AI-Based Traffic Management System

[1]Ritik Kumar Srivastav , [2]Abhishek Kumar , [3]Karan Pandey , [4]Gaurav Pawar

Department of Computer Science
R.D. Engineering College, Duhai, Ghaziabad, India
ritik022ai0062@rdec.in , abhishek022ai0008@rdec.in , karan022ai0034@rdec.in , gaurav023ai9002@rdec.in

Abstract: Traffic congestion is becoming a serious problem in modern cities with the continuous surge in the number of vehicles and the limitations of traditional traffic signal systems. In fixed-time systems, traffic signals work on fixed schedules and cannot respond to current traffic, often causing extra waiting time. This paper presents an AI-based system used for managing traffic using YOLOv5 for real-time vehicle detection. The proposed technique considers not just vehicle density but also introduces a waiting time- based fairness mechanism to ensure that no lane is ignored for long durations. Additionally, an emergency vehicle priority system is integrated to provide immediate green signals to ambulances and fire trucks. The outcomes reveal that the system significantly reduces waiting time, improves fairness among lanes, and enhances overall traffic flow compared to traditional methods. This approach is suitable for smart city applications. allocation. This system is suitable for smart city traffic management.

IndexTerms - Artificial Intelligence, Traffic Management, YOLO, YOLOv5, Traffic Signals, Waiting-Time Fairness, Emergency Vehicle Priority, Smart City, Computer Vision, Real-Time Detection

I. INTRODUCTION

Urban regions increasingly face serious traffic congestion, most notably during rush hours. The increasing more vehicles on roads and limited road facilities create delays, productivity. fuel wastage, and reduced Traditional traffic control systems mainly rely on fixed signal timings. These systems do not adapt to real-time traffic situations, which leads to inefficient traffic flow. For example, a lane with no vehicles may still receive green signal time, while another lane with heavy traffic remains congested. With the advancement of Artificial Intelligence and computer vision, smarter solutions are now possible. AI based systems can analyze live traffic footage and make dynamic decisions. A fast and effective object detection approach is YOLO (You Only Look Once), which can detect vehicles in real time. Despite this, most available systems only focus on vehicle count. This creates unfair situations where low-traffic lanes suffer long waiting times. To tackle this problem, this study proposes a fairness-based approach that considers both vehicle density and waiting time. The system ensures that no lane is indefinitely ignored, while still prioritizing lanes with higher traffic density when waiting times are comparable. A dedicated emergency vehicle priority mechanism is also integrated, which detects ambulances and fire trucks using YOLOv5 and immediately grants them a green signal, overriding the normal signal cycle. The system presented in this study is compared against fixed time and density only approaches, demonstrate improvements in both fairness, emergency response, and overall traffic flow efficiency.

II. RELATED WORK

Several researchers have worked on traffic management by applying different approaches. YOLO and AI-driven image processing are extensively used for real-time detection owing to their fast performance and accuracy. Some systems adjust signal timing based on vehicle density, which improves traffic flow. Webster et al. proposed a density-based signal timing approach that dynamically allocates green time based on vehicle count at each lane. However, these systems often ignore fairness and may cause long delays in low- density lanes. Advanced approaches like reinforcement learning provide better optimization but are complex and difficult to implement. This paper focuses on a simpler and more practical approach by combining density with waiting-time fairness.

III. SYSTEM ARCHITECTURE

The traffic management framework presented in this study is structured as a modular pipeline that processes real- time input from traffic cameras and generates signal control decisions accordingly. Each component of the system works independently, but all modules share relevant traffic data to maintain coordination. The system is specifically designed to operate at a single road intersection, where it continuously monitors and manages multiple lanes. The system consists of five main components. First, the

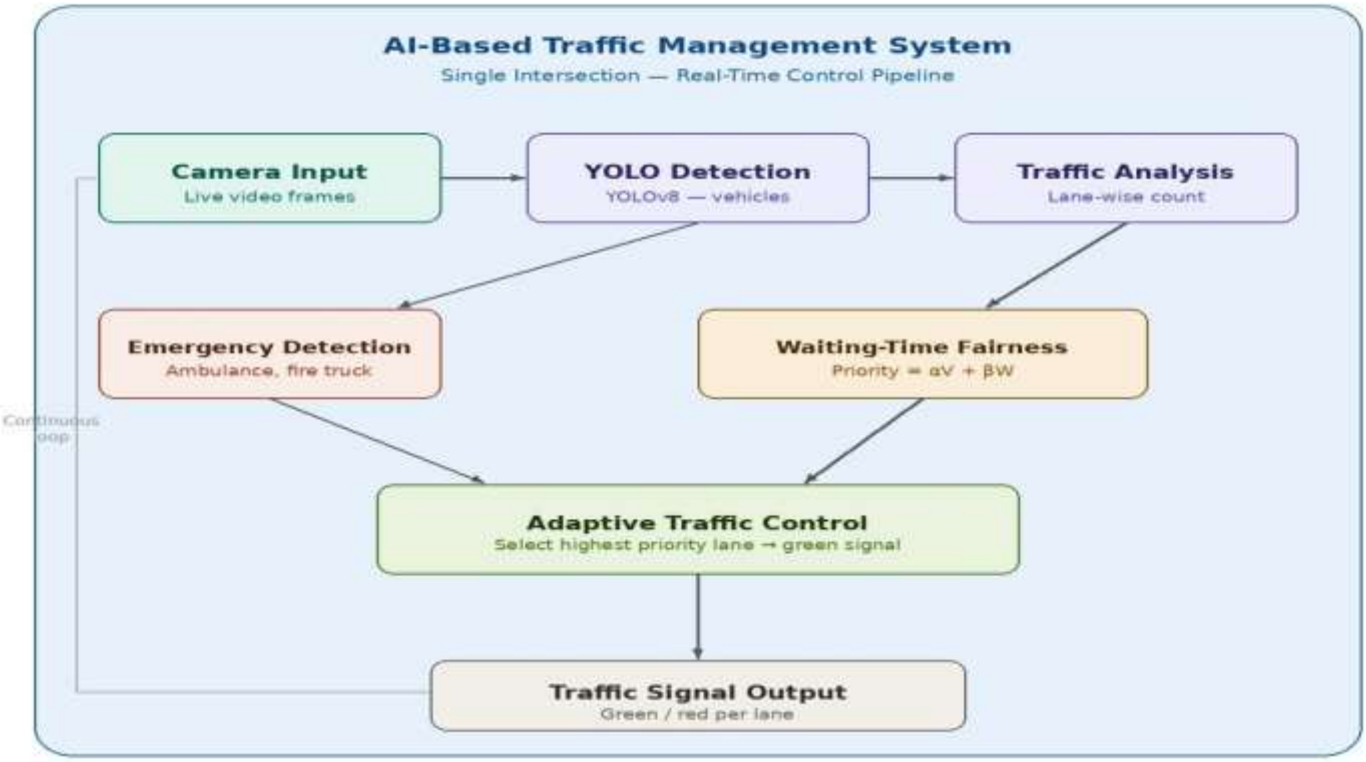

fig. system architecture of the proposed ai-based traffic management system

camera module captures live video streams from each lane. Second, the frames are analyzed by a YOLO-based detection module to detect different kinds of vehicles and their positions. After that, the traffic analysis component processes this data to calculate lane-wise vehicle counts. Third, the waiting-time module keeps track of how long each lane has been waiting and assigns a priority score based on this information. Finally, the traffic control module selects the lane with the highest priority and activates the green signal accordingly. The entire system runs in a continuous cycle, updating its decisions at regular time intervals. This allows it to

quickly adapt to changing traffic conditions, such as sudden congestion or the presence of emergency vehicles. Since the design is modular, any component can be modified or upgraded without affecting the overall system functionality.

IV. PROPOSED METHODOLOGY

A. YOLO-Based Vehicle Detection The YOLO (You Only Look Once) algorithm performs object detection and classification in one pass of a convolutional neural network, making it fast and efficient. Unlike methods such as Faster R CNN that rely on a two-step process involving region proposal and classification, YOLO processes the full image at once. This makes it highly efficient and suitable for real-time applications like traffic monitoring. In this study, YOLOv5 is used as the main detection model. The model is first trained on a general dataset and then fine-tuned using a custom dataset based on COCO, specifically prepared for traffic scenarios. This dataset contains labeled images of different vehicle types, including cars, buses, trucks, motorcycles, and emergency vehicles, captured under various environmental conditions like day, night, rain, and fog. Fine-tuning helps the model adapt better to real-world traffic conditions and specific camera viewpoints. The model produces bounding regions accompanied by class labels and confidence scores for detected vehicles.

B. Dataset and Training Training of the YOLOv5 model was carried out using a customized dataset that includes a mix of publicly available datasets and a small custom dataset compiled from traffic videos captured by surveillance cameras at urban intersections. The dataset includes diverse conditions such as daylight, nighttime, rainy, and foggy scenarios. Vehicle classes include car, motorcycle, bus, truck, and emergency vehicle (ambulance, firetruck). To enhance the model's robustness, data augmentation methods such as horizontal flipping, random cropping, brightness variation, augmentation were applied.

C. Dynamic Signal Timing
In dynamic systems, signal timing is adjusted based on current traffic conditions instead of following fixed schedules. In the proposed system, lane-wise vehicle counts obtained from YOLO helps in assessing traffic demand, allowing lanes with heavier traffic to be allocated longer green durations, while minimum and maximum limits are enforced to ensure safety and stability. This dynamic allocation improves traffic flow and serves as the base layer over which waiting- time-based fairness is applied. The minimum green time is set to 10 seconds to allow vehicles already in motion to clear the intersection safely. The maximum green

time is capped at 60 seconds to prevent excessive delays on other lanes. Green time is linearly interpolated between these bounds based on the normalized vehicle count of the selected lane relative to the maximum observed count across all lanes. The green signal time for a particular lane is not fixed; instead, it is determined based on the traffic in that lane using the following formula: $G(i) = T_{min} + (T_{max} - T_{min}) \times (V(i) / V_{max})$ Here, G(i) represents the green time assigned to lane i. The values of Tmin and Tmax define the minimum and maximum limits of green time, which are taken as 10 seconds and 60 seconds respectively. V(i) denotes the number of vehicles in lane i, while Vmax is the highest vehicle count observed among all lanes at a given time. This approach is inspired by adaptive signal control techniques as described in Robertson and Bretherton [21].

D. Waiting-Time–Based Fairness Mechanism

To avoid long waiting times on low-traffic roads, a waiting-time-based fairness method is incorporated. Each lane maintains a waiting-time counter that increments by one unit for every control cycle during which the lane has a red signal. When a lane receives a green signal, its counter is reset to zero. This mechanism ensures that the accumulated waiting time is always tracked and considered in signal allocation decisions. A priority score is computed for each lane at every control cycle using the following weighted formula: $\text{Priority} = \alpha \times \text{VehicleCount} + \beta \times \text{WaitingTime}$ Here, α and β are tunable weighting parameters that control the relative importance of vehicle count versus waiting time. In the default configuration, $\alpha = 0.6$ and $\beta = 0.4$, giving slightly more weight to traffic density while still ensuring fairness. The lane with the highest priority score receives the green signal in the next cycle This approach guarantees that lanes with fewer vehicles will still build up sufficient waiting time to compete effectively with high-density lanes for signal access.

E. Waiting-Time Fairness Traffic Control Algorithm

Algorithm 1: Waiting-Time Fairness Traffic Control
Input: VehicleCount[i] and WaitingTime[i] for each lane. Output: Lane selected for green signal.
Steps are :
1. Capture traffic frames for all lanes using cameras
2. Detect vehicles using YOLOv5
3. Count the number of vehicles in each lane (VehicleCount[i])

4. If a lane has a red signal, increase its waiting time
5. Calculate priority for each lane using: $\text{Priority}[i] = \alpha \times \text{VehicleCount}[i] + \beta \times \text{WaitingTime}[i]$
6. Select the lane with the highest priority value
7. Give green signal to the selected lane
8. Reset the waiting time of that lane to zero
9. Extend the waiting duration for all other lanes
10. Repeat this process continuously

F. System Flow Diagram

Figure illustrates the complete system flow for the proposed AI-based traffic management system. The diagram shows the sequential processing pipeline beginning from traffic frame capture through YOLO- based vehicle detection, counting, waiting time update, emergency vehicle check, and final signal allocation decision. Fig. System Flow Diagram of the Proposed AI-Based Traffic Management System As shown in the figure, the system starts by capturing a traffic frame from the camera. The YOLOv5

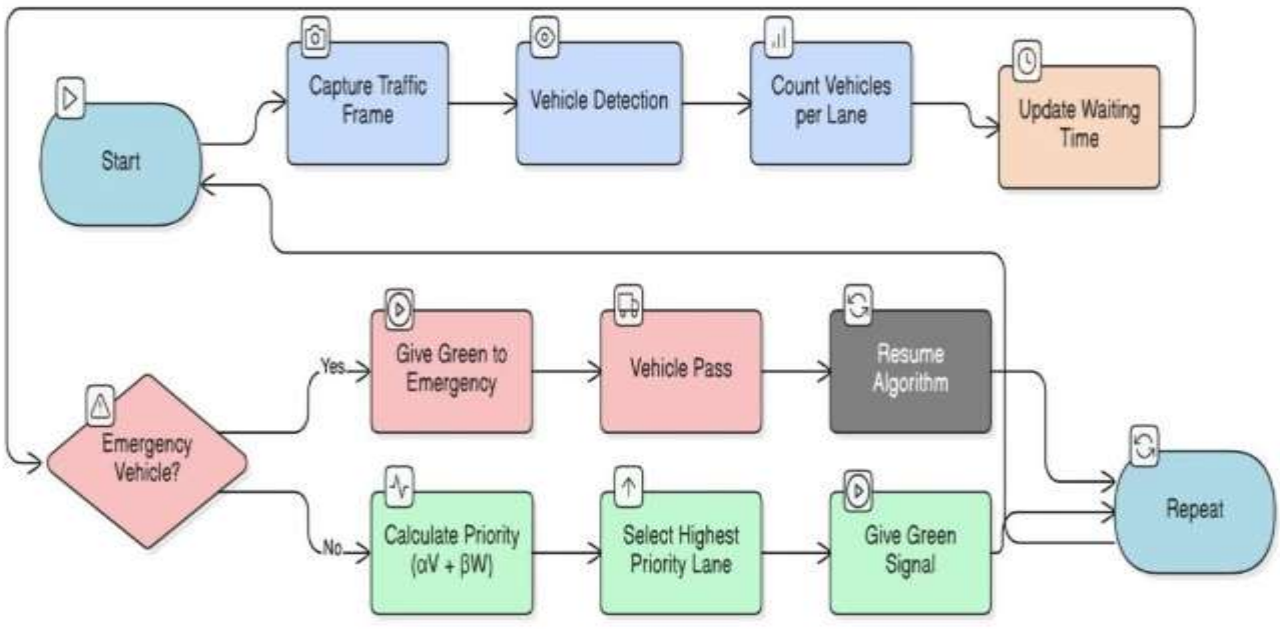

fig. system architecture of the proposed ai-based traffic management system

model is applied to the frame to recognize and count vehicles across each lanes, while also updating the waiting time for every lane. After that, the system checks whether any emergency vehicle is present. If an emergency vehicle is detected, the system immediately assigns a green signal to that lane. Otherwise, it computes priority using the formula ($\alpha V + \beta W$). The lane with the highest priority is selected and given the green signal. This process keeps running continuously so that the system can respond to changing traffic conditions in real time. This method guarantees that even lanes with fewer vehicles are not ignored and get a chance after waiting for some time. In this way, it solves the fairness problem that is common in systems based only on vehicle density.

G. Emergency Vehicle Priority Mechanism In real traffic situations, emergency vehicles like ambulances and fire trucks need to get priority allowing them to arrive at their destination quickly. To handle this, the YOLOv5 model is trained to also recognize emergency vehicles. Whenever an emergency vehicle is detected in any lane, the system temporarily ignores the normal signal logic. It immediately gives a green signal to that lane so that the vehicle can pass without delay. After the emergency vehicle crosses the intersection, the system returns to its normal working process.
If multiple emergency vehicles are arrived, system uses FCFS(First Come First Serve).

Algorithm 2: Emergency Vehicle Priority Handling

Input: Traffic camera frames Output: Immediate green signal for the lane with an emergency vehicle

Steps are:

1. Capture traffic frames from the camera
2. Detect all vehicles using YOLOv5
3. Check if any emergency vehicle is present
4. If an emergency vehicle is found in lane i:
5. Assign highest priority to that lane

6. Stop the current signal process
7. Immediately give green signal to lane i.
8. Keep the signal green until the vehicle passes
9. After that, continue the normal waiting-time fairness algorithm.

H. Mathematical Model
The complete priority score used by the system is defined as: $Priority_i = \alpha \cdot V_i + \beta \cdot W_i + \gamma \cdot E_i$ Where V_i is the number of vehicles in lane i, W_i is the accumulated waiting time of lane i, E_i is the emergency vehicle indicator (1 if an emergency vehicle is present, 0 otherwise), and α, β, γ are weighting parameters. The default values are $\alpha = 0.6$, $\beta = 0.4$, and $\gamma = 1000$, ensuring emergency vehicles always receive the highest possible priority score.The large value of γ guarantees that emergency vehicle detection overrides all normal priority calculations regardless of vehicle count or waiting time.

V. EXPERIMENTAL SETUP

To assess the proposed system, a simulation environment was created using Python and OpenCV. It models a single traffic intersection with multiple lanes, each capable of accommodating up to 30 vehicles. Vehicle arrivals are generated using a Poisson distribution to simulate different traffic conditions, from low traffic to peak hours. Three types of scenarios were considered:
(1) Balanced scenario, where all lanes have similar traffic; (2) Imbalanced scenario, where one lane has higher traffic compared to others; and (3) Emergency scenario, where emergency vehicles are introduced randomly during the simulation. All scenarios were performed for a period of multiple cycles and repeated several times to ensure consistent results. To evaluate performance, metrics like average waiting time, peak waiting time, fairness index, and throughput (vehicles/hour) were used. The results from the simulation demonstrate that the proposed fairness-based approach performs better than fixed- time and density-based methods. It reduces waiting time, improves fairness among lanes, and increases overall traffic flow efficiency. These results are based on simulation and may vary in real-world conditions

VI. COMPARATIVE ANALYSIS

The proposed system is compared with three traffic control strategies to evaluate its performance comprehensively. Fixed-time systems show high average delay due to their static nature. Density-based systems reduce overall delay but may result in significant delays for low-density lanes. The proposed fairness-based system balances efficiency and fairness by considering both vehicle density and accumulated waiting time.

table i. performance comparison of traffic control methods

Method	Avg Wait (s)	Max Wait (s)	Fairness Index	Throughput
Fixed-Time	High	High	Low	Low
Density-Based	Medium	Medium	Medium	Med ium
Propose d Fairness	Low	Low	High	High

Table I shows a comparison between fixed-time, density-based, and the proposed traffic control system.
From the table, it is evident that the fixed-time method has higher waiting time because it works on fixed signals and does not consider actual traffic conditions. The density-based method performs better, as it adjusts signals in relation to the number of vehicles, but it may still cause some delay in less crowded lanes.

The proposed system gives better overall results. Both the average waiting time and the maximum delay are lower, and the fairness is higher compared to the other methods. The system considers two main factors: the vehicle count and the duration each lane has been waiting.

Also, the throughput is higher, which means traffic moves more smoothly. These results are based on simulation and give a general idea of how the system can perform in real situations..

table ii. effect of parameter tuning (α, β) on system performance

α	β	Avg Wait (s)	Fairness Index	Through put
1.0	0.0	52	0.76	681 vph
0.8	0.2	44	0.85	701 vph
0.6	0.4	39	0.93	720 vph

0.4	0.6	41	0.95	691 vph
0.0	1.0	60	0.98	594 vph

The data in this table is based on assumed simulation conditions and is mainly used to understand the behavior of the system. It can be noticed that when α is higher, the system focuses more on the number of vehicles, which helps in improving traffic flow. On the other hand, when β is higher, more importance is given to waiting time, which improves fairness among lanes.A balanced setting of α = 0.6 and β = 0.4 gives stable results, as it considers both traffic density and waiting time. This approach maintains an effective balance between fairness and efficiency.
However, these results apply only for understanding the system behavior and may change after full implementation and real-world testing. of efficiency and fairness. This configuration is used as the default in all other experiments.

VII . RESULTS AND DISCUSSION

The findings show that the proposed system outperforms than fixed-time and density-based methods in most cases. In the balanced scenario, all methods show similar traffic flow, but the proposed system provides better fairness because it also considers waiting time.
In the imbalanced scenario, the difference becomes more noticeable. The proposed system helps in reducing long waiting times, especially for lanes with fewer vehicles, compared to the other methods.
One limitation observed during testing is that when all lanes have very high traffic simultaneously, the benefit of the waiting-time factor becomes less effective. In such situations, all lanes build up waiting time at a similar rate, and the system behaves more like a density-based approach. These results are based on simulation and provide an understanding of system behavior. Further improvements can be made by adjusting parameters dynamically to handle heavy traffic conditions more efficiently.

VIII. REAL-WORLD IMPLEMENTATION CONSIDERATIONS

To use this system in a real city environment, some practical factors need to be considered along with the algorithm. Things like camera placement, hardware setup, network connection, and linking the system with existing traffic signals are very important. Cameras need to be installed in a manner that each lane is clearly visible, either from a top view or at an angle. A resolution of around 1080p is enough for detecting vehicles properly in most conditions. For night use, cameras with infrared support can be helpful. Using a wired connection such as Ethernet is preferable to minimize delay. The system can run on an edge device placed near the intersection. This device should have enough processing power, preferably with GPU support, so that vehicle detection and signal control can work smoothly in real time. The software running on this device includes vehicle detection, traffic analysis, and signal decision- making. It should also be connected to the traffic lights so that signals can be handled automatically. If this system is connected to a central platform, it becomes easier to monitor traffic at multiple intersections. Data collected from different locations can help in understanding traffic patterns and improving signal timing in the future. Overall, it can serve as an independent traffic management system as well as be incorporated into a wider smart city network.

IX. CONCLUSION

The study introduces an AI-based traffic management system that uses YOLO for detecting vehicles, along with a waiting-time-based mechanism and support for emergency vehicles. The system is developed to resolve the limitations of methods that depend only on vehicle count, which can sometimes cause long delays in less crowded lanes. The results from the study suggest that the proposed system can help in reducing waiting time and improving fairness among lanes when compared to traditional methods. It also demonstrates that maintaining efficient traffic flow while enhancing fairness is achievable at the same time. The system can run on edge devices and can be connected with existing traffic signal infrastructure. Since the design is modular, various parts of the system can be updated or improved over time without altering the complete system.
Overall, the proposed approach provides a practical solution for smarter and more efficient traffic management.

X. FUTURE SCOPE

In the future, this system can be improved in many ways. One useful idea is to connect nearby intersections so that traffic signals can work together. This can help vehicles move more smoothly and reduce unnecessary stops. Another improvement could be making the system more flexible. Instead of using fixed values of α and β, the system can adjust these values automatically based on traffic conditions. This can help maintain better performance without manual changes. It can be further developed to identify pedestrians and cyclists as well. This will make it more suitable for busy city areas where different types of road users are present. The system can also be connected with smart vehicle communication in the future. This can help in predicting traffic movement and adjusting signals in advance. Finally, testing the system in real-world conditions at different locations will give a better idea of its performance. It can also help in understanding how much it reduces traffic delays and improves overall traffic flow.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” Proc. IEEE CVPR, pp. 779-788, 2016.
https://arxiv.org/abs/1506.02640

[2] G. Jocher et al., "YOLOv5: You Only Look Once v5," GitHub Repository, 2020.
https://github.com/ultralytics/yolov5
[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137-1149, 2017.
https://arxiv.org/abs/1506.01497

[4] T.-Y. Lin, M. Maire, S. Belongie, et al., "Microsoft COCO: Common Objects in Context," Proc. European Conference on Computer Vision (ECCV), pp. 740-755, 2014.
https://arxiv.org/abs/1405.0312

[5] S. S. Chavan, J. Balaji, and R. K. Venkatesan, "Real-Time Traffic Management Using Image Processing," International Journal of Engineering Research and Technology, vol. 3, no. 2, pp. 1-5, 2014.
https://www.ijert.org/real-time-traffic-management-using-image-processing

6] H. Wei, G. Zheng, H. Yao, and Z. Li, "PressLight: Learning Max Pressure Control to Coordinate Traffic Signals in Networks," Proc. ACM SIGKDD, pp. 1290-1298, 2019.
https://dl.acm.org/doi/10.1145/3292500.3330949

[7] A. Liao, A. Bochkovskiy, C.-Y. Wang, and H.-Y. M., 'YOLOv4: Optimal Speed and Accuracy for Object Detection,' preprint arXiv:2004.10934, 2020.
https://arxiv.org/abs/2004.10934

[8] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," Proc. IEEE CVPR, pp. 2117-2125, 2017.
https://arxiv.org/abs/1612.03144

[9] S. Devarakonda, P. Bhaskara, and M. Srinivas, "Real-Time Vehicle Detection and Counting Using OpenCV," International Journal of Innovative Research in Science, Engineering and Technology, vol. 6, no. 5, pp. 8265-8271, 2017. https://www.ijirset.com/upload/2017/may/109_Real.pdf

[10] F. V. Webster, "Traffic Signal Settings," Road Research Technical Paper No. 39, Her Majesty's Stationery Office, London, 1958. https://trid.trb.org/view/694881 [11] C. Shorten and T. M. Khoshgoftaar, "A Survey on Image Data Augmentation for Deep Learning," Journal of Big Data, vol. 6, no. 60, pp. 1-48, 2019.
https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0

[11] C. Shorten and T. M. Khoshgoftaar, "A Survey on Image Data Augmentation for Deep Learning," Journal of Big Data, vol. 6, no. 60, pp. 1-48, 2019.
https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0

[12] R. Jain, D. Chiu, and W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems," DEC Technical Report TR-301, 1984.
https://arxiv.org/abs/cs/9809099

[13] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345-1359, 2010.
https://ieeexplore.ieee.org/document/5288526

[14] D. I. Robertson and R. D. Bretherton, "Optimizing Networks of Traffic Signals in Real Time: The SCOOT Method," IEEE no. Transactions on Vehicular Technology, vol. 40, 1, pp. 11-15, 1991.
https://ieeexplore.ieee.org/document/69453