International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(21s) 2026 pp. 1641-1651
https://doi.org/10.70917/ijcisim-2026-5916
Article

IoT Enabled Traffic Signal Monitoring and Control System using Machine Learning

Mr. Shrinivas A. Paivernekar[1], Dr. Shirish V Pattalwar[2], Dr. Ashay I. Rokade[3]

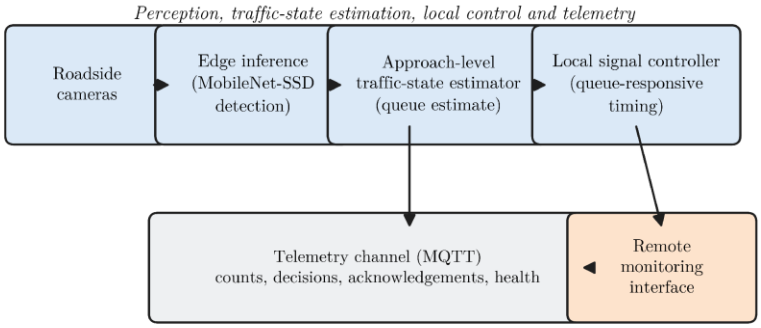
[1]Student, Prof. Ram Meghe Institute of Technology and Research, Badnera, Amravati, India, spaivernekar@gmail.com
[2]Professor, Prof. Ram Meghe Institute of Technology and Research, Badnera, Amravati, India, svpattalwar@mitra.ac.in
[3]Assistant Professor, Prof. Ram Meghe Institute of Technology and Research, Badnera, Amravati, India, airokade@mitra.ac.in

Abstract: The quality of traffic observations and the capacity of a controller to redistribute green time as demand changes are the key factors in adaptive traffic-signal control. This study proposes an Internet of Things (IoT) based traffic-monitoring architecture with MobileNet-SSD vehicle detection, approach-level queue estimation, and bounded queue-responsive signal timing and tests the timing policy in a reproducible four-approach point-queue simulation. The architecture is placed in the context of lightweight vehicle detection and vision-based traffic control, by reviewing ten empirical studies. Thirty paired replications are used for four demand scenarios and three timing policies, resulting in 360 simulation runs. In the case of switching demand, the queue-adaptive policy achieves a 59.41% decrease in mean total queue compared to a single demand-matched fixed-time policy with a demand of scenario-average demand, while the discharge increases by 1.84% compared to the single demand-matched fixed-time policy. The benefit is conditional: it grows by 15.01% in the balanced case, and by 1.27% in the stationary unequal case, but it does not seem to improve in the oversaturated case.
Keywords: Internet of Things; MobileNet-SSD; queue-responsive traffic signal control; vehicle detection; edge computing; point-queue simulation

1. Introduction

The allocation of a limited green-time budget among competing movements is called traffic-signal control. Fixed-time control is based on a fixed plan, while actuated control adjusts services based on the input from the detectors [1]. A fixed plan is good for stable demand but is inefficient if the dominant traffic approach changes. Adaptation is not just about replacing fixed timing, but about reacting to variations that a preselected allocation is not able to cope with sufficiently. This response can be achieved using camera-based monitoring. An operational system needs to be able to correlate detections with approaches, understand them as a time series, forecast demand, and alert others before the forecasted demand goes stale. The controller is then responsible for making admissible timing decisions based on the observations. This leads to two separate evaluation issues: accuracy and speed of perception, and impact of timing decisions on traffic operation. Good performance in one does not guarantee good performance in the other. MobileNet-SSD is a feature extractor that is both economical and one-stage object detection. MobileNet uses depthwise-separable convolutions [2] to lower the feature-extraction cost, and SSD makes predictions of categories and bounding boxes at multiple feature scales [3]. However, the deployment performance is still dependent on the model, data, runtime, hardware, and scene. MobileNet-SSD traffic monitoring [4] and integrated edge-based adaptive control [5] already exist, and a new system should be able to define its accuracy–latency–control trade-off, not just by combining familiar components. This study has three limited contributions. First, it defines a MobileNet-SSD/IoT architecture, which is a separation of the perception, traffic-state estimation, local control and telemetry. Second, it establishes a repeatable bounded queue-responsive timing rule and compares it to the equal-time and scenario-average demand-matched fixed-time baselines. Thirdly, it describes operating regimes in which adaptation is beneficial, neutral or detrimental in balanced, stationary unequal, switching and oversaturated demand.

Copyright: © 2026 by the authors



Architecture Overview. Schematic of the proposed IoT-based traffic-monitoring and control architecture described in this section: roadside cameras and MobileNet-SSD edge inference feed an approach-level traffic-state estimator and a queue-responsive local signal controller, with counts, decisions, acknowledgements and health status reported to a remote monitoring interface over an MQTT telemetry channel.

2. Literature review

Ten empirical publications from 2019 to 2025 related to lightweight vehicle detection, vision-based intersection measurements, or adaptive traffic-signal operation are included in the literature review. It is a targeted narrative review and not a systematic review, so it does not claim to be comprehensive of the field, but rather contextual evidence. Table 1 summarizes the reported numerical findings, but leaves the metric definitions and application context unchanged from each study. The accuracy, precision, average precision, traffic-density accuracy and waiting-time reduction are not considered as equivalent metrics of system quality, but rather as separate outcomes. Initial comparisons set the baseline of the relevance of the speed–accuracy trade-off for MobileNet-SSD. Biswas et al. [4] reported the mean detection accuracy of 92.97% and 79.30% for SSD and MobileNet-SSD respectively on images from 59 traffic cameras. In highway monitoring, Husein et al. [6] achieved mAP scores of 81.9% for YOLOv3 and 67.9% for MobileNet-SSD. These results confirm the need to evaluate the detector for application instead of assuming that a lightweight detector is enough to achieve good recognition quality. This separation is further supported by the results of the TrafficSensor experiments of Fernández et al. [7] which obtained 54.84% mAP at 10 ms with SSD-MobileNet, while the evaluated YOLOv4 configuration obtained 90.56% mAP at 13 ms. In that experiment, a relatively small difference in the latency was accompanied by a significant difference in detection performance. Later research explored the possibility of enhancing lightweight feature extraction without sacrificing its computational benefits. To enhance the performance of the SSD detector, Chen et al. [8] introduced attention and feature fusion to the model, achieving AP scores of 82.59% on BDD100K and 84.83% on KITTI. MobileNetV2 was integrated into an attention mechanism based YOLOv4 model by Zhang et al. [9] and achieved 85.79% mAP at 35 frames per second (FPS) on PASCAL VOC. Bouazizi et al. [10] reported the F1-score of 0.801, mAP of 65.41% and 71 FPS for an SSD-MobileNet road-object application. While these studies show useful configurations, the results do not provide a common-dataset benchmark nor can they be used to rank the papers across the papers. The correlation between detections and traffic-control inputs is also significant. To go beyond isolated detection scores, Shirazi et al. [11] linked computer-vision measurements to SUMO-based intersection analysis. Nocua et al. [12] studied embedded motorcycle monitoring on a Jetson Xavier NX and obtained precision of 0.90, recall of 0.66 and F1 of 0.76 for MobileNet-v1-SSD. Selected results of that study are reproduced in figures 1 and 2. The high precision and low recall rate highlights the need for assessing the detector's performance by taking missed vehicles into account as well as false detections. The latency results also differentiate between runtimes: A model name is not enough to describe the speed of deployment. At the control level, Fahrunnisa et al. [13] integrated the traffic observations obtained from the YOLO with fuzzy signal timing and found that the waiting time was reduced by about 21% and the number of vehicles discharged from the queue was increased by 18% in simulation. Raza et al. [5] combined the detection using YOLO

with the Passenger Car Equivalent (PCE) demand estimation and edge/IoT control, achieving 90% mAP, 74 FPS, and a waiting-time reduction of around 23%. These studies confirm that vision to control integration is a current research direction. The opportunity explored here is more specific: to assess the operating conditions under which a bounded controller designed for a light-weight sensing architecture can enhance traffic service. This means that a credible fixed-time comparator and a separation between detector evidence and controller evidence must be established.

Table 1. Empirical literature on vehicle detection, traffic monitoring, and adaptive control.

Study	**Method and setting**	**Reported numerical findings**	**Interpretation**
Biswas et al. [4] 2019	SSD and MobileNet-SSD; 59 traffic cameras	Detection accuracy: SSD 92.97%; MobileNet-SSD 79.30%	Traffic-density precedent; accuracy is not mAP.
Husein et al. [6] 2020	YOLOv3 and MobileNet-SSD; 42 highway videos	mAP: YOLOv3 81.9%; MobileNet-SSD 67.9%	Within-study comparison; small-vehicle detection remains relevant.
Fernández et al. [7] 2021	TrafficSensor; evaluated detector configurations	SSD-MobileNet: mAP 54.84%, 10 ms; YOLOv4: mAP 90.56%, 13 ms	Model response time excludes the complete control pipeline.
Chen et al. [8] 2022	MobileNetV2-SSD with attention and feature fusion	AP: 82.59% on BDD100K; 84.83% on KITTI	Detector optimisation, not a signal-control evaluation.
Zhang et al. [9] 2022	MobileNetV2-based YOLOv4; attention and feature fusion	PASCAL VOC mAP 85.79%; 35 FPS	Alternative lightweight detector; different task data.
Shirazi et al. [11] 2023	YOLO, tracking, and SUMO intersection analysis	YOLOv5 mean vehicle precision 0.91; recall 0.73	Links video measurements to intersection-level analysis.
Bouazizi et al. [10] 2024	Retrained SSD-MobileNet; road-object / ADAS application	Reported F1 0.801; mAP 65.41%; 71 FPS	Application and evaluation context differ from signal control.
Fahrunnisa et al. [13] 2024	YOLO-based observations and fuzzy signal control	Simulated waiting time ↓ about 21%; queue discharge ↑ about 18%	Direct control precedent; observations are sensitive to occlusion.
Nocua et al. [12] 2025	Embedded motorcycle detection; Jetson Xavier NX	MobileNet-v1-SSD: P 0.90, R 0.66, F1 0.76; optimised response about 10 ms	Single-class task; response time depends on runtime.
Raza et al. [5] 2025	YOLO, PCE-weighted demand, and edge/IoT control	mAP 90%; 74 FPS; waiting time ↓ 23%; congestion ↓ up to 33%	Integrated-system precedent; outcomes use study-specific definitions.

AP: average precision; mAP: mean average precision; FPS: frames per second; P: precision; R: recall. Metrics and experimental settings follow the cited studies.

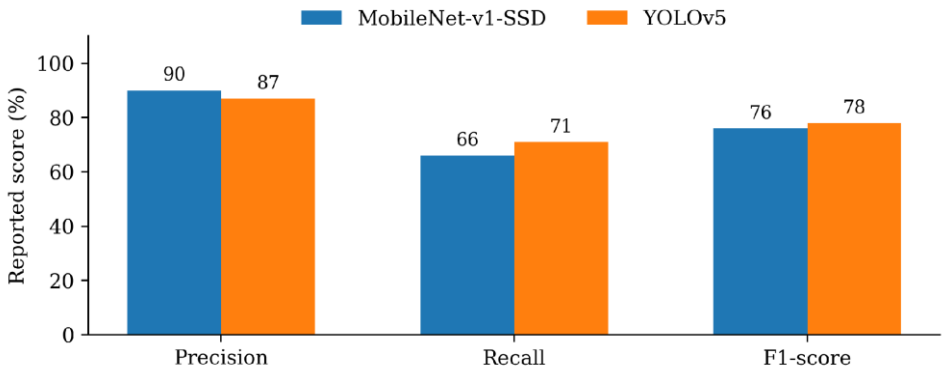


Figure 1. Precision, recall, and F1-score for motorcycle detection, redrawn from Nocua et al. [12], Table 8.

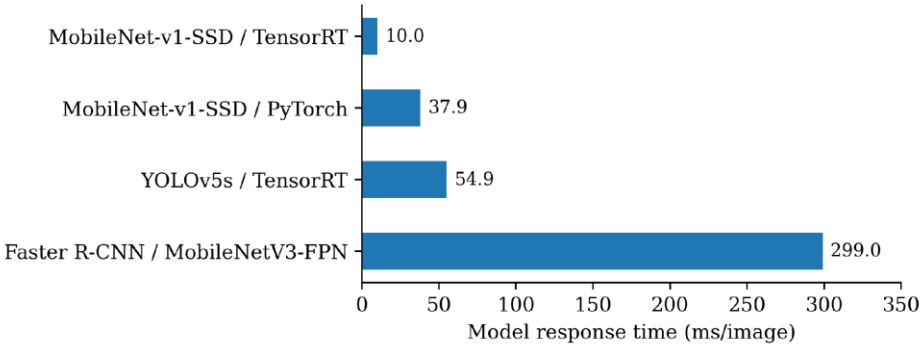


Figure 2. Model response time on Jetson Xavier NX, redrawn from [12], Table 6. Runtime and input resolution differ, with SSD using 300 × 300 inputs and YOLOv5 using 640 × 640 inputs.

3. Methodology

The proposed integration architecture includes roadside cameras, edge inference process, approach-level traffic-state estimator, local signal controller, and remote monitoring interface. Vehicle observations are processed through preprocessing and MobileNet-SSD detection, and then vehicle observations are assigned to the spatial domain and tracked to generate control inputs. These inputs are used by the local controller to allocate green time, while counts, decisions, acknowledgements and health information are sent to the monitoring interface via a separate telemetry channel. The reference perception configuration is a MobileNet-v1-SSD detector with an input of 300 × 300. MobileNet provides the feature representation, and SSD predicts class scores and bounding-box offsets at various scales [2] [3]. The typical SSD training goal is to minimize the classification loss and the localisation loss, both divided by the number of matched positive examples:

$$L = \frac{L_{conf} + \alpha L_{loc}}{N_{pos}}, \quad N_{pos} > 0 \qquad (1)$$

In this case, Lconf and Lloc represent classification loss and localisation loss, respectively, and α is a weight for the localisation term, while Npos is the number of matched positives. The chosen implementation determines how images that don't have any positive matches are treated. Low confidence and redundant predictions are eliminated by confidence filtering and non-maximum suppression. The categories of vehicles supported are dependent on the checkpoint and label map; other categories will need to be labeled with appropriate training data. The present numerical experiment does not include any detector training or inference. A calibrated image region is assigned to each approach. A bottom-centre point of a detection offers a consistent spatial assignment

rule, and temporal association helps to count unique vehicles and identify stopped vehicles. To estimate the queue, a speed threshold and persistence interval must be specified. If not calibrated and temporally validated, the count is not a queue length measurement, but a demand proxy. Likewise, vehicles per frame is not a measure of physical traffic density unless the length of the road observed is known. The controller simulation is based on the actual number of queues, which allows to separate the timing behaviour from errors caused by occlusion, missed detections, duplicate tracks or camera geometry. MQTT is proposed for publish–subscribe telemetry [14]. Messages include intersection and approach identifiers, capture time, sequence number, queue estimate, model identifier and health status. The local controller discards stale or out-of-order data and triggers a pre-defined fallback in case of communication failure. Acknowledgment means that the command has been accepted; the lamp-state verification needs to be given separately. The deployment specification includes authenticated access, encrypted transport, acceptable transitions, and site-specific clearance and pedestrian requirements. They are not tested by the simulator, and the timing values of the simulator are not field prescriptions [1]. The controller implemented is used for four approaches in a fixed non-overlapping phase order. Each phase is followed by a clearance interval ℓ, and a clearance interval of C seconds is reserved, which leaves a green budget $G = C - 4\ell$. The allocation weight wi of approach i is the number of queued aircraft at the start of each cycle. Green times are taken from

$$g_i = min\{g_{max}, g_{min} + \kappa w_i\}, \quad \sum_{i=1}^{4} g_i = G \qquad (2)$$

The green interval is bounded by gmin and gmax and κ is chosen by bisection to meet the budget. Intersections with no vehicles are given equal weights. If not, a small regulariser is added to the weights of zeros to ensure that there is still some residual time available after positive-demand approaches have been reached. The largest-remainder rounding method maintains the budget within two seconds, and ties are broken by the phase order. Each approach is allocated a minimum green time of a positive value, but there is no specific vehicle waiting-time bound. Decisions are made once a cycle, and service does not end as soon as a queue is empty. Equal fixed control is 26 s per approach. Demand matched fixed control allocates the bounded greens in proportion to the known scenario average arrival rates, using the same cycle, limits and rounding. It is not a globally optimal fixed-time plan, but rather it is based on the expected scenario-average demand, not on the realised future arrivals. In the switching scenario, it is still one time-invariant plan that is based on the average of the two demand regimes, but it does not switch plans when the dominant approach changes. The adaptive rule allocates time above a minimum-green, while the demand-matched rule cuts a proportional allocation to the bounds. This distinction is significant when reading the switching-demand comparison, as it is a comparison to a single static plan, not an optimised schedule-aware or actuated plan. The two rules are both defined in the code that accompanies them. The simulation models an isolated intersection as four un-bounded first in first out point queues. The arrival of the independent Poisson processes is spaced 1 second apart. A green approach allows for the discharge of one vehicle every two eligible seconds after a two second start up interval with no discharge during clearance. The timing assumptions, demand patterns and replication design are defined in Table 2. The nominal maximum service is 48 vehicles in 120 s cycle or 1,440 vehicles/h. This is exceeded in the oversaturated scenario on purpose. The model does not include road geometry, turning interactions, pedestrian phases, finite storage, downstream blocking, or network coordination.

Table 2. Simulation configuration and demand scenarios.

Parameter / scenario	**Specification**
Intersection and timing	Four approach phases in fixed order; 1 s time step; 120 s cycle.
Green allocation	104 s total green per cycle; 10–50 s per phase; 2 s allocation increments.
Startup and clearance	2 s startup within each green; 4 s no-service clearance after each phase.

Discharge and capacity	One vehicle per 2 eligible seconds; nominal maximum 1,440 vehicles/h.
Observation and replication	Exact queue counts; no network model; 600 s warmup + 3,600 s measurement; 30 paired seeds.
Balanced demand	1,080 vehicles/h; approach shares (0.25, 0.25, 0.25, 0.25).
Stationary unequal demand	1,080 vehicles/h; approach shares (0.55, 0.20, 0.15, 0.10).
Switching demand	1,224 vehicles/h; alternates every 900 s between (0.55, 0.20, 0.15, 0.10) and (0.20, 0.55, 0.10, 0.15).
Oversaturated demand	1,656 vehicles/h; equal approach shares; demand exceeds nominal service.

Thirty independent random seeds are used for each scenario. All three policies have the same realised arrivals for a given scenario–seed pair, resulting in 360 runs and paired comparisons. There is a 600 s warmup followed by a 3,600 s measurement window. Switching demand alternates between two approach-share patterns, with the first pattern used for warmup. The random generator is NumPy default_rng(SeedSequence([20260905, scenario_index, seed])). The archived environment contains Python 3.13.5, NumPy 2.3.5, and pandas 2.2.3. The integrated-system evaluation separates the perception quality, processing latency and traffic operation. Precision, recall and F1 are calculated based on the one-to-one matching of predicted boxes and reference boxes of the same class at a specified intersection-over-union (IoU) threshold:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F_1 = \frac{2TP}{2TP + FP + FN} \qquad (3)$$

TP, FP and FN represent true positive, false positive and false negative, respectively. The averaging convention is crucial, as is per-class values, and macro-F1 is the average of class-level F1 values, which is not necessarily the harmonic average of macro-precision and macro-recall. The average precision is calculated over the confidence thresholds, and mAP@0.50 and mAP@0.50:0.95 are reported separately under the selected evaluator [15], [16]. A valid detector experiment also splits the recording sessions or cameras into training, validation and test sets, so that there is no leakage between adjacent video frames. Confidence thresholds are chosen from validation data and set prior to testing.

Inference latency is not the whole control loop, but only the time taken to execute the model. It needs a declared device, input resolution, batch size, runtime, precision format, warmup and synchronisation procedure to be evaluated. Mean, median and 95th percentile latency are different measures of responsiveness. Capture, preprocessing, tracking, transmission, decision and acknowledgement are also part of end-to-end latency. These detector and latency measures are not estimated from the point-queue simulation, but are used in the following experiment with the integrated system.

The main result of the performed experiment is the horizon-limited queueing delay. Let av be the arrival time of the vehicle v that arrives during the measurement window and let dv be its departure time and let H be the end of the observation window. The measure is

$$W_H = \frac{1}{N_A} \sum_{v \in A} [min(d_v, H) - a_v] \qquad (4)$$

where A is the NA measurement-window arrivals and dv is assumed to be infinite for unfinished vehicles. So, all vehicles arriving contribute, including those still in the queue at H, but not waiting after the horizon. This is a limited queue-system service and not a completed-trip travel time. The outcomes of time-average total queue, discharge during the measurement hour, and terminal queue are complementary. The latter include warmup carryover where applicable, and so their populations are not necessarily the same as the arrival cohort used for restricted delay. Thirty replications are used to compute means. The 95% confidence intervals are based on a Student-t distribution with 29 degrees of freedom and policy contrasts are based on within-seed differences. Relative changes are based on the unrounded group means. These exploratory intervals are not adjusted for multiple comparisons, and represent simulation sampling uncertainty, not uncertainty in transfer to a real intersection.

4. Results and Comparative Discussion

The numerical results for all the demand scenarios and control policies are presented in Table 3. The overall trend is not necessarily positive for adaptation. The benefit of queue-responsive timing is significant when compared to the single scenario-average fixed plan used here, but is not superior in all operating conditions. This stronger of the two fixed-time baselines available in the current experiment is the focus of figures 3 and 4, but the switching baseline is not schedule-aware. To illustrate the difference in interpretation, the equal-time results are included in Table 3.

Table 3. Traffic-control outcomes averaged over thirty replications per scenario. All values are from the synthetic point-queue simulation.

Demand scenario	Controller	Delay (s/arrival)	Mean queue (vehicles)	Discharge (vehicles/h)	Final queue (vehicles)
Balanced	Equal fixed	53.67	16.09	1,086.50	17.57
	Demand-matched fixed	53.67	16.09	1,086.50	17.57
	Queue-adaptive	61.73	18.58	1,086.83	17.83
Stationary unequal	Equal fixed	545.55	166.16	840.53	291.60
	Demand-matched fixed	48.30	14.36	1,078.63	18.67
	Queue-adaptive	48.92	14.55	1,079.07	18.90
Switching demand	Equal fixed	539.20	187.53	994.43	295.30
	Demand-matched fixed	163.87	56.51	1,198.07	67.07
	Queue-adaptive	67.59	22.94	1,220.10	28.20
Oversaturated	Equal fixed	370.13	172.03	1,437.93	282.03
	Demand-matched fixed	370.13	172.03	1,437.93	282.03
	Queue-adaptive	368.76	171.31	1,439.87	280.87

The adaptive control is able to achieve a 58.75% reduction in restricted delay (163.87 s per arrival vs. 67.59 s per arrival) compared to the single demand−matched fixed plan using the scenario−average demand. The average queue length drops from 56.51 to 22.94 vehicles and from 67.07 to 28.20 vehicles in the terminal queue. Discharge increases are less pronounced, from 1,198.07 to 1,220.10 vehicles/h, suggesting that the primary advantage is a reduction in the length of the queues. This is the biggest effect in the study but not to be interpreted as being better than an optimised time−of−day, actuated or other schedule−aware controller, as these were not tested. The paired delay difference is −96.28 s per arrival, with a 95% interval from −106.62 to −85.94 s (Table 4). The opposite is true for balanced demand: adaptive delay is 61.73 s per arrival, while the fixed plan is 53.67 s per arrival, a 15.01% increase. Addressing the randomness of queues at a short time scale can cause a reallocation that is not aligned with the symmetric demand.A response to the

short-term randomness in queues can disrupt an allocation that is already aligned with the symmetric demand. The temporary imbalance can affect the entire allocation, as it can be done with once-per-cycle updates, but this would need experiments that change the update frequency and smoothing. The importance of the baseline is demonstrated in stationary unequal demand. If the timing is equal, then 545.55 s per arrival will be produced since the dominant approach is not getting enough service. This can be decreased to 48.30 s, which is very close to the adaptive value of 48.92 s, by using fixed timing, but adjusting the timing to match the demand. The paired difference is +0.62 s, with an interval from +0.18 to +1.05 s. The big advantage over equal timing is therefore a correction of an allocation imbalance, rather than an advantage over a suitable fixed plan. With oversaturation both policies converge to the service ceiling with significant terminal queues. The restricted delay is adaptive with a value of 368.76 s per arrival, and fixed with a value of 370.13 s per arrival. The paired interval includes zero, showing no clear delay advantage. The arrival-service imbalance inherent in this situation cannot be eliminated through green-time redistribution, so near-capacity discharge is not a sign of congestion being resolved.

Table 4. Paired restricted-delay contrasts relative to demand-matched fixed control. Negative values favour adaptation.

Demand scenario	Adaptive − fixed (s/arrival)	95% confidence interval (s/arrival)	Relative change of means
Balanced	+8.06	+7.43 to +8.68	+15.01%
Stationary unequal	+0.62	+0.18 to +1.05	+1.27%
Switching demand	−96.28	−106.62 to −85.94	−58.75%
Oversaturated	−1.38	−6.13 to +3.38	−0.37%

Differences and percentages are calculated from unrounded values. Confidence intervals are exploratory and unadjusted for multiple comparisons.

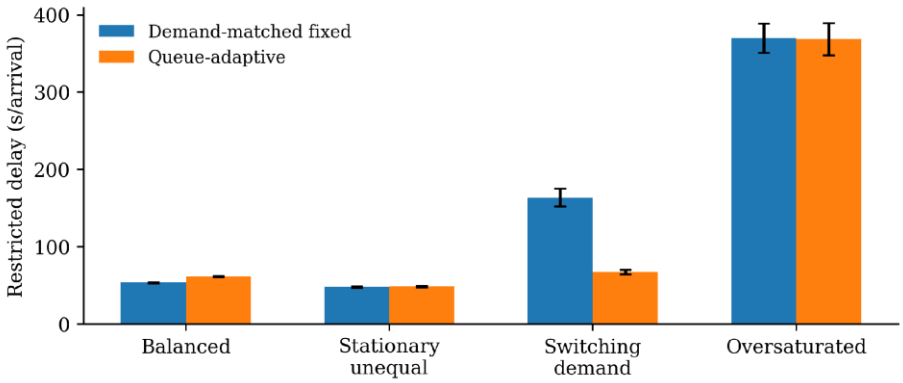


Figure 3. Restricted queueing delay for the adaptive and demand-matched fixed policies in the synthetic simulation. Bars show means over thirty paired replications; error bars show 95% confidence intervals.

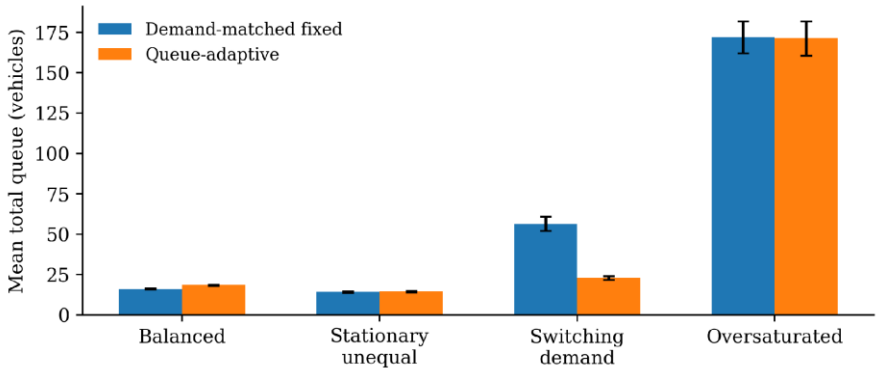


Figure 4. Time-average total queue under the same simulation conditions. Error bars show 95% confidence intervals. The equal-time baseline is retained in Table 3 but omitted here for clarity.

The same distinction between perception and control is needed when comparing with the literature. The precision, recall, F1, and latency results reported by Nocua et al. [12] for embedded monitoring are summarised in figures 1 and 2. Similarly, the waiting-time enhancements reported by Fahrunnisa et al. [13] and Raza et al. [5] are due to various demand conditions, controllers, baselines, and outcome definitions. The present 58.75% reduction in restricted delay should be considered in the context of the present experiment. The qualitative comparison is that these studies look at demand-responsive service, whereas the present experiment looks at both good and bad regimes compared to a demand-matched regime. The detector results still point to a practical problem for integration. A high precision perception module might underestimate a crowded approach, and duplicated tracks might allocate green time for a demand that does not exist. Stale observations add another mismatch between observed and current conditions. These are possible failure modes and not measured effects in this study. The importance of their can be assessed by substituting the exact simulated queues with camera-derived estimates of queues, while keeping the controller fixed, and then testing with measured communication delays. Meaningful detector comparisons require common held-out scenes, class mappings, matching rules, hardware, and runtime disclosures [11, 12]. The demand-matched fixed controller does not use a globally optimised plan, but rather a plan based on the average demand of the scenarios, and in the switching scenario, it does not use separate time-of-day plans for the two known scenarios. The current experiment is thus not able to conclude if the adaptive rule is better than a well-tuned schedule-aware or actuated controller. The point-queue model does not include platooning, heterogeneous vehicle behaviour, spillback, geometry, turning interactions, pedestrian phases or downstream interactions; the four demand patterns are exploratory and not representative of a road network. The results are further limited by the fixed cycle, phase order, timing quantisation and warmup duration. The restricted delay is censored at the terminal horizon, especially in oversaturation interpretation.

Figures 5–8 give a full graphical representation of Table 3, showing all three controllers across all four demand scenarios for each outcome measure in turn, and Figure 9 gives a graphical representation of the paired contrasts of Table 4.

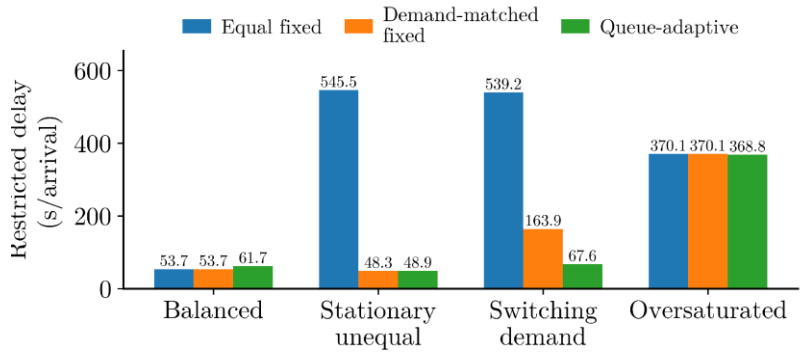


Figure 5. Restricted delay (s/arrival) by demand scenario for equal-fixed, demand-matched fixed, and queue-adaptive control, redrawn from Table 3.

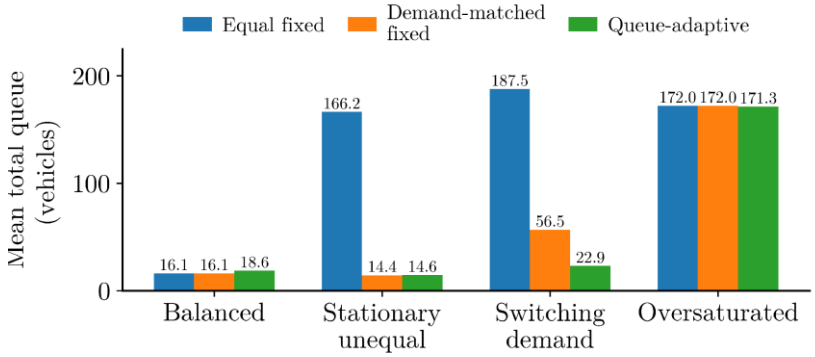


Figure 6. Mean total queue (vehicles) by demand scenario for equal-fixed, demand-matched fixed, and queue-adaptive control, redrawn from Table 3.

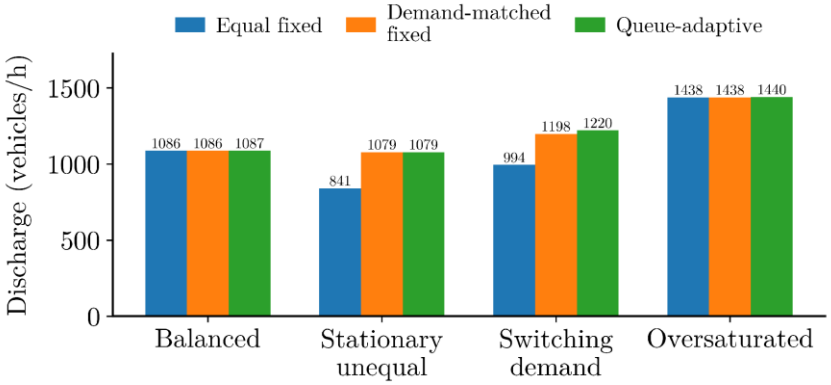


Figure 7. Discharge (vehicles/h) by demand scenario for equal-fixed, demand-matched fixed, and queue-adaptive control, redrawn from Table 3.

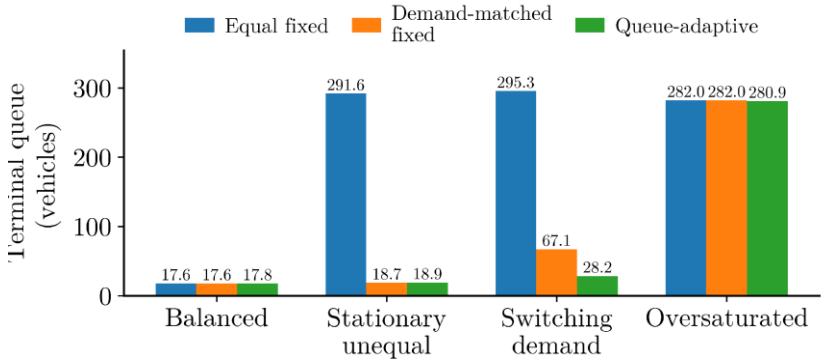


Figure 8. Final (terminal) queue (vehicles) by demand scenario for equal-fixed, demand-matched fixed, and queue-adaptive control, redrawn from Table 3.

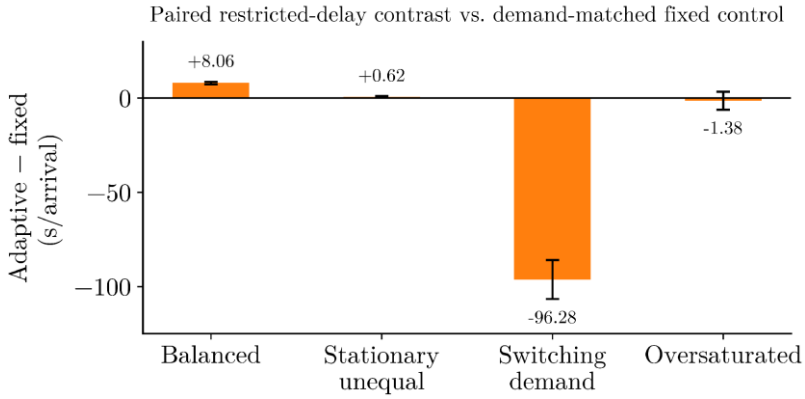


Figure 9. Paired restricted-delay contrast (adaptive − fixed) between queue-adaptive and demand-matched fixed control, by demand scenario, with 95% confidence intervals, redrawn from Table 4.

5. Conclusion

This study specifies an IoT-enabled traffic-monitoring and control architecture centred on MobileNet-SSD and evaluates its bounded queue-responsive timing policy in a reproducible preliminary point-queue simulation. The literature establishes both lightweight detection precedents and existing vision-to-control integration, making the relevant contribution a transparent evaluation of controller operating regimes and integration requirements rather than novelty from combining familiar components.

Across thirty paired replications, queue-responsive timing reduces restricted delay by 58.75% and mean total queue by 59.41% under switching demand relative to the single scenario-average demand-matched fixed-time plan used in the experiment. However, delay increases by 15.01% under balanced demand and by 1.27% under stationary unequal demand, while oversaturation yields no clear advantage. These findings support selective adaptation to changing demand within the simplified simulation and demonstrate the importance of comparator choice. Future work can extend the evaluation with stronger control baselines, perception and latency sensitivity tests, and camera-based experiments.

REFERENCES

1. P. Koonce et al., *Traffic Signal Timing Manual*, Federal Highway Administration, report FHWA-HOP-08-024, 2008, ch. 5.

2. A. G. Howard et al., “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” arXiv:1704.04861, 2017. Preprint.
3. W. Liu et al., “SSD: Single Shot MultiBox Detector,” in *Computer Vision – ECCV 2016*, LNCS 9905, pp. 21–37, 2016. DOI: 10.1007/978-3-319-46448-0_2.
4. D. Biswas, H. Su, C. Wang, A. Stevanovic, and W. Wang, “An automatic traffic density estimation using Single Shot Detection (SSD) and MobileNet-SSD,” *Physics and Chemistry of the Earth, Parts A/B/C*, vol. 110, pp. 176–184, 2019. DOI: 10.1016/j.pce.2018.12.001.
5. M. Raza, M. Kazmi, H. M. Kidwai, H. R. Khan, S. A. Qazi, K. Arshad, and K. Assaleh, “An Edge-Deployed Real-Time Adaptive Traffic Light Control System Using YOLO-Based Vehicle Detection and PCE-Aware Density Estimation,” *IEEE Access*, vol. 13, pp. 153586–153613, 2025. DOI: 10.1109/ACCESS.2025.3602844.
6. A. M. Husein, C. Christopher, A. Gracia, R. Brandlee, and M. H. Hasibuan, “Deep Neural Networks Approach for Monitoring Vehicles on the Highway,” *Sinkron*, vol. 4, no. 2, pp. 163–171, 2020. DOI: 10.33395/sinkron.v4i2.10553.
7. J. Fernández, J. M. Cañas, V. Fernández, and S. Paniego, “Robust Real-Time Traffic Surveillance with Deep Learning,” *Computational Intelligence and Neuroscience*, vol. 2021, article 4632353, 2021. DOI: 10.1155/2021/4632353.
8. Z. Chen, H. Guo, J. Yang, H. Jiao, Z. Feng, L. Chen, and T. Gao, “Fast vehicle detection algorithm in traffic scene based on improved SSD,” *Measurement*, vol. 201, article 111655, 2022. DOI: 10.1016/j.measurement.2022.111655.
9. Y. Zhang, A. Zhou, F. Zhao, and H. Wu, “A Lightweight Vehicle-Pedestrian Detection Algorithm Based on Attention Mechanism in Traffic Scenarios,” *Sensors*, vol. 22, no. 21, article 8480, 2022. DOI: 10.3390/s22218480.
10. O. Bouazizi, C. Azroumahli, A. El Mourabit, and M. Oussouaddi, “Road Object Detection using SSD-MobileNet Algorithm: Case Study for Real-Time ADAS Applications,” *Journal of Robotics and Control*, vol. 5, no. 2, pp. 551–560, 2024. DOI: 10.18196/jrc.v5i2.21145.
11. M. S. Shirazi, B. T. Morris, and S. Zhang, “Intersection analysis using computer vision techniques with SUMO,” *Intelligent Transportation Infrastructure*, vol. 2, article liad003, 2023. DOI: 10.1093/iti/liad003.
12. F. Nocua M., W.-J. Pérez-Holguín, and C. Pardo-Beainy, “Urban traffic monitoring based on deep learning on an embedded GPU,” *Expert Systems with Applications*, vol. 273, article 126847, 2025. DOI: 10.1016/j.eswa.2025.126847.
13. Z. Fahrunnisa, Rahmadwati, and R. A. 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*, vol. 10, no. 2, pp. 235–251, 2024. DOI: 10.26555/jiteki.v10i2.28712.
14. OASIS, *MQTT Version 5.0*, OASIS Standard, 7 March 2019.
15. T.-Y. Lin et al., “Microsoft COCO: Common Objects in Context,” in *Computer Vision – ECCV 2014*, LNCS 8693, pp. 740–755, 2014. DOI: 10.1007/978-3-319-10602-1_48.
16. COCO Consortium, COCO API: pycocotools/cocoeval.py, official evaluation implementation, accessed 5 September 2026. Repository path: cocodataset/cocoapi/PythonAPI/pycocotools/cocoeval.py.