Automated License Plate Detection and Recognition using YOLOv8 and OCR With Tello Drone Camera

Hanif Fakhrurroja
School of Industrial Engineering
Telkom University
National Research and Innovation Agency
Bandung, Indonesia
haniff@telkomuniversity.ac.id

Dita Pramesti
School of Industrial Engineering
Telkom University
Bandung, Indonesia
ditapramesti@telkomuniversity.ac.id

Abdul Rofi Hidayatullah
Department of Electrical Engineering
UIN Sunan Gunung Djati
Bandung, Indonesia
1207070001@student.uinsgd.ac.id

Ahda Arif Fashihullisan
Department of Electrical Engineering
UIN Sunan Gunung Djati
Bandung, Indonesia
1207070006@student.uinsgd.ac.id

Harry Bangkit
Research Center for Smart Mechatronics
National Research and Innovation Agency
Bandung, Indonesia
harr003@brin.go.id

Nanang Ismail
Department of Electrical Engineering
UIN Sunan Gunung Djati
Bandung, Indonesia
nanang.is@uinsgd.ac.id

2023 International Conference on Computer, Control, Informatics and its Applications (IC3INA) | 979-8-3503-9487-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/IC3INA60834.2023.10285750

***Abstract*— This paper proposes an Automated License Plate Recognition (ALPR) system using YOLOv8 and Optical Character Recognition (OCR). This system is developed with the Python programming language and uses the OpenCV library for image processing and Pytesseract library as an OCR engine. The camera used to detect vehicle license plates is a DJI Tello Drone camera. The purpose of this research is to implement an electronic ticketing system for vehicles that violate traffic rules. The ALPR system consists of several stages. First, YOLOv8 is used to detect the vehicle license plate in the image captured by the DJI Tello Drone camera. YOLOv8 is an object detection method that is efficient and accurate in recognizing various objects, including vehicle license plates. Once the vehicle license plate is detected, the license plate image will be processed using the OpenCV library to improve image quality and facilitate the character recognition process. Furthermore, the system will use OCR, specifically the Pytesseract library, to recognize the characters on the vehicle number plate. From 50 photo samples, the success rate for license plate detection is 100%, and character recognition is 66%. The proposed ALPR system not only achieves high accuracy rates, but it is also efficient, versatile, cost-effective, and scalable. Because of these benefits, it is a useful contribution to the field of ALPR and traffic management systems, with the potential for future developments and broader applications.**

***Keywords*—ALPR, YOLOv8, OCR, Detection, Recognition**

I. INTRODUCTION

Automatic License Plate Recognition (ALPR) systems have become an essential component of modern transportation services, allowing for the automated management of traffic, parking, toll stations, and other road operations, as well as surveillance and threat detection [1].

Automatic Number Plate Recognition (ANPR) is one of many information systems used for data extraction from given vehicle photos. ANPR is a technique that reads car registration plates by using optical character recognition on photographs. Police departments use ANPR systems for law enforcement purposes, such as determining whether a vehicle is registered or licensed [2][3].

Deep learning-based approaches for license plate identification and recognition have recently been deployed. The YOLO (You Only Look Once) algorithm is one such method. YOLO is an object detection system that can recognize things in photos and videos in real time. Because of its speed and accuracy, YOLO is a popular license plate detection and identification method [2].

Another method used for license plate detection and recognition is OCR (Optical Character Recognition). OCR is a technology that recognizes text within images. OCR is used to convert the captured image into text, which can be used for further analysis. OCR is an important component of license plate detection and recognition systems as it allows for the extraction of license plate numbers from images [1][2].

In terms of accuracy, You Only Look Once (YOLO) is the finest real-time object detection algorithm among the several available. Currently, YOLOv8 is the most recent YOLO method, released on January 10, 2023, with values on the Microsoft COCO dataset ranging from 37.3 mAP for the trained model's nano version (YOLOv8n) to 53.9 mAP for the trained model's extra-large version (YOLOv8x) [4]. The YOLO 8 algorithm is employed for object detection, enabling for real-time license plate identification. The OCR process is then used to turn the acquired image into text, which may subsequently be analyzed further.

The Automated License Plate Recognition (ALPR) system is a critical component of modern traffic management and law enforcement systems. Its primary purpose is to accurately and efficiently identify vehicle license plates and recognize the characters on them. However, the existing ALPR systems face several challenges and limitations, including:

1. Accuracy in License Plate Detection: Many ALPR systems struggle with accurately detecting license plates, particularly in challenging conditions such as low light, adverse weather, or unusual angles. This can result in missed violations and compromised security.
2. Character Recognition Robustness: Accurate character recognition is paramount for identifying vehicles and issuing electronic tickets. Variations in font, size, and style on license plates, along with potential image distortions, pose significant challenges to character recognition accuracy.
3. Real-Time Processing: In traffic enforcement scenarios, real-time processing is crucial for immediate action. Delays in license plate detection and character recognition can hinder the timely issuance of electronic tickets.

4. Adaptability to Diverse Environments: ALPR systems should be adaptable to diverse geographical and environmental conditions, as well as different types of vehicles and license plate designs. A one-size-fits-all approach may not be effective.
5. Cost and Accessibility: Developing ALPR systems that are cost-effective and accessible for deployment in various locations, including urban and remote areas, is a significant concern.

The aforementioned challenges and limitations necessitate the development of more accurate, efficient, and adaptable ALPR systems. This research aims to address these issues by proposing an ALPR system that leverages YOLOv8 for precise license plate detection and Pytesseract for robust character recognition. The objective is to create a comprehensive solution that not only increases accuracy but also enhances real-time processing capabilities, making it suitable for the implementation of an electronic ticketing system for vehicles violating traffic rules.

By tackling these challenges and providing a solution that significantly improves ALPR system performance, this research contributes to the advancement of traffic management and law enforcement technologies, ultimately enhancing road safety and compliance with traffic regulations.

II. System Model

The software model is the most significant component of this system. The software model employs a number of image processing techniques that are written in Python. The ALPR algorithm is separated into three sections:

- Image capturing using a drone
- Plate extraction from the captured images
- Recognition of the extracted plate numbers [5].

In building the system, we use Python 3.7.6 as the programming base with several modules needed, including the following:

- OpenCV (cv2)
- YOLOv8 (ultralytics)
- Tello SDK (djitellopy)
- Google's Tesseract OCR Engine (PyTesseract)

Below is a specific description of this system model illustrating the ALPR system flow, from image input to final recognition output. Each module plays its role in achieving accurate license plate detection and recognition:

1. Input Module
    - The input module receives an image containing one or more vehicles with visible license plates from the drone's camera.
    - The input image is provided as an input to the ALPR system.
2. Object Detection Module:
    - The object detection module utilizes the YOLOv8 model for license plate detection.
    - The YOLOv8 model is pre-trained on a large dataset and capable of identifying objects in real-time.
    - The module takes the input image and predicts bounding boxes around license plates.
    - Detected bounding boxes are visualized on the image for reference.
3. Region of Interest (ROI) Extraction:
    - The ROI extraction module crops the region of interest based on the predicted bounding boxes.
    - For each detected license plate, the corresponding ROI is extracted from the input image.
    - The ROI represents a smaller image region containing the license plate.
4. License Plate Preprocessing:
    - The license plate preprocessing module applies different preprocessing techniques based on the average color of the ROI.
    - If the average color suggests a black or red license plate, the following steps are applied:
        - Conversion to grayscale.
        - Denoising using fast non-local means denoising.
        - Gaussian blurring to reduce noise.
        - Normalization of pixel values.
        - Thresholding to create a binary image.
        - Bilateral filtering for edge preservation.
        - Inversion of pixel values.
    - If the average color suggests a white or yellow license plate, the following steps are applied:
        - Conversion to grayscale.
        - Gaussian blurring to reduce noise.
        - Adaptive thresholding to create a binary image.
5. Optical Character Recognition (OCR):
    - The OCR module utilizes the pytesseract library to perform text recognition on the processed ROI.
    - The pytesseract library is configured with the Tesseract OCR engine.
    - The preprocessed ROI is passed to the OCR engine for text extraction.
    - The recognized text from each license plate is obtained.
6. Output Module:
    - The output module presents the results of the ALPR system.
    - The processed ROI, along with the recognized text, is displayed for each license plate.

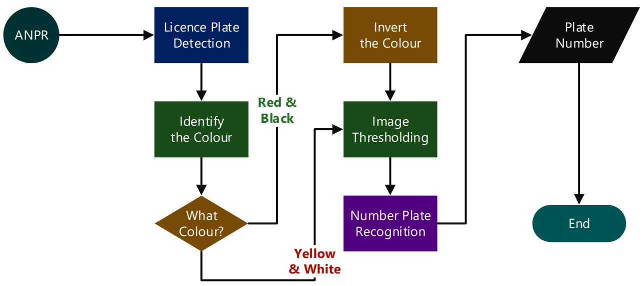

Fig. 1 Flowcharts of automatic number plate recognition model.

III. Methodology

A. Dataset

Image dataset containing vehicles with visible license plates is collected and pre-processed, including resizing and annotation. Furthermore, the YOLOv8 model is trained using the prepared dataset, with division into training and validation sets, the YOLOv8 train model used is the MEDIUM or YOLOv8m model, the number of images used for training is 1024 plate images, and the number of epochs used is 1000 times.

B. *Automatic Number Plate Recognition*

The initial step involves capturing images using a DJI Tello Drone programmed to photograph vehicles and transmit the images directly to a PC. The captured image is then detected with YOLOv8 which will then process the plate extraction. The choice of the DJI Tello Drone camera for license plate detection is a deliberate decision based on its unique capabilities, including the aerial perspective, wide area coverage, flexibility, mobility, and cost-effectiveness. These attributes collectively contribute to the system's performance by potentially improving license plate detection accuracy and enabling efficient traffic monitoring. The use of a drone camera represents an innovative approach to ALPR system deployment, addressing specific challenges in traffic management and law enforcement.

The second step of the ANPR (Automatic Number Plate Recognition) system is the extraction of the license plate on the image. A bounding box obtained from the previous detection results is utilized to extract the Region of Interest (ROI) that contains the license plate within the image. Such as license plates with black, white, red, or yellow backgrounds, and alphanumeric characters written in black or white. After identifying the ROI, the image is then cropped so that it only contains the license plate. coordinates and size calculations are performed as well as image cropping based on these equations. The coordinates and size are used to calculate the upper, lower, left, and right boundaries of the region of interest (ROI), which is the area containing the detected license plate. The resulting cropped ROI image can be used for further processing, with the following equation:

for the upper and lower boundaries of ROI:

$$\begin{aligned} y1 &= y - 0.5 * h \\ y2 &= y - 0.5 * h \end{aligned} \tag{1}$$

for the left and right boundaries of ROI:

$$\begin{aligned} x1 &= x - 0.5 * w \\ x2 &= x - 0.5 * w \end{aligned} \tag{2}$$

Next, the image undergoes filtering using two distinct techniques based on the background color of the license plate, utilizing conditional statements (IF conditioning). The pre-processing stage consists of several steps, including cropping, grayscale conversion, image inversion, blurring, and thresholding with inversion. The purpose of pre-processing is to transform images captured by cameras into processed images that are suitable for subsequent processing [6].

The third step for character recognition used Optical Character Recognition (OCR) technology to recognize the vehicle number. OCR uses a correlation mechanism to match individual characters, allowing the license plate number to be identified and saved as a string in a variable. The string is then compared to the vehicle authorisation database. The resulting signal is determined by the outcome of the comparison. The product of OCR used in this research is PyTesseract. PyTesseract is a Python-based OCR tool that acts as a wrapper for Google's Tesseract-OCR Engine, allowing text recognition in images. It can read and recognize text embedded in images, making it useful in image-to-text OCR use cases in Python. A complete overview of the workings and workflow is shown in Figure 2 and Figure 3 [5].

For the output of the ANPR system, the processed ROI, along with the identified text, is displayed for each license plate. The extracted text can be used for various purposes, such as vehicle identification, tracking, or issuing electronic tickets to traffic violators.

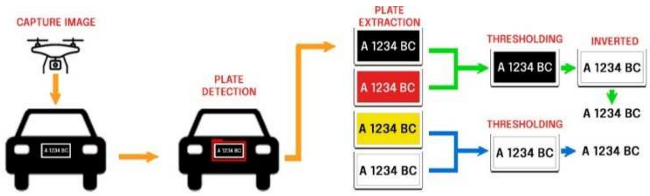

Fig. 2 Illustration of ANPR working system.

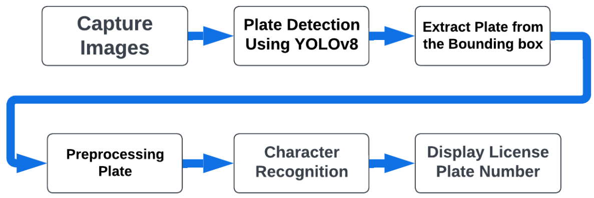

Fig. 3 ANPR workflow.

IV. RESULT AND DISCUSSION

For license plate object detection, we use two main models: YOLOv8 for object detection and PyTesseract for optical character recognition. In this project The YOLOv8m (medium) model is used to train a dataset that we own and get from the Roboflow open-source website for license plate detection, a sample of the dataset used is shown in Figure 4. The model uses 640-pixel images, which is sufficient to detect and recognize license plates. The model is tuned by training with 1000 epochs for good accuracy. For character recognition, we used the PyTesseract library.

Fig. 4 Example dataset used to be trained by the YOLOv8m model

Previous research utilized the YOLO versions 4 to 7 along with Easyocr library for the ALPR program development [7]. The test results showed high accuracy, although no preprocessing techniques were employed to enhance image quality. In our research we used the YOLOv8 version, YOLOv8 is an improved version of YOLOv2, which uses a model with 19 convolutional layers and 5 max-pooling layers [8]. YOLOv8 can be used for real-time object detection to detect and track other vehicles, pedestrians, and traffic signals.

PyTesseract is utilized for OCR due to its popularity and robustness, making it one of the most powerful OCR libraries for Python. It leverages the optical character recognition model developed by Google Tesseract OCR and offers a user-friendly Python interface. PyTesseract is a good choice as it can recognize vehicle license plate characters well. However, it is important to note that the quality of the license plate image can affect the OCR performance. one of the drawbacks of PyTesseract is that it requires some pre-processing operations before it can be used, such as grayscale image transformation and median blur.

A. Program Simulation

License plate detection is done by applying the ALPR code that has been designed. So that it accurately detects the license plate and forms a bounding box from the detected license plate. The proposed method is carried out through the stages described in the system model. With the first steps taking pictures using the DJI Tello Drone camera and the results of the vehicle photos are forwarded to the Python module for license plate detection and extraction.

The second step is the license plate detection process using the YOLO method from the trained dataset. This stage will display the bounding box on the detected license plate, shown in Figure 5. The third step is the detected license plate will be cut, illustrated in Figure 6. In the fourth step, the detected and separated license plate will be pre-processed in this case the pre-processing will be determined according to the background color of the plate. The only differentiator of this process is the inverse color of the plate. If the background color of the plate is black or red, an inverse color process will be added to change the background color of the plate to white and the characters on the plate to black. However, for plates with a white background, the inverse process will not be added, as shown in Figure 7. The last step is PyTesseract will perform optical character recognition on the plate that has been processed, the results can be seen in Figure 8.

Fig. 5 The plate detection result is marked with a bounding box.

Fig. 6 Cropped license plates (a)Black; (b)Red; (c)White

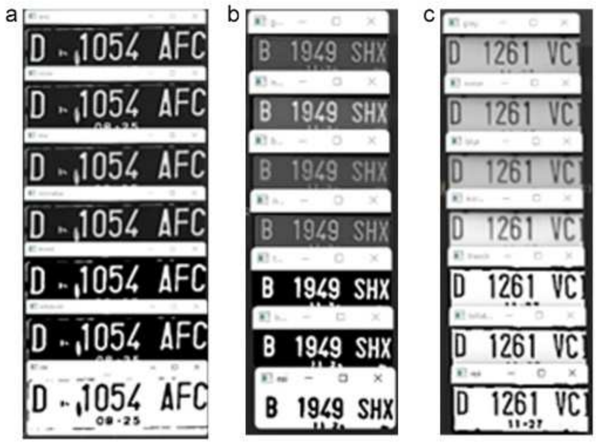

Fig. 7 Preprocessing before character recognition. (a)Black; (b)Red; (c)White.

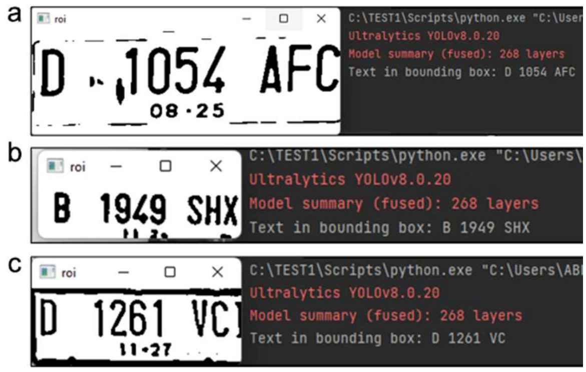

Fig. 8 Preprocessed ROI is passed to the OCR engine for text extraction. (a)Black; (b)Red; (c)White.

The number of vehicle photo samples taken for testing the detection and recognition program is 50 images of vehicles consisting of cars and motorcycles. Photos taken by the DJI Tello Drone were taken at different times in the morning, afternoon, evening, and night. 5 photos were taken at night and the rest were taken during the day. The success of character recognition on license plates is influenced by the quality of the photos and the circumstances that occur when the photos are taken. in this test, some of the photos taken experienced a slight decrease in quality due to a lack of place conditions and lighting conditions, the detection and recognition accuracy is shown in Table 1.

TABLE I. SUCCESS RATIO OF DETECTION AND RECOGNITION

Operation	Total No. of Samples	No. of successful Samples	Failure Samples	Success Ratio
License Plate Detection	50	50	0	100%
Character Recognition	50	33	17	66%

The percentage of success ratio is calculated from the equation cited in [9] where the number of successful samples is divided by the total number of samples [10][11].

B. Discussion

The use of DJI Tello cameras to take pictures of vehicles is done for the development of electronic ticketing. This DJI Tello drone supports programming in the Python programming language. The DJI Tello drone camera is adequate to take pictures but must be in good conditions such as during the day and good lighting conditions. If it is in a closed room and minimal light, the resulting image is not good. Although the image quality of the vehicle is poor the license plate can still be detected but during the character recognition process it is not possible to produce good and appropriate results. However, the license plate detection algorithm has shown limitations when dealing with yellow license plates in certain scenarios. In this research, the dataset used for training the YOLOv8 model is open-source data from the Roboflow website and there are no yellow license plates that are part of the data to be trained.

License plate detection using the YOLOv8 model that has been trained is able to detect vehicle plates even though the vehicle is at a distance. For character recognition on the vehicle photo plate, you must be at least 2-4 meters away from the drone so that the photo plate taken is not broken.

The study observed a relatively low success rate of 66% in the character recognition phase, primarily attributed to image conditions and quality. 17 failed sample photos taken were taken in low lighting conditions and the rest were images taken when the drone was flying and the resulting photos were blurry. The success rate in the character recognition process can also be influenced by the shape or condition of different plates, such as light reflections and dirt on the plate.

The proposed ALPR system offers several advantages compared to previous research methods:

1. High Accuracy: The integration of YOLOv8 for license plate detection and Pytesseract for character recognition significantly enhances the accuracy of the ALPR system. The addition of 100% accuracy in license plate detection and 66% accuracy in character recognition demonstrates the system's reliability in accurately identifying license plates and characters, contributing to more effective electronic ticketing for traffic violations.
2. Efficiency: YOLOv8 is known for its efficiency in object detection. By using YOLOv8, the ALPR system can quickly and accurately identify vehicle license plates, making it suitable for real-time applications such as traffic enforcement.
3. Versatility: YOLOv8 is capable of recognizing various objects, not just license plates. This versatility allows the system to potentially be used for other object detection tasks, making it a valuable tool beyond its primary application.
4. Cost-Effective: The use of a DJI Tello Drone camera for image capture is cost-effective compared to other specialized camera systems. This lowers the overall cost of implementing the ALPR system, making it more accessible for deployment in different locations.
5. Open-Source Tools: The system relies on open-source libraries like OpenCV and Pytesseract, which are readily available and continuously maintained by the community. This ensures that the system can be easily adopted and adapted by researchers and developers.
6. Scalability: The paper suggests that the system can be further improved by incorporating more advanced deep-learning models and OCR engines. This scalability allows for future enhancements to adapt to evolving technology and research advancements.

V. Conclusion

Our ALPR (Automatic License Plate Recognition) system successfully encompasses the character recognition process on vehicle plates, incorporating image enhancement techniques such as noise reduction, filtering, and thresholding. In this system, YOLOv8 is employed as the detection method, utilizing a convolutional architecture to generate bounding boxes that mark the detection results with a red square outline. And to separate the detected objects we use equations to cut the part. This ALPR system still does not provide maximum results in the yellow plate detection process and in the character recognition process. Several external factors impact these results, including the shape, condition, and size of the license plate, as well as the quality of the captured photos. In future developments, the vehicle plate dataset will be expanded by obtaining data from original sources that include a wider range of plate variations. The drone used should use a professional drone if it will be implemented for electronic ticketing and then add the right image enhancement process so that the character recognition results get the appropriate results.

Acknowledgment

The authors would like to thank Telkom University for supporting some of this research. The authors would also like to thank the anonymous reviewers for their insightful comments.

References

[1] Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic License Plate Recognition (ALPR): A State-of-the-Art Review," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 2, pp. 311-325, Feb. 2013, doi: 10.1109/TCSVT.2012.2203741.

[2] Fernandes, L.S. et al. (2020). A Robust Automatic License Plate Recognition System for Embedded Devices. In: Cerri, R., Prati, R.C. (eds) Intelligent Systems. BRACIS 2020. Lecture Notes in Computer Science, vol 12319. Springer, Cham. https://doi.org/10.1007/978-3-030-61377-8_16.

[3] Vi, G.V., Faudzi, A.A.b.M. (2022). A Study on Different Techniques in ALPR System: The Systems Performance Analysis. In: Ab. Nasir, A.F., Ibrahim, A.N., Ishak, I., Mat Yahya, N., Zakaria, M.A., P. P. Abdul Majeed, A. (eds) Recent Trends in Mechatronics Towards Industry 4.0. Lecture Notes in Electrical Engineering, vol 730. Springer, Singapore. https://doi.org/10.1007/978-981-33-4597-3_56.

[4] J. Solawetz dan F., “What is YOLOv8? The Ultimate Guide.,” 11 Januari 2023. [Online]. Available: https://blog.roboflow.com/whats-new-in-yolov8. [Accessed May 13, 2023].

[5] Qadri, M.T. & Asif, Muhammad. (2009). Automatic Number Plate Recognition System for Vehicle Identification Using Optical Character Recognition. 335 - 338. 10.1109/ICETC.2009.54

[6] Sugeng S, Syamsuddin EY. Designing automatic number plate recognition (ANPR) systems based on K-NN machine learning on the raspberry pi embedded system. JTEV (Jurnal Teknik Elektro dan Vokasional). 2019 Sep 23;5(1.1):19-26. https://doi.org/10.24036/jtev.v5i1.1.106135

[7] Laroca, Rayson & Severo, Evair & Zanlorensi, Luiz & Soares de Oliveira, Luiz & Gonçalves, Gabriel & Schwartz, William & Menotti, David. (2018). A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector. 10.1109/IJCNN.2018.8489629.

[8] Vedhaviyassh, D.R., R. Sudhan, G. Saranya, Mozhgan Safa and D Arun. “Comparative Analysis of EasyOCR and TesseractOCR for Automatic License Plate Recognition using Deep Learning Algorithm.” 2022 6th International Conference on Electronics, Communication and Aerospace Technology (2022): 966-971.

[9] Y. dan A. Sulchan, “The Effectiveness against Traffic Violations with Electronic Traffic Law Enforcement (ETLE),” Law Development Journal, 28 Desember 2021.

[10] Fakhrurroja, H., Machbub, C., & Prihatmanto, A. S. (2020). Multimodal Interaction System for Home Appliances Control. International Journal of Interactive Mobile Technologies, 14(15).

[11] Fakhrurroja, H., Purwarianti, A., Prihatmanto, A. S., & Machbub, C. (2018, November). Integration of indonesian speech and hand gesture recognition for controlling humanoid robot. In 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV) (pp. 1590-1595). IEEE.