Dynamic Time Allocation for Traffic Controller Using IoT, Deep Learning and Reinforcement Learning

April Castro
Caraga State University

Mary Rose O. Raz
Caraga State University

Lyn M. Villacorta
Caraga State University

Paulo A. Duga
Caraga State University

他

https://doi.org/10.5109/7395599

出版情報：Proceedings of International Exchange and Innovation Conference on Engineering & Sciences (IEICES). 11, pp.762-769, 2025-10-30. International Exchange and Innovation Conference on Engineering & Sciences
バージョン：

Dynamic Time Allocation for Traffic Controller Using IoT, Deep Learning and Reinforcement Learning

April Castro[1], Mary Rose O. Raz[2], Lyn M. Villacorta[3], Paulo A. Duga[4], Jundee Mark Molina[5], James Earl Cubillas[6], Rudolph Joshua U. Candare[7], Junrie B. Matias[8]
[1]Caraga State University, [2]Caraga State University, [3]Caraga State University, [4]Caraga State University, [5]Caraga State University, [6]Caraga State University, [7]Caraga State University
Corresponding author email: jbmatias@carsu.edu.ph

Abstract: *Urbanization in the Philippines presents challenges such as traffic congestion and road safety, particularly with the increasing number of vehicles on the road. This study proposes an intelligent traffic management system utilizing IoT and Convolutional Neural Networks (CNNs). Specifically, YOLO models were adapted to detect local vehicle types unique to Butuan City. A hybrid YOLO classifier and DeepSORT are integrated for real-time vehicle detection, classification, and tracking, with transfer learning on local datasets to enhance model precision. To dynamically control traffic signals, the system incorporates multiple deep reinforcement learning (RL) methods, including Deep Q-Learning (DQL), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), and Advantage Actor-Critic (A2C). The RL-based controller learns and adapts real-time traffic signal timing based on vehicle density, waiting times, and phase duration at intersections. Among the methods, A2C demonstrated the highest efficiency, significantly reducing vehicle waiting times and enhancing traffic throughput. Comparisons showed that both DQL and CMA-ES provided robust performance, each contributing unique advantages in different traffic scenarios, though A2C emerged as the optimal solution in simulations. This comprehensive approach highlights the potential of combining IoT, CNN-based vehicle detection, and adaptive RL controllers, offering a scalable solution to improve urban traffic efficiency and support sustainable city planning initiatives.*

Keywords: Dynamic Time Allocation; Traffic Controller; IOT; Transfer Learning; YOLO.

1. INTRODUCTION

Urbanization in the Philippines results in many opportunities that serve as engines of growth if cities are adequately planned and managed [1]. However, modernized society poses significant challenges for city authorities [2], [3], including increased traffic collisions, road congestion, and road traffic emissions due to rising vehicle numbers [4], [5], [6]. Implementing intelligent traffic systems becomes crucial in such scenarios to alleviate urban traffic congestion. Intelligent traffic systems play a crucial role in urban planning development and every city's growth [7]. Approaches and technologies like the application of artificial intelligence and internet-of-things will help systems be more analytical, prescriptive, and self-driven, contributing to the efficiency of traffic systems [8]. Modern traffic management systems rely heavily on vision-based methods, specifically Convolutional Neural Networks (CNNs) and their variants, for robust object detection, precise localization, and accurate classification tasks [9]. You Only Look Once (YOLO) is one of the excellent object detection methods that have emerged and are often used in vehicle detection [10], [11]. YOLO family performs very well in detecting objects[12], [13] and is paired with Deep SORT for vehicle tracking as it performed well in tracking precision and accuracy [14].
However, the implementation of intelligent and dynamic timed traffic systems deployed in road intersections utilizing vision-based methods can only be seen mostly in foreign countries [15], [16], [17] and commonly used datasets from MS-COCO, OpenImage, PASCAL VOC 12, ILSVRC [18], KITTI [19], and ImageNet [20].
Moreover, Reinforcement Learning (RL) has proven to be effective in numerous applications, particularly in the domain of artificial intelligence [21] and optimal traffic signal control and the study shows that an RL-based approach can operate in a traffic-responsive manner [22]. An RL-based model in [23] can dynamically adapt traffic flow data in real time, boosting the flexibility and efficiency of traffic signal control and successfuly alleviating traffic congestion. RL-based approach to traffic management systems has been quite effective in determining optimal traffic light configuration and synchronization of multiple traffic light systems at adjacent intersections [24], [25]. Yet, a complex traffic situation in real cases could lead to inferior traffic adjusting performance [26].
Reinforcement Learning (RL) involves an agent learning to make sequential decisions by interacting with its environment and receiving rewards based on its actions. It is effective for learning optimal strategies over time. Deep Learning (DL) focuses on extracting complex patterns from raw data by building hierarchical representations. When combined with RL, it forms Deep Reinforcement Learning (Deep RL), where deep neural networks approximate key RL components like value functions or policies. Deep RL has been successfully applied in diverse fields, such as biological data analysis, communications, robotics, power systems, industrial processes, and transportation [27], [28]. In transportation, Deep RL is used to optimize traffic management by dynamically adjusting signal timings based on real-time data, aiming to improve efficiency and reduce congestion.
However, publicly available DL and RL models cannot be directly applied in specific locales due to contextual differences. These models are typically trained on generalized datasets and may not account for unique local

factors, such as environmental, social, or economic conditions. For example, a model developed using data from urban areas may not perform well in rural or developing regions due to differences in infrastructure, resources, or data distribution. Additionally, these models often assume uniform global or regional conditions, which may not align with local cultural, legal, or societal needs, reducing their accuracy and relevance. Butuan City, Philippines, currently lacks an intelligent traffic management system utilizing local datasets. Implementing pre-trained models directly is impractical due to variations in vehicle types not accounted for in existing studies, which predominantly use different datasets. Local datasets include vehicles specific to Butuan City, such as jeepneys, buses, tricycles, vans, motorcycles, and more—vehicles that are absent from standard pre-trained models. This disparity underscores the need for developing customized models trained on locally relevant data to effectively address transportation challenges specific to the city.

Consequently, this study proposes the development of a vision-based sensor system for dynamically adjusting traffic signals using IoT and Convolutional Neural Networks (CNNs). Each IoT device will be equipped with a camera utilizing a hybrid YOLO classifier for vehicle detection and classification, complemented by DeepSORT for accurate vehicle counting and tracking. To ensure operational efficiency on small devices like Jetson Nano, YOLOv3, YOLOv4, and YOLOv5 models will be trained using transfer learning techniques with local datasets and evaluated using standard criteria.

Furthermore, this study will use deep reinforcement learning to create a system that dynamically adjusts traffic light timing based on real-time traffic conditions at intersections. To address the challenges of applying publicly available DL and RL models, the models will be fine-tuned or retrained using localized data and will be designed to comply with local traffic rules and regulations.

Lastly, the traffic data captured by the sensors will be transmitted to a lightweight device housing a deep reinforcement learning-based controller. This controller will utilize algorithms such as Covariance Matrix Adaptation Evolution Strategy, Advantage Actor-Critic (A2C), and Deep Q-learning. This adaptive approach allows the system to dynamically adjust signal timings in real-time, aiming to maximize traffic flow efficiency and minimize congestion. The integration of deep reinforcement learning into urban traffic management systems represents a cutting-edge approach to improving city infrastructure and enhancing transportation efficiency.

2. METHODOLOGY

2.1 Dataset

The datasets were obtained by video recording vehicles at Butuan City's intersections. Vehicle types were classified as shown in Table 1.

Table 1. Class Indexation

Index	Class	Index	Class
0	Scrum	*15*	Fire Truck
1	Sedan	*16*	Ambulance
2	CUV	*17*	Fuel Truck
3	SUV	*18*	Box Truck
4	Bus	*19*	Garbage Truck
5	Hatchback	*20*	Backhoe
6	Multicab-Pickup	*21*	Flatbed
7	Pickup	*22*	Motorcycle
8	Van	*23*	Electric bus
9	Coupe	*24*	L-Tricycle
10	Mini-Truck	*25*	Bike
11	Mini-Van	*26*	Shuttle
12	Truck	*27*	PUJ
13	Dump Truck	*28*	C-Tricycle
14	Big Truck		

The collected videos were converted into images, extracted using the VLC screenshot function, and manually checked and sorted out images having vehicles only used for training and annotated using LabelImg based on the custom classes. In data cleaning, images must correspond to the number of labels being present in the folder, if something is missing, it must be traced before splitting the data.

The total number of annotated data was 5,386, a purely local dataset from Butuan City with a camera angle of 45° and 90° and stationed with a 14ft high stand above the ground containing 29 unique vehicle types. The 428 images were occluded and used in testing. The video streams of the cameras, provide a stable streaming of 30 frames per second, supporting a resolution of 2592x1944 pixels.

A camera with a resolution of 2592x1944 was utilized due to the limited availability of alternative camera options. Nevertheless, this resolution of 2592x1944 still provides valuable benefits for object detection due to high-quality images.

2.2 Models for Detection and Tracking of Vehicles

A hybrid implementation of vehicle detection and tracking was used to detect, classify, track, and count the vehicles in every lane. Thus, this study evaluates three YOLO variants, namely, YOLOv3, YOLOv4, and YOLOv5s. These models were chosen since they are state-of-the-art and in terms of architecture. These models are lightweight and can be deployed in smaller devices such as Jetson Nano and RasPi.

- **The YOLO Classifier:** Each version of YOLO introduces improvements from its predecessor in terms of accuracy, speed, and feature extraction capabilities. YOLOv4 builds upon YOLOv3 by incorporating state-of-the-art techniques and architectural changes, while YOLOv5 is a separate implementation that offers further improvements and is implemented in PyTorch for greater accessibility. The main differences between YOLOv3, YOLOv4, and YOLOv5 lie in their architecture and feature enhancements. YOLOv3 uses the Darknet-53 backbone and introduces multi-scale predictions for small object detection, alongside anchor boxes and skip connections for better predictions. YOLOv4 builds upon YOLOv3 with an upgraded CSPDarknet53 backbone, incorporating CSPNet for improved information flow, state-of-the-art techniques (BoF, BoS) for accuracy and speed, and PANet for feature aggregation along with SPP for spatial information. YOLOv5, an unofficial separate implementation, features a custom CSPNet-based backbone and multiple model sizes for trade-offs. It also includes automatic anchor box generation and mosaic data augmentation and is implemented in PyTorch for greater accessibility.

- **Deep SORT:** The Deep SORT was released in 2017. It is a combination of a deep learning-based appearance model with Simple Online and real-time tracking. It is a powerful tool for tracking objects in real-time since it incorporates a Kalman filter to predict the motions of objects that handle occlusions and missed detections. The combination of deep learning-based appearance modelling and the Kalman filter, allows the algorithm to track objects in a video accurately. This study developed a hybridized model that encompasses both end-to-end object detection and tracking pipelines. First, the YOLO classifier was used to detect objects in a video frame, providing the initial set of detections. Then, Deep SORT takes these detections and associates them over time to track the movement of objects. This hybridized model improves accuracy and robustness due to the combination of detection and tracking algorithms, leading to more reliable results.
- **Transfer Learning:** This work used transfer learning through freezing layers that use the base model as a feature extractor. By freezing the layers, the pre-trained weight of YOLOv5s is preserved, allowing the model to use the features learned by the pre-trained YOLOv5s model while preventing updates during training. Since the local datasets were highly different from the datasets being trained in the pre-trained model, freezing the lower layers of the model was the best approach to train the new datasets. The researchers employed transfer learning for its remarkable efficiency in object detection, which conserves time and resources, enhances performance, and offers a solid foundation for this study. By capitalizing on pre-trained models, the resulting model can deliver superior performance with limited training data, making it adaptable to various datasets and applications tailored to local vehicle classifications in Butuan City and nearby areas. Furthermore, this approach contributes to reduced training time, improved accuracy with small datasets, and the ability to generalize to new data.

2.3 Deep Reinforcement Learning-based Traffic Controller

To identify and implement an effective traffic control strategy, this study conducted a comprehensive evaluation of various reinforcement learning (RL) methods, each offering distinct advantages for dynamic traffic management. The following RL techniques were tested and assessed for their suitability in optimizing traffic flow at intersections, adjusting signal timings in response to real-time conditions, and minimizing vehicle wait times [29]:

- **Deep Q-Learning (DQL):** combines Q-Learning with deep neural networks to learn optimal policies in complex, high-dimensional environments. Unlike traditional Q-learning, DQL uses a deep neural network, often a convolutional neural network (CNN), to approximate the Q-function, predicting expected future rewards for each action in a given state. Through interaction with the environment, the agent updates Q-values based on received rewards and future reward estimates [30], [31].
- **Covariance Matrix Adaptation Evolution Strategy (CMA-ES):** a gradient-free optimization method that uses random variations to optimize real-valued parameters relative to an objective function [32]. It generates a covariance matrix to model parameter distributions, sampling from a multivariate normal distribution based on the mean and covariance of top-performing candidates, similar to genetic algorithms. Unlike crossover-based methods, CMA-ES samples "children" directly from this distribution, aiming to minimize the objective function while maintaining consistent performance across linear transformations [33], [34].
- **Advantage Actor-Critic (A2C):** a method that combines policy-based and value-based approaches. While policy-based agents determine actions based on probability distributions, value-based algorithms select actions based on state or action values. A2C improves training stability by using an advantage function to update weights, which assesses an action's benefit over the average for a given state. The advantage function is often approximated with estimators like temporal-difference error for efficient computation [35], [36].

2.4 Training

The object detector and classifier were trained using a total dataset of 4,958 images, excluding 428 occluded datasets. These were divided into 80% for training (3,967 images) and 20% for validation (991 images). The models were trained on hardware consisting of an Intel Xeon 216B Workstation running Windows 10, equipped with a GTX 1060 GPU. Python 3 in the Anaconda environment was used for coding, leveraging the CUDA v10 Toolkit for efficient GPU acceleration.

To train the RL-based controller, we used a Dell T640 HPC with four Tesla V100 GPUs, leveraging PyTorch as the deep learning framework. SUMO-Webots provided realistic traffic simulation, while Deepbots acted as a middleware to interface Webots with PyTorch for reinforcement learning. This setup enabled effective training of RL models to optimize traffic management strategies.

Additionally, the setup for reinforcement learning involves three main components: State (observations), Action, and Reward. The State includes Phase ID (one-hot encoded to denote the current green phase), Lane Waiting Time (cumulative waits per lane), and Lane Density (occupancy near the intersection). Phase ID specifies active traffic lanes, Lane Waiting Time indicates congestion levels, and Lane Density shows lane occupancy. These inputs guide the Neural Network Policy. Actions for the RL-based Traffic Control Agent include setting green-phase durations (0, 15, 30, 45 seconds) and potentially skipping phases to prioritize lanes. Rewards are based on total vehicle waiting time per phase: 100 for no waiting time, or 100 divided by the total waiting time otherwise. This system encourages the agent to minimize congestion and optimize traffic flow effectively using real-time data from a State

Representation Model that tracks vehicles in specific road segments.

2.5 Wireless Sensor Network and IoT Protocols

The IoT Traffic Controller is one of the essential modules in developing the Intelligent Traffic Control System. The traffic state estimator devices' primary function is to collect insights into the state or situation of the traffic (number of vehicles, speed, vehicle types), which will then be fed as input to the Deep RL-based traffic controller. As shown in Figure 1, the system has several microcomputer systems as the main part and other electronic peripherals such as a traffic light, camera (sensor), solar panel, and LED display counter. The design is portable and can be towed by any vehicle for urgent deployment in case of a traffic problem.

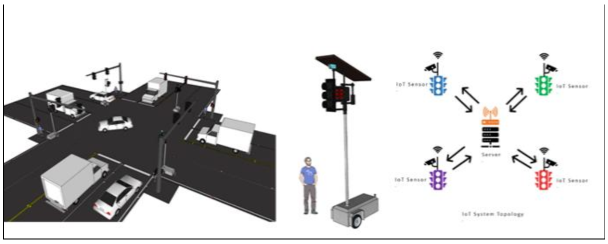


Fig. 1. Design of Mobile-Solar Powered Intelligent Traffic System

3. RESULTS AND DISCUSSION

3.1 Performance of the Trained Models

The performance of the trained model is based on the commonly used parameters that include the mAP, precision, recall, and F1 Score which are undertaken in almost all analyses. Accordingly, YOLOv5s outperformed its predecessors, YOLOv4 and YOLOv3 [37] , when trained using COCO and other various datasets [38]. Moreover, this study supports the work [39] by training the models using Transfer Learning and localized traffic datasets.

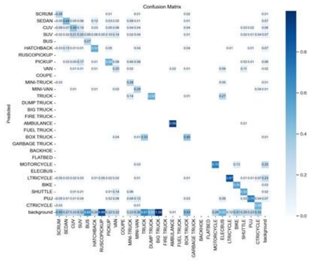
Table 2. Performance of YOLO Variants with Transfer Learning Method

Measure	YOLOv3	YOLOv4	YOLOv5s
Precision	1.0	1.0	1.0
Recall	0.87	0.75	0.94
F1-Score	0.41	0.41	0.81
mAP@0.5	0.53	0.410	0.858

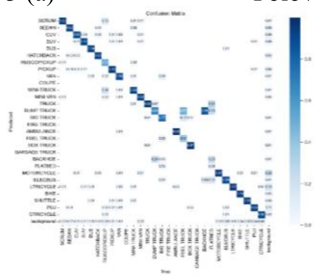
Table 2 shows all models have the same precession, and YOLOv5s has the highest recall, followed by YOLOv3, and the worst in YOLOv4. F1 Score and mAP also state that YOLOv5s is better compared to its predecessors. However, YOLOv5s required more time to train despite having the same number of epochs. For further experimentation, the researchers compared the parameter values of the three versions using transfer-learned models with occluded custom datasets (Table 3). Occluded datasets refer to the images of vehicles partially or completely obstructed by other vehicles. In general, a situation where a vehicle is not fully visible from a particular vantage point or the vehicle's visual information is unclear. Precision, recall, F1 Score, runtime, and mAP were shown in Table 2 to measure the performance of the model. In Table 3, Intersection over Union shows the measuring parameters of YOLOv3, YOLOv4, and YOLOv5s with its corresponding results. The results indicate that the highest mAP belongs to YOLOv5s, followed by YOLOv4 and YOLOv3. In terms of precision, recall, and F1 Score, YOLOv5s outlearned its predecessors. Each confusion matrix is shown in Figure 2.

Table 3. Performance of Transfer Learned Model tested on Occluded Datasets

Measure	YOLOv3	YOLOv4	YOLOv5s
Precision	1.0	1.0	1.0
Recall	0.59	0.94	0.97
F1-Score	0.36	0.67	0.86
mAP@0.5	0.369	0.706	0.88



Yolov3 (a) Yolov4(b)



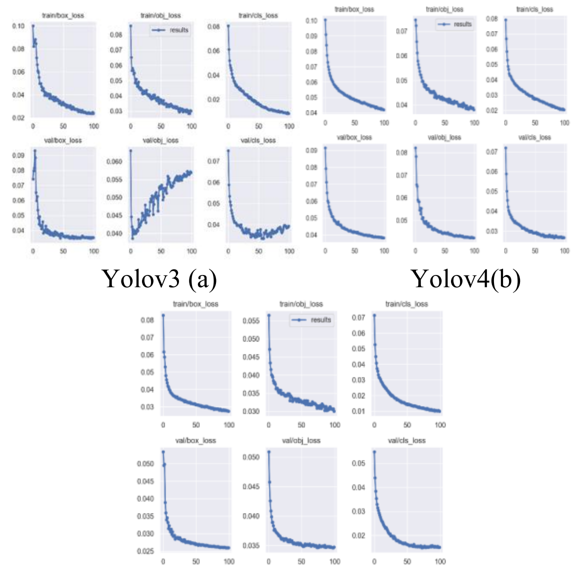
Yolov5s (c)

Fig 2. YOLO Variants Training Loss Statistics/Graphs

3.2 Overfitting

As shown in Figure 3, YOLOv3 (a) demonstrates strong performance on both the training and validation sets, with relatively low losses across all three loss functions. The training box_loss, obj_loss, and cls_loss are 0.10, 0.08, and 0.08, respectively, while the validation box_loss, obj_loss, and cls_loss are 0.09, 0.064, and 0.07. The small difference between the training and validation losses suggests that the model is not overfitting and can generalize well to new, unseen data.

Similarly, in YOLOv4 (b), the training loss values are lower compared to the validation loss values, particularly for the box_loss and obj_loss functions. The training box_loss, obj_loss, and cls_loss are 0.10, 0.078, and 0.08, while the validation box_loss, obj_loss, and cls_loss are 0.091, 0.082, and 0.072. This suggests that while the model predicts bounding boxes with lower accuracy during training, it can accurately predict objectness and object class labels.



Yolov3 (a) Yolov4(b)

Yolov5s (c)

Fig. 3. YOLO Variants Training Loss Statistics/Graphs

However, the model's performance on unseen data is not as strong as on the training data. The higher obj_loss on the validation set implies difficulties in detecting objects in validation images, and the slightly elevated validation object loss compared to the training object loss (0.082 vs. 0.078) may indicate some overfitting. The model performs well for YOLOv5s (c), with both training and validation losses being relatively low and comparable across the different loss functions. Specifically, the training box_loss, obj_loss, and cls_loss are 0.082, 0.058, and 0.071, respectively, indicating accurate predictions for bounding boxes, objectness, and class labels during training. The validation box_loss, obj_loss, and cls_loss are 0.054, 0.054, and 0.059, respectively, indicating that the model generalizes effectively to new, unseen data, with no significant overfitting observed.

3.3 Example Detection

The sample detection of each model was visually compared using both pre-trained and transfer-learned models, as shown in Figure 4. A threshold of 0.65 was used for all three models with similar pictures with 45° and 90° camera angles from the validation set to compare the differences in detection. The comparison is presented with bounding boxes above the vehicles to show the detection accuracy.

Fig. 4. Sample Detections of Pretrained and Transfer Learning

3.4 State Estimator using DEEPSORT

The state is a feature representation of the observed environment and may be represented by queue length, waiting time, volume, delay, speed, position of vehicles, and phase duration. Since our Traffic Controller utilizes a camera for traffic observation, and the model is limited only to tracking and classifying objects, the feature representation will focus only on vehicle position over time. The speed feature is derived from the vehicle's initial and final grid position, shown in Figure 5.

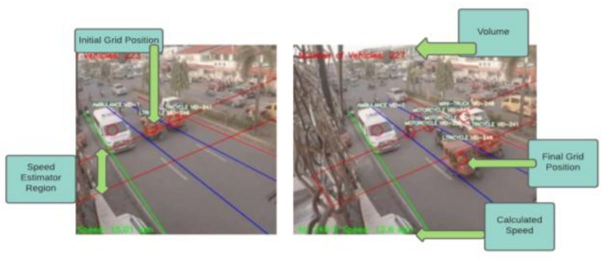


Fig. 5. The State Estimator: Speed, Volume, and Vehicle Position

3.5 Dynamic Timed Traffic Controller

We evaluated the performance of our trained RL traffic controllers through simulations, focusing on minimizing total vehicle waiting times during each green-phase state. Over a simulation period of 250,000 seconds, the A2C (Advantage Actor-Critic) method emerged as the most effective traffic controller in terms of reducing waiting times. Shown in Table 6 are the comparisons with fixed-time controllers, which revealed significant improvements in traffic flow efficiency, clearly demonstrated in comparative plots using the same vehicle routing pattern in SUMO simulations. The superior performance of A2C indicates enhanced vehicle throughput at the intersection, translating to smoother traffic operations. A2C, known for its combined approach of Direct Policy Search and Value-based RL, represents our optimal Deep Reinforcement Learning-based traffic control solution for forthcoming testing phases in this project.

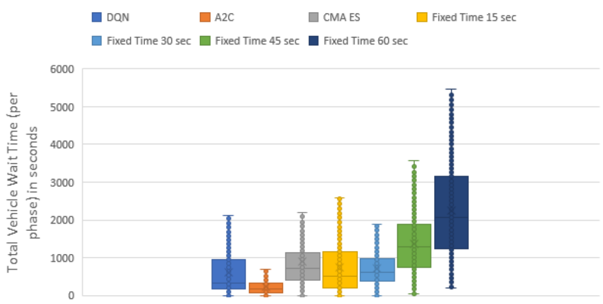


Fig. 6. Performance of RL-based and fixed-timed traffic controller

3.6 Deployment of the Models

Figure 7 illustrates the IoT setup, consisting of several components: Jetson Nano, responsible for data capture and analysis; RasPi, which receives vectorized data from Jetson Nano and communicates with the server via a wireless router; Relay, facilitating communication with the traffic controller; and Power Supply, providing specific voltage to the system. The setup includes a Box containing electronic devices and a 720-pixel camera. The system's power supply will be sourced from solar panels, enabling real-time monitoring. To enhance operational efficiency, the deployment will leverage YOLOv5s object detection, trained using custom datasets. The A2C controller was integrated into the system; however, the device was designed with reconfigurability, allowing it to accommodate various model types, whether based on reinforcement learning algorithms, fixed-timing, or rule-based controllers. This flexibility ensures adaptability to different operational requirements and enhances its capability to optimize performance based on evolving traffic conditions and environmental factors.

Fig. 7. Mobile intelligent traffic system based on A2C and Internet-of-things

Additionally, the traffic light has a height of 14 feet, a light with dimensions of 750x250x155mm, and a countdown timer with dimensions of 800x600mm. Additionally, the traffic light is fitted with a solar panel with a capacity of 130W, a battery with a capacity of 12V/55AH, and a DC-AC converter with a maximum output of 220V and a maximum power of 900W.
For testing purposes, traffic is currently deployed on University Avenue. The traffic lights would be able to detect vehicles and adjust the amount of time that should be spent at each traffic light accordingly. The time allotment is dynamic and can be between 15, 30, and 45 seconds, depending on the traffic volume.

4. CONCLUSION

This study presents a mobile-solar-powered Intelligent Traffic Control System designed to enhance urban traffic management through dynamic, real-time traffic light control. By integrating a traffic state estimator using advanced object detection models (YOLOv5s) and deep reinforcement learning (RL), specifically the A2C (Advantage Actor-Critic) algorithm, the system demonstrates significant improvements in traffic flow efficiency compared to traditional fixed-time controllers. The system's ability to adjust traffic light timings based on real-time traffic conditions, coupled with the use of solar power for sustainability, highlights its potential for deployment in diverse environments, including those with limited infrastructure. The results show that YOLOv5s, trained using transfer learning on custom traffic datasets, outperforms its predecessors (YOLOv3 and YOLOv4) in terms of precision, recall, F1 score, and mean Average Precision (mAP), even under occlusion scenarios. The state estimator effectively tracks vehicle positions and calculates critical traffic metrics such as speed, volume, and delay, enabling the dynamic adjustment of green-phase durations. The performance of the A2C controller, as demonstrated in SUMO simulations, validates the feasibility of RL-based traffic control in real-world applications, significantly reducing vehicle waiting times and enhancing overall intersection throughput. However, despite the promising results, there are certain limitations in the current design and implementation. First, the model was trained on a limited set of custom datasets primarily focused on vehicle detection and tracking. This may affect the model’s generalizability to other regions or environments with different traffic patterns or road configurations. Additionally, while the occluded datasets were effective in evaluating the system’s robustness, they may not fully represent all real-world traffic conditions. Second, while the system operates efficiently in simulations, real-time data processing and control adjustments in a highly dynamic urban environment may face latency issues due to the complexity of deep learning models and sensor data processing. Future work will involve expanding the dataset to include diverse traffic scenarios, road types, and environmental conditions. Lastly, this study makes a significant contribution by introducing an innovative, scalable, and sustainable solution for intelligent traffic control that leverages state-of-the-art AI techniques and renewable energy to address urban traffic management challenges.

5. REFERENCES

[1] H. M. Nguyen and L. D. Nguyen, “The relationship between urbanization and economic growth an empirical study on ASEAN countries,” *Int J Soc Econ*, vol. 45, no. 2, pp. 316–339, 2018, doi: 10.1108/IJSE-12-2016-0358.

[2] A. Fedorov, K. Nikolskaia, S. Ivanov, V. Shepelev, and A. Minbaleev, “Traffic flow estimation with data from a video surveillance camera,” *J Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0234-z.

[3] B. Trask, “Migration, urbanization, and the family dimension,” *Division for Inclusive Social Development, Focal Point on the Family*, 2022.

[4] S. A. H. Alhuthali, M. Y. I. Zia, and M. Rashid, “A Simplified Traffic Flow Monitoring System Using Computer Vision Techniques,” *Proceedings of 2022 2nd International Conference on Computing and Information Technology, ICCIT 2022*, pp. 167–170, 2022, doi: 10.1109/ICCIT52419.2022.9711550.

[5] K. Khazukov *et al.*, “Real-time monitoring of traffic parameters,” *J Big Data*, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-00358-x.

[6] B. M. Metran, J. P. Elumba, M. C. Alvardo, and M. M. Bermoy, “Mapping and Estimation of Road Traffic Emissions in Caraga Region using Sentinel 5P Satellite Images,” in *International Exchange and Innovation Conference on Engineering and Sciences*, Kyushu University, 2023, pp. 6–12. doi: 10.5109/7157933.

[7] T. Yuan, W. Da Rocha Neto, C. E. Rothenberg, K. Obraczka, C. Barakat, and T. Turletti, “Machine learning for next-generation intelligent transportation systems: A survey,” *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 4, pp. 1–35, 2022, doi: 10.1002/ett.4427.

[8] U. V Ramesh *et al.*, “International Journal of Research Publication and Reviews Metaverse : Future of the Internet,” vol. 3, no. 2, pp. 93–97, 2022.

[9] D. Sirohi, N. Kumar, and P. S. Rana, “Convolutional neural networks for 5G-enabled Intelligent Transportation System: A systematic review,” *Comput Commun*, vol. 153, pp. 459–498, 2020, doi: 10.1016/j.comcom.2020.01.058.

[10] J. qing Luo, H. sheng Fang, F. ming Shao, Y. Zhong, and X. Hua, “Multi-scale traffic vehicle detection based on faster R–CNN with NAS optimization and feature enrichment,” *Defence Technology*, vol. 17, no. 4, pp. 1542–1554, Aug. 2021, doi: 10.1016/j.dt.2020.10.006.

[11] J. A. Kim, J. Y. Sung, and S. H. Park, “Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition,” in *2020 IEEE International Conference on Consumer Electronics - Asia, ICCE-Asia 2020*, Institute of Electrical and Electronics Engineers Inc., 2020. doi: 10.1109/ICCE-Asia49877.2020.9277040.

[12] A. John and D. Meva, “A Comparative Study of Various Object Detection Algorithms and Performance Analysis,” *International Journal of Computer Sciences and Engineering*, vol. 8, no. 10, pp. 2347–2693, 2020, doi: 10.26438/ijcse/v8i10.158163.

[13] A. K. Shetty, I. Saha, R. M. Sanghvi, S. A. Save, and Y. J. Patel, “A Review: Object Detection Models,” *2021 6th International Conference for Convergence in Technology, I2CT 2021*, pp. 1–8, 2021, doi: 10.1109/I2CT51068.2021.9417895.

[14] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” *Proceedings - International Conference on Image Processing, ICIP*, vol. 2017-Septe, pp. 3645–3649, 2018, doi: 10.1109/ICIP.2017.8296962.

[15] O. Bourja, H. Derrouz, H. A. Abdelali, A. Maach, R. O. H. Thami, and F. Bourzeix, “Real Time Vehicle Detection, Tracking, and Inter-vehicle Distance Estimation based on Stereovision and Deep Learning using Yolov3,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 8, pp. 915–923, 2021, doi: 10.14569/IJACSA.2021.01208101.

[16] P. Wei, H. Shi, J. Yang, J. Qian, Y. Ji, and X. Jiang, “City-Scale vehicle tracking and traffic flow estimation using low frame-rate traffic cameras,” *UbiComp/ISWC 2019- - Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, no. 1, pp. 602–610, 2019, doi: 10.1145/3341162.3349336.

[17] R. Suryadithia, M. Faisal, A. S. Putra, and N. Aisyah, “Technological Developments in the Intelligent Transportation System (ITS),” pp. 837–843, 2021.

[18] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, “A survey of modern deep learning based object detection models,” *Digital Signal Processing: A Review Journal*, vol. 126, pp. 1–18, 2022, doi: 10.1016/j.dsp.2022.103514.

[19] C. jin Li, Z. Qu, S. ye Wang, and L. Liu, “A method of cross-layer fusion multi-object detection and recognition based on improved faster R-CNN model in complex traffic environment,” *Pattern Recognit Lett*, vol. 145, pp. 127–134, May 2021, doi: 10.1016/j.patrec.2021.02.003.

[20] A. A. Husain, T. Maity, and R. K. Yadav, “Vehicle detection in intelligent transport system under a hazy environment: A survey,” *IET Image Process*, vol. 14, no. 1, pp. 1–10, 2020, doi: 10.1049/iet-ipr.2018.5351.

[21] A. Shaqour and A. Hagishima, “Recent Advances in Reinforcement Learning Applications for Building Energy Management: A Mini Review.,” *International Exchange and Innovation Conference on Engineering and Sciences*, pp. 239–245, 2022, doi: 10.5109/5909098.

[22] K. Bálint, T. Tamás, and B. Tamás, “Deep Reinforcement Learning based approach for Traffic Signal Control,” *Transportation Research Procedia*, vol. 62, no. Ewgt 2021, pp. 278–285, 2022, doi: 10.1016/j.trpro.2022.02.035.

[23] Z. Huang, “Reinforcement learning based adaptive control method for traffic lights in intelligent transportation,” *Alexandria Engineering Journal*, vol. 106, no. July, pp. 381–391, 2024, doi: 10.1016/j.aej.2024.07.046.

[24] Kranti Shingate, Komal Jagdale, and Yohann Dias, “Adaptive Traffic Control System using Reinforcement Learning,” *International Journal of Engineering Research and*, vol. V9, no. 02, pp. 443–447, 2020, doi: 10.17577/ijertv9is020159.

[25] S. M. Masfequier Rahman Swapno, S. M. Nuruzzaman Nobel, A. C. Ramachandra, M. B. Islam, R. Haque, and M. M. Rahman, “Traffic Light Control Using Reinforcement Learning,” *2nd International Conference on Integrated Circuits and Communication Systems, ICICACS 2024*, no. August, 2024, doi: 10.1109/ICICACS60521.2024.10498933.

[26] H. Wei, H. Yao, G. Zheng, and Z. Li, “IntelliLight: A reinforcement learning approach for intelligent traffic light control,” *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2496–2505, 2018, doi: 10.1145/3219819.3220096.

[27] A. G. Barto, P. S. Thomas, and R. S. Sutton, “Some recent applications of reinforcement learning,” in *Proceedings of the Eighteenth Yale Workshop on Adaptive and Learning Systems*, 2017.

[28] A. T. D. Perera and P. Kamalaruban, “Applications of reinforcement learning in energy systems,” *Renewable and Sustainable Energy Reviews*, vol. 137, p. 110618, 2021.

[29] R. J. Candare and J. Matias, “Evaluating Deep Reinforcement Learning Methods to Develop an Intelligent Traffic Controller,” in *2022 5th*

International Conference of Computer and Informatics Engineering (IC2IE), Sep. 2022, pp. 173–178. doi: 10.1109/IC2IE56416.2022.9970176.
[30] N. C. Luong *et al.*, "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019, doi: 10.1109/COMST.2019.2916583.
[31] V. François-lavet *et al.*, "An Introduction to Deep Reinforcement Learning," *Foundations and trends in machine learning*, vol. II, no. 3–4, pp. 1–140, 2018, doi: 10.1561/2200000071.Vincent.
[32] Y. Liang, X. Wang, H. Zhao, T. Han, Z. Wei, and Y. Li, "A covariance matrix adaptation evolution strategy variant and its engineering application," *Applied Soft Computing Journal*, vol. 83, p. 105680, 2019, doi: 10.1016/j.asoc.2019.105680.
[33] V. Akbarzadeh, A. H.-R. Ko, C. Gagné, and M. Parizeau, "Topography-Aware Sensor Deployment Optimization with CMA-ES," in *Parallel Problem Solving from Nature, PPSN XI*, R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 141–150.
[34] R. M. España, A. M. Hernández-Díaz, J. M. Cecilia, and M. D. García-Román, "Evolutionary strategies as applied to shear strain effects in reinforced concrete beams," *Appl Soft Comput*, vol. 57, pp. 164–176, 2017, doi: https://doi.org/10.1016/j.asoc.2017.03.037.
[35] S. Kuutti, R. Bowden, H. Joshi, R. De Temple, and S. Fallah, "End-to-end Reinforcement Learning for Autonomous Longitudinal Control Using Advantage Actor Critic with Temporal Context," *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019*, pp. 2456–2462, 2019, doi: 10.1109/ITSC.2019.8917387.
[36] R. Iwaki and M. Asada, "Implicit incremental natural actor critic algorithm," *Neural Networks*, vol. 109, no. 2002, pp. 103–112, 2019, doi: 10.1016/j.neunet.2018.10.007.
[37] G. Jocher, "YOLOV5 in PyTorch," 2020.
[38] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018.
[39] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020.