2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC) | 979-8-3503-8459-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/AIC61668.2024.10730839

Enhanced License Plate Detection using YOLOv8n: An Approach for Efficient Vehicle Identification

Vishesh Chandok
Symbiosis Institute of Technology Nagpur Campus, Symbiosis International (Deemed University),
Pune, India
vchandok73@gmail.com

Sudhanshu Maurya
Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University),
Pune, India
dr.sm0302@gmail.com

Kashish Mirza
Department of Computer Science & Engineering,
Graphic Era Hill University, Bhimtal, Campus, Uttarakhand, India
kashishmirza99@gmail.com

Ajay Kumar Yadav
Department of Computer Application, United Institute of Management,
Prayagraj, India
ajaykumar.uim@gmail.com

Sneha Thomas Varghese
Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University), Pune, India
thomassneha777@gmail.com

Rachit Garg
School of Computer Science and Engineering,
Lovely Professional University, Phagwara, India
rachit.garg.in9@gmail.com

Abstract—**An essential feature of the intelligent transportation system is the automatic detection and recognition of license plate information. The e-payment systems for parking and toll collection in traffic control security could be utilized with this application. Numerous algorithms have been created for License Plate Recognition (LPR) and Identification, each with pros and cons depending on the circumstances. Computer vision has increased in terms of techniques and developments since deep learning emerged and became popular in several artificial intelligence domains. It is one of the most efficient ways to monitor traffic signals, tax paid for parking, and police monitoring through Automatic LPR. Although it, as a technology, is widely used and developed, much more must be done to improve the system's accuracy. The current science community has excelled in efficiency and problem-solving in recent years. Along with the You Only Look Once (YOLO), targeted at the object detection area and applied to this study, our primary goal is to summarize and analyze several methods and advancements in LPR in the deep learning age.**

Keywords—License plate recognition, YOLO, Deep learning, artificial intelligence

I. INTRODUCTION

Due to rising traffic and the rapid development of urban areas, Automatic License Plate Recognition (ALPR) has received much attention in the last ten years in the field of intelligent, innovative systems for transportation [1]. Using license plates (LPs), these systems will help identify vehicles with intelligent traffic monitoring. It can be used to identify cars that are blacklisted so that law enforcement can seize them. It is also possible to monitor vehicles that go above city speed limits. Additionally, installing this system in a parking lot can provide numerous advantages, such as automatic payment processing, ticketless parking fee management, and reducing auto theft. There are two steps to recognize the text on the number plate. The first step in detecting the number plate is object detection. After the detection, it acts as an input to extract the number from it, which is the second step. Even though a lot of work has been done in this field, a lot of methods have drawbacks and only work in certain situations, such as lighting, the existence of noise, blur, distortion, tilting the camera when taking pictures, and other fundamental limitations are some examples.

The yolov8n is a paralytic product mainly used for object detection and provides very high accuracy compared to various CNN models. Compared to faster R-CNN, YOLO has higher implementations. YOLO has proven to be cleaner and more efficient in product search because it provides end-to-end training. Both algorithms are pretty accurate, but sometimes, YOLO runs faster than R-CNN in terms of accuracy, speed, and efficiency.

The proposed Automatic License Plate Detection model involves obtaining a license plate image of the vehicle, testing, and training by yolov8n. This model helps in object detection with high efficiency and accuracy. The obtained number plate image will act as input for the EasyOcr module, extracting the text from the number plate. The major aims of this manuscript are given below:

- A new license plate detection model consisting of two steps is proposed.
- Use of the YOLOv8n model to improve the accuracy of character recognition.
- Implement the model and compare the results with the existing system, testing its effectiveness and accuracy.

979-8-3503-8459-8/24/$31.00 ©2024 IEEE
DOI: 10.1109/AIC.2024.150

Authorized licensed use limited to: Polytechnic University of Bucharest. Downloaded on March 17,2025 at 18:24:37 UTC from IEEE Xplore. Restrictions apply.

II. LITERATURE REVIEW

A Back-Propagation Neural Network (BPNN) classifier class self-developed by Fei Xie et al. was used with an efficient feature extraction strategy to detect license plates. The author claims this can evade the shortcomings of low visibility and complex backgrounds. They fortified the car's image for extracting the number plate from the car image. Furthermore, speaking recognition is done through training the backpropagation neural network model.

Ravi Kiran and co-workers [2] reported a vehicle number plate processing method using various conditions, including noisy environment, low light, and vehicle number plate position. Here, the author proffered three types of image procedures: Gaussian smoothing filter, Gaussian Thresholding, and morphological transform. Nevertheless, the algorithm employed for the recognition of characters is the K Nearest Neighbor algorithm, a new method for Indian vehicles' license plates produced by Hanit Karwal et al. [3]. The algorithms used to solve the problems of finding a face and the size of the system are developed. For instance, Tejasi K et al. [4] presented an artificial-intelligent license plate-recognition system through a Sobel edge detection algorithm to acquire plate region, and morphological operation is used for its segmentation. They have referred to the IoT for every kind of routine purpose, providing them with a massive database of vehicles.

Swati Jagtap [5] uncovered a procedure whereby the license plate matches the user's database, and only permitted vehicles are allowed to pass. A histogram of the oriented gradient method was used to achieve the experimental feature extraction task.

This method is developed to explore the question: How are characters related to each other through row distance? This algorithm maps an image, pixel by pixel, at row intervals N and counts the number of edges in that image. If the amount of connections is over a threshold, then the license plate is considered to be found; if the number of the alpha is below the threshold, the threshold will be lessened [6]. This method is rapid and usually has good outcomes on fully featured pictures. The limit of this work is that the edge precisely suits those good edges and uninterrupted edges. They are seen as well off when they are applied to simple pictures. In [7], the author selects a wavelet transform algorithm to extract the critical points for later use as reference information for locating the license plate. This technique even finds multiple license plates at the same site. Firstly, the contours are detected, and then the Hough transformation is used to find the corners of the license plate. That way, it is computationally inefficient, and it is hard to recognize the number plate region if the border of the number plate is blurry or distorted or the image covers a large number of vertical and horizontal edges [8].

The GrabCut algorithm for automatic license plates is implemented by using the geometric information of the license plates as a starting point for further capture [9]. This approach has limitations because the number of licenses varies from site to site. Removing the permission process is always design-dependent and cannot be adapted to all images. So, the traditional method of searching for a driver's license is ineffective and inaccurate. The algorithm creates some competing sites, and then the competing sites are separated and processed again [10], [11].

For instance, many other categories, such as end-to-end detection algorithms, will find the target class with its coordinates and then return the estimated class probabilities using SSD [12] or YOLO [13 - 15] techniques.

III. PROPOSED MODEL

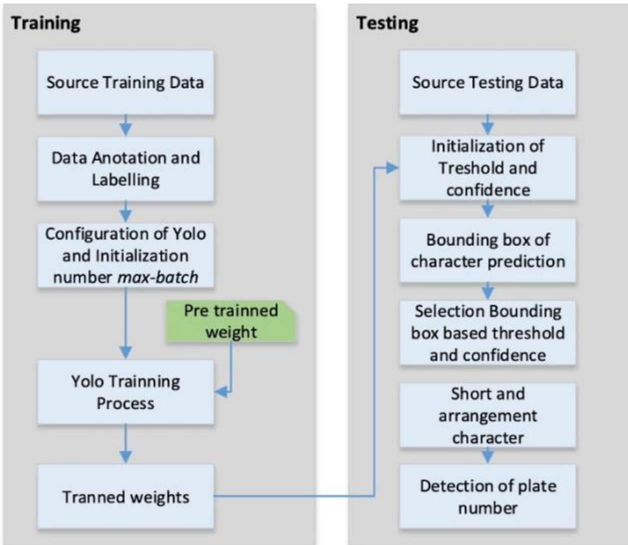
When YOLO for object detection was first proposed, there were many algorithms and methods for object recognition, but YOLO took a brand-new methodology. It was not the same old kind of classifier brought out to play as an object detector. Furthermore, the fact that the YOLO model, indicated by the name, detected the object perfectly got many new deep-learning programmers into the field. The initial version was released in 2016, and a newer version known as YOLOv9 came out in 2024. It is faster and more accurate than other traditional object detectors; it is trained on the COCO dataset and has single-shot detection, which makes it faster at detecting objects [16].

A. License plate extraction

YOLOv8 is an open-source deep-learning framework composed of yolov3, yolov4, and yolov5 for real-time object detection in computer vision systems. Yolov8, featuring efficient architecture with the current technology, has brought about disruptive advancement in object detection by allowing accurate and fast detection of objects in real-life situations. Yolov8n is the most minor and rapid, whereas yolov8x is the slowest among the other yolov8 models. We have used the yolov8n model to get the quickest and most accurate results. With the help of yolov8n, we trained the license plate datasets of cars and extracted the number plate from each dataset image to train them [17] [18].

The principal process is training and testing, as positioned in Figure 1. While in the traditional car mode, trainees are being taught relying on 2D images acquired from the video are used for training. Moreover, pretty much each character possessed a sense of being trapped in certain bounds. The figure of the letters is eight hundred one characters. The labels are alphabetical, starting with A, and Numerically with 0. The number 36 corresponds to two times the alphabet's twelve-digit (s)/characters and five-digit (s).

Authorized licensed use limited to: Polytechnic University of Bucharest. Downloaded on March 17,2025 at 18:24:37 UTC from IEEE Xplore. Restrictions apply.



o **Figure 1.** Proposed Model

B. Data design

Recognizing objects in digital images is a process built upon using training data for weights that are then applied to identifying the objects. Besides hour training data, data should also be used to evaluate the algorithm's effectiveness. Three hundred data points were checked using the model collected for this purpose. The dataset comprises number plate images extracted and cropped from the video extraction. Then, it will be divided into a ratio of 70:10:20; there is 70% code for the training process, which is 210 data, 10% for the testing process with 30 data, and 20% for the validation process with 60 data. An approaching vehicle video is taken physically using a mobile phone. The video used in our manuscript was taken in good sunlight, as shown in Figure 2. After obtaining the video, we cropped the license plate and separated unnecessary background for efficient object detection.

Figure 2. Detecting number plate

A total of 30 data sets are used for the testing process. Color intensity is our dataset's main problem, and we must evaluate it. Distraction is given on several levels for the addition or reduction in the intensity of the color of the image. This process is done as a precaution if the data that is being captured is either too dark or too bright. An example of this is depicted in Figure 3.

Figure 3 Managing color intensity

Test data, which is 30 image data in our dataset, remains undisturbed. Preprocessing is initially applied to the test data, which is then utilized to create the model derived from the training results. This is done to assess how well the model detects patterns in the training data. Preprocessing is initially applied to the test data, which is then utilized to create the model derived from the training results. This is done to assess how well the model detects patterns in the training data.

C. Data preprocessing

The data we will obtain after cropping and extracting a license plate from the video does not have good quality, for instance, if the image is very small or blurry. Therefore, data preprocessing will be necessary to improve the image quality. It includes resizing the image, sharpening the image, reducing color intensity, and removing. Hence, after installing the necessary packages, convert the image to a grayscale, remove noise by applying a Gaussian blur, and finally detect the edges using the canny edge detector. Figure 4 will be the output image.

Figure 4. Applying a canny edge detector

Authorized licensed use limited to: Polytechnic University of Bucharest. Downloaded on March 17,2025 at 18:24:37 UTC from IEEE Xplore. Restrictions apply.

D. Contours

Find contours in the photo, extract the contour on the photograph, and extract it. After finding the contours, edged photographs vary from largest to smallest and hold the five biggest ones. Then, go through them individually; locating a contour with four factors will suggest the license plate is discovered.

E. Data annotation

In this model, data annotation is done utilizing roboflow, a product offered by Ultralytics, Yolo's mother company. Visit the labeling interface by selecting a picture via an Assign or Dataset page on the Roboflow dashboard. The toolbar is available on the right side of the annotation panel and has a lot of tools for annotating images.

F. EasyOCR

EasyOcr is a free library written in Python and allows reading text from images (one of the most common methods of recognizing characters). Now, one can perform OCR by feeding the license plate number to the package called EasyOCR. The first operation is the box found around the text; the next is text detection, and the third is to tell the confidence of output. Then, the contour of the license plate is drawn, as well as the text with its probabilities of detection. In Figure 5, the number plate (MH 49 CD 5452) is detected with a probability (80.13%).

Figure 5. EasyOcr to display the text and probability

Prior to reading the class label, it is essential to gather the bounding box that encompasses the detected character's class label and confidence value. This allows for the interpretation of the class label based on the coordinates of the bounding box, starting from the top left and moving towards the bottom right. The model's correctness will be assessed by comparing the threshold and confidence values. The findings of this study are now presented as character labels. Figure 6 displays the flow diagram of the entire operation.

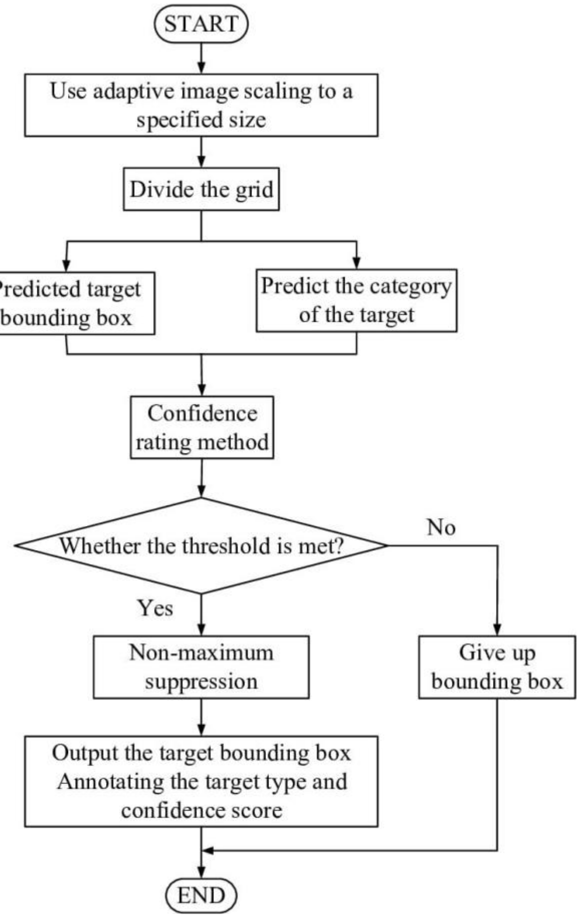


Figure 6. Flow chart of the complete process

IV. Results and Discussions

CNN involves multiple stages (e.g., region, proposal, classification), whereas YOLO performs object detection in a single stage. In YOLO, once image annotation is done, a test train folder gets created with the help of uuid (unique identifiers), as it is a trained model. In CNN, after collecting and cleaning the dataset, we have to write a separate script to split the dataset into tests and training. Unlike the two-step object detection system in YOLOv8n, the single-shot detector is a real-time object detection system. It goes one step ahead by making the entire image scanning process in a single forward pass and hence is much faster while predicting bounding boxes and classes for a given image than other methods. Images are preprocessed to improve accuracy and speed, as the methodology suggests. After preprocessing, a high rise in accuracy was observed. Therefore, two tables are created based on the accuracy of the recognized number plate and the accuracy of the characters that have been identified, depicted in Figure 7.

Authorized licensed use limited to: Polytechnic University of Bucharest. Downloaded on March 17,2025 at 18:24:37 UTC from IEEE Xplore. Restrictions apply.

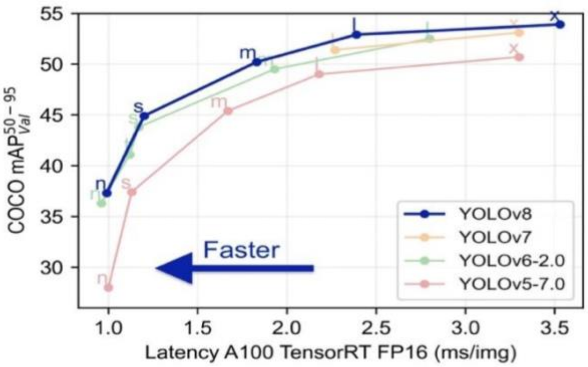


Figure 7. Speed and accuracy of different Yolo models

Table 1 shows the accuracy obtained without the preprocessing of images, and Table 2 shows the accuracy obtained by the preprocessing of images.

TABLE 1. Accuracy of dataset without preprocessing

Without pre-processing		Accuracy	
		Character recognition (%)	Number plate recognition
Without distraction		97.1	80
Brightness	+25	96	72
	+50	94.3	62
	+75	94.2	65
	-25	96.2	78
	-50	90.6	47
	-75	85.21	36

TABLE 2. Accuracy of dataset with preprocessing

With pre-processing		Accuracy	
		Character recognition (%)	Number plate recognition
Without distraction		98.2	88
Brightness	+25	97.3	86
	+50	95.7	82
	+75	94.1	69
	-25	98.1	88
	-50	96.8	83
	-75	97	83

V. CONCLUSION

In conclusion, the proposed system based on the YOLOv8n, which allows the real-time detection of vehicle number plates, is an essential advancement toward high road security and the implementation of traffic laws. Utilizing the pre-trained YOLOv8n model, which is fine-tuned for the custom dataset, produces accurate and quick detection results. Being the implementation object that can be adapted for traffic monitoring and surveillance systems to enhance traffic safety is an excellent feature of the suggested system. The data collected confirms that this model beats the existing state-of-the-art methods for plate recognition. The described process is a fruitful option for object detection in real-time traffic-related applications.

REFERENCES

[1] P. William, A. Shrivastava, N. S. Karpagam, T. A. Mohanaprakash, K. Tongkachok, and K. Kumar, "Crime Analysis Using Computer Vision Approach with Machine Learning," in Lecture notes in networks and systems, 2023, pp. 297–315. doi: 10.1007/978-981-19-7982-8_25.

[2] P. R. K. Varma, G. Srikanth, H. K. B, and P. Svsrk, "A Novel Method for Indian Vehicle Registration Number Plate Detection and Recognition using Image Processing Techniques," *Procedia Computer Science*, vol. 167, pp. 2623–2633, Jan. 2020, doi: 10.1016/j.procs.2020.03.324.

[3] H. Karwal and A. Girdhar, "Vehicle Number Plate Detection System for Indian vehicles," 2015 IEEE International Conference on Computational Intelligence & Communication Technology, Feb. 2015, doi: 10.1109/cict.2015.13.

[4] K. Sathya, S. Vasuhi, and V. Vaidehi, "Perspective Vehicle License Plate Transformation using Deep Neural Network on Genesis of CPNet," Procedia Computer Science, vol. 171, pp. 1858–1867, Jan. 2020, doi: 10.1016/j.procs.2020.04.199.

[5] Jagtap, S. (2015). Analysis of Feature Extraction Techniques for Vehicle Number Plate Detection. International Journal of Computer Science and Information Technologies, 6(6), 5342-5346.

[6] Thooyamani, K. P., Khanaa, V., & Udayakumar, R. (2013). Application of pattern recognition for Farsi license plate recognition. Middle-East Journal of Scientific Research, 18(12), 1768-1774. Apa

[7] C.-T. Hsieh, Y.-S. Juan, and K.-M. Hung, "Multiple License Plate Detection for Complex Background," 19th International Conference on Advanced Information Networking and Applications (AINA'05) Volume 1 (AINA Papers), Apr. 2005, doi: 10.1109/aina.2005.257.

[8] P. Prabhakar, P. Anupama, and S. R. Resmi, "Automatic vehicle number plate detection and recognition," 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Jul. 2014, doi: 10.1109/iccicct.2014.6992954.

[9] A. O. Salau, T. K. Yesufu and B. S. Ogundare, "Vehicle plate number localization using a modified GrabCut algorithm", J. King Saud Univ.-Comput. Inf. Sci., vol. 33, no. 4, pp. 399-407, May 2021.

[10] Z. Mahmood, K. Khan, U. Khan, S. H. Adil, S. S. A. Ali and M. Shahzad, "Towards automatic license plate detection", Sensors, vol. 22, no. 3, pp. 1245, Feb. 2022.

[11] S. Sweta, "Automatic car license plate detection system", IOP Conf. Ser. Mater. Sci. Eng., vol. 1116, no. 1, 2021.

[12] A. Mensch, G. Varoquaux and B. Thirion, "Compressed online dictionary learning for fast resting-state fMRI decomposition", Proc. IEEE 13th Int. Symp. Biomed. Imag. (ISBI), pp. 1282-1285, Apr. 2016.

[13] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement", arXiv:1804.02767, 2018.

[14] Y. Zou, Y. Zhang, J. Yan, X. Jiang, T. Huang, H. Fan, et al., "License plate detection and recognition based on YOLOv3 and ILPRNET", Signal Image Video Process., vol. 16, no. 2, pp. 473-480, Mar. 2022.

[15] Y. Sun, Q. Peng and D. Zhang, "Light-YOLOv3: License plate detection in multi-vehicle scenario", IEICE Trans. Inf. Syst., vol. E104.D, no. 5, pp. 723-728, May 2021.

[16] B. Setiyono, D. A. Amini, and D. R. Sulistyaningrum, "Number plate recognition on vehicles using YOLO - Darknet," Journal of Physics: Conference Series, vol. 1821, no. 1, p. 012049, Mar. 2021, doi: 10.1088/1742-6596/1821/1/012049.

[17] Y. Sun, Q. Peng and D. Zhang, "Light-YOLOv3: License plate detection in multi-vehicle scenario", IEICE Trans. Inf. Syst., vol. E104.D, no. 5, pp. 723-728, May 2021.

[18] S. S. Tabrizi and N. Cavus, "A hybrid KNN-SVM model for Iranian license plate recognition," Proc. Comput. Sci., vol. 102, pp. 588-594, Jan. 2016.

Authorized licensed use limited to: Polytechnic University of Bucharest. Downloaded on March 17,2025 at 18:24:37 UTC from IEEE Xplore. Restrictions apply.