precision, recall, average precision, and detection time in windshield and license plate detection compared to other YOLO variants. In addition, the use of CLAHE to enhance image contrast improved the detection and identification of characters on license plates, resulting in higher precision and average precision. Therefore, the developed system has the potential to be used in traffic surveillance applications, especially in the monitoring of seat belt violations. Therefore, it can be implemented in automated law enforcement systems on highways.

ACKNOWLEDGMENT

The authors are grateful to the Indonesian Ministry of Research and Higher Education through Diponegoro University, which funded this study under contract number 601-56/UN7.D2/PP/VI/2024.

REFERENCES

[1] "The top 10 causes of death," *World Health Organization*. https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death.

[2] N. K. Al-Shammari and S. M. H. Darwish, "In-depth Sampling Study of Charactersitics of Vehcile Crashes in Saudi Arabia," *Engineering, Technology & Applied Science Research*, vol. 9, no. 5, pp. 4724–4728, Oct. 2019, https://doi.org/10.48084/etasr.2939.

[3] H. Guo, H. Lin, S. Zhang, and S. Li, "Image-based seat belt detection," in *Proceedings of 2011 IEEE International Conference on Vehicular Electronics and Safety*, Beijing, China, 2011, pp. 161–164, https://doi.org/10.1109/ICVES.2011.5983807.

[4] W. Li, J. Lu, Y. Li, Y. Zhang, J. Wang, and H. Li, "Seatbelt detection based on cascade Adaboost classifier," in *2013 6th International Congress on Image and Signal Processing*, Hangzhou, China, 2013, pp. 783–787, https://doi.org/10.1109/CISP.2013.6745271.

[5] B. Zhou, L. Chen, J. Tian, and Z. Peng, "Learning-based seat belt detection in image using salient gradient," in *2017 12th IEEE Conference on Industrial Electronics and Applications*, Siem Reap, Cambodia, 2017, pp. 547–550, https://doi.org/10.1109/ICIEA.2017.8282904.

[6] R. A. Kapdi, P. Khanpara, R. Modi, and M. Gupta, "Image-based Seat Belt Fastness Detection using Deep Learning," *Scalable Computing: Practice and Experience*, vol. 23, no. 4, pp. 441–455, Dec. 2022, https://doi.org/10.12694/scpe.v23i4.2027.

[7] J. Luo, J. Lu, and G. Yue, "Seatbelt detection in road surveillance images based on improved dense residual network with two-level attention mechanism," *Journal of Electronic Imaging*, vol. 30, no. 3, Jun. 2021, Art. no. 033036, https://doi.org/10.1117/1.JEI.30.3.033036.

[8] Z. Wang and Y. Ma, "Detection and recognition of stationary vehicles and seat belts in intelligent Internet of Things traffic management system," *Neural Computing and Applications*, vol. 34, no. 5, pp. 3513–3522, Mar. 2022, https://doi.org/10.1007/s00521-021-05870-6.

[9] S. B. Khalid and B. Hazela, "Employing Real-Time Object Detection for Traffic Monitoring," in *Proceedings of the International Conference on Innovative Computing & Communication*, New Delhi, India, 2021, https://doi.org/10.2139/ssrn.3834176.

[10] S. A. Ahmed and S. Dixit, "Number Plate and Logo Identification using Machine Learning Approches," *Journal of Theoretical and Applied Information Technology*, vol. 102, no. 3, pp. 1023–1036, Feb. 2024.

[11] B. Kumar, K. Kumari, P. Banerjee, and P. Jha, "An Implementation of Automatic Number Plate Detection and Recognition using AI," in *2023 International Conference on Advances in Computing, Communication and Applied Informatics*, Chennai, India, 2023, pp. 1–9, https://doi.org/10.1109/ACCAI58221.2023.10199855.

[12] R. Antar, S. Alghamdi, J. Alotaibi, and M. Alghamdi, "Automatic Number Plate Recognition of Saudi License Car Plates," *Engineering, Technology & Applied Science Research*, vol. 12, no. 2, pp. 8266–8272, Apr. 2022, https://doi.org/10.48084/etasr.4727.

[13] A. Sufiun, Md. H. I. Bijoy, N. R. Chakraborty, and M. A. A. K. Akash, "Automatic Bengali Number Plate Detection and Authentication using YOLO-V4 and YOLO-V5," in *2023 26th International Conference on Computer and Information Technology*, Cox's Bazar, Bangladesh, 2023, pp. 1–6, https://doi.org/10.1109/ICCIT60459.2023.10441416.

[14] R. Kala, K. Dharani, R. Harini, A. Niranjanaa, and M. Sowmiya, "Automatic Number Plate Detection With Yolov5 and OCR Methods," in *2024 International Conference on Knowledge Engineering and Communication Systems*, Chikkaballapur, India, 2024, pp. 1–5, https://doi.org/10.1109/ICKECS61492.2024.10617305.

[15] H. U. Iyer and S. Dhavale, "Automatic Number Plate and Face Recognition System for Secure Gate Entry into Military Establishments," in *2024 International Conference on Smart Systems for applications in Electrical Sciences*, Tumakuru, India, 2024, pp. 1–6, https://doi.org/10.1109/ICSSES62373.2024.10561368.

[16] K. Vijayalakshmi, M. Dhanamalar, V. A. Lepakshi, and S. Jamtsho, "Smart Checkpoint Management System for Automatic Number Plate Recognition in Bhutan Vehicles Using OCR Technique," *SN Computer Science*, vol. 5, no. 5, May 2024, Art. no. 579, https://doi.org/10.1007/s42979-024-02905-2.

[17] O. Sarkar, S. Sinha, A. K. Jena, A. K. Parida, N. Parida, and R. K. Parida, "Automatic Number Plate Character Recognition using Paddle-OCR," in *2024 International Conference on Innovations and Challenges in Emerging Technologies*, Nagpur, India, 2024, pp. 1–7, https://doi.org/10.1109/ICICET59348.2024.10616305.

[18] A. Parikh, H. Parikh, K. Patel, N. Bhatt, and P. Praiapati, "Number Plate Detection and Recognition Using OpenCV," in *2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things*, Bengaluru, India, 2024, pp. 1460–1467, https://doi.org/10.1109/IDCIoT59759.2024.10467269.

[19] V. Devisurya, S. Adil Mohamed, A. Gavin, and A. Kamal, "Enhanced Number Plate Recognition for Restricted Area Access Control Using Deep Learning Models and EasyOCR Integration," in *2024 3rd International Conference on Artificial Intelligence For Internet of Things*, Vellore, India, 2024, pp. 1–6, https://doi.org/10.1109/AIIoT58432.2024.10574677.

[20] I. J. Khan, Md. F. Bin Amin, M. H. Sabbir, D. M. Nejhum, A. H. Muhammad Nanzil, and R. Rahman, "Vehicle Number Plate Detection and Encryption in Digital Images Using YOLOv8 and Chaotic-Based Encryption Scheme," in *2024 6th International Conference on Electrical Engineering and Information & Communication Technology*, Dhaka, Bangladesh, 2024, pp. 717–722, https://doi.org/10.1109/ICEEICT62016.2024.10534375.

[21] G. Jocher, J. Qiu, and A. Chaurasia, "Ultralytics YOLO11," *Ultralytics YOLO Docs*. https://github.com/ultralytics/ultralytics.

[22] A. Wanto, Y. Yuhandri, and O. Okfalisa, "Optimization Accuracy of CNN Model by Utilizing CLAHE Parameters in Image Classification Problems," in *2023 International Conference on Networking, Electrical Engineering, Computer Science, and Technology*, Bandar Lampung, Indonesia, 2023, pp. 253–258, https://doi.org/10.1109/IConNECT56593.2023.10327100.

[23] M. Widyaningsih, T. K. Priyambodo, M. E. Wibowo, and M. Kamal, "Optimization Contrast Enhancement and Noise Reduction for Semantic Segmentation of Oil Palm Aerial Imagery," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 1, pp. 597–609, Feb. 2023, https://doi.org/10.22266/ijies2023.0228.51.

[24] R. A. Pramunendar, D. P. Prabowo, D. Pergiwati, Y. Sari, P. N. Andono, and M. A. Soeleman, "New Workflow for Marine Fish Classification Based on Combination Features and CLAHE Enhancement Technique," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 4, pp. 293–304, Aug. 2020, https://doi.org/10.22266/ijies2020.0831.26.

[25] A. Ghosh, "YOLO11: Faster Than You Can Imagine!,"*LearnOpenCV*. https://learnopencv.com/yolo11/.

[26] N. Sajitha and S. P. Priya, "Optimal Artificial Neural Network-based Fabric Defect Detection and Classification," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13148–13152, Apr. 2024, https://doi.org/10.48084/etasr.6773.

[27] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics gems IV*, P. S. Heckbert, Ed. San Diego, CA, USA: Academic Press Professional, Inc., 1994, pp. 474–485.

[28] A. M. Reza, "Realization of the Contrast Limited Adaptive Histogram Equalization (CLAHE) for Real-Time Image Enhancement," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 38, no. 1, pp. 35–44, Aug. 2004, https://doi.org/10.1023/B:VLSI.0000028532.53893.82.

[29] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A Survey on Performance Metrics for Object-Detection Algorithms," in *2020 International Conference on Systems, Signals and Image Processing*, Niteroi, Brazil, 2020, pp. 237–242, https://doi.org/10.1109/IWSSIP48289.2020.9145130.

[30] Sutikno, A. Sugiharto, and R. Kusumaningrum, "Automated Detection of Driver and Passenger Without Seat Belt using YOLOv8," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 11, pp. 806–813, 2023, https://doi.org/10.14569/IJACSA.2023.0141181.

[31] H. J. Farell, K. J. Mudita, and R. Sutoyo, "Automatic Number Plate Recognition Performance Comparison in Old and New Indonesian License Plates Using Deep Learning," *ICIC Express Letters*, vol. 18, no. 8, pp. 793–799, Aug. 2024, https://doi.org/10.24507/icicel.18.08.793.

[32] A. Srivastava, P. Narote, S. Fatangare, S. Kakade, and R. Kulkarni, "Automatic Number Plate Detection System for Indian Vehicles Using Yolov5 and EasyOCR," in *Proceedings of the 12th International Conference on Soft Computing for Problem Solving*, Roorkee, India, 2024, pp. 305–314, https://doi.org/10.1007/978-981-97-3292-0_21.

[33] A. Naureen, K. C. Krishna, N. S. Teja, L. Mahesh, and K. Vamshi, "College Bus Number Plate Registration Detection Using YOLO-V8," in *2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques*, Bengaluru, India, 2023, pp. 1–7, https://doi.org/10.1109/EASCT59475.2023.10392472.