Faster R-CNN-Based Deep Q-Network for Smart Traffic Light Control

Muhammad Dzaki Salman[1], Ervin Yohannes[2]
[1,2] Teknik Informatika, Universitas Negeri Surabaya
[1]muhammaddzaki.22042@mhs.unesa.ac.id
[2]ervinyohannes@unesa.ac.id

***Abstract*—Conventional fixed-time traffic light control systems cannot accommodate real-time fluctuations in vehicle volume, making adaptive approaches based on reinforcement learning an increasingly popular alternative. This study implements a Deep Q-Network (DQN) architecture combined with a Faster R-CNN ResNet-50 FPN v2 object detection sensor for smart traffic light control. Traffic data were collected from CCTV recordings of a T-intersection in the city of Surabaya via the SITS application and processed into a Discrete Traffic State Encoding (DTSE) representation sized 4×20×3 as the model input. Evaluation was carried out under three time conditions (morning, afternoon, and night) using the average queue length, average reward, and total departed metrics. The Faster R-CNN detector achieved a recall of 96.21%, outperforming comparable single-stage detection methods and providing more reliable traffic observations for the reinforcement learning agent. Among the evaluated checkpoints, the model trained for 500 episodes delivered the best overall performance, with an average queue length of 127.34, an average reward of 161.75, and 1,046.97 departed vehicles. These findings suggest that accurate vehicle detection plays an important role in improving reinforcement learning performance for adaptive traffic signal control.**

***Keywords*—Deep Q-Network, Faster R-CNN, Adaptive Traffic Light, DTSE, Reinforcement Learning.**

I. Introduction

Rapid population growth and urbanization have driven a significant increase in the number of motor vehicles in Indonesia's urban areas. Statistics Indonesia (BPS) recorded more than 140 million motor vehicle units in 2022 [1], [2]. The imbalance between vehicle growth and road capacity causes congestion that results in economic losses exceeding IDR 1.5 trillion across major Indonesian cities [3], as well as increased pollutant emissions and reduced productivity [4].

Conventional fixed-time traffic light control systems cannot adapt to real-time variations in vehicle volume, resulting in long queues and inefficient waiting times [5], [6]. Reinforcement Learning (RL) has emerged as one of the most widely explored approaches for traffic signal optimization because it enables an agent to learn effective control policies directly through interaction with its environment instead of depending on predefined traffic models [7], [8].

One of the most widely adopted Deep Reinforcement Learning (DRL) algorithms is Deep Q-Network (DQN) [9], which has demonstrated promising results in traffic signal optimization tasks [10], [11]. However, most previous studies validate their methods only in simulation environments. Consequently, their reported performance may not fully represent real-world deployment, where decisions rely on visual information captured from actual traffic scenes.

This study implements a DQN combined with a Faster R-CNN ResNet-50 FPN v2 object detection sensor [12], [13], [14] for adaptive traffic light control based on real CCTV data. The main contributions of this study include: (1) an end-to-end pipeline implementation from real CCTV video to traffic light control decisions via DTSE representation, and (2) an analysis of the effect of object detection sensor quality on DQN architecture performance.

II. Literature Review

A. Deep Q-Network (DQN)

DQN [9] combines Q-learning with a deep neural network to approximate the value function Q(s, a), enabling the agent to learn an optimal policy from high-dimensional input. DQN's two main innovations are experience replay and a target network, which stabilize the training process. The optimal Q function underlying the DQN algorithm is defined as:

$$Q(s,a) = E[r + \gamma \cdot max^{a'} Q(s',a') \mid s,a] \quad (1)$$

where s is the current state, a is the action taken, r is the immediate reward, s′ is the next state, and γ is the discount factor ($0 < \gamma < 1$), which determines how much the agent values future rewards.

B. Faster R-CNN

Faster R-CNN [12] is a two-stage object detector that combines a Region Proposal Network (RPN) with a classifier. The ResNet-50 FPN v2 version [13], [14] leverages a Feature Pyramid Network to detect objects at multiple scales, making it more sensitive to distant or small vehicles than single-stage detectors.

C. Related Work

Nguyen and Vo [10] showed that DQN can significantly reduce vehicle waiting time compared to fixed-time systems. Park and Han [11] demonstrated the effectiveness of DQN across various traffic scenarios. Putra et al. [15] showed that a well-designed reward function can improve DQN performance by up to 90% compared to conventional systems. This study complements that literature by validating DQN using real CCTV data through a Faster R-CNN sensor.

III. Research Method

A. Research Design

This study uses a two-stage approach: a DQN model training stage within a simulation environment based on real CCTV data, and an evaluation stage on test data separated via a Stratified Temporal Split. The model built is a DQN paired with a Faster R-CNN ResNet-50 FPN v2 sensor as the source of the environment's state representation.

B. Dataset and Extraction Pipeline

The data are sourced from CCTV video recordings of a T-intersection (3 lanes: A, B, C) in the city of Surabaya via the Department of Transportation's SITS application. The recordings cover three time conditions (morning, afternoon, night). Vehicle detection uses fasterrcnn_resnet50_fpn_v2 (pretrained on COCO) with a confidence threshold of 0.1 and per-lane ROI-based filtering.

The detection results are converted into a Discrete Traffic State Encoding (DTSE) grid with dimensions (4, 20, 3). The dataset contains 5,061 samples, split sequentially per time condition using an 80/20 Stratified Temporal Split, yielding 4,048 training samples and 1,013 test samples.

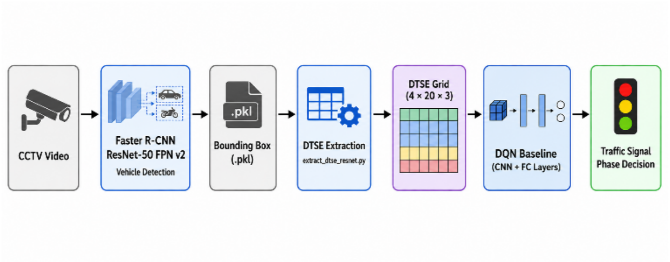

Figure 1. Smart Traffic Light Control System Pipeline Based on DQN Baseline

TABLE I
DATASET DESCRIPTION

Characteristic	Description
Data Source	CCTV via SITS, Surabaya Dept. of Transportation
Number of Lanes	3 lanes (A, B, C)
Time Conditions	Morning, Afternoon, Night
Total Samples	5,061 rows
Training Data	4,048 rows (80%)
Test Data	1,013 rows (20%)
Dataset Format	PyTorch Tensor (.pt)
DTSE Grid Dimensions	(4 × 20 × 3)

C. DQN Model Architecture

The proposed DQN receives a DTSE representation with dimensions of (4, 20, 3) as its input. Feature extraction is performed using three convolutional layers that progressively increase the number of channels from 4 to 32, then 64, and finally 64 feature maps. Each convolution is followed by a ReLU activation function, while adaptive average pooling is applied before the extracted features are flattened. The resulting 256-dimensional feature vector is transformed into 128 neurons through a fully connected layer and subsequently processed by two hidden layers containing 64 neurons each. A final linear layer produces the Q-values for the two available traffic signal actions. In contrast to the DQN Attention model, this network relies solely on features learned by the convolutional layers, without incorporating any channel attention mechanism. In addition, traffic signal transitions are permitted only after the current green phase has remained active for at least five timesteps.

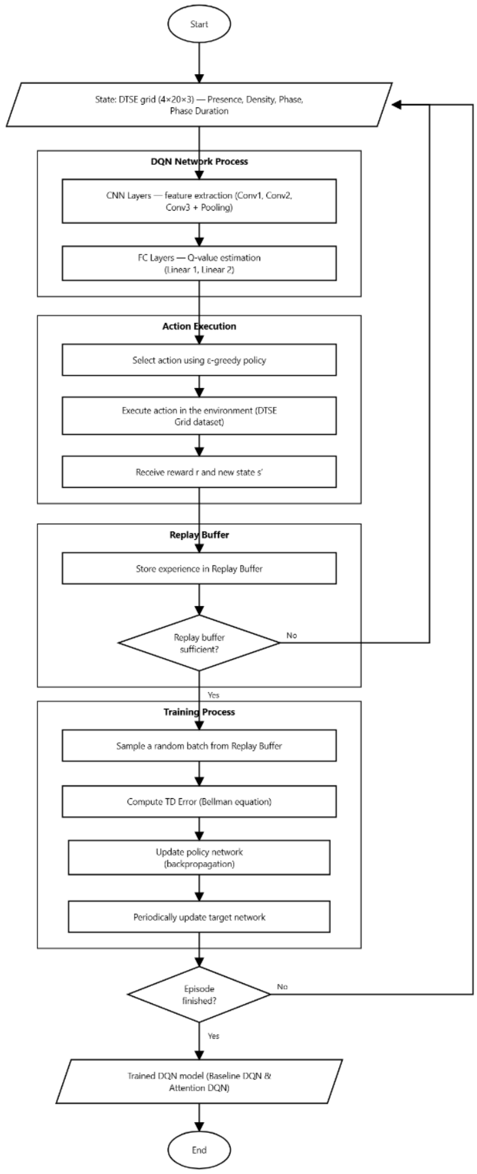

Figure 2. DQN Training Process Flowchart

D. Reward Function and Hyperparameters

The reward function is designed to encourage queue reduction and increased throughput:

$$R_t = -\frac{Q_{total}}{Q_{max}} - 0.5\,\mathbb{1}_{\text{SWITCH}} + 0.3D_t - 0.1\,\mathbb{1}_{\text{IDLE}} \quad (2)$$

TABLE II
DQN HYPERPARAMETER CONFIGURATION

Hyperparameter	Value
Learning Rate	0.001 (Adam)
Batch Size	64
Gamma (γ)	0.99
Epsilon Decay	0.995
Target Network Update	Every 10 episodes
Replay Buffer	10,000 transitions
Max Steps/Episode	150
Total Episodes	1,500
Checkpoints	500, 1,000, 1,500 ep

E. Simulation Environment

The experiments were carried out in a simulated three-lane T-intersection composed of lanes A, B, and C with a cyclic traffic signal sequence. Each training episode consisted of 150 timesteps, and the traffic condition at the beginning of every episode was randomly assigned as morning, afternoon, or nighttime. At each timestep, the agent received a four-channel DTSE representation containing vehicle presence, vehicle density, the current signal phase, and phase duration. To better reflect realistic traffic operation, signal switching was restricted until the active green phase had lasted for at least five timesteps.

F. Evaluation Metrics

Sensor evaluation uses Precision, Recall, and F1-Score against 90 ground-truth samples (10 frames × 3 lanes × 3 conditions). Control evaluation uses three metrics: Average Queue Length (the average queue length, lower is better), Average Reward (the average reward per timestep, higher is better), and Total Departed (the total number of vehicles that successfully passed through the intersection, higher is better).

F.1 Average Queue Length

Average Queue Length measures the average number of vehicles waiting across all intersection lanes during one episode:

$$\bar{Q} = \frac{1}{T}\sum_{t=1}^{T}\sum_{i\in L} n_i\,(t) \quad (3)$$

F.2 Total Departed

Total Departed measures the total number of vehicles that successfully passed through the intersection within one episode:

$$\eta = N_{\text{departed}} \quad (4)$$

F.3 Average Reward

Average Reward measures the overall quality of the agent's decisions, computed as the average reward per timestep:

$$\bar{R} = \frac{1}{T}\sum_{t=1}^{T} r_t \quad (5)$$

The reward function components that make up the value r_t at each timestep are defined by Equation (2).

F.4 Precision, Recall, and F1-Score

Precision, Recall, and F1-Score are used to measure the detection quality of the Faster R-CNN sensor against ground truth:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (6)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (7)$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (8)$$

where TP is the number of correctly detected vehicles, FP is the number of false detections, and FN is the number of undetected vehicles.

IV. RESULTS AND DISCUSSION

A. Faster R-CNN Sensor Evaluation

The object detection model was assessed using 90 manually annotated ground-truth samples. Faster R-CNN ResNet-50 FPN v2 achieved a precision of 68.65%, a recall of 96.21%, and an F1-score of 80.13%. The high recall indicates that most vehicles were successfully detected, while the lower precision suggests that false-positive detections still occurred. Detection performance also differed across traffic conditions. The afternoon dataset produced a perfect recall of 100% (FN = 0), whereas nighttime conditions proved more difficult, reducing recall to 95.23%.

TABLE III
FASTER R-CNN SENSOR EVALUATION RESULTS (90 SAMPLES)

Condition	Precision	Recall	F1-Score	TP/FP/FN
Morning	0.7785	0.9406	0.8519	935/266/59
Afternoon	0.6083	1.0000	0.7564	705/454/0
Night	0.6611	0.9523	0.7805	519/266/26
Aggregate	**0.6865**	**0.9621**	**0.8013**	**2159/986/85**

This high recall directly affects the quality of the state representation received by the RL model. The DTSE grid produced by Faster R-CNN more densely and accurately reflects actual traffic conditions, enabling the DQN agent to build a more discriminative policy.

B. Training Curve Analysis

Figure 3 through Figure 5 show the training curves of the Faster R-CNN-based DQN over 1,500 episodes. During the first 500 episodes, the model maintained an average reward of ~100–120 with a stable avg queue, indicating a productive exploration phase. Loss remained under control, staying below 55 during this period.

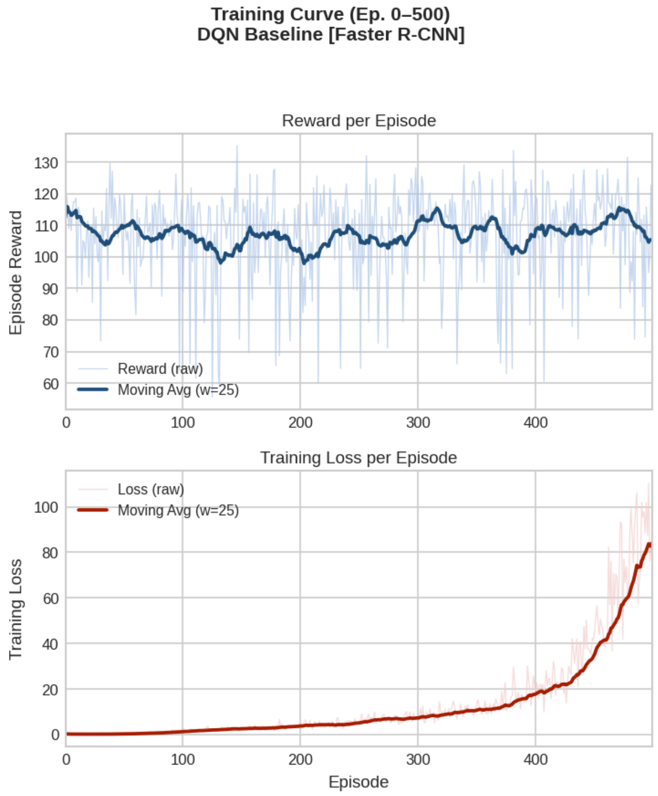

Figure 3. Training Curve of DQN [Faster R-CNN] — Episode 0–500 (Red dashed line = Checkpoint 500ep)

From episode 501–1,000, the model began to show instability. The avg queue experienced a massive spike around episode 900, reaching more than 220 vehicles, accompanied by training loss exploding to above 1,500. This is a sign of uncontrolled gradient divergence resulting from the absence of a regularization mechanism in the architecture.

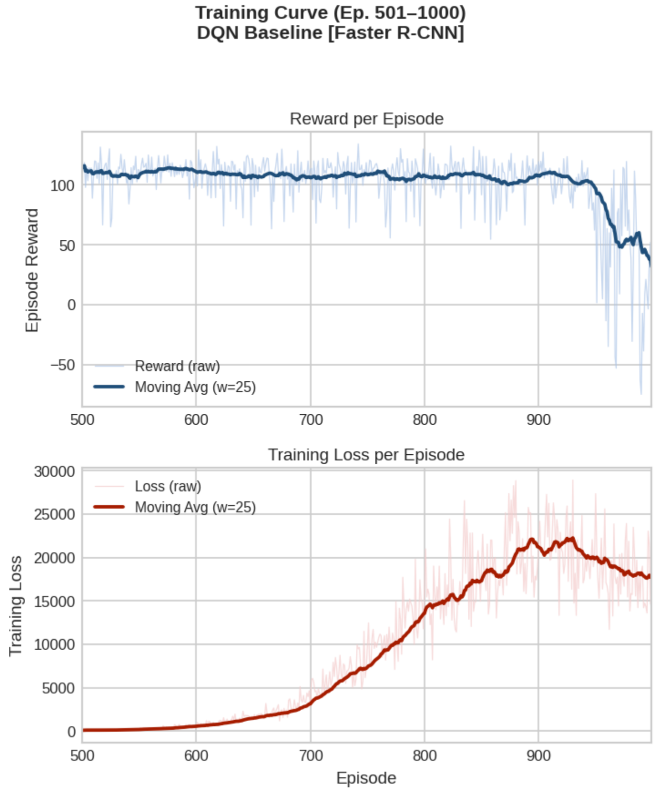

Figure 4. Training Curve of DQN [Faster R-CNN] — Episode 501–1000 (Orange dashed line = Checkpoint 1000ep)

From episode 1,001–1,500, the model underwent total divergence, with loss exceeding 40,000 and an avg queue that remained unstable throughout this training range. This finding confirms that only the 500-episode checkpoint is valid for the Faster R-CNN-based DQN model.

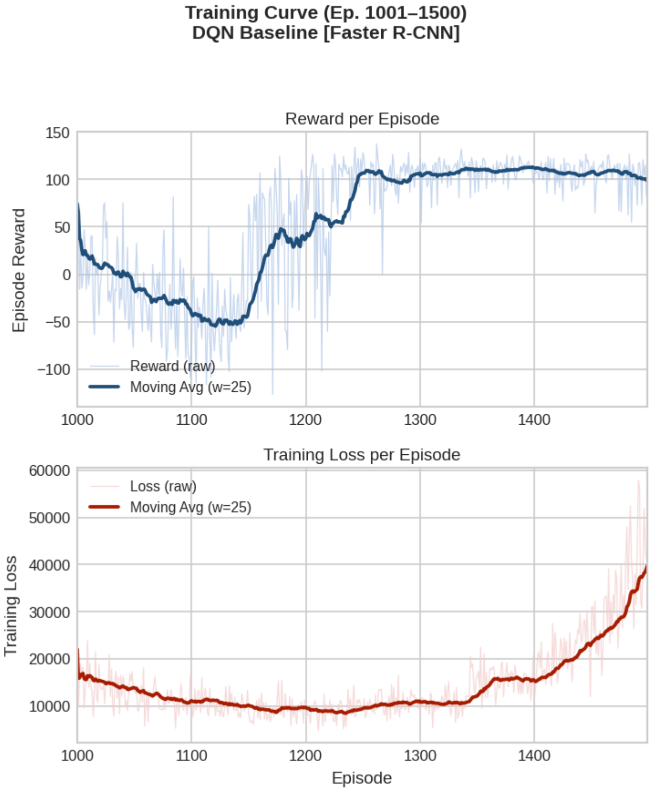

Figure 5. Training Curve of DQN [Faster R-CNN] — Episode 1001–1500 (Purple dashed line = Checkpoint 1500ep)

C. Evaluation per Time Condition

TABLE IV

EVALUATION RESULTS PER TIME CONDITION (DQN [FASTER R-CNN], CHECKPOINT 500 EP)

Condition	Avg Queue ↓	Avg Reward ↑	Total Dep. ↑
Morning	106.59	163.12	982.37
Afternoon	146.39	172.91	1,147.67
Night	176.98	135.70	1,125.60

The afternoon condition produced the highest total departed (1,147.67) with an avg reward of 172.91, indicating that the model is more effective under moderate traffic volume and optimal lighting. The night condition showed both the highest avg queue (176.98) and the lowest avg reward (135.70), reflecting the sensor's detection challenges under low-light conditions, which affect the quality of the state representation.

Figure 6. Performance Visualization per Time Condition: (a) Avg Queue, (b) Avg Reward, (c) Total Departed

D. Aggregate Evaluation and Checkpoint Selection

TABLE V

AGGREGATE EVALUATION — ALL CHECKPOINTS (DQN [FASTER R-CNN])

Checkpoint	Avg Queue ↓	Avg Reward ↑	Total Dep. ↑
500 ep ✓	**127.34**	**161.75**	**1,046.97**

Checkpoint	Avg Queue ↓	Avg Reward ↑	Total Dep. ↑
1,000 ep	129.94	161.21	1,052.77
1,500 ep	127.34	161.75	1,046.97

The final model was selected by considering both evaluation performance and training stability. Although the checkpoints trained for 1,000 and 1,500 episodes achieved evaluation metrics comparable to the 500-episode model, both experienced loss explosion during training, indicating unstable optimization. Therefore, their similar evaluation results should not be interpreted as evidence of successful convergence but rather as a consequence of policy degradation after gradient divergence. Based on these observations, the checkpoint at 500 episodes was selected as the final model. Using this model, the Faster R-CNN-based DQN achieved a total departed value of 1,046.97. This result is likely supported by the detector's high recall of 96.21%, which provides more dependable DTSE observations for the reinforcement learning agent.

E. Implementation Environment Specification

All training and evaluation processes were carried out on the hardware and software listed in Table VI.

TABLE VI
IMPLEMENTATION ENVIRONMENT SPECIFICATION

Component	Specification
Operating System	Windows 11
Processor	Intel Core i5-13450HX
RAM	12 GB
GPU	NVIDIA RTX 3050 6 GB VRAM
CUDA	11.8
Python	3.12
PyTorch	2.7.1+cu118
Ultralytics (YOLOv8)	8.4.52
OpenCV	4.13.0
RL Framework	Custom (PyTorch DQN)

V. CONCLUSION

This study implements and evaluates a Deep Q-Network combined with a Faster R-CNN ResNet-50 FPN v2 sensor for adaptive traffic light control based on real CCTV data. Three main conclusions are drawn.

First, Faster R-CNN, with a recall of 96.21%, produces an accurate and informative DTSE state representation, enabling the DQN model to build a more discriminative control policy than sensors with lower recall. Second, the Faster R-CNN-based DQN achieves optimal performance at the 500-episode checkpoint, with an avg queue of 127.34, avg reward of 161.75, and total departed of 1,046.97 — before undergoing uncontrolled gradient divergence in subsequent episodes. Third, computer-vision sensor quality is shown to act as the upper bound on DQN architecture performance — without an informative state representation from a high-quality sensor, even a DQN architecture trained with a well-designed reward function cannot produce an optimal policy.

ACKNOWLEDGMENT

The author would like to thank the Surabaya City Department of Transportation for providing access to CCTV data through the SITS application for the purposes of this study. The author also wishes to express sincere gratitude to parents, academic supervisor, and friends for their invaluable support throughout this research.

REFERENCES

[1] Badan Pusat Statistik Indonesia, “Statistik Transportasi Darat 2023.” [Daring]. Tersedia pada: https://www.bps.go.id/id/publication/2024/11/25/cdcf9b5e74dd2e9bb3458ee4/statistik-transportasi-darat-2023.html
[2] Badan Pusat Statistik Indonesia, “Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis (Unit), 2022.” [Daring]. Tersedia pada: https://www.bps.go.id/id/statistics-table/2/NTcjMg==/perkembangan-jumlah-kendaraan-bermotor-menurut-jenis--unit-.html
[3] WRI Indonesia, “Mengurai Kompleksitas Urbanisasi dan Pembangunan Kota Berkelanjutan: Mendorong Solusi untuk Masa Depan yang Lebih Hijau.” [Daring]. Tersedia pada: https://wri-indonesia.org/id/wawasan/mengurai-kompleksitas-urbanisasi-dan-pembangunan-kota-berkelanjutan-mendorong-solusi-untuk
[4] Sulistyono, “Kemacetan Kendaraan Pengguna BBM Fosil dan Dampaknya Terhadap Kerugian Ekonomi dan Lingkungan,” *Swara Patra Maj. Ilm. PPSDM Migas*, vol. 12, no. 2, hlm. 12–21, 2022.
[5] Y. Li dan J. Gao, “A Survey on Traffic Signal Control Methods.” 2019.
[6] K. Zhang dan S. Batterman, “Air pollution and health risks due to vehicle traffic,” *Sci. Total Environ.*, vol. 450–451, hlm. 307–316, 2013.
[7] R. S. Sutton dan A. G. Barto, *Reinforcement Learning: An Introduction*, 2 ed. Cambridge, MA: MIT Press, 2018.
[8] L. Zhang, Y. Chen, dan P. Li, “Reinforcement Learning in Urban Network Traffic Signal Control: A Review,” *Expert Syst. Appl.*, vol. 228, hlm. 115000, 2024.
[9] V. Mnih, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, hlm. 529–533, 2015.
[10] T. H. Nguyen dan D. T. Vo, “Improving Traffic Light Systems Using Deep Q-Networks,” *Expert Syst. Appl.*, vol. 253, hlm. 125044, 2024.
[11] S. Park dan S. Han, “Deep Q-network-based traffic signal control models,” *PLOS ONE*, vol. 16, no. 9, hlm. e0256405, 2021.
[12] S. Ren, K. He, R. Girshick, dan J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.
[13] K. He, X. Zhang, S. Ren, dan J. Sun, “Deep residual learning for image recognition,” dalam *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, hlm. 770–778.
[14] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, dan S. Belongie, “Feature pyramid networks for object detection,” dalam *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, hlm. 2117–2125.
[15] R. A. Putra, Y. A. Syahbana, dan Ananda, “Implementasi algoritma Deep Q-Network (DQN) pada lampu lalu lintas adaptif berdasarkan waktu tunggu dan arus kendaraan,” *Indones J Comput Sci*, vol. 13, no. 5, Okt 2024, doi: 10.33022/ijcs.v13i5.4372.