Journal of Information Systems and Informatics
Vol. 6, No. 3, September 2024 e-ISSN: 2656-4882 p-ISSN: 2656-5935
DOI: 10.51519/journalisi.v6i3.848 Published By DRPM-UBD

Enhancing Automated Vehicle License Plate Recognition with YOLOv8 and EasyOCR

Nurul Salsabila[1], Sriani[2]

[1]Computer Science, State Islamic University of North Sumatra, Medan, Indonesia
[2]Faculty of Science and Technology, State Islamic University of North Sumatra, Medan, Indonesia
Email: [1]nurulsalsabila528@gmail.com, [2]sriani@uinsu.ac.id

Abstract

This research focuses on the development of an automatic system for vehicle license plate recognition using YOLOv8, EasyOCR, and CNN methods for object classification. The main issue raised is the need for an accurate and efficient system for recognizing vehicle license plates in real-time in dynamic environments, especially in urban areas with high traffic levels. The method used in this study involves resizing the input image to 416x416 pixels to standardize the data, analyzing the YOLO architecture that divides the image into a 7x7 grid, and using the Convolutional Neural Network (CNN) algorithm for feature extraction and object classification. Object detection uses the YOLOv8 method which is tasked with recognizing license plates using a previously trained YOLO (pretrained model) model then implemented and tested using video with 4k quality to ensure its effectiveness in detecting vehicle license plate objects, followed by the Optical Character Recognition (OCR) process with the EasyOCR method to read text on license plates and tested to ensure its effectiveness in reading characters on license plates vehicle number. The purpose of this research is to develop a system that can improve accuracy and efficiency in vehicle license plate recognition. The results show that the accuracy, precision, recall and F1-Score for object detection reach 100% and the average percentage of detected text conformity is 74.66%, which shows that this system is reliable in real applications and contributes to the development of automatic license plate recognition technology.

Keywords: YOLOv8, EasyOCR, Vehicle License Plate Recognition, Convolutional Neural Network, Object Detection

1. INTRODUCTION

In an era filled with technological developments, the mobility of motor vehicles is becoming more and more diverse. Vehicle license plate recognition is an important aspect of traffic surveillance and safety, especially in congested urban environments. Vehicle identification through license plates allows for effective traffic monitoring, tracking of vehicles involved in criminal activities, and better law enforcement. Therefore, the development of an automatic vehicle license plate

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

recognition system is important as a solution to improve efficiency and accuracy in the vehicle identification process.

License plate recognition involves the process of reading characters from the license plate image taken by the system automatically. This technology has been an active subject of research in the field of computer vision and artificial intelligence. According to [1] Vehicle license plate recognition is useful for recognizing the identity of the vehicle. Vehicle plate detection is a technology to identify vehicle license plates with license plate data that has been obtained previously so that it can find data that will later be processed in the database [2]. Through this automation approach, the process of recording on vehicle license plates can be carried out more accurately and quickly. According to [3] The process of vehicle plate recognition as much as possible must be able to be implemented in complex image conditions with various angles of image capture.

In recent years, there have been recent developments in vehicle license plate recognition technology, especially in the use of more sophisticated and efficient deep learning models. One of the most well-known models is YOLO (You Only Look Once), which has undergone several iterations of performance improvements, including YOLOv4 and YOLOv5. YOLOv8 is the latest iteration of the YOLO model that promises a better level of accuracy and speed in the detection of objects, including vehicle license plates. Meanwhile, EasyOCR is an easy-to-use and reliable optical character recognition library. The combination of these technologies can result in fast, accurate, and reliable systems, even in diverse environmental conditions [4]. Therefore, the researcher is interested in exploring and implementing an automatic vehicle license plate recognition system using YOLOv8 and EasyOCR as part of the research to see how accurate YOLOv8 and EasyOCR are in recognizing an object and number character on a vehicle plate.

Vehicle License Plate Recognition Technique Using the YOLOv5 algorithm has also been applied to previous research, namely research by 3. [5] In this study, it is proven that the accuracy of vehicle license plate detection with YOLOv5 is 100%, the result of the accuracy of letter and number recognition on vehicle license plates is 95.83%. The accuracy of vehicle license plate recognition on opening and closing door bars with testing from 5 classes of residential license plates is 100% and the average computing time required to run the system is 0.287344 seconds.
[6] His research showed an mAP score of 84% and an F1-Score of 80%. In the test based on the distance between the object and the camera, the farthest object detection can be achieved as far as 3 meters with a low confidence value of below 60% but at a distance of 2 meters the confidence value can reach above 80%.

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

Based on previous research, this study was made with differences in terms of the algorithm to be used and the library that will be used in recognizing characters on vehicle plates, where the algorithm used from the above research is YOLOv5 and the Tesseract OCR library while in this study using the YOLOv8 algorithm and using the EasyOCR library, then it will be analyzed how the level of accuracy obtained by using the YOLOv8 algorithm with EasyOCR will be analyzed. Previous license plate recognition methods often faced problems such as inaccuracy in dynamic environmental conditions, diverse camera angles, and poor lighting. YOLOv8 [7], the latest version of the YOLO model, improves the speed and accuracy of object detection. Nonetheless, EasyOCR offers reliable character recognition capabilities in difficult lighting conditions. The goal of this research is to find a solution for automatic license plate recognition that can be used to monitor traffic in real-time.

2. METHODS

This study uses a quantitative research methodology. The following is a diagram of the stages of the research as shown in Figure 1.

Figure 1. Research Stages

Figure 1 shows the stages of research carried out from planning to the testing stage. The research uses the YOLO (You Only Look Once) algorithm which is one of the model variants of the Convolutional Neural Network (CNN) method. A convolutional neural network (CNN) is an artificial neural network (JST) in which neurons in its layers are divided into three dimensions [8].

YOLO is a new approach in object detection systems designed for real-time data processing. This method uses a single neural network to predict bounding boxes and class probabilities directly in images/frames in a single process [4]. YOLOv8 (You Only Look Once version 8) is an object detection model popular in the field of image processing and computer vision. This model is a development of a series of previous YOLO models. YOLOv8 has the advantage of real-time object detection with a high level of accuracy [9].

The EasyOCR model was chosen as a tool to read labels on cardboard [10]. According to [11]. To calculate the average percentage of character recognition accuracy from two-character recognition results in a video. According to [12] Confusion matrix is a tool that is often used in data mining to measure the level of accuracy.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

2.1. Planning

This study applies YOLOv8 algorithm to detect vehicle plates and EasyOCR in text recognition on vehicle plates [13]. There are several steps that will be carried out in this study, including data collection, data analysis, system design, and detection testing.

2.2. Data Collection

2.2.1. Pretrained Model

There are two YOLOv8 models initialized. The first yolov8n.pt to detect common objects (in this case, vehicles) and the second license_plate_detector.pt is more devoted to detecting vehicle license plates. Pretrained models or pre-trained models, namely yolov8n, which can be obtained freely on the ultralytics website and license_plate_detector.pt which can be obtained on the recognition-rxg4e/ https://universe.roboflow.com/roboflow-universe-projects/license-plate. This trained model is the basis for researchers to test vehicle recognition models to achieve the research goals that have been set.

2.2.2. Library Research

Library research is the process of searching for information carried out in libraries or through online information sources such as academic databases, scientific journals, and digital libraries [14]. In this study, literature studies were carried out by summarizing the contents of accredited journals, books in the library, and theses and previous research.

2.3. Data Analysis

The data analysis process, at this stage, the steps carried out in detecting objects will be explained which can be seen in Figure 2.

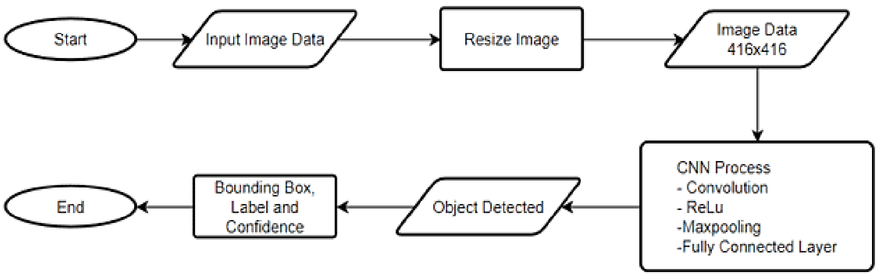


Figure 2. Yolo Algorithm Flowchart

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

Image data can be input in the form of both photos and videos, with videos being either real-time or recorded. The YOLO (You Only Look Once) process begins by resizing the input image data to 416 x 416 pixels. This resized image then undergoes feature extraction and object classification through a Convolutional Neural Network (CNN).

The process involves several steps: First, convolution operations are applied to the image using a 3x3 kernel to extract features. Following each convolution operation, the ReLU (Rectified Linear Unit) function is used to introduce non-linearity. Next, max pooling is performed as a downsampling process, which identifies the maximum value from the pixel values obtained after the ReLU operation. Finally, the Fully Connected Layer functions similarly to a standard neural network by generating class scores from the activations, which are then used for classification.

2.4 System Planning

Implementing the YOLOv8 algorithm for vehicle license plate recognition is one of the goals of the design of this research system. This license plate detection system starts by receiving a video of the vehicle and processing it using the YOLOv8 model to detect the vehicle and its license plate. Once identified, the license plate is labeled on the appropriate vehicle. The system then identifies the car. If that's true, thresholding is applied, and the video frame data is converted to grayscale format. Once the detection results are saved in a CSV file, the video is re-analyzed by reading the CSV file and selecting the license plate with the highest score. The system draws a bounding box, labels the license plate, and saves the result in a video. The labeled video output is generated as the result of the vehicle license plate detection process.

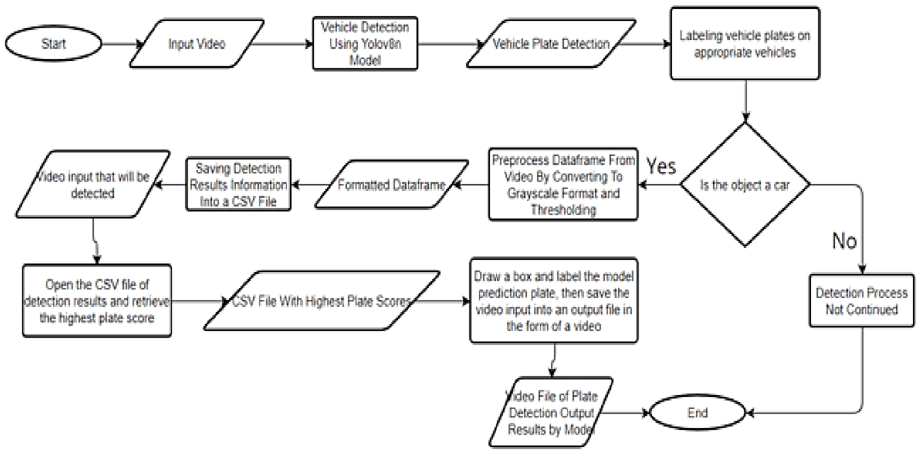


Figure 3. Flowchart of vehicle Plate Detection system

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

2.5 Testing

The test was carried out by taking 10 video data taken with 4k resolution at 30 fps and 4k at 60 fps using the iPhone 10 XR series mobile phone camera. Next, it will be tested for vehicle plate object detection using YOLO and text extraction using EasyOCR. This test analyzes the performance of the system under various daylight conditions. The results will be compared to the expected accuracy standards to assess the effectiveness of the system.

3. RESULTS AND DISCUSSION

The data analysis process involves several important stages: resizing the input image to 416x416 pixels to standardize the data, analyzing the YOLO architecture that divides the image into a 7x7 grid, and using the Convolutional Neural Network (CNN) method for feature extraction and object classification. The pre-trained YOLO model is then implemented and tested to ensure its effectiveness in detecting vehicle license plate objects. The Detection Process Analysis will further explain the process of detecting objects that go through various stages so as to get the final result of a detected object. The following are the stages of detection using the YOLO Algorithm.

3.1 Resize Image

The first step before processing the image data is to adjust the input data to the YOLO architecture configuration, namely by resizing the input data. Resize this is also important to standardize the size of various input data that has a variation in image size.

Figure 4. Original Image Image

Figure 4 shows the original image with a resolution of 1920 x 1080 pixels. Before being processed by the network, the image will be ressized to 416 x 416 pixels, as seen in Figure 5.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Figure 5. Image Image 416x416

The input data is in the form of (416, 416, 3) or an image measuring 416 x 416 with 3 channels, after going through the resizing process, will be divided into 7 x 7 squares as shown in Figure 4.3. These squares are called grid cells. Each box is responsible for predicting whether there is an object in it or not. If there is, the box is given a value of 1, and if not, it is given a value of 0. A box with a value of 1 will result in a Bounding Box. Each cell consists of 5 bounding boxes with 7 components in each box (bx, by, bw, bh, confidence, pc0, pc1). Table 4.1 will show an illustration of each grid cell on the output vector.

Figure 6. Image Imaging With 7x7 Grid

Table 1. Illustration of the contents of each grid cell

	Bx	by	Bw	brassiere	confidence	PC0	PC1
Bbox1							
Bbox2							
Bbox3							
Bbox4							
Bbox5							

3.2 YOLO Architecture Analysis

YOLO uses a reduction factor of 32 to downsample the input image. Images with a size of (416, 416, 3) or 416 x 416 pixels with 3 channels, will experience a decrease

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

in resolution during the convolutional network process resulting in an output with a resolution of 13 x 13 (416 divided by 32). This output will form a vector with size (13, 13, 35), where 13 x 13 is the final grid, and 35 is derived from the formula Bx(5+C). Here, B is the number of bounding boxes, which is 5, and C is the number of classes, which is 2 (cars and plates). The YOLO architecture used in this study is shown in more detail in Table 2. This architecture shows the summary during training according to the configuration found in Appendix A.1, with F as the filter kernel size, P as the padding, and S as the stride.

Table 2. YOLO Architecture

	Input	Operation	F	P	S
1	(416, 416, 3)	Conv with batch norm and leaky ReLU	(3,3)	1	1
2	(416, 416, 16)	Max Pooling	(2,2)	0	2
3	(208, 208, 16)	Conv with batch norm and leaky ReLU	(3,3)	1	1
4	(208, 208, 32)	Max Pooling	(2,2)	0	2
5	(104, 104, 32)	Conv with batch norm and leaky ReLU	(3,3)	1	1
6	(104, 104, 64)	Max Pooling	(2,2)	0	2
7	(52, 52, 64)	Conv with batch norm and leaky ReLU	(3,3)	1	1
8	(52, 52, 128)	Max Pooling	(2,2)	0	2
9	(26, 26, 128)	Conv with batch norm and leaky ReLU	(3,3)	1	1
10	(26, 26, 256)	Max Pooling	(2,2)	1	1
11	(13, 13, 256)	Conv with batch norm and leaky ReLU	(3,3)	1	1
12	(13, 13, 512)	Max Pooling	(2,2)	0	1
13	(13, 13, 512)	Conv with batch norm and leaky ReLU	(3,3)	1	1
14	(13, 13, 1024)	Conv with batch norm and leaky ReLU	(3,3)	1	1
15	(13, 13, 1024)	Conv with batch norm and leaky ReLU	(3,3)	0	1

In the first layer, a convolution operation is carried out with a 3x3 filter kernel, padding 1, and stride 1 on a 416x416 input. On the third layer onwards, the output is displayed according to Table 1 above. In the last layer, there is a difference, where the previous one is through convolution with ReLU activation, while the last layer uses convolution with linear activation. This last layer is in charge of predicting class probabilities and bounding boxes.

The class probability is obtained from the value of the confidence box score, which is a component of the bounding box as shown in Table 2. The confidence box will only have a value if the bounding box detects the presence of an object

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

in it. The final value is related to the IoU (Intersection over Union) of the bounding box. IoU is the comparison between the size of the bounding box and the ground truth obtained during the training period for each class, as shown in equation (3). For manual calculations, we will next try it in Figure 4.3, which has been resized and divided into a grid of 7 x 7 cells. It is known that the coordinate points (x, y) in the form of RGB pixels that indicate the existence of vehicle license plate objects are illustrated as follows:

RGB point (x,y) = (153, 100)
Width (w) = (51)
Height (h) = (50)

A piece of the image showing the coordinate points (153,100) can be seen in Figure 7.

Figure 7. Plate Object

The bounding box search process from Figure 8 which has been divided into a grid of 7 x 7 sections is carried out. Figure 8 is an illustrative example of the bounding box search process.

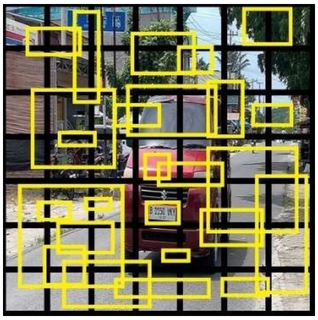


Figure 8. Bounding Box Search Illustration

In Figure 8, the bounding box search process shows that each grid cell is responsible for searching using a different-sized anchor box. The rule in the anchor box search is that if the anchor box is outside the boundary, then it doesn't count.

RGB point (x,y) = (152, 100)
Width (w) = (51)
Height (h) = (51)

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

The result of the image cut with the bounding box at the coordinates (152, 100) can be seen in Figure 9.

Figure 9. Objects Captured by Bounding Box

The calculated bounding box is the yellow box in Figure 9, and is then used for the IOU calculation, as shown in Figure 10.

Figure 10. IoU Predictions

Since the value obtained is greater than 0.2, the Bounding box data can be used with coordinate points:

RGB point (x,y) = (152,100)
Width (w) = (51)
Height (h) = (51)

The data Bounding Box obtained will continue to be used until it reaches the Fully Connected Layer network. In addition, a value of 0.48 will be used as a box confidence score value according to the equation, where a value of 0.48 is also called confidence in the bounding box. As mentioned by Joseph Redmon in his paper, the final value of the prediction (class confidence score) is the result of the multiplication of the box confidence score with the conditional probability of the class. The conditional probability values for each class (pc0, pc1) indicate the

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

likelihood that the objects in the bounding box belong to the plate class. This value ranges from 0 to 1, with 1 indicating that the box contains an object from that class, and 0 indicating the absence of an object from that class in the box. In the bounding box example, the conditional probability for the class "plate" is 1, so pc0 = 1. So, the class confidence score will be calculated as follows:

Pr(ClassPlat) = Pc0 * box confidence score = 1 * 0.48 = 0.48

Figure 11. Final Illustration Results of Detected Objects

3.3 CNN method process

To apply the CNN method to the YOLOv8 algorithm with an 8x8 input image, we will next discuss a step-by-step manual calculation that includes convolution, ReLU, and Max Pooling operations. Here is the sequence of the process:

a) Convolution
Convolution is the initial stage of image extraction performed on the YOLO network after the process of resize and IOU prediction search. After obtaining the results from the convolution, the next step in processing on Convolutional Neural Networks (CNNs) is to apply the ReLU (Rectified Linear Unit) activation function and then perform max pooling to reduce the dimensions of the resulting feature map.

b) Rectified Linear Unit (ReLu)
The ReLU (Rectified Linear Unit) function is an activation function that replaces all negative values in the matrix with zero values, while the positive values remain unchanged.

c) Max Pooling
Max pooling is a technique to reduce the dimensional size of a feature map by maintaining the maximum value in each small section (generally 2x2) of the matrix. Max pooling is done by taking the maximum value of each 2x2 sub-matrix.

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

d) Fully Connected Layer
First, it is necessary to flatten the 2D feature map into a 1D vector to be input into the fully connected layer. This vector will have a length of 36 (because 6x6 = 36 elements).

3.4 Implementation of the Pretrained Model

Object detection using YOLOv8 (You Only Look Once version 8) involves several stages from data preparation to generating detection output.

3.4.1 System Testing

Tests on video data were carried out in daylight conditions using a mobile phone camera to measure the performance of the model that had been made. In this study, 10 video data were used to evaluate the model's performance against character recognition. The trial was carried out on 10 video data which will later be used as frames with the number of frames.

Table 2. Specification of Video Frame Output 1

Video	Number of Frames
Video 1	430 Frame

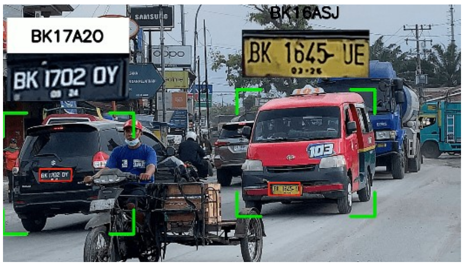


Figure 12. Video Output 1 Car 1 and 2

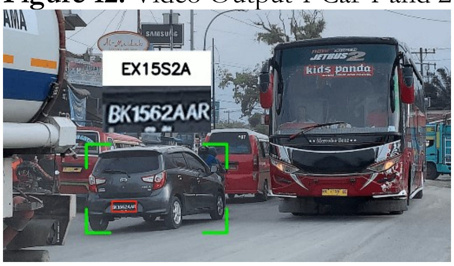
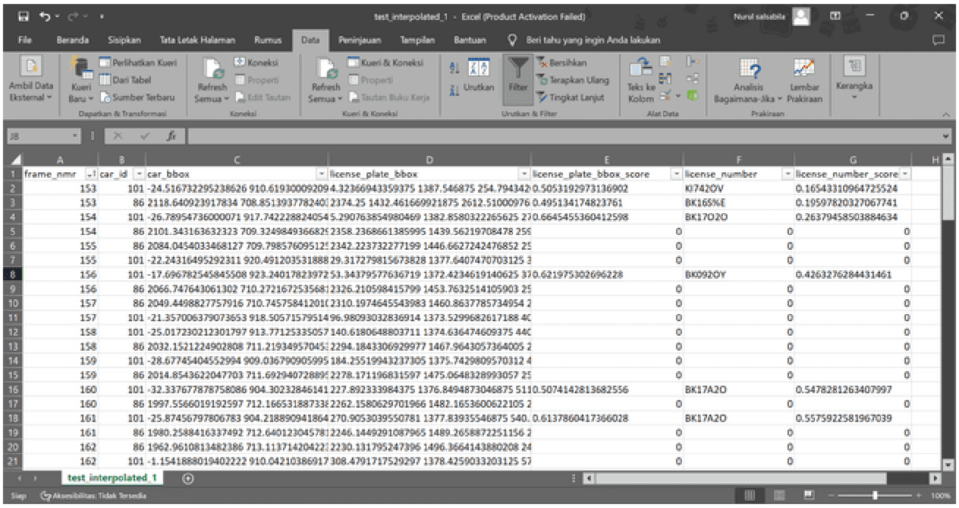


Figure 13. Video Output 1 Car 3

Character recognition accuracy is obtained on the test_interpolate_1.csv file as shown in Figure 14 and the system will automatically read the highest accuracy on

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

the test_interpolate_1.csv file and then it will be displayed on the video output 1. The overall test results for video output data 1 in daytime conditions can be seen in Table 3.



	frame_nmr	car_id	car_bbox	license_plate_bbox	license_plate_bbox_score	license_number	license_number_score
2	153	101	-24.516732295238626 910.61930009209	4.32366943359375 1387.546875 254.794342	0.5053192973136902	KI742OV	0.16543310964725524
3	153	86	2118.640923917834 708.851393778240	2374.25 1432.461669921875 2612.51000976	0.495134174823761	BK165%E	0.19597820327067741
4	154	101	-26.78954736000071 917.742228824054	5.290763854980469 1382.8580322265625 27	0.6645455360412598	BK17O2O	0.26379458503884634
5	154	86	2101.343163632323 709.32498493668	2358.2368661385995 1439.56219708478 25	0	0	0
6	155	86	2084.0454033468127 709.79857609512	2342.223732277199 1446.6627242476852 25	0	0	0
7	155	101	-22.24316495292311 920.491203531888	29.317279815673828 1377.6407470703125 3	0	0	0
8	156	101	-17.696782545845508 923.24017823972	53.34379577636719 1372.4234619140625 37	0.621975302696228	BK092OY	0.4263276284431461
9	156	86	2066.747643061302 710.27216725356	2326.210598415799 1453.7632514105903 25	0	0	0
10	157	86	2049.4498827757916 710.74575841201	2310.1974645543983 1460.8637785734954 2	0	0	0
11	157	101	-21.357006379073653 918.50571579514	96.98093032836914 1373.5299682617188 40	0	0	0
12	158	101	-25.017230212301797 913.77125335057	140.6180648803711 1374.636474609375 44	0	0	0
13	158	86	2032.1521224902808 711.21934957045	2294.1843306929977 1467.9643057364005 2	0	0	0
14	159	101	-28.67745404552994 909.036790905995	184.25519943237305 1375.7429809570312 4	0	0	0
15	159	86	2014.8543622047703 711.69294072889	2278.171196831597 1475.0648328993057 25	0	0	0
16	160	101	-32.337677878758086 904.30232846141	227.892333984375 1376.8494873046875 51	0.5074142813682556	BK17A2O	0.5478281263407997
17	160	86	1997.5566019192597 712.16653188733	2262.1580629701966 1482.1653600622105 2	0	0	0
18	161	101	-25.87456797806783 904.218890941864	270.9053039550781 1377.83935546875 540.	0.6137860417366028	BK17A2O	0.5575922581967039
19	161	86	1980.2588416337492 712.64012304578	2246.1449291087965 1489.2658872251156 2	0	0	0
20	162	86	1962.9610813482386 713.11371420422	2230.131795247396 1496.3664143880208 24	0	0	0
21	162	101	-1.1541888019402222 910.04210386917	308.4791717529297 1378.4259033203125 57	0	0	0

Figure 14. File Test_Interpolated_1.csv

Table 3. Output video Test Result 1

Video 1	Original plates	Plate detected	Character Recognition
Car 1	BK1645UE	BK16ASJ	70%
Car 2	BK1702OY	BK17A20	55%
Car 3	BK1562AAR	EX15S2A	31%

Furthermore, for the 2nd video, it will be used as frames with the number of frames per video. For output video 2, you can see in table 4 An example of the test results on output video 2 can be seen in Figure 15.

Table 4. Video Frame Output 2 Specifications

Video	Number of Frames
Video 2	278 Frame

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

Figure 15. Video Output 2 Car 1

Character recognition accuracy is obtained in the test_interpolate_2.csv file as shown in Figure 16 and the system will automatically read the highest accuracy in the test_interpolate_2.csv file and then it will be displayed in the video output 2. The overall test results for video output 2 data in daylight conditions can be seen in Table 5.

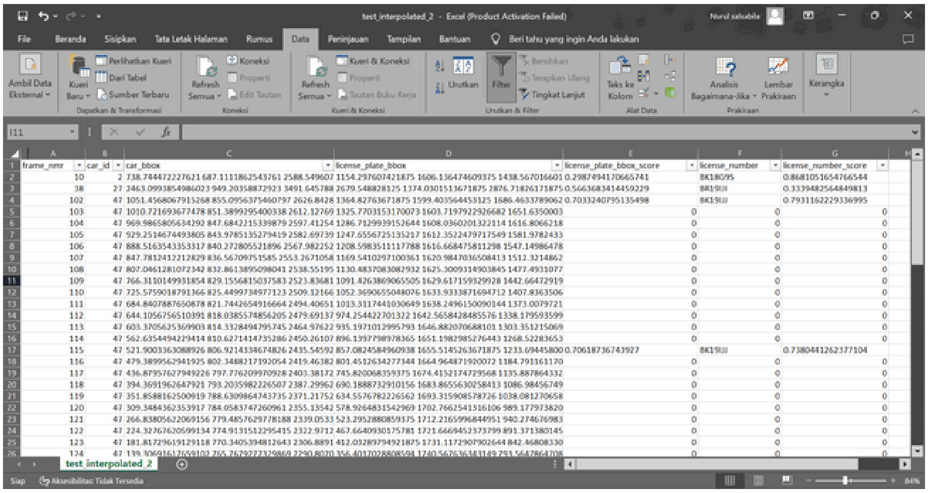


Figure 16. File Test_Interpolated_2.csv

Table 5. Output video Test Results 2

Video 2	Original plates	Plate detected	Character Recognition
Car 1	BK1869SA	BK18G9S	86%

Furthermore, for the 3rd video, it will be used as frames with the number of frames per video. For output video 3 can be seen in table 6, examples of test results on output video 3 can be seen in Figure 17.

Journal of Information Systems and Informatics
Vol. 6, No. 3, September 2024

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

Table 6. Specification of Video Frame Output 3

Video	Number of Frames
Video 3	Frame

Figure 17. Video Output 3 Car 1

The accuracy of character recognition is obtained on the test_interpolate_3.csv file as shown in Figure 18 and the system will automatically read the highest accuracy on the test_interpolate_3.csv file and then it will be displayed on the video output 3. The overall test results for video output 3 data in daylight conditions can be seen in Table 7.

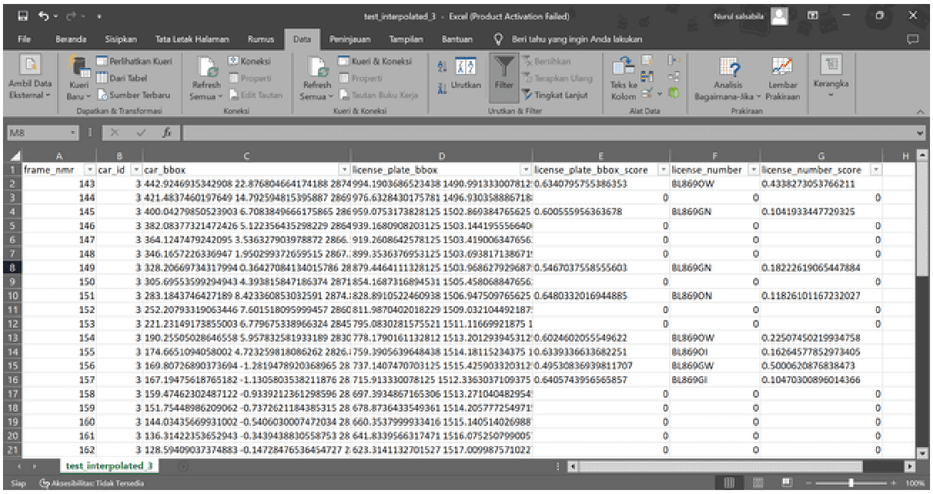


Figure 18. File Test_Interpolated_3.csv

Table 7. Output video Test Results 3

Video 2	Original plates	Plate detected	Character Recognition
Car 1	BK1150VKY	GI150VK	45%

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

This step is done until the output video is tested 10 or 10 times to detect objects so that it can be calculated with equations 5 and 6 to find the average percentage of character recognition accuracy for 1 video. The results can be seen in table 8 as below:

Table 8. Character Extraction Results Detected

Video data	Average percentage of detected characters
Video 1	52%
Video 2	85,33%
Video 3	45%
Video 4	64%
Video 5	71.75%
Video 6	82.6%
Video 7	89%
Video 8	60%
Video 9	98%
Video 10	99%
Average	**746.68%**

Based on Table 8, it can be seen that the minimum percentage of detection of each category is 90%, it can be seen that the average result of the percentage of detected text conformity is 74.66%. Before we get into the discussion of object calculation, it is important to understand some of the evaluation metrics that are often used in object detection, such as Accuracy, Precision, and Recall. 100% Precision means 0 False Positives (no wrong predictions), and 100% Recall means 0 False Negatives (all basic truth boxes are correctly predicted). Here are the general steps to calculate the above metrics from the results of the car license plate detection video.

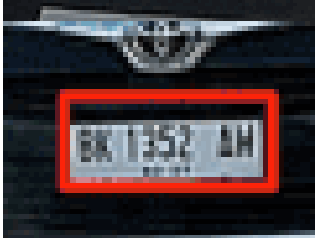
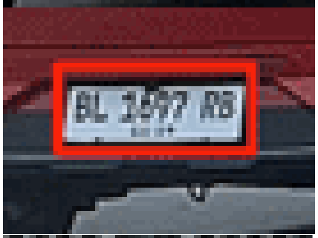
1. True Positives (TP): The correct number of detections (the license plate is detected correctly and is in the proper bounding box).
2. False Positives (FP): Number of false detections (the model detects something that is not a car plate as a car plate).
3. False Negatives (FN): The number of car plates that are not detected by the model.
4. True Negatives (TN): The number of cases in which the model correctly did not detect an object that did not exist (for example, an area without a car plate).

From the confusion matrix value obtained from the classification process which can be used to determine the value of precision, recall, and f1-score as well as to find out the level of accuracy that the system has been built. The results of 100% precision, 100% recall, 100% F-1 Score and 100% accuracy in object recognition were obtained, however, the accuracy of text character recognition only reached 74.66%. This is likely due to factors such as lighting conditions, camera viewing angles, and object movement. Compared to other methods that use the YOLOv5

Journal of Information Systems and Informatics
Vol. 6, No. 3, September 2024
p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

model and Tesseract OCR, this model is faster and more accurate in detecting vehicle license plates, although character recognition still needs further improvement. The implementation of this system is relevant for law enforcement, traffic monitoring, and toll automatic systems.

Table 9. Evaluation of Plate Objects

Detected	Confidence	Matches	Acumulative			
			TP	TN	FP	FN
	0.76	TP	1	0	0	0
	0.73	TP	2	0	0	0
	0.72	TP	3	0	0	0
	0.72	TP	4	0	0	0
	0.71	TP	5	0	0	0

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

Detected	Confidence	Matches	Acumulative			
			TP	TN	FP	FN
	0.69	TP	6	0	0	0
	0.69	TP	7	0	0	0

3.5 Discussion

The data analysis for vehicle license plate detection using the YOLO algorithm involved several key stages, each contributing to the system's overall performance in detecting and recognizing license plates from video footage. The process begins with resizing the input image to 416x416 pixels to standardize the input data for the YOLO architecture, which divides the image into a 7x7 grid for object detection. The Convolutional Neural Network (CNN) method is then employed for feature extraction and object classification. This architecture enables the pre-trained YOLO model to detect objects, such as vehicle license plates, effectively.

The first stage in the detection process involves resizing the input image to 416x416 pixels, ensuring that all input data conforms to a consistent size, which is critical for maintaining the accuracy of the YOLO model. The resized image is divided into a 7x7 grid of cells, where each cell is responsible for predicting the presence or absence of an object. If an object is detected, a bounding box is generated, and various parameters, such as location (bx, by), dimensions (bw, bh), confidence scores, and class probabilities, are calculated. This method allows the model to accurately localize objects within an image and classify them accordingly.

The YOLO architecture further refines the detection process by downsampling the input image using a reduction factor of 32. This results in an output resolution of 13x13, forming a final grid of 13x13 cells. Each cell predicts five bounding boxes with associated class probabilities, allowing for the detection of multiple objects, such as cars and license plates, within the same image. The final layer of

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

the YOLO model, which uses linear activation, predicts the class probabilities and bounding boxes, ensuring a comprehensive and accurate detection process. The predicted values are then compared to ground truth data using Intersection over Union (IoU) to assess the bounding box's accuracy in relation to the actual object location.

The implementation of the CNN method in the YOLOv8 algorithm involves several key steps, including convolution, the application of the Rectified Linear Unit (ReLU) activation function, and max pooling. These steps are essential for extracting relevant features from the input image and reducing the dimensionality of the data, thereby enhancing the detection accuracy. The final step involves flattening the 2D feature map into a 1D vector, which is then fed into the fully connected layer to perform the final classification and generate the detection output.

System testing on video data demonstrated that the YOLOv8 model could effectively detect vehicle license plates under daylight conditions with varying degrees of accuracy. Across multiple test videos, the model achieved an average character recognition accuracy of 74.66%. This accuracy, while lower than the object detection performance, indicates room for improvement in character recognition. Factors such as lighting conditions, camera angles, and object motion significantly affected character recognition accuracy. Despite these challenges, the YOLOv8 model showed a competitive advantage in speed and detection accuracy compared to other methods, such as those using the YOLOv5 model and Tesseract OCR.

The performance metrics for object detection, including precision, recall, F1-score, and accuracy, all achieved 100%, indicating that the model successfully detected all vehicle license plates without any false positives or false negatives. However, the character recognition component achieved only a 74.66% accuracy rate, primarily due to external factors affecting image quality. These results suggest that while the YOLOv8 model is highly effective in detecting vehicle license plates, further optimization is needed to improve character recognition accuracy. The high detection performance of the model demonstrates its suitability for applications in law enforcement, traffic monitoring, and toll automation systems.

The YOLOv8 model demonstrates a robust capacity for detecting vehicle license plates in video footage, with high precision and recall rates. However, the character recognition accuracy needs improvement to meet the high standards required for practical applications. Future work should focus on enhancing character recognition accuracy by addressing factors such as image quality, lighting conditions, and model training techniques. The implementation of this system shows promise for real-world applications, providing a valuable tool for law

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

enforcement, traffic monitoring, and other related fields. Further research could explore the integration of more advanced OCR techniques or hybrid models that combine YOLO with other character recognition technologies to achieve even greater accuracy.

4. CONCLUSION

Based on the results of the license plate detection accuracy research obtained from 10 videos taken by the author in analyzing the performance of the system, there are 24 vehicle plate objects detected by the vehicle license plate recognition system using the YOLOv8 pretrained model, showing excellent results with object detection accuracy using Precision, Recall, and F1-score reaching 100%, indicating that this system is very effective in recognizing and detecting vehicle license plates automatic. Character recognition from 10 videos taken by the author in analyzing the performance of the system, there are 24 vehicle plate objects with an average percentage of text conformity using the Easy OCR method detected from vehicle license plates reaching 74.66%. This shows that the developed system is capable of recognizing characters with a low error rate and this system is recommended to be applied in urban traffic surveillance. More research is needed to improve the accuracy of character recognition, especially in more diverse environmental conditions such as low lighting and extreme weather.

REFERENCES

[1] K. Rifki, J. Priambodho, and A. Musthofa, "Pengenalan Plat Nomor dan Wajah Pengendara Menggunakan Convolutional Neural Network dan Metode Absolute Difference pada Sistem Gerbang Otomatis," *J. Tek. ITS*, vol. 10, no. 2, 2021, doi: 10.12962/j23373539.v10i2.72508.
[2] M. Rosyadi, R. P. P, and F. T. Industri, "Mendeteksi Plat Nomor Kendaraan Berbasis Website," vol. 6, no. 2, pp. 936–944, 2022.
[3] M. Zakiyamani, T. I. Cahyani, D. Riana, S. Hardianti, and B. Naren, "Menggunakan Opencv Dan Deep Learning Berbasis Python Detection And Recognition Of Vehicle Number Character Plate Using Python-Based Opencv And Deep Learning, " *Universitas Nusa Mandiri*, vol. 5, pp. 56–64, 2022.
[4] T. T. H. Vu, D. L. Pham, and T. W. Chang, "A YOLO-based Real-time Packaging Defect Detection System," *Procedia Comput. Sci.*, vol. 217, no. 2022, pp. 886–894, 2022, doi: 10.1016/j.procs.2022.12.285.
[5] M. R. Rais, F. Utaminingrum, and H. Fitriyah, "Sistem Pengenalan Plat Nomor Kendaraan untuk Akses Perumahan menggunakan YOLOv5 dan Pytesseract berbasis Jetson Nano," vol. 7, no. 2, pp. 681–685, 2023.
[6] I. Maulana, N. Rahaningsih, and T. Suprapti, "Analisis Penggunaan Model Yolov8 (You Only Look Once) Terhadap Deteksi Citra Senjata

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Berbahaya," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 6, pp. 3621–3627, 2024, doi: 10.36040/jati.v7i6.8271.

[7] M. Safaldin, N. Zaghden, and M. Mejdoub, "An Improved YOLOv8 to Detect Moving Objects," *IEEE Access*, vol. 12, no. May, pp. 59782–59806, 2024, doi: 10.1109/ACCESS.2024.3393835.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once : Unified , Real-Time Object Detection".

[9] S. Tamang, B. Sen, A. Pradhan, K. Sharma, and V. K. Singh, "Exploring YOLOv8 Object Detection for Accurate Face Mask Classification," *Ijisae*, vol. 2023, no. 2, pp. 892–897, 2023, [Online]. Available: www.ijisae.org

[10] F. B. Stefanus Adhie Nugroho, Nur Kholis, Endryansyah, "Rancang Bangun Sistem Deteksi Label Kardus Berbasis Model Kecerdasan Buatan YOLO dan EasyOCR serta ESP32-CAM Rancang Bangun Sistem Deteksi Label Kardus Berbasis Model Kecerdasan Buatan YOLO dan EasyOCR serta ESP32-CAM Stefanus Adhie Nugroho Abstrak," *J. Tek. Elektro*, vol. 11, no. 2, pp. 190–200, 2022.

[11] A. M. Alqadri and F. Utaminingrum, "Pengenalan Papan Nama Ruangan untuk Kendali Kursi Roda Pintar menggunakan YOLOv7-Tiny dan EasyOCR berbasis TX2," vol. 7, no. 5, pp. 2227–2231, 2023.

[12] M. F. Rahman, D. Alamsah, M. I. Darmawidjadja, and I. Nurma, "Klasifikasi Untuk Diagnosa Diabetes Menggunakan Metode Bayesian Regularization Neural Network (RBNN)," *J. Inform.*, vol. 11, no. 1, p. 36, 2017, doi: 10.26555/jifo.v11i1.a5452.

[13] D. Nafis Alfarizi, R. Agung Pangestu, D. Aditya, M. Adi Setiawan, and P. Rosyani, "Penggunaan Metode YOLO Pada Deteksi Objek: Sebuah Tinjauan Literatur Sistematis," *J. Artif. Intel. dan Sist. Penunjang Keputusan*, vol. 1, no. 1, pp. 54–63, 2023.

[14] B. A. Habsy, N. Mufidha, C. Shelomita, I. Rahayu, and M. I. Muckorobin, "Filsafat Dasar dalam Konseling Psikoanalisis : Studi Literatur," *Indones. J. Educ. Couns.*, vol. 7, no. 2, pp. 189–199, 2023, doi: 10.30653/001.202372.266.

[15] A. I. Rizal and T. N. Suharsono, "Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Citra Jamur Berbasis Mobile," *J. Soc. Sci. Res.*, vol. 3, pp. 864–875, 2023.