Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th June 2025. Vol. 4. No. 3; e-ISSN: 2808-4519

Prediction Model Optimization on Odd-Even License Plates Using YoloV8 Algorithm

Denar Ahmaron Surya Gemilang[1]*, Rudi Kurniwan[2], Yudhistira Arie Wijaya[3], Rahmat Hidayat[4]

[1]Program Study Teknik Informatics, STMIK IKMI Cirebon
[2]Program Study Teknik Informatics, STMIK IKMI Cirebon
[3]Information Systems Study Program, STMIK IKMI Cirebon
[4]Program Study Teknik Informatics, STMIK IKMI Cirebon
denarahmaron1@gmail.com[1]*

Abstract

Traffic congestion in urban areas encourages the implementation of vehicle restriction policies based on license plate numbers, such as the odd-even system. Therefore, to support this policy, an accurate vehicle license plate detection system is needed and can work in real-time. The main challenge faced is how to develop an accurate and efficient detection model in recognizing license plates in various environmental conditions. The research method used is Knowledge Discovery in Databases (KDD) with five main stages, namely: data selection, preprocessing, transformation, data mining, and evaluation. This research method aims to develop and evaluate a vehicle license plate detection model based on the YOLOv8 algorithm, focusing on the classification of license plates into the "Odd" and "Even" categories. However, the dataset used was only obtained from the Roboflow platform and primary data in the parking environment, which was then processed through the cropping, resizing, and labeling stages using a format that was in accordance with YOLOv8's needs. The model was trained for 100 epoches with performance evaluation using precision, recall, F1-score, and Average Precision (mAP) metrics. The training results showed that the model achieved a precision of 0.879, a recall of 0.888, an mAP50 of 0.954, and an mAP50–95 of 0.830, with a fitness value of 0.843. In addition, the image resolution of 640x480 pixels results in the highest detection accuracy, which is 92% for odd plates and 85% for even plates. Tests were carried out on both images and videos, showing that the model was able to work in real-time with stable results. Based on these results, it can be concluded that YOLOv8 is effectively used to detect odd-even license plates with high accuracy. This research contributes to the development of intelligent systems based on computer vision to support efficient traffic management, especially in the implementation of odd-even policies in urban areas.

Keywords: *YOLOv8, License Plate Detection, Odd-Even, Deep Learning, Object Detection*

1. Introduction

Various efforts have been made to reduce congestion, one of which is through the implementation of an odd-even system that limits the number of vehicles that can operate on the road at a certain time [1]. This system is based on vehicle license plate numbers, where vehicles with odd license plate numbers can only operate on days with odd dates and vice versa for even license plate numbers can operate on days that are even dates. However, the implementation of the odd-even system often encounters obstacles, such as difficulties in verifying vehicle license plates in real-time. Artificial intelligence-based object detection technology, such as the YOLO (You Only Look Once) algorithm, offers an effective solution. YOLOv8, as a version of this algorithm, has a high ability to detect objects quickly and accurately, making it a great choice for the detection of odd-even plates on vehicle numbers.

Traffic congestion in urban areas encourages the implementation of vehicle restriction policies based on license plate numbers, such as the odd-even system. Therefore, to support this policy, an accurate vehicle license plate detection system is needed and can work in real-time. The main challenge faced is how to develop an accurate and efficient detection model in recognizing license plates in various environmental conditions. Why choose the YOLO algorithm? Because YOLO is an object detection algorithm created by Joseph Redmond in 2015, this YOLO (You Only Look Once) algorithm can detect in real time [2]. Evaluation metrics such as precision, recall, F1-score, and mean Average Precision (mAP) are also important to be used in measuring model performance, so that it can be known whether the YOLOv8 algorithm is optimal [3]. In addition, variations in license plate design, such as fonts, colors, and formats, add to the complexity of detection, requiring the refinement of datasets and data augmentation techniques [4].

Research by [5] discusses the character recognition system on Indonesian vehicle license plates, which provides an alternative to vehicle license plate character recognition using the Convolutional Neural Network (CNN) method. This method allows the recognition of objects in images in a human-like manner through computer learning using artificial neural networks. Meanwhile, according to [6] the YOLO approach in object recognition has significant differences from traditional methods because it combines the process of object detection and segmentation in a single step, thereby improving efficiency. A study by [7] shows that the accuracy of vehicle license plate detection using YOLOv5 reaches 100%, with the accuracy of letter and number recognition on vehicle license plates of 95.83%, which proves the potential of the YOLO method in achieving very high results in character recognition on license plates. This research continues previous work by applying the YOLOv8 model to optimize the detection of odd-even license plates.

This study uses an experimental approach to implement the YOLOv8 algorithm in detecting odd-even vehicle license plates on the Google Colab platform. The YOLOv8 algorithm was chosen in this study for its high speed and accuracy, which is suitable for detecting vehicle license plates in real-time in dynamic highway environments [8]. With features that support object recognition under various conditions, YOLOv8 can provide relevant detection results, especially to support odd-even policies. In addition, YOLO's advantages in video image processing ensure that this algorithm can be used effectively for CCTV-based traffic monitoring [9]. The process begins with the collection of a dataset consisting of images and videos of vehicle license plates in various lighting conditions, viewing angles, and resolutions. This dataset is then processed and annotated to ensure the data used is in accordance with the needs of odd-even detection. After that, the YOLOv8 model is implemented on Google Colab, where the algorithm is trained using the annotated dataset. Evaluation metrics such as precision, recall, F1-score, and mean Average Precision (mAP) are used to assess the model's performance in identifying vehicle license plates.

The results of this study have significant implications in the development of a support system for odd-even traffic restriction policies in cities with high levels of congestion. Using the Google Colab platform, the study offers a more economical cloud-based detection solution without requiring expensive hardware investments. In addition, the comparison between detection using images and video provides insight into the most effective input media in various monitoring situations. This technology allows for the application of both static cameras and real-time CCTV-based monitoring. The results of this study have the potential to support more efficient, reliable, and affordable traffic management, as well as help reduce congestion in congested traffic areas.

2. Literature Review

Various previous studies have developed vehicle license plate detection systems using a combination of object detection algorithms and Optical Character Recognition (OCR). Incorporation of YOLOv3 and Tesseract OCR to recognize license plates from images and videos [6]. Then, modify the Template Matching method in OCR to improve the accuracy of license plate detection [10], and develop an automated system using YOLOv3 and Tesseract. There are also those who use CNN to recognize the character of Indonesian vehicle license plates.

Some studies also emphasize the integration of the system with specific hardware, such as implementing YOLOv5 and Pytesseract on the Jetson Nano for residential security systems. Meanwhile, there is also the application of OCR and image segmentation in intelligent CCTV systems. Then develop a system for reading plate characters quickly with OCR. The use of YOLO is also developing in general traffic contexts, such as vehicle type classification using YOLOv7, as well as the detection of traffic violations with YOLOv3. This review shows that although various studies have utilized YOLO and OCR in the context of license plate detection, the focus on specific odd-even classifications is still limited, making it an important research gap in this study.

3. Research Methods

The research method used in this study is the Knowledge Discovery in Databases (KDD) method, which aims to test the performance of the YOLOv8 algorithm in detecting odd-even vehicle license plates. The following are the stages of the research method:

The research method is a stage in the Knowledge Discovery in Databases (KDD) process or often known as the data mining stage. The following are the stages of research using the Knowledge Discovery in Databases (KDD) method along with the activities and descriptions of the activities:

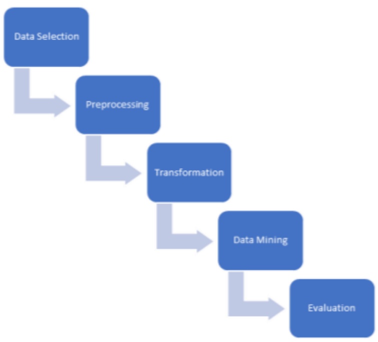


Fig. 1: Research methods

Table 1: Stages of Research Methods

No	Stages	Activity	Activity Description
1	Data Selection	Dataset Collection	The secondary data obtained from the selected Image Roboflow platform includes a variety of lighting conditions, viewing angles, and resolutions to create diverse datasets.
2	Preprocessing	Preprocessing	Crop with a scale of 1:1 then resize with a size of 640x640 pixels.
3	Transformation	Data Transformation	The selected images are processed through the labeling stage, which is adding bounding boxes and category information (odd/even) on each vehicle license plate.

No	Stages	Activity	Activity Description
4	Data Mining	Data Mining Process	Conduct YOLOv8 model training using processed data. to explore patterns or relationships contained in the data. In the epoch process it is used 100 times. The YOLOv8 algorithm is trained to detect vehicle license plates based on odd and even categories with defined evaluation parameters.
5	Evaluation	Model Evaluation	After training, a model evaluation was carried out to measure the extent to which YOLOv8 was able to correctly recognize vehicle license plates. Evaluations are conducted using metrics such as Precision, Recall, F1-Score, and Mean Average Precision (mAP).

Based on table 3.1 in the research method, in this study, the Knowledge Discovery in Database (KDD) method is used for the data analysis techniques. The following are the stages and processes in Knowledge Dissemination in Database:

1. Data Selection
   The first stage in this study is data selection, which aims to analyze and select relevant data from the Roboflow platform through https://universe.roboflow.com/i-gusti-fajar/perbandingan-plat link. The selection process involves filtering out the dataset, i.e. irrelevant images, such as license plate photos that are too small, blurry, or taken from too far, are removed from the dataset. Only license plate images that meet certain criteria, such as a large enough size and clear resolution, are passed on to the next stage. Because the YOLOv8 algorithm detects deep learning-based objects that rely heavily on the quality of data input.
2. Preprocessing
   The next stage is the preprocessing stage. At this stage, the image cropping and resizing stage is carried out for the purpose of YOLO input data. The cropping process is with 1:1 or square image dimensions, while the image resize is made to be 640x640 pixels. The YOLOv8 algorithm requires a fixed-size input image and a standard label format, i.e. a .txt format with bounding box coordinates. This stage aims to align the data so that it is readable by the YOLO model system. After this stage is completed, the next step is labeling or transformation of the image data to process the data so that it can be read by the YOLOv8 algorithm.
3. Transformation
   Pada tahap transformation data gambar akan diubah menjadi data yang sesuai untuk dilakukan pengolahan data yang dapat dibaca by the YOLOv8 algorithm to convert data into training data, data testing and data validation. At this stage, the image will be given odd label annotations for odd license plates, while even label annotations will be given for even license plates. Labels and annotations are essential because YOLOv8 works in a supervised learning format. After the labeling process is completed, the image data will be divided into 3 parts, namely 70% for train, 10% for test, and 20% for valid, this data sharing is also a mandatory stage of machine learning to prevent overfitting and maintain model generalization. After the data has been divided into 3 parts, the next step is the data training process on data mining.
4. Datamining
   In the datamining stage, this study conducted YOLOv8 model training using the processed dataset. During the training, the model will learn to recognize patterns and features associated with vehicle license plate objects. Then to achieve a good balance between accuracy and speed of training, 100 epochs were selected for the training data of the data testing.
5. Evaluation
   This stage is carried out a process of evaluation and interpretation of the rules that have been obtained in accordance with the goals that have been determined. At this stage, the model is applied to the test dataset to detect objects on the image of odd-even license plates that have never been seen before. Then use evaluation metrics such as Precision, Recall, and F1 Score to measure the model's performance. These metrics help identify the extent to which the model can correctly recognize objects and reduce detection errors.

4. Results and Discussion

The results of this study tested the effectiveness of the YOLOv8 Algorithm to detect vehicle license plates in the "Odd" or "Even" classes or labels. Then the results of the YOLOv8 model performance evaluation show that the model has a precision value of 0.879 or if it is percentaged to 87.9% and recall is 0.888 or if it is percentaged to 88.8%. This means indicating that the model can accurately identify vehicle license plate objects in the "Odd" or "Even" classes or labels contained in the dataset. In the mAP50 metric, the model showed very high performance with a value of 0.954 or approximately 95.4%, which reflects the model's accuracy in recognizing and localizing objects at the Intersection over Union (IoU) threshold of 0.5. Meanwhile, an mAP50–95 value of 0.830 or about 83% indicates that the model remains consistent and accurate in detecting objects at various levels of bounding box precision. The fitness value of 0.843 indicates that the model is in optimal condition, with a good balance of performance in terms of accuracy, efficiency, and stability during the detection process in detecting vehicle license plates, both in the "Odd" and "Even" classes.

Table 2: Metric Value

metrics/precision(B)	0.879
metrics/recall(B)	0.888
metrics/mAP50(B)	0.954
metrics/mAP50-95(B)	0.830
fitness	0.843

Based on the results of table 2, it can be concluded that Algortima YOLOv8 shows excellent performance in detecting and classifying odd and even vehicle license plates. The high mAP values, both at the low threshold (mAP50) and the wide range (mAP50–95), prove that this model can recognize odd or even vehicle license plate objects despite variations in position, size, or resolution. Data mining in the YOLOv8 model succeeded in detecting "Odd" and "Even" license plate objects, The modeling results were obtained based on data training conducted over 100 epochs. The following is the code for carrying out training and testing in Figure 2 for 100 epochs.

```
from datetime import datetime
start = datetime.now()
results = model.train(data='ganjil_genap.yaml', imgsz=640, epochs=100, batch=32, name="hasil")
end = datetime.now()
```

Fig 2: Code Runtime 100 Epoch

The code in Figure 2 is the training process of the YOLOv8 model will start using a dataset whose annotations have been defined in the "ganjil_genap.yaml" file. The processed dataset is in the form of a vehicle license plate image with a size of 640x640 pixels and a batch size of 32, after completing the training, the output results from precission, recall, and f1-score are obtained as follows:

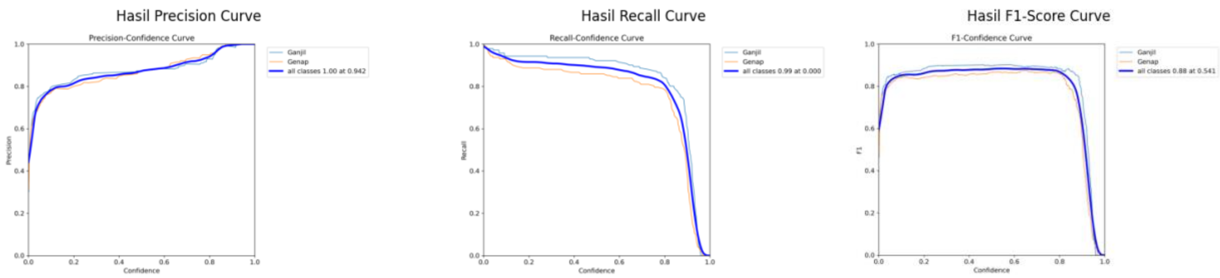


Fig 3: Precision Value, Recall Value and F1-Score Value

After completing the training on the results of the Precision-Confidence Curve and Recall-Confidence Curve graphs in figure 3, the Precision graph explains that the curves of all classes have high precision, reaching a value of 1.00 at a confidence of about 0.942, which indicates that the model has a maximum level of accuracy at the confidence. Meanwhile, the recall graph explains that all classes have a maximum value of 0.99 at 0.000 confidence and decrease as confidence increases.

The following F1-Score Confidence Curve shows the relationship between confidence level and F1-score, which is a metric that combines precision and recall into a single value to provide an idea of the balance between accuracy and completeness of detection. The F1-Score score itself for all classes reached a maximum value of 0.88 at a confidence of around 0.541. This is the point at which the model has an optimal balance between precision and recall. With confidence approaching a 1.0 value, F1-scores for all classes began to decline drastically. This shows that at very high confidence, both precision and recall become lower, which leads to a decrease in overall performance.

Overall, the model successfully learned effectively and showed a steady improvement in performance throughout the training process. The following is code to display the loss functions (box_loss, cls_loss, dfl_loss) and instances of each epoch, and help visualize the model's training progress.

```
# Create the full file path for 'results.csv' using the directory path and file name
results_csv_path = os.path.join('/content/ultralytics/runs/detect/hasil/results.csv')

# Load the CSV file from the constructed path into a pandas DataFrame
df = pd.read_csv(results_csv_path)

# Remove any leading whitespace from the column names
df.columns = df.columns.str.strip()

# Plot the learning curves for each loss
plot_learning_curve(df, 'train/box_loss', 'val/box_loss', 'Box Loss Learning Curve')
plot_learning_curve(df, 'train/cls_loss', 'val/cls_loss', 'Classification Loss Learning Curve')
plot_learning_curve(df, 'train/dfl_loss', 'val/dfl_loss', 'Distribution Focal Loss Learning Curve')
```

Fig 4: Results of Data Mining Evaluation

The code in Figure 4 will display the results of data mining evaluation from the 100 epochs training that has been carried out. The following are the results of the training code described in figure 4, which are as follows:

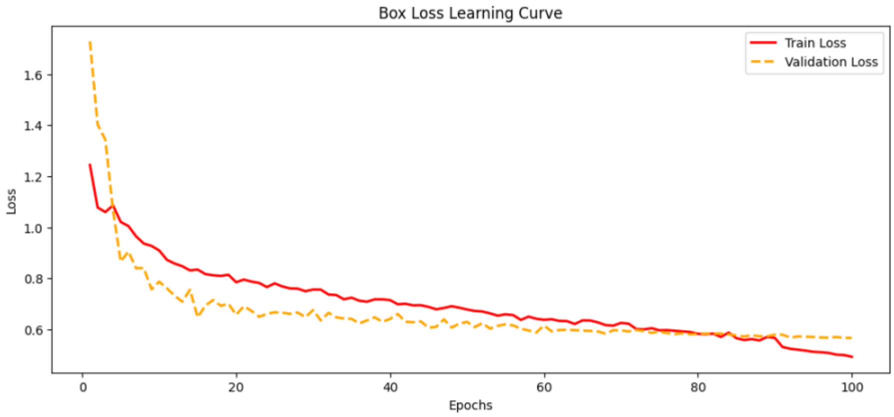


Fig 5: Box Loss Learning Curve

Figure 5 shows the results of the Box Loss Learning Curve of the YOLOv8 model for 100 epoches in the form of metric and loss graphs that indicate that the YOLOv8 model training process is running effectively. The values of train loss and validation loss decreased consistently and began to converge after about the 60th epoch, without showing significant symptoms of overfitting. This indicates that the model can learn the bounding box pattern optimally in detecting vehicle license plates. Then the Classification Loss Learning Curve is as follows:

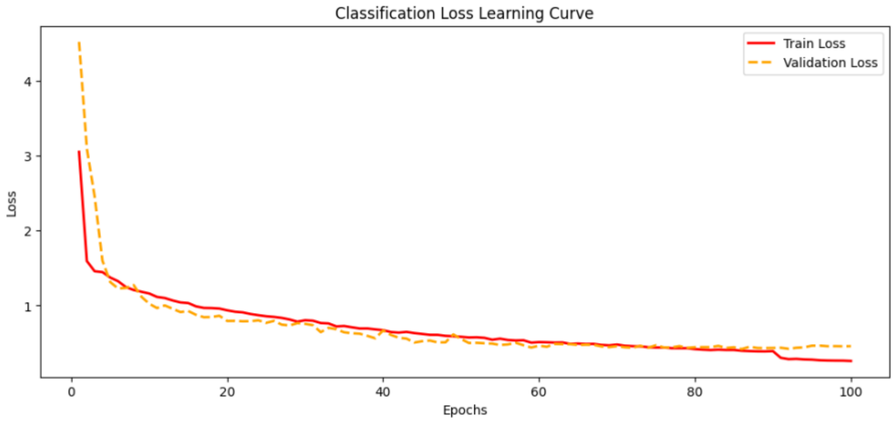


Fig 6: Classification Loss Learning Curve

Based on Figure 6, it shows a classification loss curve over 100 epochs, where both train loss and validation loss have decreased significantly since the beginning of training and tend to stabilize after about the 60th epoch. This decrease reflects the improved ability of the YOLOv8 model in classifying vehicle license plate objects into odd or even categories. The consistency between the two curves also shows that the model did not experience significant overfitting and was able to maintain good classification performance on data that had never been seen. As for the Distribution Focal Loss Learning Curve as follows:

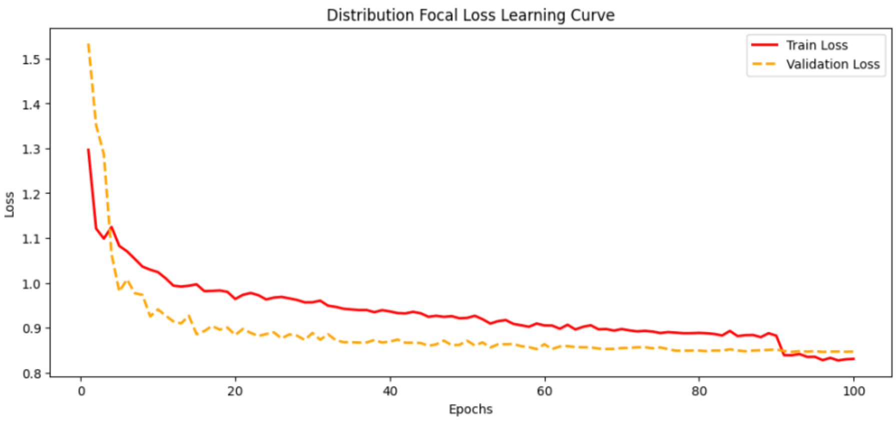


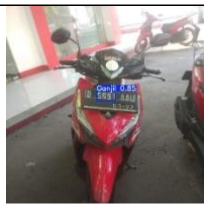
Fig 7: Distribution Focal Loss Learning Curve

Based on figure 7, the Distribution Focal Loss learning curve is shown to be over 100 epochs. The curve shows a steady downward trend in train loss (red line) and validation loss (dotted orange line), reflecting the model's improved accuracy in focusing training on hard-to-classify samples. After about the 60th epoch, the loss of value began to stabilize and was close to converging, with a small difference between the train and the validation loss. This indicates that the YOLOv8 model has good generalizations and does not experience overfitting in the training process.

4.1. Odd-even license plate detection testing and resolution comparison

The detection test of odd and even vehicle license plates aim to evaluate the performance of the YOLOv8 algorithm in accurately recognizing objects. In addition, comparisons were also made of various image resolutions to determine their effect on detection accuracy. Model performance evaluation was carried out using precision, recall, mAP50, mAP50–95, and fitness metrics to obtain a comprehensive picture of the effectiveness and efficiency of the model in the object detection process as follows:

Table 3: Detection Test Data

No.	Annotations	Image Resolution	Test Data	Detection Results	Keterangan
1.	Ganjil	640x640 px			The YOLOv8 algorithm successfully predicts "Odd" plates with a detection accuracy of 0.85 or 85%
2.	Genap	640x640 px			The YOLOv8 algorithm successfully predicts "Even" plates with a detection accuracy of 0.88 or 88%
3.	Ganjil	480x640 px			The YOLOv8 algorithm successfully predicts "Odd" plates with a detection accuracy of 0.77 or 77%
4	Genap	480x640 px			The YOLOv8 algorithm successfully predicts "Even" plates with a detection accuracy of 0.84 or 84%
5.	Ganjil	640x480 px			The YOLOv8 algorithm successfully predicted "Odd" plates with a detection accuracy of 0.92 or 92%
6.	Genap	640x480 px			The YOLOv8 algorithm successfully predicts "Even" plates with a detection accuracy of 0.85 or 85%

Based on Table 3, the resolution image resolution gets the best accuracy is 640x480 pixels, with detection accuracy for odd plates of 0.92 (92%) and detection accuracy for Even plates of 0.85 (85%). From this data, it can be concluded that the resolution of 640x480 pixels provides the highest accuracy, especially for the prediction of the "Odd" plate with a value of 92%. This shows that this resolution is the most optimal in detecting vehicle license plates compared to other resolutions. The modeling results were obtained based on data training conducted over 100 epochs.

4.2. Creating a Detection Function on Real-Time Video

To create a detection function on real-time videos, at runtime code will be executed that will utilize the YOLOv8 train model to be able to automatically detect videos as will be described in Figure 8 below.

```python
import shutil
from IPython.display import Video
from ultralytics import YOLO

# Define the source and destination paths for the video
video_path = '/content/ultralytics/sampel_video.mp4'

# Load the best model
best_model = YOLO('/content/ultralytics/runs/detect/hasil/weights/best.pt')

# Perform vehicle detection on the sample video and save the output
output_dir = '/content/ultralytics/runs/detect'
best_model.predict(source=video_path, save=True, project=output_dir, exist_ok=True)
```

Gambar 8: Detection Function Runtime Code

After the detection function code in figure 8 is executed, the YOLOv8 algorithm will automatically detect objects for every frame in the video, then the results of the detection will be in the folder "/content/ultralytics/runs/detect/predict" in the form of an AVI video format. If the detection video is successfully found, the video output will be displayed containing the identification annotation of the vehicle license plate "Odd" and "Even". The following is a comparison of the before and after detection videos as follows.

Fig 9: Before Detection

Based on the results of figure 9, it is an image before its detection where in the image there are no annotations or labels on each vehicle that has odd or even license plates. Meanwhile, in figure 10 is the result of the application of the YOLOv8 algorithm which will detect with labels and scores the model's confidence level on its detection accuracy.

Fig 10: After Detection

Based on the results of figure 4.15, each vehicle that has an odd or even license plate is detected with a label and a score of the model's confidence level on its detection accuracy. Where the plates were detected 2 license plates with even annotations with percentages of 0.74 or 74% and 0.78 or 78%. As for odd license plates, 1 plate was detected with gajil annotation with a percentage of 0.84 or 84%.

5. Conclusion

Based on the results of research that has been conducted on the detection of odd and even vehicle license plates using the YOLOv8 algorithm, it can be concluded that the YOLOv8 algorithm has excellent performance in classifying objects based on vehicle license plate numbers, both odd and even, as follows:

1. Evaluation of the accuracy of the model in detecting odd and even vehicle license plates
   The results of the evaluation of the YOLOv8 Algorithm model achieved a precision value of 0.879 or a percentage of 87.9% and a recall of 0.888 if a percentage of 88.8%, and an F1-Score value of 88% which indicates a high level of accuracy and completeness of detection. The mAP50 values of 0.954 and the mAP50–95 of 0.830 also reinforce that the model can recognize and localize objects with consistent precision at various levels of overlap bounding boxes.
2. Image resolution that affects the detection accuracy of odd and even vehicle license plates
   The resolution of the image that has a significant influence on the detection accuracy level is the resolution of 640x480 pixels provides the best results, with detection accuracy reaching 92% for odd license plates and 85% for license plates. This makes the resolution an optimal choice to improve the performance of vehicle license plate detection.

6. Suggestions

Based on the results of the research in detecting and classifying vehicle license plates as "Odd" and "Even" using the YOLOv8 Algorithm, some suggestions for improving the performance of this model are as follows:

1. Dataset Enhancement
   To further improve performance, to add a more diverse dataset in terms of lighting conditions, shooting angles, and license plate types and shapes and numbers from 0001 to 9999 to improve the model's generalization.
2. Implementation in a Real-Time Environment
   Given that the model has shown good performance on video, the implementation of YOLOv8 is in real-time conditions for CCTV surveillance of highway or highway traffic.
3. Evaluation of Different Environmental Conditions
   Test the model on a variety of environmental conditions, such as day and night, inclement weather, or crowded road conditions, to ensure model generalization in more complex situations.
4. Better Hardware Use
   For implementations in real-time or larger scale video, use a more powerful GPU such as the A100 or others to significantly improve model performance.

References

[1] N. M. Farhan and B. Setiaji, “Pengembangan Sistem Deteksi Kecelakaan di Jalan Raya Menggunakan Algoritma YOLOv8 dan Notifikasi Otomatis Melalui Telegram,” *Indones. J. Comput. Sci.*, vol. 12, no. 2, pp. 284–301, 2023, [Online]. Available: http://ijcs.stmikindonesia.ac.id/ijcs/index.php/ijcs/article/view/3135

[2] F. F. Sanubari and R. D. Puriyanto, “Deteksi Bola dan Gawang dengan Metode YOLO Menggunakan Kamera Omnidirectional pada Robot KRSBI-B,” *Bul. Ilm. Sarj. Tek. ...*, 2022, [Online]. Available: http://journal2.uad.ac.id/index.php/biste/article/view/6712

[3] K. Anwariyah, “Deteksi Objek Nomor Kendaraan Pada Citra Kendaraan Bermotor,” *JTIM J. Teknol. Inf. dan ...*, 2020, [Online]. Available: https://www.journal.sekawan-org.id/index.php/jtim/article/view/65

[4] H. Diwanti, I. S. Sumaryo, and C. Setianingsih, “Real time smart CCTV untuk mendeteksi plat nomor kendaraan menggunakan optical character recognition real time smart CCTV to detect vehicle license plate using optical character recognition,” *e-Proceeding Eng.*, vol. 6, no. 2, pp. 3045–3052, 2019.

[5] M. Zakiyamani, T. I. Cahyani, D. Riana, and S. Hardianti, “Deteksi Dan Pengenalan Plat Karakter Nomor Kendaraan Menggunakan OpenCV Dan Deep Learning Berbasis Python,” *INTECOMS J. Inf. Technol. Comput. Sci.*, vol. 5, no. 1, pp. 56–64, 2022, doi: 10.31539/intecoms.v5i1.3403.

[6] M. Jonathan, M. T. Hafidz, N. A. Apriyanti, and ..., “MENDETEKSI PLAT NOMOR KENDARAAN DENGAN METODE YOLO (You Only Look Once) DAN SINGLE SHOT DETECTOR (SSD).,” *AI dan SPK J. ...*, 2023, [Online]. Available: http://jurnalmahasiswa.com/index.php/aidanspk/article/view/320

[7] M. R. Rais, F. Utaminingrum, and H. Fitriyah, “Sistem Pengenalan Plat Nomor Kendaraan untuk Akses Perumahan menggunakan YOLOv5 dan Pytesseract berbasis Jetson Nano,” *... Teknol. Inf. dan Ilmu ...*, 2023, [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/12282

[8] K. Ibnutama and M. G. Suryanata, “Ekstraksi Karakter Citra Menggunakan Optical Character Recognition Untuk Pencetakan Nomor Kendaraan Pada Struk Parkir,” *J. Media ...*, 2020, [Online]. Available: http://www.ejurnal.stmik-budidarma.ac.id/index.php/mib/article/view/2432

[9] R. Dwiyanto, D. W. Widodo, and ..., “Implementasi Metode You Only Look Once (YOLOv5) Untuk Klasifikasi Kendaraan Pada CCTV Kabupaten Tulungagung,” *Pros. SEMNAS ...*, 2022, [Online]. Available: https://proceeding.unpkediri.ac.id/index.php/inotek/article/view/2669

[10] K. Ibnutama, Z. Panjaitan, and ..., “Modifikasi Metode Template Matching pada OCR Untuk Meningkatkan Akurasi Deteksi Plat Nomor Kendaraan,” *J. Teknol. Sist. ...*, 2019, [Online]. Available: http://ojs.trigunadharma.ac.id/index.php/jsk/article/view/121