Intelligent Traffic Management System Using Mask Regions-Convolutional Neural Network (MR-CNN)

Muhammad Kemal Pasha[1], Aldy Rialdy Atmadja[2], Muhammad Deden Firdaus[3]
Department of Informatics, Faculty of Sains and Technology, UIN Sunan Gunung Djati Bandung, Indonesia[1,2,3]

Article Info
Keywords:
Deep Learning, Mask R-CNN, R-CNN, Traffic Engineering, Traffic Management

Article history:
Received 17 August 2018
Revised 15 February 2019
Accepted 4 April 2019
Available online 4 April 2019

Cite:
Wardana, A., Rakhmatsyah, A., Minarno, A., & Anbiya, D. (2019). Internet of Things Platform for Manage Multiple Message Queuing Telemetry Transport Broker Server. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 4(3). doi:http://dx.doi.org/10.22219/kinetik.v4i3.841

* Corresponding author.
Aldy Rialdy Atmadja
E-mail address:
aldyrialdy@uinsgd.ac.id

Abstract
Urban centers worldwide continue to face challenges in traffic management due to outdated traffic signal infrastructure. This study aims to develop an intelligent traffic management system by implementing the Mask R-CNN algorithm for real-time vehicle detection and traffic flow optimization. Utilizing the CRISP-DM framework, this research processes CCTV footage from the Pasteur-Pasopati intersection in Bandung to identify and quantify vehicles dynamically. The proposed system leverages an enhanced Mask R-CNN model with a ResNet-50 FPN backbone to improve detection accuracy. Experimental results demonstrate an 80% vehicle detection accuracy, with a macro-average precision of 0.89, recall of 0.83, and an F1-score of 0.82. These findings highlight the system's capability to replace conventional fixed-time traffic signals with a more adaptive approach, adjusting green light durations based on real-time traffic density. The proposed solution has significant practical implications for reducing congestion and improving traffic flow efficiency in urban environments.

1. Introduction

Urban traffic gridlock represents a widespread challenge confronting metropolitan areas globally. The multifaceted burden of vehicular congestion extends across temporal, financial, and psychological domains, significantly impacting daily commuters. According to 2021 INRIX research[1], revealed that London topped the global congestion rankings with drivers losing 148 hours annually, while Paris followed closely behind at 140 hours. In the Indonesian context, Surabaya experienced the highest congestion levels with 62 hours lost, surpassing Jakarta's 28-hour delay. The economic impact is particularly evident in New York's 2020[2] data, where each driver suffered financial losses averaging $1,486, contributing to a citywide economic burden of $7.7 billion from 100 hours lost per driver. Beyond these tangible costs, traffic congestion takes a psychological toll, manifesting in elevated stress levels and increased instances of aggressive driver behavior.

With rapid population growth and urbanization, the number of vehicles on the roads has increased significantly. This leads to decreased transportation efficiency, longer travel times, and higher emissions of pollutants. Congestion not only disrupts the daily activities of city residents but also negatively impacts the economy and environment. In Indonesia, data from the Central Bureau of Statistics (BPS) shows that the number of motor vehicles has continued to rise year after year. In 2021, there were approximately 143 million motor vehicles in Indonesia, and this number increased to around 148 million in 2022[3]. The rising number of vehicles has a direct and significant impact on traffic congestion, especially in many large cities[4].

A significant contributing factor to urban traffic gridlock stems from inadequate traffic signal control mechanisms at road intersections[5]. Multiple research studies and comprehensive reports indicate that numerous traffic light installations continue to operate on predetermined timing schedules that fail to respond to dynamic traffic flow patterns[6][7]. As a result, situations arise where vehicles must stop at a red light even though the other direction is empty, or green lights remain active for too short a time to accommodate high vehicle volumes. These non-adaptive systems exacerbate congestion and increase driver frustration[8]. Traffic signal systems employ either fixed-time scheduling or manual regulation, neither of which accounts for real-time traffic patterns, frequently resulting in avoidable congestion. Fixed-time approaches implement predetermined green and red light durations, while manual systems depend on traffic personnel who cannot efficiently respond to fluctuating vehicle numbers. To address these limitations,

this research introduces an innovative Mask R-CNN-based solution that leverages existing CCTV infrastructure to continuously assess traffic conditions, facilitating intelligent signal timing modifications based on actual vehicular movement data. Through real-time analysis of vehicle concentration and traffic flow dynamics, this proposed system enhances green signal duration optimization, minimizes unnecessary waiting periods at crossroads, and substantially improves metropolitan traffic management efficiency[9].

To address this problem, an intelligent traffic management system is needed to adjust traffic light settings based on real-world conditions. Such a system must be capable of detecting and analyzing traffic volumes in real-time, enabling traffic lights to operate more efficiently and reduce congestion[10]. This technology requires the application of advanced algorithms and sensors that can accurately monitor traffic conditions. The innovation of smart traffic lights is divided into three main focuses: reducing congestion, prioritizing emergency vehicles, and accommodating pedestrians. This system is designed to determine the green light duration for each road lane to help reduce traffic density. To achieve this, the system needs to calculate vehicle volume and density, then learn from vehicle volume data to optimize the traffic light duration[11]. One of the technologies that support intelligent traffic management systems is object detection algorithms, such as Mask Regions-Convolutional Neural Network (MR-CNN). MR-CNN is a sophisticated and accurate algorithm for detecting, identifying, and segmenting objects in images or videos, making it highly suitable for real-time applications like traffic control[12][13].

The Mask R-CNN model represents an advanced iteration of R-CNN (Region-based Convolutional Neural Networks), enhancing the original architecture by incorporating object instance segmentation functionality to its detection capabilities[10]. This algorithm divides the detection task into three main steps: first, generating a large number of region proposals or candidate areas that may contain objects[14]. Second, classifying each candidate area using CNN (Convolutional Neural Network), and third, generating more precise segmentation maps for each detected object[15][16]. With this approach, Mask R-CNN can detect and classify objects with high accuracy and detailed segmentation. This study was carried out at the Pasteur-Pasopati intersection in Bandung, where the Mask R-CNN model was applied to real-world traffic scenarios. The research aims to assess the model's effectiveness in optimizing traffic light configurations based on vehicle volume and density data, ultimately enhancing urban traffic flow. The main contribution of this research is to develop a Mask R-CNN-based traffic management system capable of detecting and counting vehicles in real-time, using the CRISP-DM approach to ensure a systematic research process, and providing model performance analysis under various different traffic conditions.

Several related studies include research by Zhang Gongguo and Wei Junhao in 2021, which utilized an Improved YOLO V3 algorithm for small target detection in traffic flows. This algorithm successfully improved the accuracy of small target detection by 3%, the recall rate of small target detection by 5.2%, and the average accuracy of multi-category detection by 6.64%[17]. Another study by Gokalp Cinarer in 2024 used the YOLOv5 model to achieve high accuracy in traffic sign detection. In this study, three YOLOv5 models (s, m, l) were compared, and the YOLOv5m model demonstrated the highest training quality with an mAP metric of 98.1% after 200 epochs. The YOLOv5l model also achieved the highest precision rate of 99.3%[18]. Similarly, research by Hoang Tran Ngoc, Khang Hoang Nguyen, Huy Khanh Hua, Huynh Vu Nhu Nguyen, and Luyl-Da Quach in 2023 used the YOLOv8 model for traffic light detection, yielding the best results with a Mean Average Precision (mAP) of 98.5%[19]. Another study in the same year, conducted by Huaqing Lai, Liangyan Chen, Weihua Liu, Zi Yan, and Sheng Ye, focused on object detection using YOLOv5s with modifications such as MPANet, C4STB, NWD, and K-means++, which significantly improved small object detection, increasing the mean average precision (mAP) from 79.6% to 86.4%. Additionally, these modifications enhanced accuracy in detecting small objects and improved model performance in challenging conditions such as snow, fog, noise, motion blur, and partial occlusion.[20].

Another study that related to smart traffic light system was conducted by Mochammad Sahal, Zulkifli Hidayat, Yusuf Bilfaqih, Mohamad Abdul Hady, and Yosua Marthin Hawila Tampubolon in 2023 using YOLO. This research successfully developed a smart traffic light system capable of adapting to traffic conditions based on the number of vehicles. The results showed that the average vehicle queue length was reduced from 110 vehicles to 106 vehicles during the 06:00 to 09:00 time frame[21]. In addition, the research was developed by Andrea Vidali, Luca Crociani, Giuseppe Vizzari, and Stefania Bandini titled A Deep Reinforcement Learning Approach to Adaptive Traffic Lights Management demonstrates that the Reinforcement Learning (RL) approach for traffic light adaptation and management has great potential to improve global traffic flow efficiency[22].

Several previous studies have used YOLO and Faster R-CNN methods to detect vehicles in smart traffic systems. However, this study found that these methods have limitations in segmenting more complex objects. By using Mask R-CNN equipped with ResNet-50 FPN, this study fills the horizon by improving the detection accuracy and segmenting more detailed vehicles.

2. Research Method

This study implements the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology, a well-established structure for guiding data mining operations and model creation across diverse analytical scenarios. The framework encompasses six interconnected and cyclical phases[6][7][23], offering flexibility to adapt to specific project requirements and circumstances.

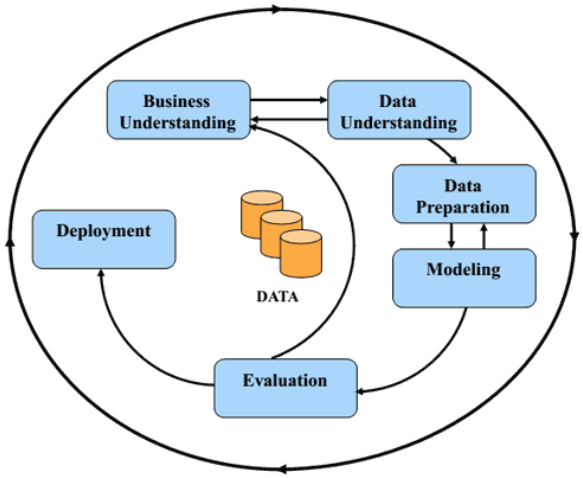

Figure 1. CRISP-DM Methodology[6]

2.1. Business Understanding

In this research, the business understanding phase encompasses a comprehensive analysis[24]. The core objective centers on creating a smart traffic management system that employs the Mask R-CNN algorithm for automated vehicle detection, specifically utilizing TorchVision's pre-trained maskrcnn_resnet50_fpn model. This initial stage incorporates the following components:

a. Vehicle Analysis: Analyze vehicles through CCTV footage recorded at the Pasteur-Pasopati intersection in Bandung in the morning on a Friday.
b. Inference Using Pre-Trained Model: Perform inference using the pre-trained maskrcnn_resnet50_fpn model for vehicle detection, which includes feature extraction and object classification without requiring model retraining[25].
c. Output Generation: Produce a new video recording containing vehicle predictions and counts as part of the output.

2.2. Data Understanding

This stage describes the data utilized in the intelligent traffic control system. It involves data collection, processing, and labeling. Activities:

a. Data Collection and Identification: Collect and identify data in the form of CCTV footage recorded at the Pasteur-Pasopati intersection in Bandung in the morning on a Friday.
b. Data Processing: Analyze the video characteristics, such as color intensity, duration, and image quality, ensuring that the video contains sufficient detail for vehicle detection and counting processes.
c. Data Labeling: Label the video based on the types of objects it contains, such as vehicles (cars, motorcycles, trucks, bicycles, and buses).

Table 1 presents the index and labels used during the object detection testing process. Out of the nine available traffic-related labels, the testing will focus on five vehicle categories Bicycle (Index 2), Car (Index 3), Motorcycle (Index 4), Bus (Index 6) and Truck (Index 8).

Table 1. Model Label

Index	Label
3	car
4	motorcycle
6	bus
8	truck
2	bicycle
7	train
9	Traffic light
12	stop sign
13	parking meter

2.3. Data Preparation

The data processed originates from CCTV footage of the intersection, and the video processing stages are as follows:

a. Video Acquisition: The video is recorded from CCTV footage at the Pasteur-Pasopati intersection in Bandung in the morning on a Friday.
b. Frame Extraction: The video is divided into individual frames.
c. Frame Processing: Each frame is processed using the pre-trained maskrcnn_resnet50_fpn model to detect and count objects.
d. Object Labeling: Detected objects are labeled according to their categories (car, motorcycle, truck, bus, bicycle) using the COCO (Common Objects in Context) annotation format.
e. Data Storage: The labeled and counted data is stored for use in simulating traffic light conditions.

Figure 2. Video per frame

Figure 2 provides a visual representation of the sequential steps involved in processing video data obtained from CCTV footage. This process includes capturing raw footage, preprocessing the video to enhance quality, applying object detection algorithms, analyzing traffic patterns, and extracting relevant data for further decision-making in intelligent traffic management systems.

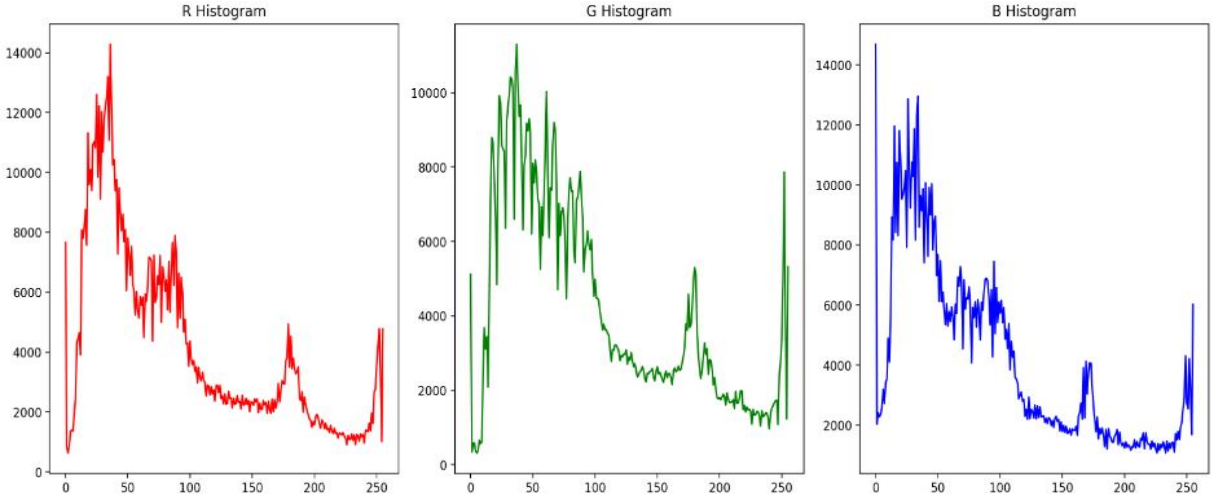

Figure 3. Frame average color analysis process

The RGB intensity distribution analysis, depicted in Figure 3, provides a histogram visualization examining the pixel characteristics within the video frame samples. Each color component—Red (R), Green (G), and Blue (B)—displays intensity values spanning from 0 to 250. The histograms reveal comparable distribution trends across all channels, characterized by pixel concentrations predominantly in the lower intensity ranges (displaying left-skewed distributions). Peak frequency measurements differ among the channels, with the Red channel reaching roughly 14,000 pixels, the Blue channel showing approximately 12,000 pixels, and the Green channel registering around 10,000 pixels at their respective maxima.

This analysis of RGB intensity distribution plays a critical role in understanding the visual characteristics of video frames, particularly regarding variations in lighting and color dominance. These factors can significantly influence subsequent image processing tasks, such as object detection and classification[26].

2.4. Modelling

As the number of vehicles in urban areas increases, intelligent traffic management systems are becoming increasingly important to optimize traffic flow and reduce congestion. Computer vision-based technologies, such as Mask R-CNN, offer a more adaptive solution compared to conventional systems that rely on fixed time or manual settings. With its pixel-level segmentation and accurate object detection capabilities, Mask R-CNN can improve the efficiency of traffic light settings based on real-time traffic conditions. However, to ensure that this model performs optimally in various real-world situations, data preprocessing and augmentation are required to improve the model's robustness to environmental variations and video quality.

1. Pra-Pemrosesan
   In order for Mask R-CNN to accurately detect vehicles in a variety of conditions, several data preprocessing techniques are applied before the video is processed by the model:
   - Contrast and Brightness Normalization, Improves object visibility in extreme lighting conditions, such as bright sunlight, sharp shadows, or night scenes.
   - Perspective Correction, Reduces distortions caused by varying camera angles, ensuring vehicles remain proportional in a variety of positions.
2. Augmentation
   To ensure the model performs well in a variety of real-world scenarios, several data augmentation techniques are applied that help improve the model's robustness:
   - Control Noise, Adding visual noise to ensure the model can still recognize vehicles even when video quality is poor or sensor noise is present.
   - Rotation and Scalling, Simulating different camera angles and vehicle sizes to improve model generalization[27].

The approach to modeling relies on the Mask R-CNN algorithm to implement intelligent traffic management. This algorithm integrates a Convolutional Neural Network (CNN) layer for extracting image features, a Region Proposal Network (RPN) for identifying objects, and a Mask Head for segmenting objects within the images[28]. Since the model is pre-trained using PyTorch and optimized for a variety of object detection tasks, it can perform object detection directly on input data without requiring additional training.

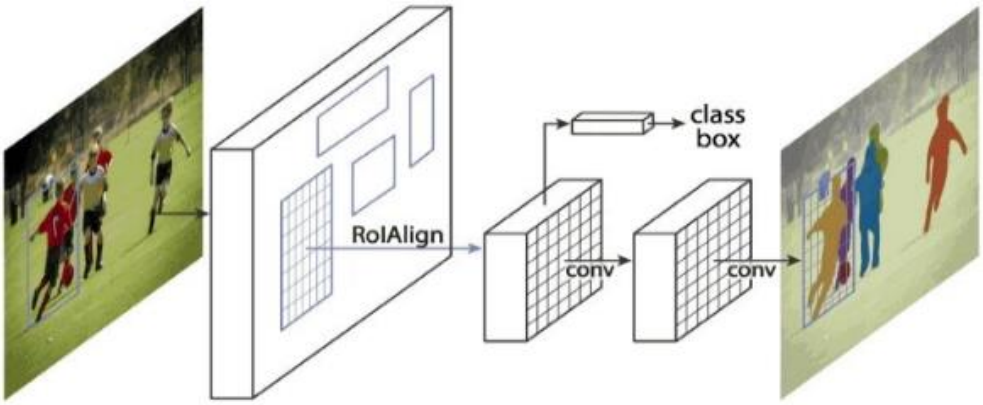

Figure 4. MR-CNN classification flow

Figure 4 illustrates the MR-CNN classification flow, which consists of several key stages: input image, RoIAlign (Region of Interest Alignment), class box prediction, and output. The process begins with an input image, where relevant regions are extracted using RoIAlign to ensure precise spatial alignment. Next, the system classifies objects and generates bounding boxes in the class box prediction stage. Finally, the output consists of detected objects with their respective classifications and segmented masks, making MR-CNN highly effective for real-time traffic analysis and management[28].

2.5. Evaluation

The model's performance is assessed using three key metrics: Precision, Recall, and F1-Score. Precision evaluates the accuracy of the model's predictions by comparing the relevant data retrieved with the total data identified by the model. Recall assesses the model's effectiveness in correctly identifying all relevant information. The F1-Score,

serving as a comprehensive measure, is the harmonic mean of Precision and Recall, offering a balanced evaluation of the model's performance[16].

1. Validation and Variability of Train Loss
   To ensure the model can generalize well to various traffic conditions, k-fold cross validation (k=5) is applied. This approach divides the dataset into five parts, where four parts are used for training and one part for testing in each iteration. The validation results show quite large variations in train loss in each fold:
   - Fold 1: 33.2418
   - Fold 2: 2.2435
   - Fold 3: 0.9201
   - Fold 4: 3.4749
   - Fold 5: 10.7194

   This variation can be caused by uneven data distribution, where some subsets of data have higher complexity, such as many stacked vehicles or less than ideal lighting. Folds with low train loss (e.g. Fold 3: 0.9201) may be overfitting, while high train loss (e.g. Fold 1: 33.2418) may indicate that the data in that fold is more complex, but does not necessarily result in low accuracy. Therefore, to understand the impact on model performance, it is necessary to compare it with the evaluation results on the test data.
2. Evaluation Metrics
   Precision, Recall, and F1-Score are calculated using the following formulas:

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

   In this context, True Positive (TP) refers to the count of instances where the model accurately predicted a positive class, aligning correctly with the desired outcome. Conversely, False Positive (FP) indicates the count of instances where the model mistakenly identified a positive class, leading to an incorrect prediction for the intended outcome.

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

   In this case, True Positive (TP) refers to the count of instances where the model accurately identified a positive class, signifying correct predictions for the intended outcome. Meanwhile, False Negative (FN) denotes the count of instances where the model incorrectly classified positive instances as negative, resulting in missed or misclassified positive outcome.

$$F1 - Score = 2\ x\ \frac{precision\ x\ Recall}{precision+recal} \quad (3)$$

3. Results and Discussion

The following are the results obtained and the discussion related to the labeling process, vehicle counting, simulation, and the analysis of the vehicle detection system using the Mask R-CNN model for intelligent traffic management. Based on a single test sample, the model achieved a perfect precision, recall, and F1-score of 1.00. For the car category, the model demonstrated a precision of 0.67, a recall of 1.00, and an F1-score of 0.80 from two test samples. Similarly, in the motorcycle category, the model attained a precision of 1.00, a recall of 0.50, and an F1-score of 0.67, also from two test samples. In total, the system achieved an overall vehicle detection and counting accuracy of 80% across five test samples. Moreover, it recorded a macro-average precision of 0.89, a recall of 0.83, and an F1-score of 0.82.

3.1. Labelling Process

The implementation of this system is developed using the Python programming language, leveraging the pre-trained maskrcnn_resnet50_fpn model from the Torchvision library. Technically, the labeling process is carried out using the class_names list, which has been defined within the pre-trained Mask R-CNN model, with COCO annotation format according to the label index that has been established. The model detects five main categories of vehicles: bicycle, car, motorcycle, bus, and truck, which are commonly encountered in road traffic. These labels are used as references in the classification and real-time vehicle counting process.

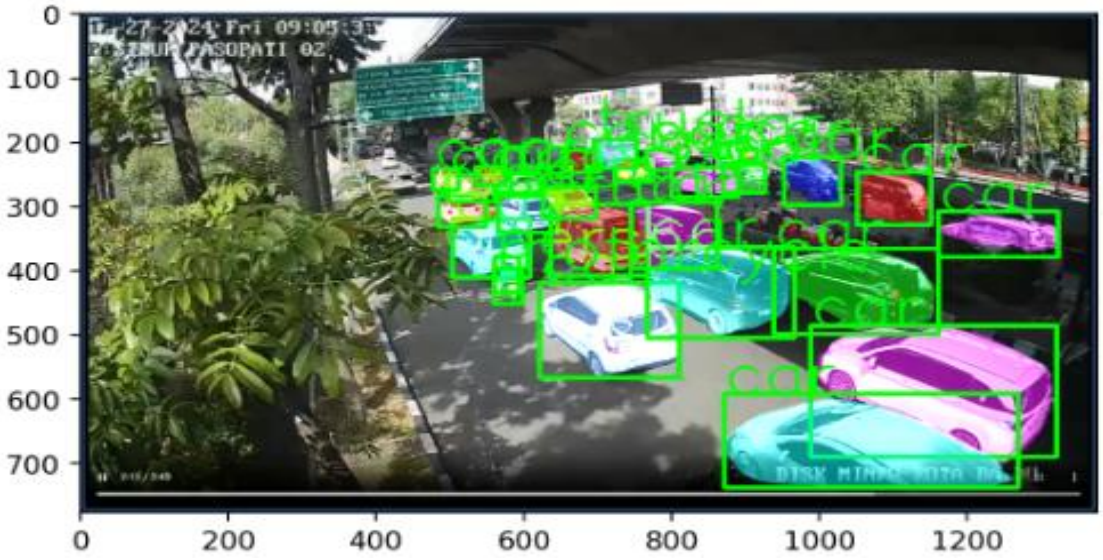

Figure 5. Labelling

Figure 5 shows the result of the labeling implementation, where each detected vehicle object is assigned a bounding box and labeled according to its category based on the predetermined index. This labeling result demonstrates the model's ability to identify and classify various types of vehicles within the video frame being analyzed.

3.2. Vehicle Counting

The process of detecting and counting vehicles, where the input video is divided into a series of frames. These frames are then analyzed by the model to detect and count the number of vehicles identified in each frame.

Figure 6. Vehicle counting by category

as shown in Figure 6 below. The data shows that the vehicles counted correctly are 13 for cars and 1 for motorcycles. However, the model identified 15 cars and 1 truck, resulting in an accuracy of 92.86%. In the image, the blurred area is only for identification to represent unnecessary footage and is intentionally excluded to prevent it from being counted. In addition to accuracy, the computational efficiency of the model is an important factor in implementing real-time traffic management. Mask R-CNN has a complex architecture with pixel-level segmentation processes, which makes its inference time higher than lighter detection methods such as YOLO or MobileNet. In this test, the model

showed an average inference time of 180 ms per frame on Cuda GPU, which is still applicable in adaptive traffic systems but may experience latency on devices with limited computing power.

3.3. Traffic Analysis Simulation (TAS)

The Traffic Analysis Simulation (TAS) is conducted to simulate the traffic monitoring process by displaying both the traffic light status and vehicle count results simultaneously in the video. This system integrates two main components: vehicle detection and traffic light status monitoring, which are displayed as an overlay on the video stream. The detected vehicle count and traffic light status are visualized in real-time, enabling live observation of the current traffic conditions.

(a) (b) (c)

Figure 7. Prediction with traffic light status

In Figure 7a, the prediction result shows the "STOP" status. In this system, the "STOP" status is triggered if the number of vehicles detected in the video frame is fewer than 20. When this condition is met, the system enters the "STOP" status. Once the "STOP" status is active, the system will wait for 60 seconds before transitioning to the next status.

After 60 seconds, as shown in Figure 7b, the system transitions to the "WARNING" status. This status occurs after the red light (STOP status) has been on for 60 seconds, signaling that the red light is about to switch to green. The "WARNING" status lasts for 5 seconds, giving drivers time to prepare for the upcoming light change.

In Figure 7c, the prediction result displays the "GO" status. This status is triggered when the number of vehicles detected exceeds 20. The "GO" status remains active for 30 seconds, allowing enough time for vehicles to pass through the intersection before the light changes again.

However, in real-world traffic scenarios, these settings need to be extended to handle more complex conditions, such as:

1. Traffic Congestion, if the number of vehicles in several consecutive frames continues to increase without decreasing, the system will detect congestion and extend the green light duration to the maximum threshold to reduce the queue of vehicles.
2. Pedestrian Crossings, if additional cameras or sensors detect a large number of pedestrians in the crosswalk area, the system can prioritize the red light for longer to allow pedestrians time to cross safely.
3. Multi-lane Intersections, at intersections with multiple lanes, the system can use priority lane analysis, where the lane with the highest vehicle density is given a longer green duration than the lane with less traffic.

3.4. Analysis of Multi-Class Vehicle Detection Results

The model demonstrated varying levels of performance across different vehicle categories. The bus category achieved perfect performance, with precision, recall, and F1-score all equal to 1.00, based on a single test sample. For the car category, the model attained a precision of 0.67, recall of 1.00, and an F1-score of 0.80 from two test samples. Similarly, the motorcycle category achieved a precision of 1.00, recall of 0.50, and an F1-score of 0.67, also from two test samples. Overall, the model achieved an 80% accuracy in detecting and counting vehicles across five test samples. Additionally, it obtained a macro-average precision of 0.89, recall of 0.83, and F1-score of 0.82. The model shows varying performance across vehicle categories.

1. Buses perform excellently (Precision, Recall, and F1-Score = 1.00) due to their large size and distinctive visual features, making them easier for the model to recognize.
2. Cars show a precision of 0.67 and a recall of 1.00, meaning the model often correctly detects cars but also produces false positives, possibly due to misclassification with trucks or similar vehicles.

3. Motorcycles have a precision of 1.00 but a recall of only 0.50, indicating that the model often fails to detect motorcycles that are actually in the frame.

Table 2. Confusion matrix for class in frame

	Car	Truck	Motorcycle
Car	13	1	0
Motorcycle	0	0	1
Truck	0	0	0

As shown in Table 2, the model achieved a detection result where the actual number of cars (13) and motorcycles (1) was compared against the predictions. The model correctly identified 13 cars but mistakenly detected 1 truck instead of a car and failed to misclassify motorcycles as cars or trucks. Additionally, the motorcycle was correctly detected, resulting in a detailed performance evaluation reflected in the confusion matrix.

To improve model performance and reduce detection errors (false positives and false negatives), several strategies can be applied, such as expanding and enriching the dataset by adding more data, especially for the motorcycle and truck categories, and increasing the variety of shooting angles and lighting conditions. The imbalance in the number of samples between vehicle categories in the dataset can be overcome by oversampling or synthetic data augmentation techniques so that the model has a better representation of each category. In addition, adjusting the detection threshold (threshold tuning) in Mask R-CNN can help reduce errors in over-detecting cars and increase sensitivity to previously under-detected motorcycles. Although Mask R-CNN has shown an accuracy of 80%, the difference in performance between vehicle categories is still a challenge, especially in detecting motorcycles which have lower recall.

4. Conclusion

This research successfully developed an intelligent traffic light control system using Mask R-CNN for real-time vehicle detection and counting, optimizing traffic flow based on vehicle density. The model, built upon maskrcnn_resnet50_fpn with a ResNet-50 backbone and Feature Pyramid Network (FPN), demonstrated high accuracy in detecting and categorizing vehicles. The Car category achieved 100% recall but lower precision (67%), resulting in an F1-score of 80%, while the Motorcycle category exhibited perfect precision (100%) but a lower recall (50%), leading to an F1-score of 67%. The system successfully adjusted traffic light durations dynamically, achieving an accuracy of 92.86% in real-time frame identification. The scalability of this system holds significant potential for managing larger and more complex traffic scenarios, such as multi-lane intersections or city-wide traffic control. Compared to YOLO (You Only Look Once), Mask R-CNN offers complementary strengths in certain applications. YOLO is widely recognized for its high inference speed and efficiency, making it well-suited for real-time detection tasks with limited computational resources. However, in scenarios that require detailed object localization and segmentation, such as dense or wide-area traffic scenes, Mask R-CNN provides additional advantages. Its ability to perform pixel-wise segmentation (masking) allows for more precise identification of vehicles, including those located at greater distances from the camera. This makes Mask R-CNN particularly effective for traffic monitoring systems where both detection accuracy and spatial detail are important. Rather than replacing YOLO, Mask R-CNN serves as a robust alternative for use cases that demand richer contextual information and finer spatial resolution. While the model can generalize across various traffic conditions, adaptation may be required for different cities due to variations in traffic behavior, infrastructure, and road regulations. Future improvements could explore automated adaptation techniques, allowing the system to learn and optimize itself dynamically based on localized traffic patterns without requiring manual adjustments.

To further enhance model robustness, collecting diverse traffic video datasets is recommended, particularly for edge cases like heavy rain, nighttime, and foggy conditions. Model quantization or pruning techniques could also be implemented to reduce model size and accelerate inference without sacrificing accuracy. Future research should address technical challenges and potential solutions for real-world system deployment, including hardware considerations, edge deployment, and integration with existing traffic management infrastructure. Further research can focus on expanding object detection capabilities to include pedestrians, non-motorized vehicles, and public transport, providing a more comprehensive traffic analysis. Additionally, integrating reinforcement learning could enable adaptive traffic signal control, where the system continuously learns from real-time data to optimize light timing autonomously. Implementing edge computing solutions could also enhance real-time processing, reducing dependency on cloud infrastructure and minimizing computational latency. Overall, this research demonstrates the feasibility and effectiveness of AI-driven traffic management, offering a foundation for future advancements in smart city traffic control that is scalable, adaptive, and efficient in reducing congestion and improving urban mobility.

Acknowledgement

We would like to express our deepest gratitude to the Faculty of Science and Technology, Department of Informatics Engineering, UIN Sunan Gunung Djati Bandung for publishing this research and would also like to thank the supervisor for his guidance and contribution to this research.

References

[1] B. Pishue, "2021 INRIX Global Traffic Scorecard," p. 21, 2021.

[2] B. Pishue, "2021 INRIX Global Traffic Scorecard," p. 23, 2021.

[3] Badan Pusat Statistik, "Jumlah Kendaraan Indonesia 2022." Accessed: Jun. 12, 2024. [Online]. Available: https://www.bps.go.id/id/statistics-table/3/VjJ3NGRGa3dkRk5MTlU1bVNFOTVVbmQyVURSTVFUMDkjMw==/jumlah-kendaraan-bermotor-menurut-provinsi-dan-jenis-kendaraan--unit---2022.html?year=2022

[4] K. Lubis, "Analysis Of The Characteristics Of Public Transportation Modes To Users Of Land Transportation Modes As City Transportation Within The Province," Online, 2019. doi: https://doi.org/10.30743/but.v15i1.1877.

[5] J. Dwijoko Ansusanto and S. Tanggu, "Analisis Kinerja Dan Manajemen Pada Simpang Dengan Derajat Kejenuhan Tinggi Performance Analysis And Management On Saturated Traffic Intersection." [Online]. Available: http://dinarek.unsoed.ac.id

[6] J. Li, Y. Zhang, and Y. Chen, "A Self-Adaptive Traffic Light Control System Based on Speed of Vehicles," in *2016 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, IEEE, Aug. 2016, pp. 382–388. doi: 10.1109/QRS-C.2016.58.

[7] A. Muralidharan, R. Pedarsani, and P. Varaiya, "Analysis of fixed-time control," *Transportation Research Part B: Methodological*, vol. 73, pp. 81–90, Mar. 2015, doi: 10.1016/j.trb.2014.12.002.

[8] S. Zhao, G. Qi, P. Li, and W. Guan, "The aggressive driving performance caused by congestion based on behavior and EEG analysis," *J Safety Res*, vol. 91, pp. 381–392, Dec. 2024, doi: 10.1016/j.jsr.2024.10.004.

[9] Z. Fahrunnisa and R. Arief Setyawan, "Adaptive Traffic Light Signal Control Using Fuzzy Logic Based on Real-Time Vehicle Detection from Video Surveillance," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 10, no. 2, pp. 235–251, 2024, doi: 10.26555/jiteki.v10i2.28712.

[10] F. Zahwa, C.-T. Cheng, and M. Simic, "Novel Intelligent Traffic Light Controller Design," *Machines*, vol. 12, no. 7, p. 469, Jul. 2024, doi: 10.3390/machines12070469.

[11] A. N. Aulia Yusuf, A. Setyo Arifin, and F. Yuli Zulkifli, "Recent development of smart traffic lights," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, no. 1, p. 224, Mar. 2021, doi: 10.11591/ijai.v10.i1.pp224-233.

[12] et al Abigail See, "Mask-RCNN for object detection and instance segmentation on Keras and TensorFlow." Accessed: Jun. 19, 2024. [Online]. Available: https://github.com/matterport/Mask_RCNN

[13] S. Sumahasan, "Object Detection using Deep Learning Algorithm CNN," *Int J Res Appl Sci Eng Technol*, vol. 8, no. 7, pp. 1578–1584, Jul. 2020, doi: 10.22214/ijraset.2020.30594.

[14] M. Li, H. Zhu, H. Chen, L. Xue, and T. Gao, "Research on Object Detection Algorithm Based on Deep Learning," *J Phys Conf Ser*, vol. 1995, no. 1, p. 012046, Aug. 2021, doi: 10.1088/1742-6596/1995/1/012046.

[15] F. Charli, H. Syaputra, M. Akbar[3], S. Sauda, and F. Panjaitan, "Implementasi Metode Faster Region Convolutional Neural Network (Faster R-CNN) Untuk Pengenalan Jenis Burung Lovebird," 2020. [Online]. Available: https://journal-computing.org/index.php/journal-ita/index

[16] L. Du, R. Zhang, and X. Wang, "Overview of two-stage object detection algorithms," *J Phys Conf Ser*, vol. 1544, no. 1, p. 012033, May 2020, doi: 10.1088/1742-6596/1544/1/012033.

[17] Z. Gongguo and W. Junhao, "An improved small target detection method based on Yolo V3," in *Proceedings - 2021 International Conference on Electronics, Circuits and Information Engineering, ECIE 2021*, Institute of Electrical and Electronics Engineers Inc., Jan. 2021, pp. 220–223. doi: 10.1109/ECIE52353.2021.00054.

[18] G. ÇINARER, "Deep Learning Based Traffic Sign Recognition Using YOLO Algorithm," *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, vol. 12, no. 1, pp. 219–229, Jan. 2024, doi: 10.29130/dubited.1214901.

[19] H. Tran Ngoc, K. Hoang Nguyen, H. Khanh Hua, H. Vu Nhu Nguyen, and L.-D. Quach, "Optimizing YOLO Performance for Traffic Light Detection and End-to-End Steering Control for Autonomous Vehicles in Gazebo-ROS2." [Online]. Available: www.ijacsa.thesai.org

[20] H. Lai, L. Chen, W. Liu, Z. Yan, and S. Ye, "STC-YOLO: Small Object Detection Network for Traffic Signs in Complex Environments," *Sensors*, vol. 23, no. 11, Jun. 2023, doi: 10.3390/s23115307.

[21] M. Hidayat, Z. Bilfaqih, Y. A. Hady, and M. A. Tampubolon, "Smart Traffic Light Using YOLO Based Camera with Deep Reinforcement Learning Algorithm," 2023. doi: 10.12962/jaree.v7i1.335.

[22] A. Vidali, L. Crociani, G. Vizzari, and S. Bandini, “A Deep Reinforcement Learning Approach to Adaptive Traffic Lights Management,” 2019. [Online]. Available: https://population.un.org/wup/
[23] C. Schröer, F. Kruse, and J. M. Gómez, “A Systematic Literature Review on Applying CRISP-DM Process Model,” *Procedia Comput Sci*, vol. 181, pp. 526–534, 2021, doi: 10.1016/j.procs.2021.01.199.
[24] S. Jaggia, A. Kelly, K. Lertwachara, and L. Chen, “Applying the CRISP-DM Framework for Teaching Business Analytics,” *Decision Sciences Journal of Innovative Education*, vol. 18, no. 4, pp. 612–634, Oct. 2020, doi: 10.1111/dsji.12222.
[25] K. Abedi, J. Codjoe, R. Thapa, and V. Gopu, “Making Data-Driven Transportation Decisions for Freight Operations,” *J Transp Technol*, vol. 13, no. 03, pp. 411–442, 2023, doi: 10.4236/jtts.2023.133020.
[26] S. Zia, B. Yuksel, D. Yuret, and Y. Yemez, “RGB-D Object Recognition Using Deep Convolutional Neural Networks,” in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, IEEE, Oct. 2017, pp. 887–894. doi: 10.1109/ICCVW.2017.109.
[27] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.10934
[28] M. A. Khasawneh and A. Awasthi, “Intelligent Meta-Heuristic-Based Optimization of Traffic Light Timing Using Artificial Intelligence Techniques,” *Electronics (Basel)*, vol. 12, no. 24, p. 4968, Dec. 2023, doi: 10.3390/electronics12244968.