electronics

Article

Autonomous Traffic System for Emergency Vehicles

Mamoona Humayun [1,*], Maram Fahhad Almufareh [1] and Noor Zaman Jhanjhi [2]

[1] Department of Information Systems, College of Computer and Information Sciences, Jouf University, Sakakah 72388, Saudi Arabia; mfalmufareh@ju.edu.sa
[2] School of Computer Science and Engineering (SCE), Taylor's University, Subang Jaya 47500, Malaysia; noorzaman.jhanjhi@taylors.edu.my
* Correspondence: mahumayun@ju.edu.sa

Abstract: An emergency can occur at any time. To overcome that emergency efficiently, we require seamless movement on the road to approach the destination within a limited time by using an Emergency Vehicle (EV). This paper proposes an emergency vehicle management solution (EVMS) to determine an efficient vehicle-passing sequence that allows the EV to cross a junction without any delay. The proposed system passes the EV and minimally affects the travel times of other vehicles on the junction. In the presence of an EV in the communication range, the proposed system prioritizes the EV by creating space for it in the lane adjacent to the shoulder lane. The shoulder lane is a lane that cyclists and motorcyclists will use in normal situations. However, when an EV enters the communication range, traffic from the adjacent lane will move to the shoulder lane. As the number of vehicles on the road increases rapidly, crossing the EV in the shortest possible time is crucial. The EVMS and algorithms are presented in this study to find the optimal vehicle sequence that gives EVs the highest priority. The proposed solution uses cutting-edge technologies (IoT Sensors, GPS, 5G, and Cloud computing) to collect and pass EVs' information to the Roadside Units (RSU). The proposed solution was evaluated through mathematical modeling. The results show that the EVMS can reduce the travel times of EVs significantly without causing any performance degradation of normal vehicles.

Keywords: emergency vehicles; shoulder lane; IoT; cloud computing; 5G; roadside unit

check for updates

Citation: Humayun, M.; Almufareh, M.F.; Jhanjhi, N.Z. Autonomous Traffic System for Emergency Vehicles. *Electronics* **2022**, *11*, 510. https://doi.org/10.3390/electronics11040510

Academic Editor: Arturo de la Escalera Hueso

Received: 6 January 2022
Accepted: 4 February 2022
Published: 9 February 2022

1. Introduction

Roads have grown increasingly crowded in the last few decades due to the rapid increase in population. Excessive traffic has created many problems such as property damage, air pollution, time wastage, and loss of life due to accidents. Careful city planning can alleviate transportation problems, but planning does not always function effectively in the face of unforeseen population and vehicle usage development [1]. Various intelligent traffic control systems have been introduced in the past few years to better manage and control rapidly increasing traffic [2]. However, passing EVs in peak traffic hours is a challenge that needs to be addressed. An EV is a vehicle that provides emergency services in an incident. These vehicles are usually exempted from conventional traffic rules to reach their destination as soon as possible. EVs are mainly categorized into three types: medical, firefighting, and law enforcement. These EVs are equipped with audible and visual warning devices that help them reach their destination as soon as possible [3,4].

Different countries have developed various intelligent systems for passing EVs without disrupting the other traffic. These systems use Global Positioning Systems (GPS), Computer-Aided Dispatch (CAD), and traffic management information to determine the location of EVs and the estimated time of reaching the junction. These systems automatically turn red lights into green until EVs cross that intersection; once an EV passes the intersection, the traffic lights operate in a normal mode to minimize traffic disruptions [5,6]. These systems pass the EV in normal situations when the road is not congested and there is only a single EV, but what if the road is highly congested and there is no space for the other vehicles to change lanes? Or if more than one EV is coming from opposite directions of the signal? In addition, the criticality level of all emergencies is not the same. Therefore, a proper mechanism is required that may pass multiple EVs arriving at an intersection simultaneously with minimum delay through EV scheduling.

EVs are usually at high speed. Therefore, there is a great chance of crashes occurring if there is congestion on the road. A key motivation behind the EV scheduling and management mechanism is to reduce EV crashes. According to the statistics provided by National Safety Council (NSC) [7], the US experienced 170 deaths in 2019 which involved EVs. The vast majority (63%) of those killed were passengers in non-emergency vehicles. Figure 1 provides the statistics for the last seven years. These statistics show that most of the fatalities occurred in multiple-vehicle crashes. Most deaths were caused by incidents involving law enforcement vehicles 114, followed by ambulances 33, and fire trucks 28. According to paper [8], when ambulances run with lights and sirens, there is a possibility for an increase in collision risk. The collision rate is 4.6 per 100,000 replies when an ambulance responds to an emergency call without lights and sirens. When lights and sirens are employed, the accident rate rises to 5.5. When the ambulance is transporting a person, the risk is significantly higher. Without lights and sirens, the accident risk is 7.0 per 100,000 transports; however, when lights and sirens are employed during the trip, the risk rises to 16.5 per 100,000 transports. To reduce these fatalities and improve passengers' security on the road, there is a need to develop a system that may overcome the existing problems associated with the passing of EVs.

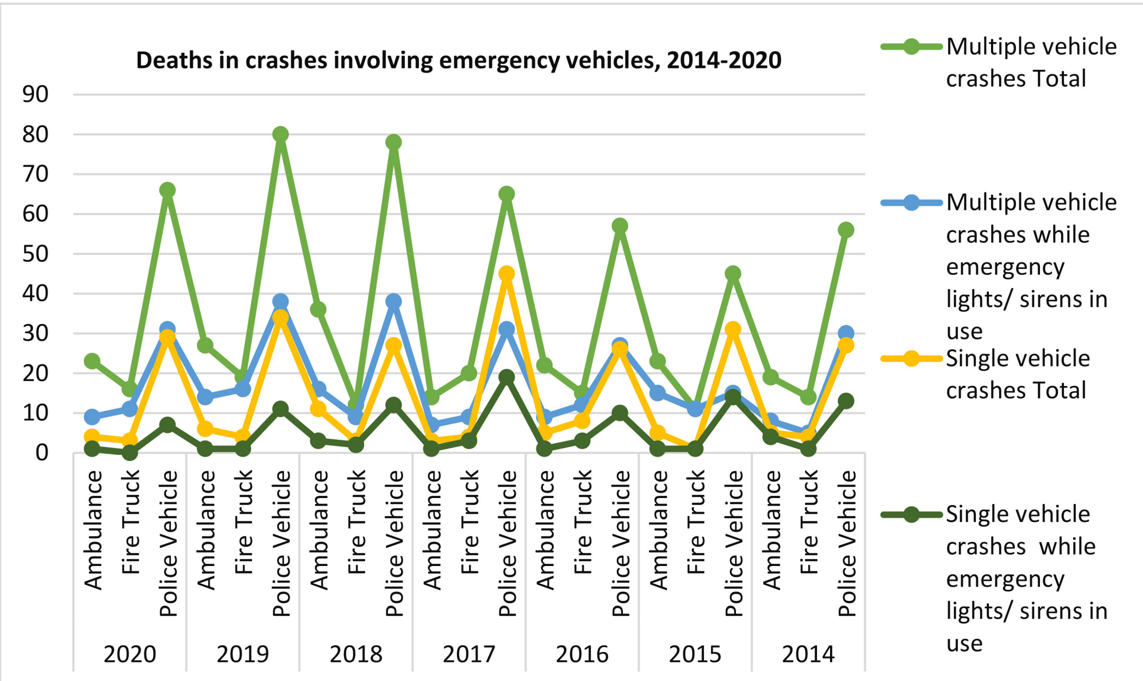

Figure 1. Death statistics of crashes involving EVs [7].

This paper provides a solution for passing EVs in various situations. The EV in the proposed system is an intelligent vehicle that can sense its surroundings with minimum human interference. EVs are prioritized based on their criticality, and this information is stored on the cloud which is accessible to RSU. When multiple EVs arrive at the intersection simultaneously, the decision is based on the priority of the EV. If both vehicles with the same priority reach the intersection simultaneously, the proposed system calculates the distance between EV and destination and gives priority to the EV with more distance. To pass EVs in congestion, a shoulder lane will be used to move vehicles to vacate EV space. Thus, the proposed solution provides a significant contribution, as follows:

- Explore the existing traffic management strategies and find the gaps for routing EV, as described in Section 2.
- Prioritize EVs based on their criticality, propose a model for passing single and multiple EVs based on priority and distance, and propose multiple algorithms for passing EVs with the same priority and varying priority; all this is discussed in Section 3.
- Mathematical modeling of the proposed system is provided in Section 4.
- Evaluating the model and algorithms mathematically is in Section 5.

Before moving towards the proposed solution, we provide insights into existing solutions for managing EVs and for identifying the existing gaps in the next section. Table 1 provides the list of abbreviations used in this research paper for a better understanding.

Table 1. List of abbreviations used in this study.

Abbreviations	Used for	Abbreviations	Used for
EVMS	Emergency vehicle management solution	DICA	DTOT-based Intersection Control Algorithm
EV	Emergency vehicle	DTOT	Discrete-Time Occupancies Trajectory
RSU	Roadside unit	TMC	Traffic Management Center
CAD	Computer aided dispatch	ITS	Intelligent traffic system
GPS	Global positioning system	EVPS	Emergency Vehicle Priority System
NSC	national safety council	IoT	Internet of Things
RFID	Radio Frequency Identification	PLC	Programmable Logic Controller
SUMO	Simulation of Urban Mobility	TLS	Traffic Light Systems
BFMG	Budget and Financial Management Guidance	SL	Shoulder Lane
DSRC	Dedicated short-range communications	OBUs	On-board units
ED	Euclidian distance	MD	Manhattan distance
CD	Canberra distance	L&L	Latitude and longitude

2. Literature Review

This section explores the existing traffic management strategies for routing EVs and provides a comparative analysis of these studies to identify the possible research gaps related to traffic management based on traffic accidents.

In paper [5], an evolutionary method is suggested to handle the optimization issue of finding the ideal vehicle sequence that provides precedence to EVs. Comparisons with the DTOT-based Intersection Control Algorithm (DICA) and a reactive traffic signal algorithm and comprehensive simulations are used to verify the effectiveness of the suggested technique for expediting EVs' crossings. There is no discernible impact on regular vehicle performance from the suggested genetic algorithm, which dramatically reduces the journey durations of EVs in light and medium traffic loads without degrading their performance.

When an EV approaches a junction, a Radio Frequency Identification (RFID)-based system is suggested, as in paper [9], to monitor and regulate traffic signals so that the vehicle may easily exit the traffic jam and go to its destination. An experimental setup utilizing Arduino and light-emitting diode (LED) displays was chosen to mimic a real-world traffic situation to test the suggested framework. Simulated findings show that the authors' EV detection and management framework performs better than the current system in peak hours.

Paper [10] presents a method for scheduling EVs in traffic. The technology integrates optical sensing, vehicle counting, and time-sensitive warning transmission inside the sensor network to calculate the distance between EVs and junctions. An EV and intersection are measured with visual data and compared using Euclidean, Manhattan, and Canberra approaches. There is evidence that the Euclidean distance is superior to other distance measuring methods and may be used in real-time.

Usually, the department establishes a target travel time for an EV, but it is difficult to meet in most cases. Study [11] proposes a novel Intelligent Traffic System (ITS) that takes into account the priorities of EVs depending on the kind of event and a mechanism for detecting and reacting to traffic signal hacking to address this problem directly. A simulation experiment was carried out. For EVs, this system outperforms both already operating and newly planned ITS in terms of congestion avoidance and trip time savings. Budget and Financial Management Guidance (BFMG)'s reaction time objective for both conventional and hacked traffic lights is met by the proposed method.

Based on the kind and severity of an event, paper [12] introduces an Emergency Vehicle Priority System (EVPS) that determines an EV's priority based on its type and severity and estimates the number of required signal interventions while taking into account their influence on traffic on nearby roads. A novel technique to estimate the number of green signal interventions needed to achieve the fastest incident response time while also minimizing the effect on others is presented, along with an explanation of how EVPS arrives at the priority code. Simulation of Urban Mobility (SUMO) uses actual traffic data from sensors in Melbourne, Australia to create a simulation model. The proposed system's findings suggest that implementing the recommended number of interventions may dramatically shorten its time to respond to an emergency. Paper [13] proposes a novel way for managing EV traffic using the Internet of Things (IoT). With the traffic signal controller positioned at a junction, EVs may notify it about arrival to manage traffic. The EV's passengers must utilize the Android app installed on their phones to communicate with the traffic controller hardware. A cutting-edge approach is also suggested in this paper for automatically managing traffic.

An automated traffic management system with EV control is the focus of paper [14]. In the event of an emergency signal, the system automatically maintains the condition of the regular sequence. It provides the associated route a green light signal for the duration of the signal. When the emergency signal is no longer strong enough, the system immediately returns to the previously saved condition of the usual sequence. RFID technology is used to identify emergency signals. Programmable Logic Controller (PLC) ladder logic is used to create the system. PLC simulation was used to implement the system. After that, it was put to the test in a working prototype of the hardware. The results of the tests showed that the original code does not need to be altered to make it work in the real world. This PLC-based system may replace the standard traffic control system. A complete city's traffic system may be dynamically managed using a PLC system with extra input and output modules.

An IoT-based platform for EV priority and self-organized traffic control at crossings is presented in the article [15]. A new platform and protocol known as EVP-STC is proposed, which consists of three basic components. First, the junction controller, which is mounted at traffic signals, gathers EV location data and vehicle density data on each road segment approaching an intersection. Based on this real-time traffic data, the junction controller subsequently makes adjustments to the timing of the traffic lights. A second system with force resistive sensors is put at each road section to identify automobiles. ZigBee is used to communicate with the intersection controller to send the information it has detected. EVs are equipped with a third system that gives the junction controller with GPS coordinates so that emergency vehicles do not have to wait at intersections. The simulation results show the proposed platform's ability to reduce overall delays, lane opening times, and waiting times for emergency vehicles.

Paper [16] claims that a variety of options for dealing with EVs have been proposed, including special lanes for EVs in cities. However, EVs sometimes find it difficult to meet their optimal goal timings even with these lanes. Existing technology may be used in conjunction with present infrastructure to create an ITS that can assist in tackling this issue. TLS, RFID, and IoT are compared in this research. According to the study's findings, using IoT, data on traffic congestion may be gathered more rapidly and correctly. A "User Interface" for mobile applications may be used to identify congestion in various locations, as well as give users other paths. These strategies are designed to provide drivers with more information about traffic and road conditions so that they may make better-informed decisions. Through ITS, EVs may be given precedence over non-emergency cars.

Paper [17] proposes a way to alleviate transportation congestion in a smart city environment. According to the proposed solution, every junction of the road should have a sensor node and a camera installed to collect real-time traffic data. Central servers may be used for a variety of purposes, including storing traffic data and providing high-priority vehicles with updates on their status and the best possible alternative routes to avoid traffic jams in real-time.

The foregoing discussion and analysis of other current research, such as [4,18–21], reveal that passing EVs at intersections is a major difficulty, particularly in crowded metropolitan locations. Table 2 compares prior research to understand the gaps better. Researchers and practitioners have offered a variety of remedies, yet the issue persists. In the next section, an autonomous traffic management system for EVs is offered as a solution to this problem.

Table 2. Research Gap based on Literature.

Pap#	Problem Discussed	Proposed Solution	Technology Used	Contribution	Research Gap
[5]	EV crossing at the intersection	Reactive DICA	Genetic algorithm	Optimize the sequence of vehicles reducing travel times of EVs minimal effect on the performance for normal vehicles	Only address autonomous vehicles
[9]	EV crossing at peak hours	Emergency vehicle detection and management framework	RFID-based system	Better performance in detection as well as management of emergency vehicle	The Lane clearing process in case of congestion is not discussed
[10]	Distance measure of EV from the intersection	Visual sensing techniques developed VANET model	PE-MAC protocol, NS-2	Emergency vehicle information is measured accurately, the measured information is delivered to the TMC in less time	Distance measurement in bad weather and high traffic conditions
[11]	Meeting the target travel time of an EV	An Intelligent Traffic Signal system	IoT	Meet the target travel time of an emergency vehicle set by the Department	Do not discuss the normal traffic flow and the process of vacating lane for EV
[12]	Sending EV quicker to the incident place	EVPS	SUMO	Recommend an appropriate intervention number that assists an EV in reducing response time	Do not discuss the process of vacating lane for EV
[13]	Better management of EV traffic	Prototype	IoT	Management of EV in an effective way	Do not discuss the normal traffic flow and the process of vacating lane for EV
[14]	Emergency Vehicle Detection and Management	Prototype	RFID, High Priority Encoder	Emergency Vehicle Detection and Management system	Validation in a real setting is missing, Do not discuss the process of vacating lane for EV
[15]	EV priority and self-organized traffic control	EVP-STC protocol	IoT Zigbee GPS	EVs do not have to wait at intersectionsreduce overall delays, lane opening times, and waiting times for emergency vehicles	Do not discuss the normal traffic flow and the process of vacating lane for EV during congestion

Table 2. *Cont.*

Pap#	Problem Discussed	Proposed Solution	Technology Used	Contribution	Research Gap
[16]	Traffic management	Review of existing technologies used in ITS	RFID tag, magnetic sensor, IR sensor, WSN, IoT, etc.	Provide the comparison of existing ITS along with their pros and cons	Only suggestions are mentioned without validation
[17]	Congestion handling and controlling of EVs	The idea of a self-configurable system is proposed	PIC microcontroller, RF transceiver, ultrasonic sensors	Provide the comparison of existing ITS along with their pros and cons, The idea of a self-configurable system is proposed	Validation is missing

3. Proposed Methodology

In major cities, traffic congestion is a problem, and EVs have to contend with it as well. For the sake of preserving human life, a delay in the arrival of an EV is sometimes unavoidable. To address this issue, we propose a solution named EVMS, as illustrated in Figure 2.

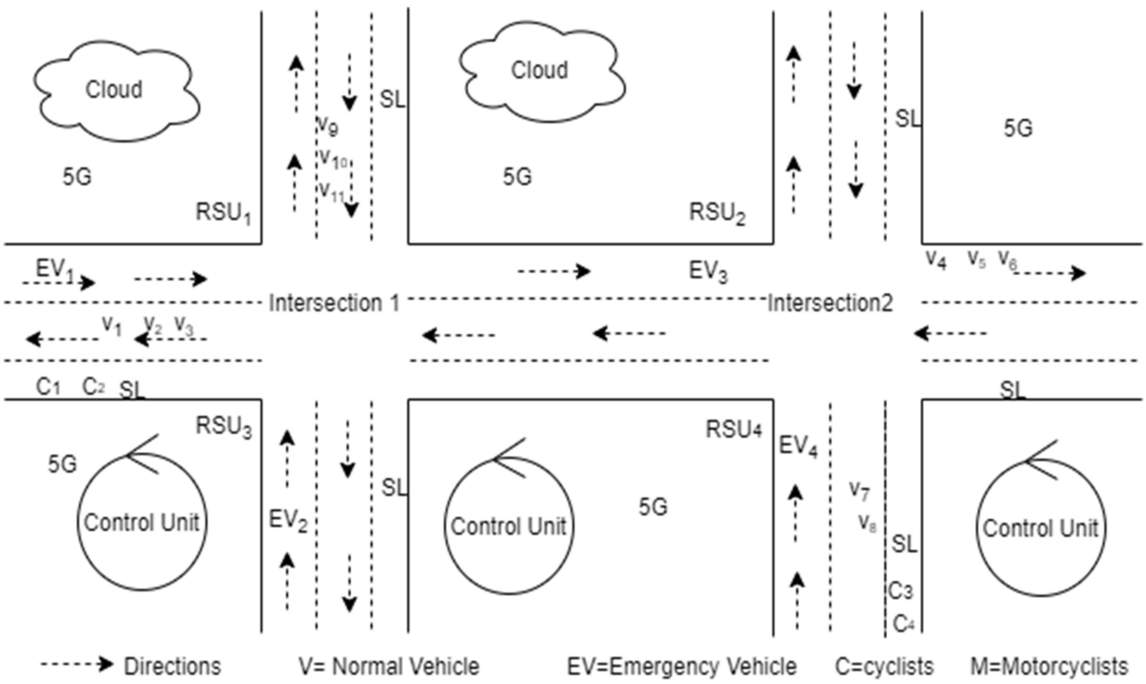

Figure 2. An overview of EVMS.

3.1. Components of the EVMS

The proposed EVMS consists of seven components along with priority rules. The details of these components are discussed in the following.

3.1.1. Emergency Vehicle (EV)

An EV is a vehicle that emergency personnel use to react to a situation. These vehicles are normally run by recognized agencies, which are often government entities, but they may also be non-governmental groups or commercial firms that have been specially authorized by legislation [22]. To get to their destinations as quickly as possible, EVs may be excluded from several standard road restrictions, such as passing through a junction while the traffic signal is red or exceeding the speed limit. We categorize EVs into three groups in our suggested solution: medical, firefighting, and law enforcement [23]. The EVs in the EVMS are equipped with IoT sensors that collect real-time data of EVs and transmit it to the cloud using 5G.

3.1.2. Shoulder Lane (SL)

In the proposed system, every crowded road has an SL that is reserved for cyclists and motorcyclists in normal situations. In the case of an EV entering the intersection range, the lane adjacent to the SL is shifted towards the SL to make room for the EV. The SL is narrower than other lanes [24]. The benefits of having an SL are many [25]: first, cyclists and motorcyclists may quickly move to the pavement or other lanes in case of EV arrival. Secondly, the separate lane for cyclists and motorcyclists will help in reducing congestion in the other lanes. Third, the EV can pass through the intersection even during congestion.

3.1.3. Roadside Unit (RSU)

An RSU is a Dedicated Short-Range Communications (DSRC) transmitter that is attached to a roadway or pedestrian walkway. Any vehicle or hand-carried unit may have an RSU attached to it; however, the RSU can only work while the vehicle or hand-carried unit is at a standstill. In addition, an RSU operating under this section is limited to the area in which it has been granted a license to do so [26,27]. RSUs that are small enough to be carried around and not interfere with a fixed installation are allowed to function. For example, an RSU transmits or exchanges data with other Onboard units (OBUs) in its communications area through radio waves. OBUs in an RSU's communications zone may also request channel allocations and operational instructions from the RSU. In the EVMS, RSUs will be used by the control unit to manually interfere with traffic signals when there is an EV on the road [26,28]. It will gather traffic data from a static sensing region along a road and communicate it to traffic control devices and a central traffic management center [28].

3.1.4. Control Unit

The control unit in the EVMS collects vehicle information from the cloud and takes action accordingly. It is responsible for managing road traffic, pedestrian traffic, and EVs on the road.

3.1.5. IoT

Connected physical items (or groups of such objects) with sensors, computing power, software, and other technologies that exchange data via the Internet or other communications networks are known as IoT [19,29]. In the EVMS, these IoT sensors are embedded in the EV for the collection of real-time data, and they transmit it on the cloud using 5G from where the control unit accesses it to take decisions.

3.1.6. Cloud

IoT devices attached/embedded in EVs generate a huge amount of real-time data, the cloud is used for the storage of this data [30,31] that can be further used for decision making. The EVMS uses the public cloud for storing EV information, road traffic, and other related data.

3.1.7. 5G

As the name suggests, 5G refers to the next generation of mobile networks, which will be able to provide consumers with faster peak data speeds, ultra-low latency, greater dependability, huge network capacity, better availability, and a more consistent user experience [32,33]. The proposed system leverages the benefits of 5G and uses it for data transmission between IoT devices and the cloud, and from the cloud to control units.

3.1.8. Priority Rules

EVs are of different types and have various levels of emergency. In the EVMS, medical vehicles have the highest priority [34], then comes the firefighting vehicles, followed by law-enforcement vehicles. Medical EVs (MEVs) are further categorized into three levels named as: critical, less critical, and moderate. A critical EV is one which carries a critical patient, a less critical medical EV is one which moves a less critical patient (such as a delivery case, etc.) from one place to another, while a moderate EV is one that is empty or carries a dead body or medical equipment that are not so urgent. The firefighting EVs (FEVs) and law enforcement EVs (LEVs) will have only one priority that is a high priority. Equations (1) and (2) show the priority of the three types of EVs.

If a medical EV is critical or less critical, then the priority will be as shown in Equation (1).

$$P(\mathrm{MEV}) > P(\mathrm{FEV}) > P(\mathrm{LEV}) \tag{1}$$

If a medical EV is moderate, then the priority of EVs will be as shown in Equation (2).

$$P(\mathrm{FEV}) > P(\mathrm{LEV}) > P(\mathrm{MEV}) \tag{2}$$

where MEV represents medical EV, FEV represents firefighting EV, and LEV represents law enforcement EV.

3.2. Working Mechanism of EVMS

The roads which are usually congested in peak hours will have a shoulder lane for traffic adjustment in case of EVs on the road. When any EVs approach the junction, the data of that EV (EV number, destination, priority, velocity, speed) will be sent to the cloud through IoT sensors using 5G Internet service. The control unit will fetch the data from the cloud and will come to know about the details of the EV; if there is only one EV on the road, the normal traffic on the road will move to the SL and space will be vacated for the EV to move towards the junction. The control unit will turn red lights into green lights until the EV passes the junction successfully. Once the EV passes the junction, the traffic signal will operate normally. If two EVs are coming from varying directions, the decision of passing the EV through the junction will be decided based on the priority of the EVs as mentioned in Equations (1) and (2). If both EVs with the same priority are coming from varying directions, the control unit will give priority to the EV that is further from its destination. If two EVs are coming from varying directions but their priority is different, then the vehicle with high priority will be given preference according to Equations (1) or (2). Algorithms 1 and 2 show the working of a proposed approach.

Algorithm 1 shows the working of the EVMS in the case of 1 or more EVs, while Algorithm 2 shows how medical EVs with varying priorities will be passed through the junction. As mentioned in the description of priority rules, MEVs have three priorities named critical, less critical, and moderate. Algorithm 2 shows the process of passing EVs with varying priorities of MEVs

Case 1 of Algorithm 1 is modeled in Figure 3, where two MEVs are coming from different directions in such a way that the signal of both sides cannot be green at the same time. Further, both EVs have the same priorities; in such a situation, the EV with more distance from its destination will be given priority.

Algorithm 1. Passing of EV with varying priority from intersection.

$Let\ \ \mathcal{EV} = emergency\ vehicle, \mathcal{EV}\rangle\lceil = emergency\ vehicle\ id,\ \mathcal{V}\dashv\sqcup = vehicle\ arrival\ time,$
$\mathcal{V} = vehicle,\ \mathcal{V}\rangle\lceil = vehicle\ id, \mathcal{V}\updownarrow = velocity,\ \ \lceil = distance,\ c = case$
$\mathcal{MEV} = medical\ emergency\ vehicle,\ \mathcal{FEV} = firefighting\ emergency\ vehicle$
$\mathcal{LEV} = law - enforcement\ emergency\ vehicle, \mathcal{P}\nabla = priority,$
$\mathcal{K} = point\ of\ intersection, \mathcal{NV} = normal\ vehicle,\ \mathcal{SL} = shoulder\ lane,$

1. *begin*
2. $for\ each\ \ \mathcal{V}i\ passing\ through\ \mathcal{K}$
3. {
    $if\ (\mathcal{V}\rangle\lceil = \mathcal{EV}\rangle\lceil)$
    $Then\ count\ N(\mathcal{EV})$
        {
            $if\ (N(\mathcal{EV}) > 1))go\ to\ step\ 4$
            else
                $broadcast\ \mathcal{EV}(\mathcal{EV}\rangle\lceil,\ \mathcal{V}\updownarrow,\ \lceil,\ \mathcal{P}\nabla)$
                $verify\ passage\ (\mathcal{EV}\rangle\lceil)$
                $Move\ \mathcal{NV}\ to\ \ \mathcal{SL}$
                $checksignal(R, G, Y)$
                $Set\ signal\ to\ green\ on\ arrival\ of\ \mathcal{EV}\rangle\lceil$
                    $passout(\mathcal{EV}\rangle\lceil)$
                $send\ data\ to\ next\ \mathcal{K}$
        }
    else
    $Resume\ normal\ traffic$
   }
4. $switch\ (c)$

$case\ 1\ :\ there\ are\ two\ \mathcal{EV}s\ named\ as\ \mathcal{EV}i\ and\ \mathcal{EV}\ j$
    $if\ (\mathcal{P}\nabla(\mathcal{EV}i) = \mathcal{P}\nabla(\mathcal{EV}j)\ \&\&\ \mathcal{V}\dashv\sqcup\ (\mathcal{EV}i) = \mathcal{V}\dashv\sqcup\ (\mathcal{EV}j))$
    $then$
        $calculate\ \lceil\ of\ each\ \mathcal{EV}\ from\ destination$
        $if\ d(\ \mathcal{EV}i > \mathcal{EV}j)$
            $passout\ \mathcal{EV}i\ at\ \mathcal{K}\ first$
            $else$
            $passout\ \mathcal{EV}j\ at\ \mathcal{K}\ first$
            $send\ data\ to\ next\ \mathcal{K}$
    $break$

$case\ 2\ :\ there\ are\ two\ \mathcal{EV}s\ named\ as\ \mathcal{EV}i\ and\ \mathcal{EV}\ j$
    $if\ (\mathcal{P}\nabla(\mathcal{EV}i) \neq \mathcal{P}\nabla(\mathcal{EV}j)\ \&\&\ \mathcal{V}\dashv\sqcup\ (\mathcal{EV}i) = \mathcal{V}\dashv\sqcup\ (\mathcal{EV}j))$
    $then$
        $if\ \mathcal{P}\nabla(\ \mathcal{EV}i > \mathcal{EV}j)$
        $passout\ \mathcal{EV}i\ at\ \mathcal{K}first$
        $else$
        $passout\ \mathcal{EV}j\ at\ \mathcal{K}\ first$
        $send\ data\ to\ next\ \mathcal{K}$
    $break$

5. $Resume\ normal\ traffic$
6. *end*

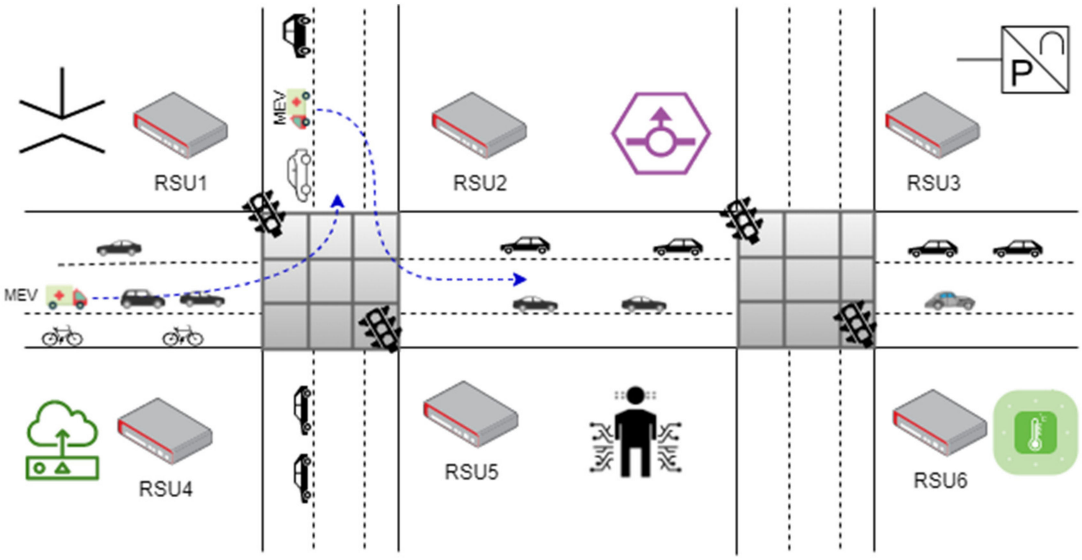

Figure 3. Case 1, mentioned in Algorithm 1 (when both vehicles have the same priorities).

Figure 4 models the situation in which two EVs, FEV and MEV on junction K1, and PEV and FEV on junction K2, are coming from varying directions and cannot cross simultaneously. In such a situation, the decision will be taken based on the process mentioned in Algorithm 2. The detailed working of both algorithms is also shown in the flow chart of Figure 5, where VAT refers to vehicle arrival time.

Algorithm 2. Passing of EVs based on priority.

---

1. $begin$
2. $for\ multiple\ \mathcal{EV}s\ passing\ through\ \mathcal{K}$
3. $switch\ (c)$

$case\ 1\ :\ both\ \mathcal{EV}i\ and\ \mathcal{EV}j\ are\ \ \mathcal{MEV}$
$if\ (\mathcal{P}\nabla(\mathcal{EV}i) = \mathcal{P}\nabla(\mathcal{EV}j)\ \&\&\ \mathcal{V}\dashv\sqcup\ (\mathcal{EV}i) = \mathcal{V}\dashv\sqcup\ (\mathcal{EV}j))$
$then$
$calculate\ \lceil\ of\ each\ \mathcal{EV}\ from\ destination$
$if\ d(\ \mathcal{EV}i > \mathcal{EV}j)$
$passout\ \mathcal{EV}i\ at\ \mathcal{K}\ first$
$else$
$passout\ \mathcal{EV}j\ at\ \mathcal{K}\ first$
$send\ data\ to\ next\ \mathcal{K}$
$break$
$case\ 2\ :\ \mathcal{EV}i\ is\ \mathcal{MEV}\ while\ \mathcal{EV}j\ is\ \ \mathcal{FEV}$
$if\ (\mathcal{P}\nabla(\mathcal{EV}i)\ is\ critical\ or\ less\ critical)$
$then$
$passout\ \mathcal{EV}i\ at\ \mathcal{K}$
$passout\ \mathcal{EV}j\ at\ \mathcal{K}$
$else$
$if\ (\mathcal{P}\nabla(\mathcal{EV}i)\ is\ moderate)$
$then$
$passout\ \mathcal{EV}j\ at\ \mathcal{K}$
$send\ data\ to\ next\ \mathcal{K}$
$break$

$case\ 3\ : \mathcal{EV}i\ is\ \mathcal{MEV}\ while\ \mathcal{EV}j\ is\ \ \mathcal{LEV}$
$if\ (\mathcal{P}\nabla(\mathcal{EV}i)\ is\ critical\ or\ less\ critical)$
$then$
$passout\ \mathcal{EV}i\ at\ \mathcal{K}$
$passout\ \mathcal{EV}j\ at\ \mathcal{K}$
$send\ data\ to\ next\ \mathcal{K}$
$else$
$if\ (\mathcal{P}\nabla(\mathcal{EV}i)\ is\ moderate)$
$then$
$passout\ \mathcal{EV}j\ at\ \mathcal{K}$
$send\ data\ to\ next\ \mathcal{K}$
$break$
4. $Resume\ normal\ traffic$
5. end

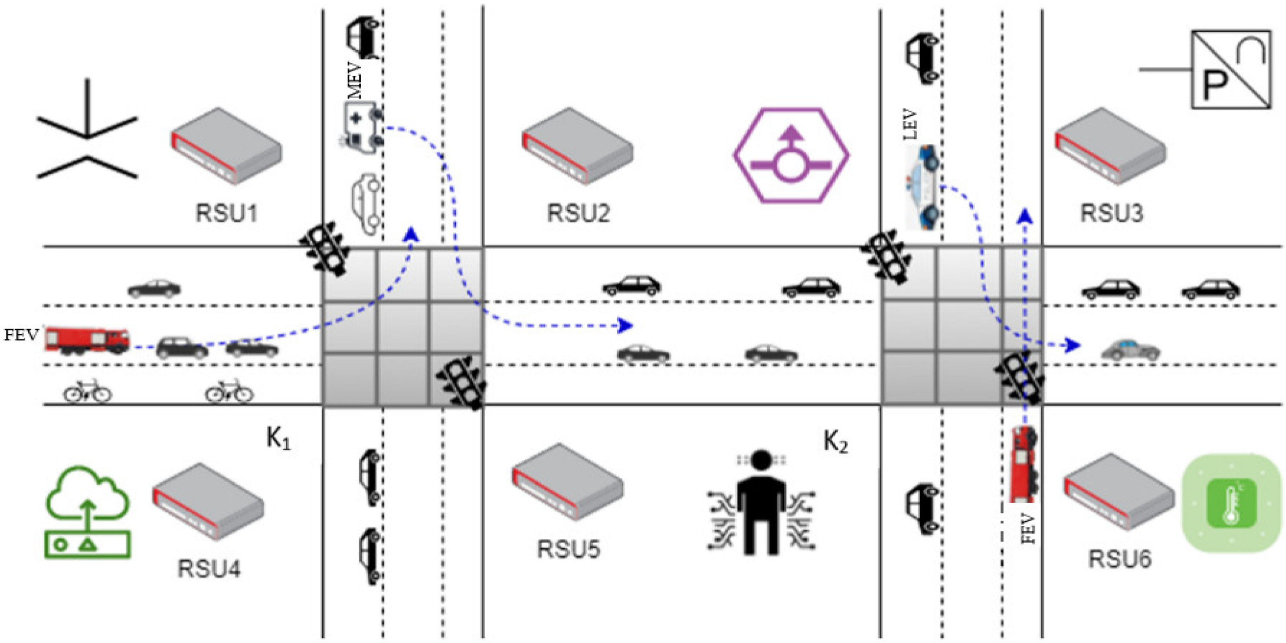

Figure 4. Case 2, mentioned in Algorithm 1 (when both vehicles have the same priorities).

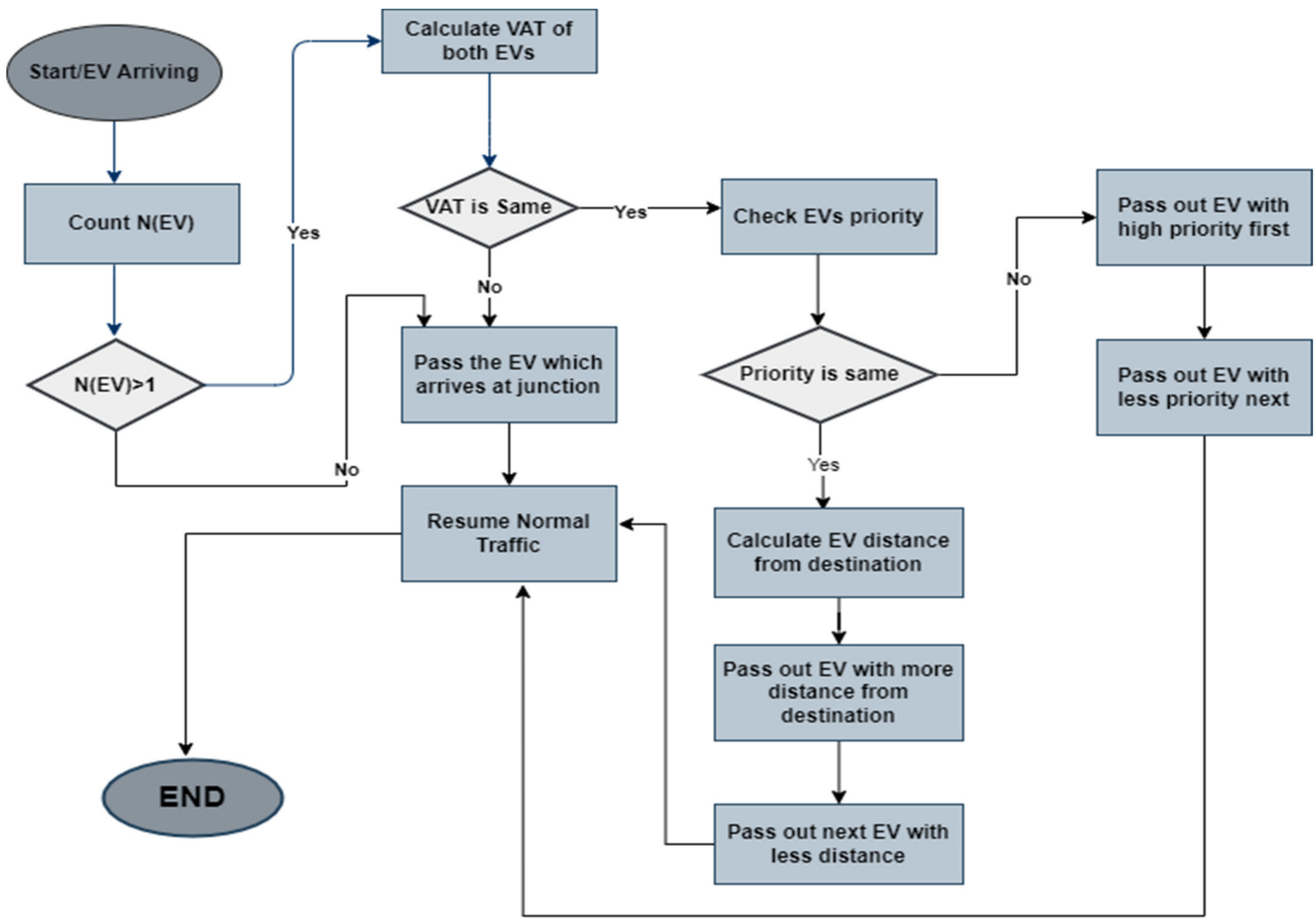

Figure 5. Working of EVMS.

4. Mathematical Modeling of EVMS

It is essential to investigate the traffic at the intersection in order to pass the EV quickly and preserve social order. It is practical to examine the traffic flow of intersections since the capacity of a junction directly impacts the efficiency of a road network. Analysis of the traffic conditions of junctions was carried out by setting up a mathematical model and applying it to the values of road statistics. Below we describe the details of the mathematical model. Table 3 shows the symbols and notations used throughout the paper.

Table 3. Symbols and notations used.

Symbols	Used for	Symbols	Used for
vl	Velocity	$\mathcal{A}$	Time taken in adjusting SL traffic
$\overline{vl}$	Average velocity	$\mathbb{S}$	Time taken in signal switching
α	Arrival rate of EV	$\mathcal{TT}$	Total time needed to adjust and pass EV
t_p	Time taken to entertain an EV	Δx	Change in displacement
Wv	Waiting time for normal vehicles	u	Initial velocity
ρ	Performance parameters	∂	Acceleration
$\mathcal{M}$	Time taken by normal vehicle to move to SL	t	Time
m/s	Meter per second	m/s^2	Meter per second squared
n	Normal vehicles on the road	t_n	Time taken to entertain normal vehicles

To determine the precise moment of an EV's arrival at a junction, the distance between the EV and the intersection must be calculated. Furthermore, when there are multiple EVs with the same priority on a junction, the distance to the destination will aid in the prioritization of the EVs. As a result, the distance calculation is a key aspect of the EVMS. The distance from an EV to an intersection in the EVMS is calculated using the three most commonly used techniques of distance calculation, named Euclidean distance (ED), Canberra distance (CD), and Manhattan distance (MD). The formulas for the calculation of ED, CD, and MD are given in Equations (3)–(5), respectively. The distance from an EV to its destination is calculated by computing the real length of the route to the destination using a map distance calculator. Further, the symbols used in the mathematical model are mentioned in Table 3. The steps followed for this mathematical model are mentioned in Figure 6.

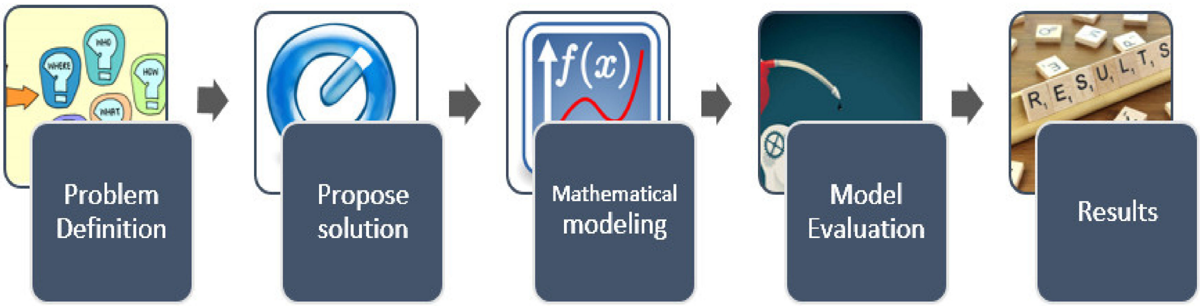

Figure 6. Steps of Mathematical modeling to validate the proposed model.

ED formula is used to compute the distance between two or more points [35,36], as given in Equation (3). Let a = (x_1, y_1) and b = (x_2, y_2) be two points on a plane, then the ED between two points will be calculated as shown in Equation (3).

$$\mathrm{ED} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3}$$

In the EVMS, EVs will be equipped with GPS that will calculate the latitude and longitude (L&L) values to determine the location of a vehicle. The distance from the intersection will be calculated by using the L&L of the EV and intersection.

The CD distance is a weighted variation of the MD that is often used for data that is dispersed around a point of origin [37,38]. The formula for calculating CD by using the above-mentioned two coordinates is as given in Equation (4).

$$\mathrm{CD} = \frac{|x_1 - y_1|}{|x_1| + |y_1|} + \frac{|x_2 - y_2|}{|x_2| + |y_2|} \tag{4}$$

The MD between two points is calculated by taking the absolute sum of the differences between the coordinates [39,40]. The formula for calculating MD by using the above-mentioned two coordinates is as given in Equation (5).

$$\mathrm{MD} = |x_1 - y_1| + |x_2 - y_2| \tag{5}$$

As the vehicles do not follow the same speed on the road due to congestion on the road and other traffic-related issues, instead of speed, the velocity of EVs will be calculated [41,42] using the general formulae of velocity as shown in Equation (6). Let *vl* represent velocity, then average velocity can be computed as

$$\overline{vl} = \frac{\Delta x}{\Delta t} \tag{6}$$

where Δx is displacement while Δt represents the change in time.

The performance parameter of EVs will be calculated by using the formula of Equation (7).

$$\rho(\mathrm{EV}) = \frac{\alpha}{t_p} \tag{7}$$

The average number of EVs on the road can be measured using the formula mentioned in Equation (8), where the average number of EVs is equal to the average number of EVs in the line plus those who are being served.

$$Avg(\mathrm{EV}_i) = \frac{\alpha}{\alpha - t_p} \tag{8}$$

The aim of the EVMS is not only to pass the EV efficiently but to minimize the average waiting time for the normal vehicles on the road. The waiting time for normal vehicles on the road will be measured using the formula mentioned in Equation (9).

$$Wv = \frac{n}{n\left(n - t_p\right)} \tag{9}$$

On the arrival of an EV at an intersection, normal traffic moves towards the SL to vacate space for the EV, and the red light is turned to green. The whole time it takes to vacate space for the EV and pass the signal onto the EV can be measured using Equation (10).

$$\mathcal{TT} = f(\mathcal{M},\ \mathcal{A},\ \mathcal{S}) \tag{10}$$

The objective function in our case will be to minimize TT and Wv; thus, the objective function in our case will be as mentioned in Equations (11) and (12).

$$\text{Objective function1 (OBF1)} = Minimize(\mathcal{TT}) \tag{11}$$

$$\text{Objective function2 (OBF2)} = Minimize(Wv) \tag{12}$$

In the case of OBF1, there are three determinants of $\mathcal{TT}$ as mentioned in Equation (10), the individual impact of each determinant can be found using the formulas mentioned in Equations (13)–(15).

The individual impact of each determinant can be found by taking the partial derivatives of Equation (10) as given in Equations (13) to (15).

$$\frac{\partial}{\partial\mathcal{M}}(TT) = \frac{\partial}{\partial\mathcal{M}}(\mathcal{M}) + \frac{\partial}{\partial\mathcal{M}}(\mathcal{A}) + \frac{\partial}{\partial\mathcal{M}}(\mathcal{S}) + \varepsilon \tag{13}$$

$$\frac{\partial}{\partial\mathcal{A}}(TT) = \frac{\partial}{\partial\mathcal{A}}(\mathcal{M}) + \frac{\partial}{\partial\mathcal{A}}(\mathcal{A}) + \frac{\partial}{\partial\mathcal{A}}(\mathcal{S}) + \varepsilon \tag{14}$$

$$\frac{\partial}{\partial\mathcal{S}}(TT) = \frac{\partial}{\partial\mathcal{S}}(\mathcal{M}) + \frac{\partial}{\partial\mathcal{S}}(\mathcal{A}) + \frac{\partial}{\partial\mathcal{S}}(\mathcal{S}) + \varepsilon \tag{15}$$

In the same way, the determinant of average weighting times for normal vehicles and their impact can be found in Equations (16) and (17).

$$W_a = \frac{\alpha\left(\alpha - t_p\right)\frac{\partial}{\partial\alpha}(\alpha) - \alpha\frac{\partial}{\partial\alpha}\left(\alpha\left(\alpha - t_p\right)\right.}{\left[\alpha\left(\alpha - t_p\right)\right]^2} \tag{16}$$

$$W_{t_p} = \frac{\alpha\left(\alpha - t_p\right)\frac{\partial}{\partial t_p}(\alpha) - \alpha\frac{\partial}{\partial t_p}\left(\alpha\left(\alpha - t_p\right)\right.}{\left[\alpha\left(\alpha - t_p\right)\right]^2} \tag{17}$$

4.1. Data Analysis and Results

This section will evaluate the above mathematical model with the help of data. We assume the following:

- Each signal remains green for 30 s, yellow for 10 s, and red for 30 s.
- The traffic at the intersection comes from four directions (east, west, north, south).
- EV transfers its data to the cloud through built-in IoT sensors.
- Control unit fetches EV data and acts accordingly.
- The time to adjust normal traffic to SL to vacate space for EV is assumed to be 3 s.
- The average time to entertain a normal vehicle is 15 s.

4.1.1. Case 1: There Is Only One EV

Let's evaluate the above parameters by using the data mentioned in Table 4 and assuming that only one EV is approaching the intersection. The control unit fetches EV coordinates from the cloud and calculates the distance of the EV from the intersection to know the expected arrival time for the EV.

Table 4. Road statistics (case 1).

Parameters	Values	Source
Number of normal Lanes	3	RSU
Vehicle accumulated	14	RSU
SL status	Moderately occupied	RSU
Signal status while EV arrives	Red	Control unit
EV source coordinates	29.989808, 40.229722	GPS
Intersection coordinates	29.977561, 40.214250	RSU
EV speed	27 m/s	IoT sensor
Signal switching time	2 s	RSU

The distance is calculated as below

$$(\mathrm{X}_1, \mathrm{Y}_1) = (29.989808, 40.229722)$$

$$(\mathrm{X}_2, \mathrm{Y}_2) = (29.977561, 40.21425)$$

$$d = \sqrt{(29.977561 - 29.989808)^2 + (40.21425 - 40.229722)^2}$$
$$d = 0.0197$$

The distance can also be calculated using the CD or MD method. Once the distance is known, the velocity of the EV will be calculated to know the exact arrival time, the reason for calculating velocity instead of speed is that speed varies with regards to congestion and other road situations. Velocity will be calculated by an IoT-based velocity sensor using distance covered. To evaluate the performance measure of the EV, the performance parameter will be calculated using the formula mentioned in Equation (7). In our case, the value of α is 1 as there is only one EV. The value of t_p is the sum of $\mathcal{A}$ and $\mathbb{S}$, which is equal to 5 in our case, so the performance parameter will be calculated as

$$\rho(\mathrm{EV}) = \frac{\alpha}{t_p} = \frac{1}{5} = 0.2$$

For normal and stable systems, the value of t_p will always exceed the arrival rate; therefore, the value of $\rho(\mathrm{EV})$ will always be less than 1. The Wv will be calculated by using the formula of Equation (9), as follows:

$$Wv = \frac{n}{n(n - t_n)} = \frac{16}{16(16 - 15)} = \frac{16}{16} = 1$$

4.1.2. Case 2: There Are Two EVs with the Same Priority

By taking the same assumptions as above and the data in Table 5, we handle the situation of multiple EVs arriving at the same intersection at the same time with the same priority. In this case, the distance between the coordinates of the intersection and EV destination will be calculated for both EVs. The EV that is further from the destination will be given priority, as shown below.

Table 5. Road Statistics (Case 2).

Parameters	Values	Source
Signal status while EVs arrives	Red	Control unit
Intersection coordinates	29.977573, 40.214200	GPS
EV1 Destination address	Address	Google map
EV2 Destination address	Address	Google map
EV1 priority	High	IoT sensor
EV2 Priority	High	IoT sensor

Let d_1 be the distance of EV1 from the intersection to its destination and d_2 be the distance of EV2 from the intersection to its destination. The destination address of both EVs will be given as an input to the map distance calculator. It will calculate the distance of EV1 and EV2 from their destinations.

If $d_1 > d_2$, then EV1 will pass from the intersection first, otherwise EV2 will pass from the intersection.

4.1.3. Case 3: There Are Two EVs with Varying Priorities

In this scenario, the decision of passing EV will be based on the priority of the EV. IoT sensors will send the data of EV to the cloud, which will include the speed of the vehicle, velocity, priority, and destination address. The control unit will use this data to make decisions.

5. Discussion

Traffic congestion is a common problem in urban areas due to excessive population growth and the movement of people towards urban areas. This congestion creates problems for normal traffic as well as for pedestrians. However, one of the key challenges faced due to congestion is the passing of EVs on intersections during peak congestion hours. EVs sometimes carry critical patients or move to provide life safety and protection to people. The delay in passing EVs sometimes leads to fatal consequences. To overcome this issue, an EVMS is proposed based on modern cutting-edge technologies for the early transmission of EVs' information to the traffic control unit to take timely action. The EVMS in this study provides the idea of introducing an SL on congested roads to accommodate traffic in case of EV arrival. This SL will be used by cyclists and motorcyclists in normal conditions and will be used to adjust normal traffic flow in case of an EV on the road. Two algorithms are also proposed in this study to discuss the various cases of EVs, including only one EV on the junction, more than one EV on the junction with the same priority, and more than one EV on the junction with varying priorities.

The EVMS categorized EVs into three main types, namely medical EVs (ambulances), firefighting EVs, and law-enforcement EVs. The priority of these EVs is mentioned in Equations (1) and (2). The medical EV is further categorized into three levels based on the criticality of the patient inside: highly critical, less critical, and moderate. This priority helps the control unit in decision making while there is more than one EV at the intersection at the same time. According to the EVMS, all three EV types are equipped with GPS and IoT sensors to collect EV information. This collected information is sent to the cloud through IoT sensors using a 5G network. The control unit fetches the EV data from the cloud and makes a decision accordingly. In the case of a single EV at the intersection, the control unit turns the signal green when the EV reaches the intersection. In the case of more than one EV from varying directions at the intersection, the decision is either based on the priority of the EV or its distance from the destination. When both EVs have different priorities, the control unit passes the EV with high priority first. In the case of the same priority, the distance of EV from the destination is calculated and the EV with more distance from the destination is given priority.

To evaluate the proposed methodology, we have proposed various mathematical equations to calculate the EV parameters, including distance, velocity, performance parameters, the average number of EVs on the roads, and the effect of various factors. The proposed methodology is different from existing studies in different ways; firstly, existing studies do not consider the issue of multiple EVs with the same priority [43]. Secondly, they just assign a high priority to medical EVs without classifying it into various levels [4]. Table 6 compares the proposed system with a few existing studies on EVs. It shows that the proposed system provides a precise mechanism for handling EVs.

Table 6. Comparison with existing studies.

Paper	Single EV	Two EVs with the Same Priority	Two EVs with Varying Priority
[4]	Yes	No	Yes
[10]	Yes	No	Yes
[43]	Yes	No	No
EVMS	Yes	Yes	Yes

6. Conclusions and Future Work

The passing of EVs on intersections especially in urban areas is a key challenge due to traffic congestion. Various solutions address the problem of traffic congestion, but the passing of EVs with minimal impact on normal traffic is still a challenge. To overcome this issue, an EVMS has been proposed in this research; the proposed EVMS uses modern cutting-edge technologies for getting timely information about EVs. According to EVMS, EVs are equipped with GPS and IoT sensors, these sensors fetch EV data and transmit it to the cloud using 5G Internet service. The traffic management unit collects this data from the cloud and makes decisions accordingly. Two algorithms of passing single and multiple EVs with the same and different priorities have been developed. The EVMS is evaluated using mathematical modeling, and results suggest that the proposed system outperforms in the case of single and multiple EVs.

We are planning to apply the EVMS on real-time traffic data for a better evaluation in the future. We are also planning to strengthen this EVMS by adding more cutting-edge technologies to address the security and safety issues of the autonomous traffic system as well.

Author Contributions: Conceptualization, N.Z.J.; Data curation, M.H.; Formal analysis, N.Z.J.; Funding acquisition, M.F.A.; Investigation, M.H. and M.F.A.; Methodology, M.H.; Project administration, N.Z.J.; Resources, M.F.A.; Software, M.F.A.; Writing—original draft, M.H.; Writing—review & editing, N.Z.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Deanship of Scientific Research at Jouf University under grant No (DSR-2021-02-0327).

Data Availability Statement: Data will be available upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nieuwenhuijsen, M.J. Urban and transport planning pathways to carbon neutral, liveable and healthy cities; A review of the current evidence. *Environ. Int.* **2020**, *140*, 105661. [CrossRef] [PubMed]
2. AlAttar, M.; Al-Mutairi, N. Quantification of time and fuel losses due to daily traffic congestion in Kuwait. *Int. J. Crashworthiness* **2021**, *26*, 258–269. [CrossRef]
3. Younes, M.B.; Boukerche, A. An efficient dynamic traffic light scheduling algorithm considering emergency vehicles for intelligent transportation systems. *Wirel. Netw.* **2018**, *24*, 2451–2463. [CrossRef]
4. Sumi, L.; Ranga, V. Intelligent traffic management system for prioritizing emergency vehicles in a smart city. *Int. J. Eng.* **2018**, *31*, 278–283.

5. Lu, Q.; Kim, K.-D. A genetic algorithm approach for expedited crossing of emergency vehicles in connected and autonomous intersection traffic. *J. Adv. Transp.* **2017**, *2017*, 1–15. [CrossRef]
6. Bhate, S.V.; Kulkarni, P.V.; Lagad, S.D.; Shinde, M.D.; Patil, S. IoT based intelligent traffic signal system for emergency vehicles. In Proceedings of the 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 20–21 April 2018.
7. Traffic Safety Facts Annual Report Tables. Available online: https://cdan.nhtsa.gov/tsftables/tsfar.htm (accessed on 6 January 2021).
8. Watanabe, B.L.; Patterson, G.S.; Kempema, J.M.; Magallanes, O.; Brown, L.H. Is use of warning lights and sirens associated with increased risk of ambulance crashes? A contemporary analysis using National EMS Information System (NEMSIS) data. *Ann. Emerg. Med.* **2019**, *74*, 101–109. [CrossRef] [PubMed]
9. Naik, T.; Roopalakshmi, R.; Ravi, N.D.; Jain, P.; Sowmya, B.H.; Manichandra. RFID-based smart traffic control framework for emergency vehicles. In Proceedings of the 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 20–21 April 2018.
10. Nellore, K.; Hancke, G.P. Traffic management for emergency vehicle priority based on visual sensing. *Sensors* **2016**, *16*, 1892. [CrossRef] [PubMed]
11. Chowdhury, A. Priority based and secured traffic management system for emergency vehicle using IoT. In Proceedings of the 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, Morocco, 22–24 September 2016.
12. Karmakar, G.; Chowdhury, A.; Kamruzzaman, J.; Gondal, I. A smart priority based traffic control system for emergency vehicles. *IEEE Sens. J.* **2020**, *21*, 15849–15858. [CrossRef]
13. Tammishetty, S.; Ragunathan, T.; Battula, S.K.; Rani, B.V.; Ravi Babu, P.; Nagireddy, R.; Jorika, V.; Reddy, V.M. IOT-based traffic signal control technique for helping emergency vehicles. In *Proceedings of the First International Conference on Computational Intelligence and Informatics*; Springer: Berlin/Heidelberg, Germany, 2017.
14. Amir, S.; Kamal, M.S.; Khan, S.S.; Salam, K.M.A. PLC based traffic control system with emergency vehicle detection and management. In Proceedings of the 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kerala, India, 6–7 July 2017.
15. Khan, A.; Ullah, F.; Kaleem, Z.; Rahman, S.U.; Anwar, H.; Cho, Y.-Z. EVP-STC: Emergency vehicle priority and self-organising traffic control at intersections using Internet-of-things platform. *IEEE Access* **2018**, *6*, 68242–68254. [CrossRef]
16. Avatefipour, O.; Sadry, F. Traffic management system using IoT technology—A comparative review. In Proceedings of the 2018 IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, USA, 3–5 May 2018.
17. Ghosal, S.; Chatterjee, T. Controlling Emergency Vehicles During Road Congestion—A Survey and Solution. In *Computational Intelligence in Pattern Recognition*; Springer: Singapore, 2020; pp. 529–538.
18. Awan, K.A.; Din, I.U.; Almogren, A.; Kim, B.S.; Altameem, A. vTrust: An IoT-Enabled Trust-Based Secure Wireless Energy Sharing Mechanism for Vehicular Ad Hoc Networks. *Sensors* **2021**, *21*, 7363. [CrossRef]
19. Humayun, M.; Jhanjhi, N.; Hamid, B.; Ahmed, G. Emerging smart logistics and transportation using IoT and blockchain. *IEEE Internet Things Mag.* **2020**, *3*, 58–62. [CrossRef]
20. Humayun, M.; Jhanjhi, N.Z.; Alamri, M.Z.; Khan, A. Smart Cities and Digital Governance. In *Employing Recent Technologies for Improved Digital Governance*; IGI Global: Hershey, PA, USA, 2020; pp. 87–106.
21. Jadoon, G.; Din, I.U.; Almogren, A.; Almajed, H. Smart and agile manufacturing framework, a case study for automotive industry. *Energies* **2020**, *13*, 5766. [CrossRef]
22. Zeng, Z.; Yi, W.; Wang, S.; Qu, X. Emergency vehicle routing in urban road networks with multistakeholder cooperation. *J. Transp. Eng. Part A Syst.* **2021**, *147*, 04021064. [CrossRef]
23. Oubbati, O.S.; Lakas, A.; Lorenz, P.; Atiquzzaman, M.; Jamalipour, A. Leveraging communicating UAVs for emergency vehicle guidance in urban areas. *IEEE Trans. Emerg. Top. Comput.* **2021**, *9*, 1070–1082. [CrossRef]
24. Wu, J.; Kulcsár, B.; Ahn, S.; Qu, X. Emergency vehicle lane pre-clearing: From microscopic cooperation to routing decision making. *Transp. Res. Part B Methodol.* **2020**, *141*, 223–239. [CrossRef]
25. Li, B.; Zhang, Y.; Jia, N.; Zhou, C.; Ge, Y.; Liu, H.; Meng, W.; Ji, C. Paving green passage for emergency vehicle in heavy traffic: Real-time motion planning under the connected and automated vehicles environment. In Proceedings of the 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Shanghai, China, 11–13 October 2017.
26. Akabane, A.T.; Immich, R.; Bittencourt, L.F.; Madeira, E.R.; Villas, L.A. Towards a distributed and infrastructure-less vehicular traffic management system. *Comput. Commun.* **2020**, *151*, 306–319. [CrossRef]
27. Fogue, M.; Sanguesa, J.A.; Martinez, F.J.; Marquez-Barja, J.M. Improving roadside unit deployment in vehicular networks by exploiting genetic algorithms. *Appl. Sci.* **2018**, *8*, 86. [CrossRef]
28. Fakirah, M.; Leng, S.; Chen, X.; Zhou, J. Visible light communication-based traffic control of autonomous vehicles at multi-lane roundabouts. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 1–14. [CrossRef]
29. Ullah, A.; Azeem, M.; Ashraf, H.; Alaboudi, A.A.; Humayun, M.; Jhanjhi, N. Secure healthcare data aggregation and transmission in IoT—A survey. *IEEE Access* **2021**, *9*, 16849–16865. [CrossRef]
30. Humayun, M. Role of emerging IoT big data and cloud computing for real time application. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 1–13. [CrossRef]

31. Alayda, S.; Almowaysher, N.; Humayun, M.; Jhanjhi, N. A Novel Hybrid Approach for Access Control in Cloud Computing. *Int. J. Eng. Res. Technol.* **2020**, *13*, 3404–3414. [CrossRef]
32. Humayun, M.; Jhanjhi, N.; Alruwaili, M.; Amalathas, S.S.; Balasubramanian, V.; Selvaraj, B. Privacy protection and energy optimization for 5G-aided industrial Internet of Things. *IEEE Access* **2020**, *8*, 183665–183677. [CrossRef]
33. Humayun, M.; Hamid, B.; Jhanjhi, N.; Suseendran, G.; Talib, M.N. 5G Network Security Issues, Challenges, Opportunities and Future Directions: A Survey. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2021.
34. Créput, J.-C.; Hajjam, A.; Koukam, A.; Kuhn, O. Dynamic vehicle routing problem for medical emergency management. In *Self Organizing Maps-Applications and Novel Algorithm Design*; Intechopen: London, UK, 2011; pp. 233–250.
35. Mesquita, D.P.; Gomes, J.P.; Junior, A.H.S.; Nobre, J. Euclidean distance estimation in incomplete datasets. *Neurocomputing* **2017**, *248*, 11–18. [CrossRef]
36. Tabaghi, P.; Dokmanić, I.; Vetterli, M. Kinetic Euclidean distance matrices. *IEEE Trans. Signal Process.* **2019**, *68*, 452–465. [CrossRef]
37. Gultom, S.; Sriadhi, S.; Martiano, M.; Simarmata, J. Comparison analysis of K-means and K-medoid with Ecluidience distance algorithm, Chanberra distance, and Chebyshev distance for big data clustering. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2018.
38. Santosh, M.; Sharma, A. A proposed framework for emotion recognition using Canberra distance classifier. *J. Comput. Theor. Nanosci.* **2019**, *16*, 3778–3782. [CrossRef]
39. Lubis, A.R.; Lubis, M. Optimization of distance formula in K-Nearest Neighbor method. *Bull. Electr. Eng. Inform.* **2020**, *9*, 326–338. [CrossRef]
40. Sitompul, O.; Nababan, E. Measuring the accuracy of simple evolving connectionist system with varying distance formulas. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2017.
41. Li, L.; Coskun, S.; Zhang, F.; Langari, R.; Xi, J. Energy management of hybrid electric vehicle using vehicle lateral dynamic in velocity prediction. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3279–3293. [CrossRef]
42. Borowczyk, A.; Nguyen, D.-T.; Nguyen, A.P.-V.; Nguyen, D.Q.; Saussié, D.; Le Ny, J. Autonomous landing of a multirotor micro air vehicle on a high velocity ground vehicle. *Ifac-Papersonline* **2017**, *50*, 10488–10494. [CrossRef]
43. González, C.L.; Pulido, J.J.; Alberola, J.M.; Julian, V.; Niño, L.F. Autonomous Distributed Intersection Management for Emergency Vehicles at Intersections. In *Practical Applications of Agents and Multi-Agent Systems*; Springer: Berlin/Heidelberg, Germany, 2021.

engineering proceedings

Proceeding Paper

Traffic Management System Using YOLO Algorithm †

Pankaj Kunekar , Yogita Narule, Richa Mahajan, Shantanu Mandlapure, Eshan Mehendale * and Yashashri Meshram

Vishwakarma Institute of Technology, Savitribai Phule Pune University, Pune 411037, Maharashtra, India; pankaj.kunekar@vit.edu (P.K.); yogita.narule@vit.edu (Y.N.); richa.mahajan21@vit.edu (R.M.); shantanu.mandlapure21@vit.edu (S.M.); yashashri.meshram21@vit.edu (Y.M.)

* Correspondence: eshan.mehendale21@vit.edu

† Presented at the International Conference on Recent Advances in Science and Engineering, Dubai, United Arab Emirates, 4–5 October 2023.

Abstract: The issue of traffic congestion is becoming worse day by day. The typical traffic lights are unable to effectively regulate the growing number of vehicular traffic; therefore, we mixed computer vision and machine learning to mimic complicated incoming traffic at signalized intersections. This was accomplished using the cutting-edge, real-time object detection system You Only Look Once (YOLO), which is built on deep convolutional neural networks. In order to maximize the number of vehicles that can cross safely with the least amount of waiting time, this paper presents an efficient method to use this algorithm, where traffic signal phases are based on the data obtained, primarily queue density and waiting time per vehicle. Embedded controllers that adopt the transfer learning methodology can implement YOLO.

Keywords: computer vision; machine learning; neural network; traffic density; waiting time

Citation: Kunekar, P.; Narule, Y.; Mahajan, R.; Mandlapure, S.; Mehendale, E.; Meshram, Y. Traffic Management System Using YOLO Algorithm. *Eng. Proc.* **2023**, *59*, 210. https://doi.org/10.3390/engproc2023059210

Academic Editors: Nithesh Naik, Rajiv Selvam, Pavan Hiremath, Suhas Kowshik CS and Ritesh Ramakrishna Bhat

Published: 23 January 2024

1. Introduction

Nearly all facets of contemporary systems and their basics involve technology. Automation is therefore now necessary rather than just a luxury. The typical person in today's world sits in traffic for at least 8 to 10 days annually. This necessitates a significant amount of time that could be spent productively working and adding to fuel consumption, which is a huge problem right now. Several cities struggle with congestion, and stationary traffic light signal controllers are not capable of reducing the lengthy wait times at crossings. Rather than a traffic light, we have often seen a police officer controlling traffic who inspects the road conditions and calculates the authorized length for every route. This feat of sentient accomplishment motivates us to create smart traffic signal control that proactively handles intersections while taking into consideration current traffic circumstances. To put such a framework in place, two basic components are required: a gaze ahead to monitor virtual traffic conditions as well as a processing mind. The top priorities of a traffic signal system are to move as many vehicles through a junction as possible while minimizing traffic congestion. An instantaneous object recognition called YOLO version 7 (You Only Look Once) accurately recognizes specific things in images, streaming broadcasts, and videos. YOLOv7 employs properties that a deep convolutional neural network has learned to identify the objects and recognize them. After handling the input images and neural network, the resulting software will be activated by attaching the neural network to the hardware.

2. Literature Review and Methodology

2.1. Literature Review

For the purpose of determining the viability of our proposal and exploring the various ways it might be carried out, we read a considerable number of research articles. We gained knowledge from these papers, which helped us clearly define our project's vision and scheme of action. There are numerous ways to execute this project, but in order to do so, one needs to be knowledgeable in a variety of areas, including Python, image processing, computer vision, and machine learning. While browsing through numerous blogs, we found many that provided detailed explanations of how Yolo functions and how to use it in projects. We consulted this paper to gain a sense of how it might be carried out; it is advised to count and detect automobiles in a chaotic traffic situation [1]. Having read the research stating that deep learning-based methods are useful for visual vehicle recognition and counting in highway scenarios [2], we were able to better understand the YOLO algorithm. This research investigated how to compute an automobile's highway pace whilst attempting to avoid speeding charges. The same idea can be used to ensure that all the traffic laws are observed in order to stay safe on a typical city street. We determined which YOLO paradigm should be employed for enabling image processing after analyzing these and other works. To acquire a general understanding of how to estimate traffic variables, we referred to [3], an article explaining " YOLOv3 with deep learning techniques for Automated Moving-based Analysis of Location Information Vehicle Spectator Method" [4]. The proposed system aims to count the number of vehicles passing through a given road segment using real-time video analysis, which gave us good insight into how to calculate the number of vehicles in a specific section. Also, the research paper [5] showed that the proposed system using the YOLO algorithm for real-time vehicle counting, speed estimation, and classification achieved high accuracy and real-time performance, outperforming the existing methods in terms of both detection and classification accuracy, as well as speed estimation precision.

2.2. Methodology

I. Workflow of the process:

Below Figure 1 shows Traffic estimation workflow of YOLOv7 model. So, first of all, an image input or video input will be given to the program in the microcontroller through a camera present at the junction. Then, the traffic density will be calculated using the algorithm, including the number of vehicles and the type of vehicle present, like a car, a bike, or a truck. Identifying the type of vehicle helps determine its speed and the time it takes to cross the road. There are four straightforward stages capable of illustrating the results, which are as follows:

1. Choose a live picture of every track.
2. Evaluate and estimate the traffic pattern.
3. Put this information in the timer calculation algorithm.
4. The result is going to recommend the intervals on every track and side appropriately.

YOLO Object Detection Path

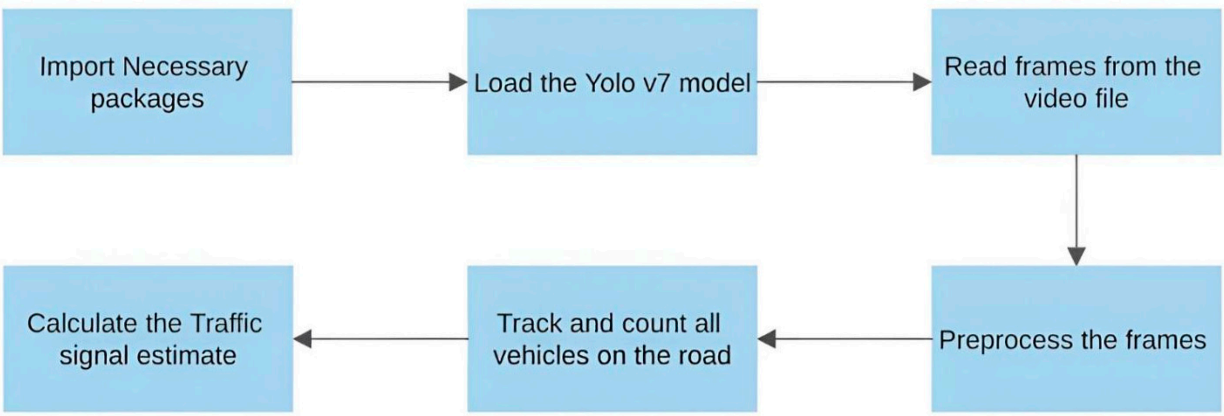

Figure 1. Traffic estimation workflow.

II. Theory Algorithm:

YOLO version 7 is the quickest and most precise context of the physical tracking system and identification framework for digital imaging workloads. Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao published the official YOLOv7 paper, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors", in July 2022. YOLO version 7 is a fundamental model that lends itself to conventional GPU computing the fastest. The YOLOv7-tiny model is a straightforward one, and its edge GPU is optimized. Tiny computer vision models are easier to execute on dispersed edge servers and devices, or mobile computing devices. Additionally, they are optimized for deep learning and edge AI applications. This strategy is essential for networked real-world computer vision applications. YOLOv7-W6 is a foundational model designed for cloud GPU computation. Figure 2 here demonstrates YOLOv7 architecture.

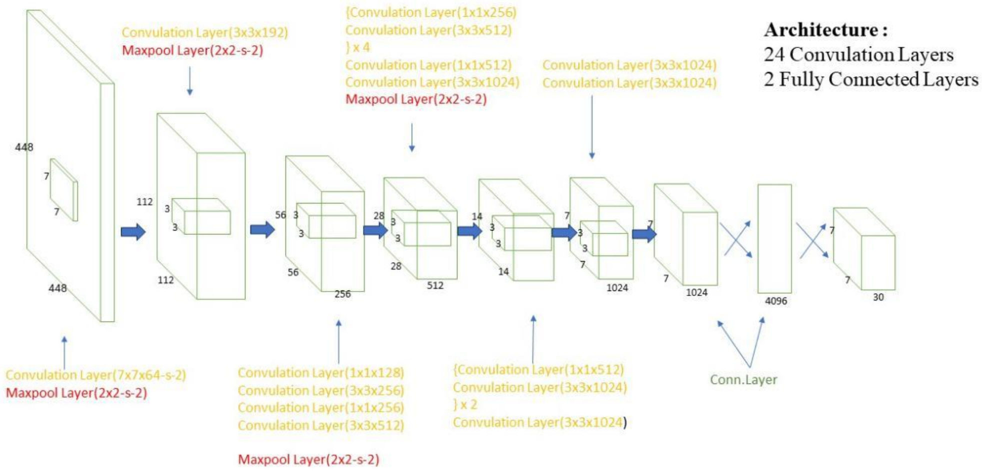

Figure 2. YOLOv7 infrastructure [6].

YOLO's Deep CNN comprises 24 convolutional layers altogether, the final 2 of which are bonded, which is inspired by the Google Net architecture. Below Figure 3 depicts image's division and packaging using YOLO.

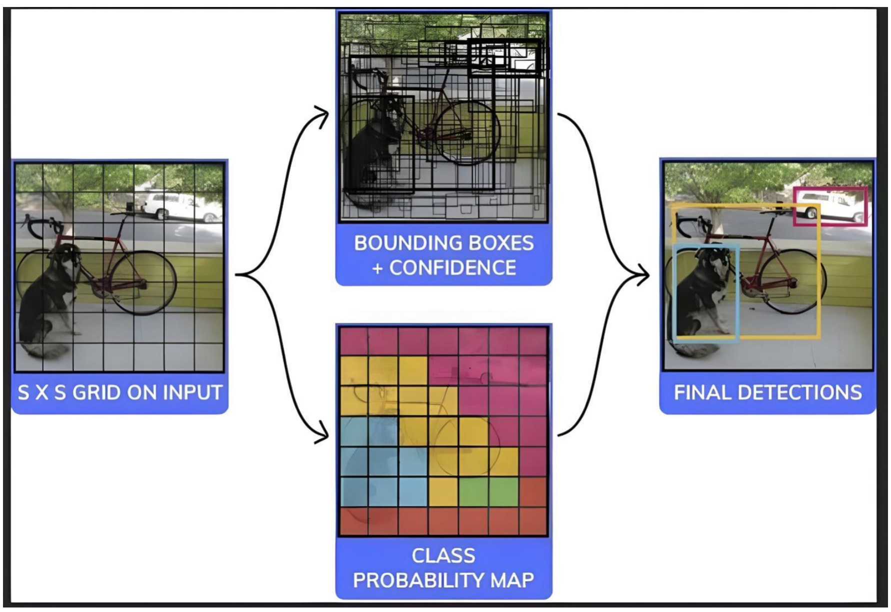

Figure 3. An illustration of the image's division and packaging using YOLO [7].

The YOLOv7 workflow is as follows:

- Pre-processing: The algorithm takes an input image or video frame and pre-processes it by resizing it to a fixed size and normalizing the pixel values.
- Feature extraction: The algorithm then extracts features from the pre-processed image using a convolutional neural network (CNN). The CNN is trained on a large dataset of images and learns to extract features that are useful for object detection.
- Grid creation: The algorithm divides the image into a grid of cells, where each cell is responsible for detecting objects that fall within its boundaries.
- Object detection: For each grid cell, the algorithm predicts a set of bounding boxes and a confidence score for each box. The confidence score represents how likely it is that the box contains an object.
- Non-maximum suppression: The algorithm applies non-maximum suppression to the predicted boxes to remove duplicates and overlapping boxes.
- Classification: For each remaining box, the algorithm classifies the object inside it using a SoftMax function. The SoftMax function assigns a probability to each possible object class.
- Output: Finally, the algorithm outputs the coordinates of the bounding boxes, the class labels, and the confidence scores for each detected object.

Traffic Signal Calculations:

We calculate and allocate the signal time for each road based on its traffic density. When no vehicle awaits, the road is completely skipped.

For the calculation, the following parameters are used:

(a) Maximum allowed time (Tmax).
(b) Column density (Cd).
(c) Crossing time (CrT).
(d) Road width (Rw).
(e) Crossing distance (AVl).
(f) (d) Allotted time.
(g) Leftover vehicles (Lo).
(h) Buffer distance (Bd).

Waiting time = total time that the number of vehicles is waiting for on the other three roads and the leftover of the current road allotted time.

3. Results

In Figure 4, input is given to the YOLOv7 model. Different sorts of cars have been successfully spotted and distinguished in Figure 5. The program may also keep track of how many vehicles are parked at the traffic light at any given moment. From there, readers can shorten the moment that the automobiles must wait during the indication as a result of this initiative. Fewer vehicles can pass under the static system (with a fixed green signal period) as compared to the proposed system. This can be performed by adjusting the indication for "go", which is dependent on the volume of the vehicles during the indication. Make sure that the green signal is in place for a longer period of time in the direction with more traffic than it is in the way with less traffic. So, by dynamically adjusting the duration of the green signal based on the volume of traffic, the proposed system could help reduce congestion and improve the overall traffic flow. This could lead to shorter travel times and reduced frustration for drivers. Also, this proposed system could lead to significant energy savings, as the traffic lights would only stay green for as long as needed based on the number of vehicles at the intersection. This would reduce energy consumption and greenhouse gas emissions. The system can be connected to a central control center that monitors traffic across the city or region, allowing for coordinated management of traffic signals and other traffic-related activities.

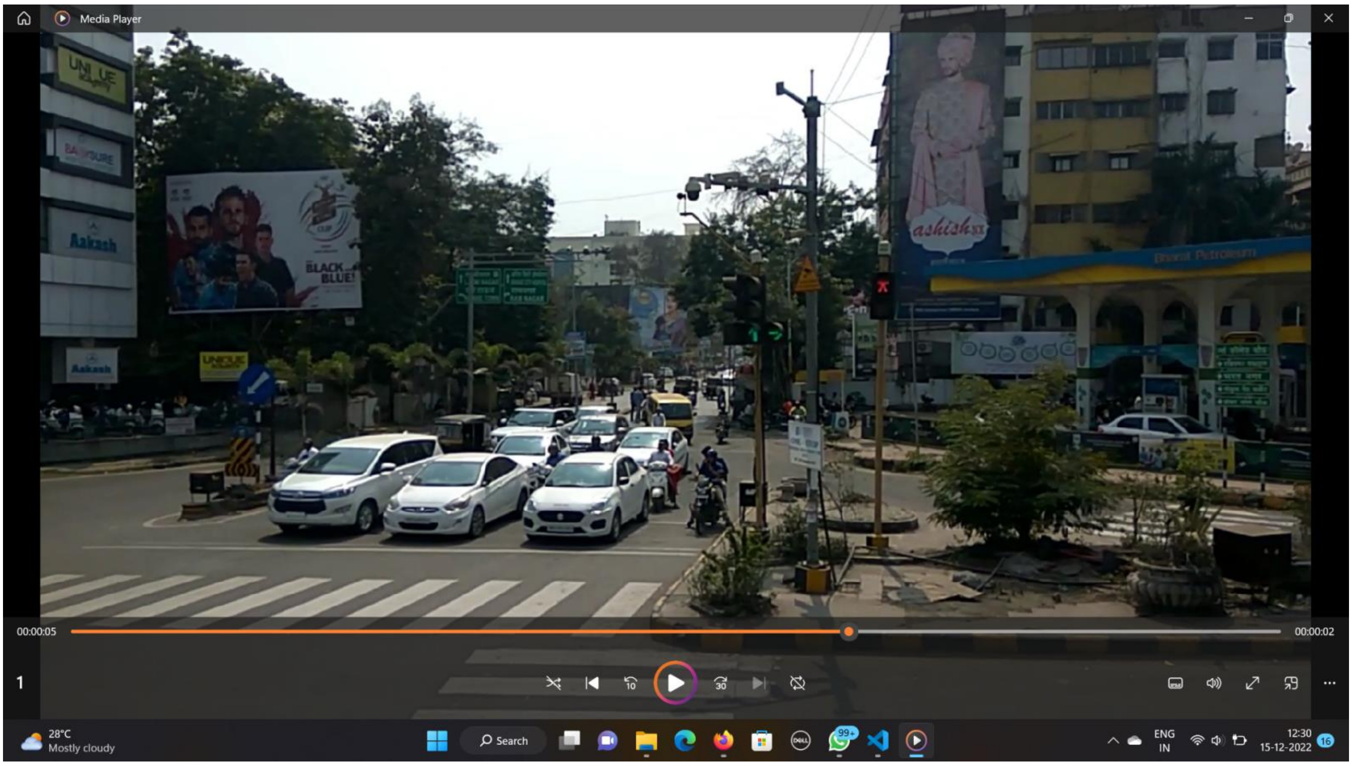

Figure 4. Input given to the project.

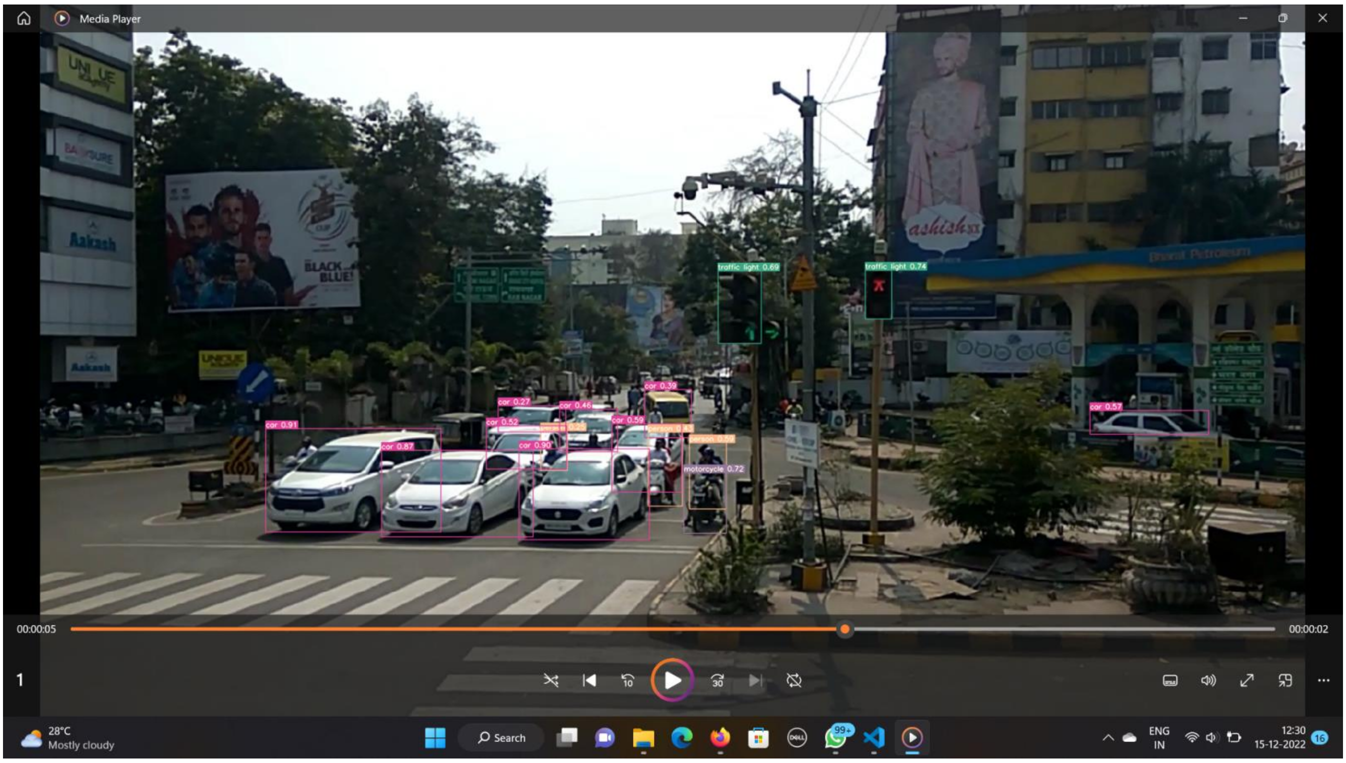

Figure 5. Output generated.

4. Limitations and Future Scope

Due to the increased cost of installation as well as the upkeep of recording devices and certain equipment, this project has a high starting cost. A substantial investment will be necessary for this kind of technology. The other restriction is that poor weather conditions like rain or fog reduce road visibility, which directly affects the accuracy of traffic counts and leads to inaccurate traffic estimates. The system must also have the right street lights in place for it to work correctly at night. The design of this system must be faultless because a human life is at stake, and even a small flaw in it can cause an accident. It is challenging to determine the precise number of cars because there is no designated lane for each type of vehicle. This is so that the two-wheelers can be seen even with a large vehicle in the front. The accuracy of the vehicle count will be affected [8–10].

Over the years, technology has progressively migrated across fields, but in the area of traffic management and control, inventions have been considerably less noticeable. Intelligent transportation systems (ITS) have gained popularity recently thanks to their many benefits, which go beyond traffic information and control to include effective infrastructure use and road safety. The knowledge gained from censored traffic lights can be applied to operational management to keep it current and make adjustments. By operating on the cloud, traffic control centers can lower expenses while proactively responding to changing traffic circumstances and outages, enhancing the citizen experience, enhancing traffic safety, and improving signal performance. Also, this system can be integrated with other traffic management technologies, such as adaptive cruise control and lane departure warning systems, to further enhance traffic safety and efficiency. Also, we can implement this further. In addition to traffic volume, the system can also take into account other factors that affect traffic flow, such as weather conditions, road closures, and special events. Finally, the system can be enhanced using machine learning algorithms that can learn from historical data to predict traffic patterns and adjust traffic signals accordingly [11,12].

5. Conclusions

This project was performed with the intention of developing a smart transportation management system that makes use of a self-adaptive and robust optimizer for deep learning-based congestion control and a convolutional neural network. The proposed model is expected to perform better than the traditional systems as the amount of data collected grows. More accurate forecasting will be possible. By utilizing cutting-edge tools like machine learning and vision for computing, we will be able to automate traffic. This new system makes it easier for cars to move through crossing points, leading to less overcrowding, lower releases of carbon dioxide, and so on. Considering the wealth of information pertaining to the clip, it emphasizes the implications of pushing fundamental limits for pattern recognition, classification, and monitoring of existing works. YOLOv7 has a lightning-fast reasoning rate at a minor loss of reliability, particularly for tinier items and less clarity.

Author Contributions: Y.N. contributed to conceptualization, validation, and project administration; P.K. oversaw validation, supervision, and project administration; Y.M. handled original draft preparation, review, and editing; E.M. conducted formal analysis, investigation, and resource management; S.M. led methodology and software development; R.M. contributed to conceptualization, software, and validation. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The following are idealistic avenues and novel dimensions that will aid in the accomplishment of this project, which we would like to acknowledge. We would like to thank Pune's Vishwakarma Institute of Technology for trusting us to engage towards this work and for your generosity. We would also like to extend our appreciation to our instructors for their help and support while experimenting with different areas. We would also like to express our appreciation to Pankaj Kunekar and Yogita Narule, who served as our project advisors and whose direction and criticism made this effort feasible.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Asha, C.S.; Narasimhadhan, A.V. Vehicle Counting for Traffic Management System using YOLO and Correction filter. In Proceedings of the 2018 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 16–17 March 2018; p. 18149219.
2. Song, H.; Liang, H.; Li, H.; Dai, Z.; Yun, X. Vision-based vehicle detection and counting system using deep learning in highway scenes. *Eur. Transp. Res. Rev.* **2019**, *11*, 51. [CrossRef]
3. Guerrieri, M.; Parla, G. Deep Learning and YOLOv3 Systems for Automatic Traffic Data Measurement by Moving Car Observer Technique. *Infrastructures* **2021**, *6*, 134. [CrossRef]
4. Lin, J.-P.; Sun, M.-T. A YOLO-Based Traffic Counting System. In Proceedings of the 2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI), Taichung, Taiwan, 30 November–2 December 2018; p. 18366282.
5. Lin, C.-J.; Jeng, S.-Y.; Lioa, H.-W. A Real-Time Vehicle Counting, Speed Estimation, and Classification System Based on Virtual Detection Zone and YOLO. *Math. Probl. Eng.* **2021**, *2021*, 1577614. [CrossRef]
6. YOLO: Algorithm for Object Detection Explained [+Examples]. Available online: https://www.v7labs.com/blog/yolo-object-detection (accessed on 15 April 2023).
7. Kabara, N.; Nere, D.; Niturkar, G.; Patil, N.; Patil, D.; Bhorge, S. ATMOS: Advanced Traffic Management and Optimizing System. *ISOR J.* **2022**, *24*, 55–59. [CrossRef]
8. Lanke, N.; Sheetal, K. Smart Traffic Management System. *Int. J. Comput. Appl.* **2013**, *75*, 19–22. [CrossRef]
9. Oltean, G.; Florea, C.; Orghidan, R.; Oltean, V. Towards Real-Time Vehicle Counting using YOLO–Tiny and fast Motion Estimation. In Proceedings of the IEEE 25th International SIITME, Cluj-Napoca, Romania, 23–26 October 2019; p. 19359208.
10. Mahmood, M.T.; Ahmed, S.R.A.; Ahmed, M.R.A. Detection of the vehicle with Infrared images in Road Traffic using YOLO computational mechanism. In Proceedings of the IOP Conference Series: Materials Science and Engineering, 2nd International Scientific Conference of Al-Ayen University (ISCAU-2020), Thi-Qar, Iraq, 15–16 July 2020; p. 022027.
11. Krishnamoorthy, R.; Kumar, N.; Grebennikov, A.; Ramiah, H. A high-efficiency Ultra-Broadband mixed-mode Gan HEMT power amplifier. *IEEE Trans. Circuits Syst. II Express Briefs* **2018**, *65*, 1929–1933. [CrossRef]
12. Vitee, N.; Ramiah, H.; Mak, P.-I.; Yin, J.; Martins, R.P. A 3.15-MW + 16.0-DBM IIP3 22-DB CG inductively source degenerated balun-lna mixer with integrated transformer-based gate inductor and IM2 injection technique. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2020**, *28*, 700–713. [CrossRef]

Artificial Intelligence Based Autonomous Traffic Regulator

Dr. Sreelatha R[1,] Mahalakshmi B S[2], Riya Yadav[3], Shreyam Pandey[4], Vandit Agarwal[5]

[1,2]Assistant Professor, Department of Information Science, BMSCE, Bangalore, India
[3, 4, 5] Students Department of Information Science, BMSCE, Bangalore, India

Abstract - Artificial intelligence (AI)-based autonomous traffic regulation refers to the management and control of traffic flow. In order to collect real-time data on traffic conditions, sensors, cameras, and communication networks are used. This data is then evaluated and processed by AI algorithms to produce insights and make judgement. AI-powered autonomous traffic regulation aims to increase system efficiency by reducing congestion, increasing safety, and all of the above. The advantage of using autonomous traffic regulation utilizing AI is the ability to process and collect large real time data and conclusions are drawn. This enables the system to adjust the traffic flow fast in response to shifting traffic circumstances. Algorithms based on AI can also be used learn from previous traffic patterns and situations to create future forecasts and conclusions that are more accurate. For autonomous traffic regulation, a variety of AI algorithms, which includes reinforcement learning machine learning, deep learning, can be applied. Algorithms based on Deep learning can be used to interpret photos, video data from cameras, spotting patterns and trends in traffic data can be achieved through machine learning algorithms. Algorithms for reinforcement learning can be used to learn from the past and make choices based on reward signals. To guarantee their dependability and safety, it is crucial to make sure that these systems are designed and deployed with the proper protections. This AI-powered system can also adjust in real-time to shifting traffic patterns and road conditions, making the traffic regulating process more responsive and dynamic. As a result, there may be an improvement in traffic-related emissions reductions and fuel efficiency. Overall, the AI is used for the development of intelligent transportation systems which has advanced significant, which has the potential to revolutionize traffic management and assure a more effective, safe, and sustainable transportation system.

I. INTRODUCTION

The automation of traffic management and control is accomplished here by development of an autonomous traffic regulator. It enhances the safety and effectiveness of roadways by using a variety of technologies such as cameras, signal controllers and artificial intelligence algorithms to detect and adapt to traffic patterns in real-time. Reduced traffic congestion, lower accident risk, and improved vehicle flow are the objectives of an autonomous traffic regulator. Image detection, image processing, density calculation, communication networks, an efficient signal switching algorithm and a centralized control system are essential parts of an autonomous traffic regulation system. Autonomous Traffic Regulators use a combination of such technologies and algorithms to collect and analyze data about the flow of traffic. This information is then used to control the traffic lights, which in turn help us in regulating the flow of traffic, time spent by each vehicle on the road and lesser time delays. This further helps in reducing congestion and hence reducing the carbon emissions on the road.

The ATR system's ability to reduce travel time and fuel consumption is one of its main advantages. The ATR represents a significant advance in the future of traffic management given the rising demand for smart cities and the creation of intelligent transportation systems. It has the ability to fundamentally alter how we control traffic and guarantee a more effective, secure, and sustainable transportation system. Increased road safety is a benefit of the ATR system as well. The system can make decisions that reduce the danger of crashes and other traffic-related occurrences by assessing real-time data on traffic patterns and road conditions. This can aid in lowering the amount of collisions and fatalities on the roads, hence enhancing the safety of the roadways for all users. But there are enormous potential advantages, and technology is developing quickly. The ATR system offers a viable answer to one of the most critical issues facing modern cities as they continue to grow and traffic congestion worsens.

The necessity to overcome the difficulties traditional traffic management systems confront is driving the development of autonomous traffic regulators. The increased needs of modern transportation and the growing complexity of urban traffic networks have shown the current traffic management methods to be insufficient.

SCCTT 2023: Proceedings of CEUR Workshop Proceedings, Month 07–12-2023, New Delhi, India
sree.ise@bmsce.ac.in, (Dr. Sreelatha), bsmahalakshmi.ise@bmsce.ac.in (BS Mahalaxmi)
0000-00002-3181-6104 (Dr. Sreelatha); 0000-0002-9144-7021 (BS Mahalaxmi);

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

One of the main problems that autonomous traffic regulators seek to address is traffic congestion. In addition to wasting time and fuel, it also causes air pollution and traffic collisions. Traditional traffic management systems rely on time-consuming, ineffective manual interventions to control traffic. Road safety is another problem that autonomous traffic regulators seek to solve. As per the World Health Organization records, road accidents are the ninth main common cause of mortality worldwide and the main reason for the death among young people. Real-time detection and accident prevention are limitations of conventional traffic management systems. To improve traffic safety, autonomous traffic regulators make use of cutting-edge technologies including object identification, weighted assignments to objects, and real-time traffic monitoring. Furthermore, conventional traffic management systems are frequently created for a particular site and are not adaptable enough to accommodate shifting traffic patterns. This may result in an ineffective utilization of the road system, especially during rush hour. By minimizing travel time, fuel use, and emissions, autonomous traffic regulators can also increase the overall effectiveness of the transportation network. Autonomous traffic regulators can save travel times and use less fuel by enhancing traffic flow and minimizing congestion. Additionally, autonomous traffic regulators can lower car emissions by lowering the amount of traffic accidents.

II. LITERATURE SURVEY

Zaatouri et al. [1] introduces a traffic light control system which accepts the YOLO (You Only Look Once) algorithm for detecting the objects. The system dynamically adjusts traffic light timings by analyzing vehicle and pedestrian presence, aiming to enhance traffic flow and alleviate congestion. By utilizing YOLO's efficiency and accuracy, the proposed system helps to self-adaptive approach for optimizing traffic signal operations.

Liu et al. [2] presents an approach that combines the YOLO network with the anchor box mechanism to improve object detection accuracy and efficiency. Experimental results demonstrate the effectiveness of the proposed method in detecting objects in real-time scenarios. The research contributes to the advancement of object detection techniques by leveraging the capabilities of the YOLO network and introducing the anchor box mechanism.

The authors Pratama B et al. [3] present a system for traffic density calculation with a help of road pattern analysis using adaptive traffic light control. The model aims to optimize traffic flow by dynamically adjusting signal timings according to the current traffic density. Experimental results from a case study in Manado, Indonesia, demonstrate the effectiveness of the proposed approach in reducing traffic congestion and improving overall traffic management.

Bhave N et al. [4] proposes a smart traffic signal control system that combines reinforcement learning and object detection. The system dynamically adjusts signal timings based on real-time traffic conditions and vehicle detection. By applying reinforcement learning algorithms, the system learns optimal traffic signal policies for different traffic scenarios. Experimental results from Palladam, India, demonstrate the effectiveness of the proposed approach in reducing traffic congestion and improving overall traffic management by adapting to changing traffic patterns.

Garg et al. [5] the authors explored the multi-agent deep reinforcement learning approach for optimizing traffic flow at multiple road intersections. By utilizing live camera feeds, the system learns optimal traffic signal control policies, resulting in improved traffic efficiency and reduced congestion.

Kwon J et al. [6] focuses on traffic data classification using machine learning algorithms in Software-Defined Networking (SDN) networks. The study proposes a classification framework to classify network traffic based on machine learning techniques. By analyzing traffic patterns, the proposed approach enables efficient traffic management and improves network performance in SDN environments.

Lee et al. [7] designs intelligent traffic control techniques for autonomous vehicle systems using machine learning. The paper discusses the application of machine learning algorithms to predict traffic conditions and optimize traffic signal timings for improved traffic flow. The proposed approach aims to enhance the performance and efficiency of autonomous vehicle systems by leveraging machine learning capabilities in traffic control.

Tiwari et al. [8] focuses on real-time traffic management utilizing machine learning techniques. The study proposes a system that employs machine learning algorithms to analyze traffic data and make intelligent decisions for traffic control and management. The goal is to enhance the efficiency of traffic flow and reduce congestion by dynamically adjusting signal timings based on real-time traffic conditions.

SCCTT 2023: Proceedings of CEUR Workshop Proceedings, Month 07–12-2023, New Delhi, India
sree.ise@bmsce.ac.in, (Dr. Sreelatha), bsmahalakshmi.ise@bmsce.ac.in (BS Mahalaxmi)
0000-00002-3181-6104 (Dr. Sreelatha); 0000-0002-9144-7021 (BS Mahalaxmi);

Lorencik D et al. [9] discusses the object recognition techniques in traffic monitoring systems. The study explores the use of computer vision algorithms and machine learning methods for accurately detecting and classifying objects in traffic scenarios. The proposed approach aims to enhance the effectiveness of traffic monitoring systems by enabling automated object recognition, which can contribute to improved traffic analysis, management, and safety.

Asha C S et al. [10] presents a vehicle counting system for traffic management. The system combines the YOLO (You Only Look Once) algorithm and correlation filter techniques to detect and count vehicles in real-time. The proposed approach aims to provide accurate and efficient vehicle counting for traffic analysis and management systems, which can assist in making informed decisions and improving overall traffic flow.

De Oliveira L F P et al. [11] presents the development of a smart traffic light control system with real-time monitoring capabilities. The system utilizes Internet of Things (IoT) technologies to monitor traffic conditions and dynamically adjust signal timings based on traffic flow. The paper discusses the design and implementation of the system, highlighting its ability to improve traffic efficiency, reduce congestion, and enhance overall traffic management through real-time monitoring and control.

Peiyuan Jiang et al.[12] provides a comprehensive review of the developments in the YOLO (You Only Look Once) algorithm. The paper discusses the evolution and improvements of the YOLO algorithm over time, including different versions and variations. It covers various aspects such as network architecture, training techniques, object detection performance, and applications. The review aims to provide an understanding of the advancements in the YOLO algorithm and its relevance in the field of computer vision and object detection.

Maqbool S et al.[13] presents an approach that combines computer vision techniques with image processing algorithms to detect vehicles, track their movement, and accurately count the number of vehicles in a given area. The proposed system has potential applications in traffic monitoring, congestion management, and urban planning. The research contributes to the field of intelligent transportation systems by providing an effective solution for vehicle detection and tracking.

Kumari R et al.[14,15] The first paper focuses on analyzing the PyGameGUI modules and their functionalities, while the second paper demonstrates the use of Pygame for implementing a trained model for autonomous driving using deep reinforcement learning. Together, they contribute to the understanding and utilization of Pygame in different contexts such as user interface development and autonomous driving simulations.

III. STUDY OF SUCCESSIVE TECHNOLOGIES

1) The flow of Self Adaptive Traffic Light Control by Adapting YOLO Algorithm

This research suggests a real-time method of traffic signal control which is based on traffic movement. They have the features of the opposing traffic flows at the signalized road crossing thanks to computer vision and machine learning. You Only Look Once, a cutting-edge real-time item detection system, does this. It is built on deep conventional neural networks (YOLO). Then, traffic signal phases are optimized based on data that has been gathered, namely line length and waiting time per vehicle, to allow the greatest number of vehicles to pass safely with the shortest amount of waiting time. YOLO's accuracy and real-time efficiency made it possible to substitute the policeman in traffic control optimization.

Deep learning is used to create a novel adaptive traffic light control algorithm that complies with safety standards. Real-time detection and vehicle monitoring with duration before exiting the intersection are made feasible by YOLO v2. In fact, the controller uses the YOLO model to determine how many vehicles are in each lane and how long they will wait when the light turns yellow. The duration of the following phase is determined to reduce waiting time based on the maximum and average waiting times for each lane and the length of the queue.

Without disrupting the cycle order, our approach gives preference to those who have waited the longest. See Fig. 1 for a discussion of this algorithm.

SCCTT 2023: Proceedings of CEUR Workshop Proceedings, Month 07–12-2023, New Delhi, India
sree.ise@bmsce.ac.in, (Dr. Sreelatha), bsmahalakshmi.ise@bmsce.ac.in (BS Mahalaxmi)
0000-00002-3181-6104 (Dr. Sreelatha); 0000-0002-9144-7021 (BS Mahalaxmi);

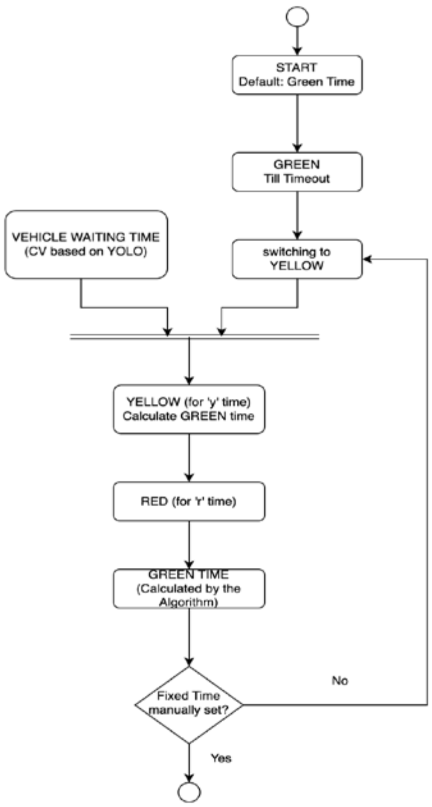

Figure 1: Activity Diagram of the proposed ATR

2) Use of YOLO algorithm to detect the Objects

We explored and simulated several visual degenerative processes. The excellent results have been produced by Deep learning-based object detection. As there are many numerous problems with issues like photographs when shooting happens in the real world, which includes issues such as noise, blurring, rotational jitter, etc. The advantages of these issues on object detection is important.. In the beginning, they developed the models for degraded photos mostly by applying mathematical models to produce degraded images based on common data sets. They then trained the network to adapt to the challenging real-world environment using these models. Based on the YOLO network developed image degradation model and incorporated conventional image processing techniques to emulate the issues present in real-world shooting, using traffic signs as an example. We examined the impacts of various degradation models on object detection after developing the various degradation models. In order to increase the average precision (AP) of traffic sign detection in real scenarios, we trained a strong model using the YOLO network. In addition to improving the accuracy of object detection, our work has also shown that it is possible to train models that are robust to a variety of visual degradations. This is important for applications such as self-driving cars, where the ability to detect objects in degraded conditions is essential for safety.

Finally, we enhanced the model's capacity for generalizing complex images. We used the YOLO neural network to assess the traffic signs as our study object. A new picture degradation model was created as a result, using various deteriorated photographs as test sets. After that, they altered the source network and used several degradation techniques to the training set. Then, they used more intricate degradation processes to the training sets to produce an improved and broadly applicable detection network. In conclusion, the model's capacity for generalization had been strengthened, and object detection had become more precise.

3) The Traffic Density Calculation done for Road Patterns

Through estimations of traffic density on road layouts, we suggest adaptive traffic signals to regulate their timing. Several different road designs are subjected to image processing to determine the traffic density. Later, the traffic density is used to determine when the traffic signal will turn on. To assess the performance of their suggested method and compare it to a fixed-time traffic light system, the authors created a simulation model. The simulation's findings demonstrated that, in comparison to the fixed-time system, the adaptive traffic signal system was able to decrease the average vehicle waiting time and increase traffic flow. A server that collects data and manages traffic light operations at crossroads is also present. Real-world road conditions are used to validate the whole set of algorithms, including those for calculating traffic density and timing of traffic lights. The results obtained demonstrate the accuracy with which the traffic density sensing system can accurately determine the time of a traffic light. By ensuring that the green light is on for a longer period of time when there is a high volume of traffic on the road and a shorter period of time when there is less traffic, the system was able to lessen congestion.

A. *Canny Edge Detector*

Canny edge detection algorithm is one of the important used techniques in image processing.

SCCTT 2023: Proceedings of CEUR Workshop Proceedings, Month 07–12-2023, New Delhi, India
sree.ise@bmsce.ac.in, (Dr. Sreelatha), bsmahalakshmi.ise@bmsce.ac.in (BS Mahalaxmi)
0000-00002-3181-6104 (Dr. Sreelatha); 0000-0002-9144-7021 (BS Mahalaxmi);

using the Contrast Limited Adaptive Histogram Equalization (CLAHE) equation The image is processed before the edge of the image is determined. By adaptively adjusting the contrast difference, this technique can be utilized to lessen the noise that results from using a camera with low performance or from taking photos at night.

B. *Bilateral Filtering*

A method of image screening known as bilateral filtering provides a smoothing operation while preserving the image's edge structure. In other words, the image is edge-preserving smoothing via bilateral filtering .The two processes that make up bilateral filtering are selection and filtering. Here, Bilateral filtering is employed in this study to eliminate noise on coloring that was created in the first step. The goal of the selection procedure is to take the surrounding pixels into account. A delimiter function based on the difference in pixel values is the criteria function that is utilized. The filtering procedure itself then applies linear (using kernel box or Gaussian) or nonlinear (median filter) filtering techniques. The range of pixels included in the selection process and the maximum distance that passes the selection process are two parameters for the bilateral filtering algorithm that must be manually defined.

C. *Binary Threshold*

The last method to identify traffic congestion is the binary threshold. This procedure's major goal is to separate the automobiles from the background (road). In order to clearly identify the region that includes the object and backdrop of the image, the binary threshold converts the image to a binary or black-and-white image. The Region of Interest (ROI) of the path, which serves as the observation's focal point, is where the segmentation process is restricted.

By counting the black and white pixels in the ROI, one can determine the traffic density. The following formula is used to determine the traffic density formula:

$$Traffic\ density(\%) = \frac{black\ pixels}{black\ pixels + white\ pixels} x1$$

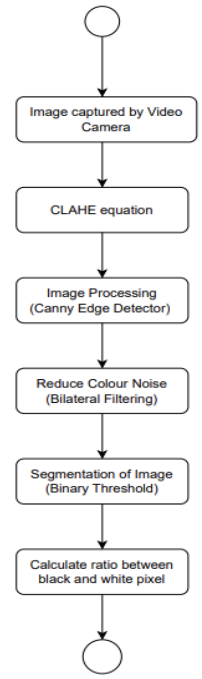

Figure 2: Traffic Density Calculation Algorithm

4) Self-Adaptive Traffic Signal Control incorporating Reinforcement Learning

The use of Reinforcement Learning (RL) and Object Detection to improve traffic flow and reduce congestion is explored. The system uses object detection algorithms to detect and count vehicles at intersections and RL algorithms to determine the optimal signal timings. The signal timings are then adjusted in real-time based on the traffic conditions to reduce wait times and improve traffic flow.Our suggested system is a fully functional model that includes hardware, software, algorithms for object identification and reinforcement learning. Following is a description of how each component like Actions and State-Action Pair work. The agent's Actions are determined by how the agent perceives the environment. The potential green phase timings of the traffic signal are the activities of our RL agent. These show how many seconds have passed since the green phase began whereas the State-Action pair is a mapping which is associated with Q-values, known as the state space representation. The State space is represented by a Matrix in our implementation. Each cell displays a Q-value for a possible State and action pair.The authors

SCCTT 2023: Proceedings of CEUR Workshop Proceedings, Month 07–12-2023, New Delhi, India
sree.ise@bmsce.ac.in, (Dr. Sreelatha), bsmahalakshmi.ise@bmsce.ac.in (BS Mahalaxmi)
0000-00002-3181-6104 (Dr. Sreelatha); 0000-0002-9144-7021 (BS Mahalaxmi);

evaluated the performance of the proposed system using simulations and compared it to a traditional fixed-time signal control system.
The results showed that the proposed system was able to significantly reduce average wait times and improve traffic flow compared to the fixed-time system. In conclusion, the study demonstrates the potential benefits of using RL and object detection in traffic signal control systems to improve traffic flow and reduce congestion. The authors suggest that their proposed system could be implemented in real-world scenarios and further research is needed to validate the results and refine the algorithms.

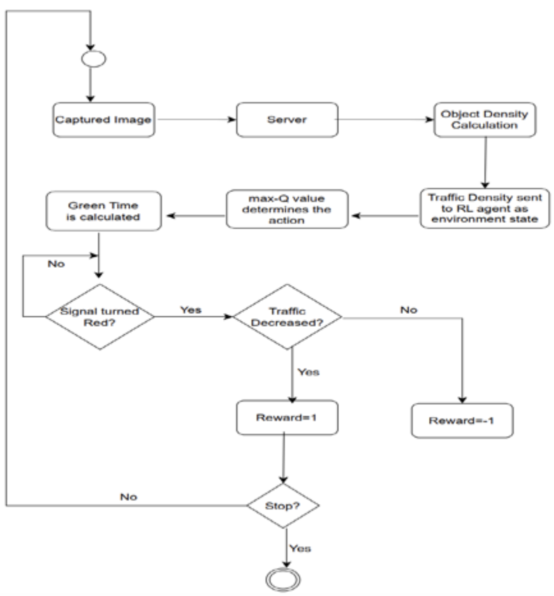

Figure 3: Flowchart for the reinforcement based system

5) Traffic optimization can be done through Multiple Road Intersections adapting Multi-Agent Deep Reinforcement Learning using Live Camera

A system of numerous, coordinated traffic signal control systems is suggested to be used. It presents a study on a traffic optimization system that uses multi-agent deep reinforcement learning (RL) to control the traffic lights at multiple road intersections. The system uses live camera feeds to detect and count vehicles at each intersection and adjust the signal timings in real-time to optimize traffic flow. The authors used a multi-agent deep RL algorithm to train the system, where each intersection was treated as an independent agent. In this study, multi-agent deep reinforcement learning (DRL) is applied for the first time to real-time traffic optimization over several road crossings using just the raw pixel input from CCTV cameras. By enhancing traffic flow and decreasing the average amount of time a vehicle spends at an intersection, it is demonstrated that this set of traffic control agents significantly outperforms independently running adaptive signal control systems. In a scenario where each agent only has access to the partial state of the traffic environment, they have shown that a centralized controller is capable of fostering a principled learning strategy between the signal control agents, leading to the positive emergence of cooperative behavior among them.The performance of the proposed system was evaluated using simulations and compared to a traditional fixed-time signal control system. The results showed that the proposed system was able to significantly reduce average wait times and improve traffic flow compared to the fixed-time system.

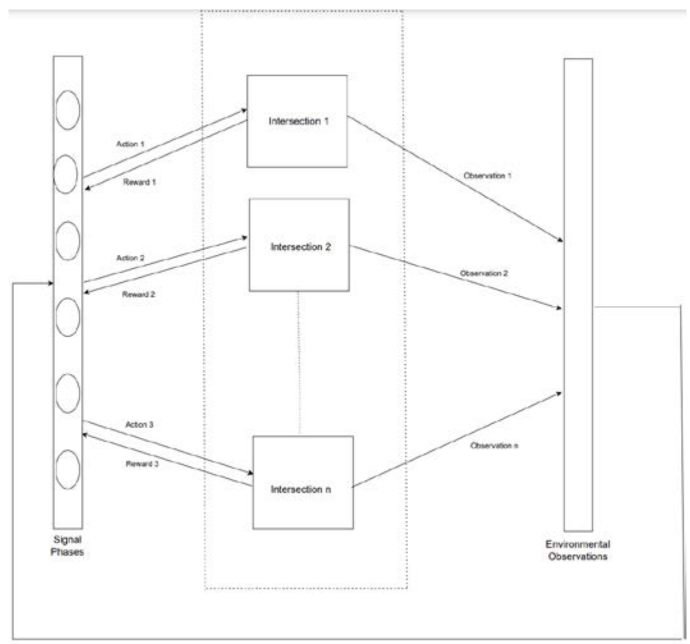

Figure 4: Flowchart for the reinforcement based system

6) Usage of YOLO and Correlation Filter for Vehicle Counting for Traffic Management System

In order to comprehend the flow of traffic and make judgments about traffic control, vehicle counting is a crucial component of traffic management. The current techniques

SCCTT 2023: Proceedings of CEUR Workshop Proceedings, Month 07–12-2023, New Delhi, India

sree.ise@bmsce.ac.in, (Dr. Sreelatha), bsmahalakshmi.ise@bmsce.ac.in (BS Mahalaxmi)
0000-00002-3181-6104 (Dr. Sreelatha); 0000-0002-9144-7021 (BS Mahalaxmi);

for counting vehicles take a long time, require a lot of work, and are inaccurate. This method locates and recognises automobiles in real-time video footage by using the object detection algorithm YOLO, which is based on deep learning. Then, to precisely count the number of vehicles on the road, the Correlation Filter method is applied. It can be concluded that the YOLO and Correlation Filter algorithms can be used to automate vehicle counting in traffic control systems. The suggested approach works well for precisely counting automobiles and following their movements. Future research should concentrate on enhancing the algorithms' accuracy and integrating the approach with other traffic control systems.

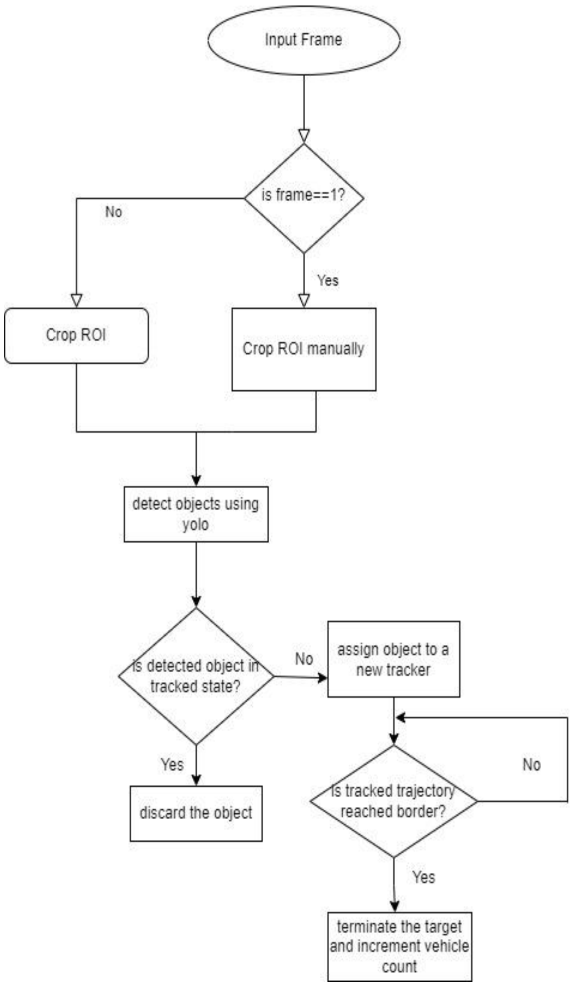

Figure 5: Flowchart of the proposed mechanism

Factors to be considered for developing the algorithm:

a. **Number of lanes**

b. Traffic density is calculated by using processing time of the algorithm similarly image need to be acquired which is decided by the green light duration.

c. For each class the total count of vehicles is maintained.

d. The above factors are used to calculate the traffic density.

e. Due to lag time added for each vehicle suffers during starting stage and the non-linear increase in lag is suffered by the vehicles which are at the back.

f. The average speed of each class of vehicle when the green light starts i.e. the average time required to cross the signal by each class of vehicle.

g. The minimum and maximum time limit for the green light duration -to prevent starvation.

IV. WORKING OF THE ALGORITHM

When the algorithm is initially run, it sets the default time for the first signal of the first cycle and all following cycles' signals as well as the times for all other signals of the first cycle. The main thread manages the timer of the current signal, and a second thread is initiated to handle vehicle detection for each direction. The detecting threads take a snapshot of the next direction when the current signal's green light timer (or the subsequent green signal's red light timer) reaches zero seconds. The next green signal's timer is set when the result has been parsed. While the main thread is reducing the time remaining on the current green signal's timer, all of this is occurring in the background. As a result, there won't be any latency during the timer's assignment. The next signal turns green for the duration specified by the algorithm when the current signal's green timer reaches zero.

To improve traffic management, it is possible to specify the average amount of time it takes for each class of vehicle to

SCCTT 2023: Proceedings of CEUR Workshop Proceedings, Month 07–12-2023, New Delhi, India
sree.ise@bmsce.ac.in, (Dr. Sreelatha), bsmahalakshmi.ise@bmsce.ac.in (BS Mahalaxmi)
0000-00002-3181-6104 (Dr. Sreelatha); 0000-0002-9144-7021 (BS Mahalaxmi);

$$GST = \frac{\sum_{vehicleClass} (NoOfVehicles_{vehicleClass} * AverageTime_{vehicleClass})}{(NoOfLanes + 1)}$$

cross an intersection based on the location, i.e., the region, the city, the locality, or even the intersection itself. For this, information from the relevant transportation authorities can be analyzed. The picture is taken when there are exactly zero seconds till the signal that will turn green next. This allows the system to process the image, count the number of vehicles in each class present in the image, and determine the green signal duration in a total of 5 seconds (equivalent to the value of the yellow signal timer). and set the red signal time for the following signal as well as the times for this signal appropriately. The average speeds of vehicles at startup and their acceleration times were utilized to determine the best green signal time based on the number of vehicles of each class at a signal, and from there, an estimate of the average time each class of vehicle takes to cross an intersection was found. The following formula is then used to get the green signal time.

where:

- NoOfVehicles of Class indicates the number of vehicles of each class of vehicle at the signal as detected by the vehicle detection module,
- Green Signal Time(GST)
- averageTimeOfClass is the average time the vehicles of that class take to cross an intersection,
- noOfLanes is the number of lanes at the intersection

Summary of the Algorithm

The vehicle detection module's traffic density data is used by the Signal Switching Algorithm to set the green signal timer and update other lights' red signal timers. Additionally, it cycles through the signals in accordance with the timers. The detection module's information on the vehicles that were picked up by the algorithm, as described in the preceding section, serves as its input. This data is presented in JSON format, with the confidence and coordinates serving as the values and the label of the object being detected as the key. To determine the total number of vehicles in each class, this data is analyzed next. Following this, the signal's green signal time is determined and assigned, and the red signal times of other signals are calculated. To accommodate any number of signals at an intersection, the algorithm can be scaled up or down.

Simulation Module

To model actual traffic, Pygame was used to create a simulation from scratch. It helps with system visualization and comparison with the current static system. There are 4 traffic lights at a 4-way intersection there. Each signal has a timer on top that displays the amount of time until it changes from green to yellow, yellow to red, or red to green. The quantity of vehicles that have passed through the intersection is also shown next to each light. There are cars, bikes, buses, trucks, rickshaws, and other vehicles coming from all directions. Some of the vehicles in the rightmost lane turn to cross the intersection to increase the realism of the simulation. When a vehicle is generated, random numbers are also used to determine whether or not it will turn. It also has a timer that shows how much time has passed since the simulation began.

V. RESULT: ATR VS EXISTING SYSTEM

In this study, we examined that Autonomous Traffic Regulator reduced travel times by up to 28%.

A study by the Indian Institute of Technology, Bombay examined that the traffic congestion in India cost the country an estimated $100 billion per year in lost productivity and fuel cost. So if our model is implemented we can save the fuel cost by an estimated figure of $26 billion.

The below graphs help us in understanding the efficiency and effectiveness of our proposed system vs. the traditional automatic traffic light control system that is already in use, by comparing the number of vehicles crossing the signal per second unit of time:

SCCTT 2023: Proceedings of CEUR Workshop Proceedings, Month 07–12-2023, New Delhi, India
sree.ise@bmsce.ac.in, (Dr. Sreelatha), bsmahalakshmi.ise@bmsce.ac.in (BS Mahalaxmi)
0000-00002-3181-6104 (Dr. Sreelatha); 0000-0002-9144-7021 (BS Mahalaxmi);

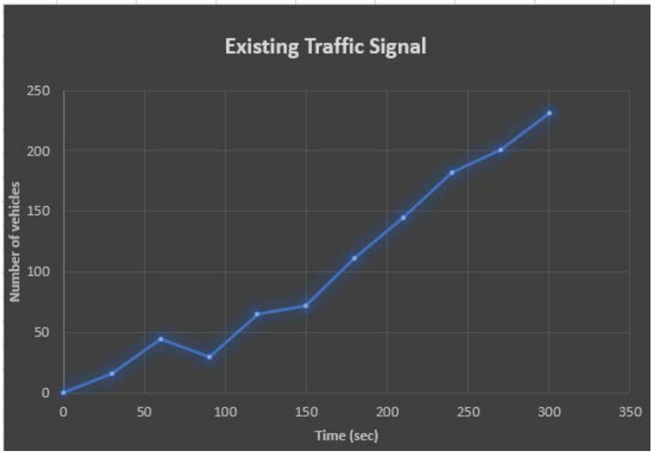

Figure-6: Indicating the vehicles crossing the signal per unit time in the existing traffic system

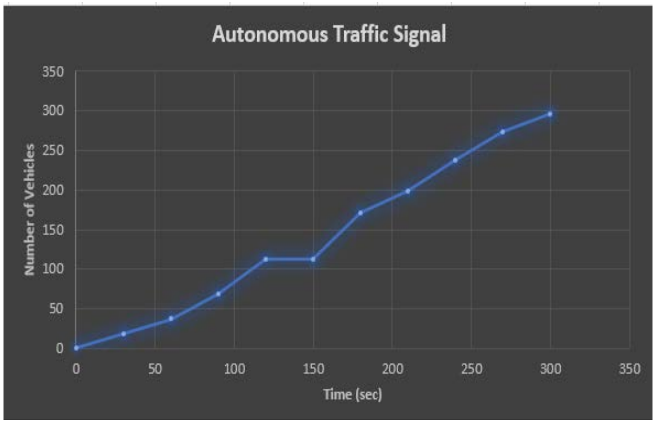

Figure-7: Indicating the vehicles crossing the signal per unit time in the proposed system

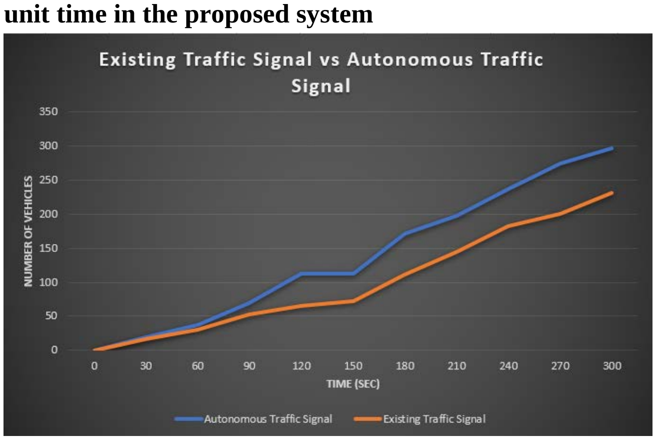

Figure-8: The efficiency Comparison of autonomous traffic control system v/s traffic control system which are traditional

VI. APPLICATIONS

a. Traffic Control: Controlling traffic is one of the main uses for autonomous traffic regulators. Autonomous traffic regulators monitor and manage traffic flow using real-time data from sensors and algorithms, which helps to ease congestion and improve traffic flow. As a result, the transportation system becomes more effective and less time and fuel are lost due to congestion.

b. Road Safety: By detecting and averting incidents on the road in real time, autonomous traffic regulators also improve road safety. For the purpose of preventing accidents, sensors and algorithms can identify risky driving practices, poor road conditions, and seasonal patterns.

c. Environmental Sustainability: By lowering transportation-related pollutants, autonomous traffic controllers can help support environmental sustainability. Autonomous traffic regulators can optimize traffic flow and ease congestion while cutting down on fuel consumption, which lowers emissions.

d. Emergency Response: Autonomous traffic controllers can help with emergency response initiatives. Autonomous traffic regulators can modify traffic lights and infrastructure in the case of a natural disaster, auto accident, or other emergency circumstance to enable a smooth flow of emergency vehicles and help the evacuation of impacted areas.

VII. CONCLUSION

Autonomous traffic regulation using AI has the potential to greatly increase road safety and traffic flow. In order to improve traffic flow, AI algorithms can study traffic patterns, forecast congestion, and dynamically change signal timings. This may lead to shorter travel distances and less fuel use, as well as fewer pollution and accidents. Additionally, To improve road safety by detecting and responding to potential hazards, such as vehicles driving erratically or pedestrians crossing the street illegally AI-Powered Traffic management system can be implemented..

SCCTT 2023: Proceedings of CEUR Workshop Proceedings, Month 07–12-2023, New Delhi, India
sree.ise@bmsce.ac.in, (Dr. Sreelatha), bsmahalakshmi.ise@bmsce.ac.in (BS Mahalaxmi)
0000-00002-3181-6104 (Dr. Sreelatha); 0000-0002-9144-7021 (BS Mahalaxmi);

However, the implementation of AI-based traffic regulating systems also brings up significant ethical and privacy issues, in addition to technical difficulties such as ensuring the resilience, dependability, and explain ability of AI algorithms. In addition, thorough examination of a number of legal and regulatory issues, such as culpability in the event of accidents or system failures, is necessary for the implementation of autonomous traffic regulation using AI. A lot of money must be invested, and many parties, including communities, businesses, and governments, must work together to integrate AI-based systems with current infrastructure. Despite these difficulties, autonomous traffic management using AI offers a lot of potential for the future of mobility and transportation. Artificial intelligence (AI)-based technologies can offer real-time, data-driven solutions to enhance traffic flow and safety on our roads by utilizing the power of machine learning and computer vision. But it's crucial to make sure that the deployment of these technologies is carried out in a morally righteous, accountable, and open way, taking into account the potential risks and advantages for all stakeholders.

It is essential to make sure that autonomous traffic regulators are deployed in a way that is visible, accountable, and beneficial to society as a whole. Additionally, it's crucial to approach the deployment of autonomous traffic regulators holistically, taking into account not only the technological elements but also the social, economic, and political ramifications.

VIII. REFERENCES

[1] Zaatouri, Khaled & Ezzedine, Tahar. (2018). A Self-Adaptive Traffic Light Control System Based on YOLO. 16-19. 10.1109/IINTEC.2018.8695293.

[2] C. Liu, Y. Tao, J. Liang, K. Li and Y. Chen, "Object Detection Based on YOLO Network," 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 2018, pp. 799-803, doi: 10.1109/ITOEC.2018.8740604.

[3] B. Pratama, J. Christanto, M. T. Hadyantama and A. Muis, "Adaptive Traffic Lights through Traffic Density Calculation on Road Pattern," 2018 International Conference on Applied Science and Technology (iCAST), Manado, Indonesia, 2018, pp. 82-86, doi: 10.1109/iCAST1.2018.8751540.

[4] N. Bhave, A. Dhagavkar, K. Dhande, M. Bana and J. Joshi, "Smart Signal – Adaptive Traffic Signal Control using Reinforcement Learning and Object Detection," (I-SMAC), Palladam, India, 2019, pp. 624-628, doi: 10.1109/I-SMAC47947.2019.9032589.

[5] D. Garg, M. Chli and G. Vogiatzis, "Multi-Agent Deep Reinforcement Learning for Traffic optimization through Multiple Road Intersections using Live Camera Feed," 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 2020, pp. 1-8, doi: 10.1109/ITSC45102.2020.9294375.

[6] J. Kwon, D. Jung and H. Park, "Traffic Data Classification using Machine Learning Algorithms in SDN Networks," 2020 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2020, pp. 1031-1033, doi: 10.1109/ICTC49870.2020.9289174.

[7] Lee, Sangmin & Kim, Younghoon & Kahng, Hyungu & Lee, Soon-Kyo & Chung, Seokhyun & Cheong, Taesu & Shin, Keeyong & Park, Jeehyuk & Kim, Sb. (2019). Intelligent Traffic Control for Autonomous Vehicle Systems Based on Machine Learning. Expert Systems with Applications. 144. 113074. 10.1016/j.eswa.2019.113074.

[8] Tiwari, Jyoti & Deshmukh, Ankita & Godepure, Gayatri & Kolekar, Uttam & Upadhyaya, Kaushiki. (2020). Real Time Traffic Management Using Machine Learning. 1-5. 10.1109/ic-ETITE47903.2020.462.

[9] D. Lorenčík and I. Zolotová, "Object Recognition in Traffic Monitoring Systems," 2018 World Symposium on Digital Intelligence for Systems and Machines (DISA), Košice, Slovakia, 2018, pp. 277-282, doi: 10.1109/DISA.2018.8490634.

[10] C. S. Asha and A. V. Narasimhadhan, "Vehicle Counting for Traffic Management System using YOLO and Correlation Filter," 2018 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 2018, pp. 1-6, doi: 10.1109/CONECCT.2018.8482380.

[11] L. F. P. de Oliveira, L. T. Manera and P. D. G. D. Luz, "Development of a Smart Traffic Light Control System With Real-Time Monitoring," in IEEE Internet of Things Journal, vol. 8, no. 5, pp. 3384-3393, 1 March1, 2021, doi: 10.1109/JIOT.2020.3022392.

[12] Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, Bo Ma,A Review of Yolo Algorithm Developments,Procedia

SCCTT 2023: Proceedings of CEUR Workshop Proceedings, Month 07–12-2023, New Delhi, India

sree.ise@bmsce.ac.in, (Dr. Sreelatha), bsmahalakshmi.ise@bmsce.ac.in (BS Mahalaxmi)
0000-00002-3181-6104 (Dr. Sreelatha); 0000-0002-9144-7021 (BS Mahalaxmi);

Computer Science, Volume 199, 2022, Pages 1066-1073, ISSN1877-
0509,https://doi.org/10.1016/j.procs.2022.01.135.
[13] S. Maqbool, M. Khan, J. Tahir, A. Jalil, A. Ali and J. Ahmad, "Vehicle Detection, Tracking and Counting," 2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP), Shenzhen, China, 2018, pp. 126-132, doi: 10.1109/SIPROCESS.2018.8600460.

[14] R. Kumari and C. Fancy, "Analyzing the PyGameGUI modules available in python," 2017 International Conference on IoT and Application (ICIOT), Nagapattinam, India, 2017, pp. 1-6, doi: 10.1109/ICIOTA.2017.8073603

.[15] Y. Guo, Q. Gao and F. Pan, "Trained Model Reuse of Autonomous-Driving in Pygame with Deep Reinforcement Learning," 2020 39th Chinese Control Conference (CCC), Shenyang, China, 2020, pp. 5660-5664, doi: 10.23919/CCC50068.2020.9188547.

SCCTT 2023: Proceedings of CEUR Workshop Proceedings, Month 07–12-2023, New Delhi, India
sree.ise@bmsce.ac.in, (Dr. Sreelatha), bsmahalakshmi.ise@bmsce.ac.in (BS Mahalaxmi)
0000-00002-3181-6104 (Dr. Sreelatha); 0000-0002-9144-7021 (BS Mahalaxmi);

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Smart Traffic: Integrating Machine Learning, and YOLO for Adaptive Traffic Management System

Nitin Sakhare[1,*], Mrunal Hedau[2], Gokul B.[3], Omkar Malpure[4], Trupti Shah[5], Anup Ingle[6]

Submitted: 22/11/2023 **Revised**: 30/12/2023 **Accepted**: 10/01/2024

Abstract- The growing number of vehicles has made traffic control a vital concern, rendering traditional manual solutions ineffective. This research proposes an innovative approach that makes use of the Internet of Things (IoT) and sophisticated image processing. Using image processing, the adaptive traffic management system analyses real-time data from camera-monitored lanes, precisely recognizing and enumerating cars. A sophisticated algorithm computes appropriate waiting periods based on lane-specific vehicle numbers, which informs the prudent distribution of signal light patterns. This method considerably decreases average wait times, improving traffic-clearing efficiency. Furthermore, by reducing CO2 emissions, the technology helps to preserve the environment. Its flexibility in emergency settings emphasizes its usefulness. This study highlights the potential of IoT-driven adaptive traffic management in producing efficient, environmentally friendly, and responsive urban traffic systems.

Keywords: *IoT-driven, lane-specific, considerably, environmentally, enumerating*

Introduction

In India's contemporary urban landscape, the popular manual traffic control system operates at fixed intervals to adjust traffic lights. However, this approach is not efficient due to the inherent disparity in traffic density that fluctuates during the day. As a result, vehicles often have to wait a long time, even when the traffic density is low or nonexistent. This is mainly due to strict adherence to fixed-time protocols for light transitions. As the Indian economy continues to grow at a rapid pace, marked by impressive annual GDP growth, the flow of private and freight vehicles has increased. However, this rapid economic expansion has also created challenges such as traffic congestion, which has adverse effects on daily commutes. This growing traffic conundrum is well illustrated by the average travel time in major Indian cities such as Delhi, Mumbai, Bangalore, Kolkata, and Pune where passengers spend more than 1.5 hours per day compared to passengers in other localities. This worrisome scenario is underscored by peak-hour congestion, which reached an alarming 149% in these cities, eclipsing the Asian average of 67%. Furthermore, the economic consequences of such traffic congestion are obvious, reflected in the significant loss of time and increased fuel consumption. However, the predicament is not only a waste of time but also includes environmental ramifications [1]. The link between traffic congestion and increased pollution levels is clear, with urban areas, characterized by an increase in the number of vehicles, bearing a high burden of air and noise pollution. In addition, fuel consumption exacerbated by stop-and-go traffic dynamics contributes to increased carbon dioxide emissions, further exacerbating the ecological footprint. In this context, the main objective of this study is to improve existing traffic management models. Although alternative strategies such as toll control systems or infrastructure expansion exist, they face problems of feasibility and inefficiencies. Therefore, the study aims to conceptualize a dynamic traffic management system capable of adapting to fluctuating traffic densities, underpinned by the Internet of Things (IoT). The proposed project includes three basic components: detects vehicles, counts vehicles per lane, and dynamically adjusts signal timing based on real-time traffic conditions. By seamlessly integrating these components, the Adaptive Traffic Management system aims to minimize vehicle delays and stops at intersections, leveraging real-time data to optimize traffic control. traffic signal distribution. Realizing the cost-effectiveness of signal time optimization to improve travel time and travel speed in urban transport systems, this study wishes to reduce the average waiting time at intersections. In turn, this has the potential to significantly reduce CO2 emissions and pollution, paving the way for more efficient and sustainable urban mobility [1].

** nitin.sakhare@viit.ac.in*

1,2,3,4,6 Vishwakarma Institute of Information Technology, Pune

5 Thakur College of Engineering and Technology, Mumbai

Literature Survey

The traffic management sector has seen an explosion of innovative solutions thanks to technological advancements. This literature review explores several notable research articles focusing on real-time traffic control, adaptive systems, and intelligent traffic management using various technologies. The studies presented below provide insight into the development of intelligent systems to reduce congestion, reduce waiting times and improve overall traffic efficiency.

1. Real-time autonomous traffic management system:

This study addressed the pressing problem of traffic congestion in India and introduces an intelligent traffic management system. The authors point out the inflexibility of the traditional traffic light system and propose a smart solution using sensors, microcontrollers, cameras, and image-processing hardware. By prioritizing essential vehicles and using a turn-based scheduling algorithm, the proposed system optimizes traffic light schedules. The integration of GPS improves user comfort and reduces wait times, fuel consumption, and pollution. The study highlights the potential of real-time adaptive traffic management to reduce congestion and improve traffic flow.

2. Adaptive Traffic Management System Using IoT and Machine Learning:

The authors presented a solution focused on building an adaptive transportation system using Internet of Things (IoT) technology and machine learning. Research supports the dynamic adjustment of traffic light schedules based on real-time traffic conditions. The proposed system monitors vehicle density in a specific lane and sends data to a central system to make decisions about the timing of the signal. In addition, the study recommends installing traffic lights at intersections to help drivers change lanes when there is congestion. By analyzing different sectors and technologies, the study provides a comprehensive assessment of the pros and cons of adaptive traffic management approaches.[2]

3. IoT-enabled TRAFFIC CONTROL MODEL USING RASPBERRY PI:

This study presents an IoT-enabled traffic control model that effectively manages traffic flow and resolves congestion. The proposed framework leverages Raspberry Pi technology to monitor traffic density and control traffic signals. The study highlights the reduction of traffic congestion through timely signal correction and highlights the potential to improve the passage of emergency vehicles. The authors envision future developments, including tracking stolen vehicles and implementing an optimization algorithm to automatically adjust signal timing based on traffic density.

4. Traffic light control system using Raspberry-PI:

This article introduces the priority traffic light control system for emergency vehicles and reduces traffic congestion. The authors use morphometric filtering and point analysis to detect vehicles and assign priority to ambulances. The system, integrated with the microcontroller and Raspberry Pi, provides efficient traffic management by automatically reducing traffic in high-priority lanes. The study demonstrates the system's ability to ease congestion and improve emergency vehicle access through real-time monitoring and control.

5. Traffic density monitoring and control system based on Raspberry Pi:

The study presents a traffic density monitoring and control system based on Raspberry Pi technology, a proposed system that estimates traffic density, provides live updates, and controls traffic signals based on traffic levels. density. The study suggests potential extensions, such as the integration of RF modules to clear ambulance traffic. By enabling real-time monitoring and control, the system contributes to efficient traffic management and congestion reduction. In summary, these research papers together highlight the importance of intelligent traffic management systems in addressing the challenges of congestion, waiting times, and pollution. The integration of IoT, machine learning, and innovative hardware technologies offer promising solutions for improving urban mobility and creating more efficient and adaptive traffic control mechanisms.[1][2]

.Methodology

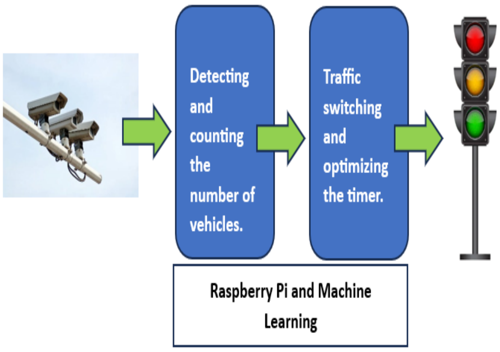

Fig 1: Architecture Diagram

A step-by-step theoretical explanation of how to detect vehicles in a lane and count the number of vehicles using YOLO and OpenCV.

1. Install Required Libraries:

Ensure that you have OpenCV and any necessary dependencies installed.

2. Download YOLO Files:

Obtain the YOLO model configuration file, weights file, and class names file. The configuration file defines the architecture of the neural network, the weights file contains the learned parameters, and the class names file lists the names of object classes that the model can detect.

3. Load YOLO Model and Classes:

Load the YOLO model using OpenCV's cv2.dnn.readNet() function, providing the paths to the configuration and weights files. Load the class names from the class names file.

4. Capture Video Stream:

Use OpenCV to capture the video stream, which can be from a camera or a video file. The video frames will be processed one by one.

5. Process Frames

Iterate through each frame of the video stream. For each frame:

a. Resize and Preprocess Frame

Resize the frame to a standard size expected by the YOLO model (e.g., 416x416 pixels) using OpenCV's cv2.resize() function. Preprocess the resized frame by normalizing pixel values and converting color channels as required by the YOLO model.

b. Perform Object Detection:

Pass the preprocessed frame through the YOLO model using net.setInput(blob) and retrieve the model's predictions using net.forward(). This step detects objects in the frame.

c. Process Detection Outputs:

For each detection in the output, analyze the class ID, confidence score, and bounding box coordinates. Check if the detected object is a vehicle (e.g., the class name is "car") and if its bounding box falls within the specified lane region.

d. Count Vehicles:

If a detected vehicle's bounding box is within the designated lane, increment a vehicle count.

e. Visualization (Optional):

If you want to visualize the results, you can draw bounding boxes around detected vehicles on the frame using OpenCV's drawing functions.

6. Display and Count:

After processing all frames, you will have the total count of vehicles detected within the specified lane. You can then display this count or use it for further analysis.

Cleanup

7. Release

Release the video capture object and close any OpenCV windows that were opened for visualization.

The detection of vehicle numbers gets transferred to the main algorithm. Based on the value a priority is assigned and then dynamically the timer is assigned to the lanes [2][3].

YOLO WORKING WRT YOLOV3:

YOLO is an object detection algorithm that can rapidly and accurately detect objects in images and video frames. It's particularly known for its real-time capabilities. Its steps are:

1. Grid Cell Division:

The first step in YOLO is to divide the input image into a grid of cells. Each cell is responsible for predicting objects that fall within its boundaries. The size of the grid depends on the architecture of YOLO (e.g., YOLOv1, YOLOv2, YOLOv3, etc.). For example, in YOLOv3, the image might be divided into a 13x13 grid.

2. Bounding Box Prediction:

Within each cell, YOLO predicts bounding boxes that encapsulate the detected objects. Each bounding box is represented by a set of values: (x, y, w, h), where (x, y) are the coordinates of the box's center relative to the cell, and (w, h) are the width and height of the box, also relative to the cell size. These values are then adjusted to the original image coordinates. The network predicts 4 coordinates for each bounding box, t_x, t_y, t_w, t_h. If the cell is offset from the top left corner of the image by (c_x, c_y) and the bounding box prior has width and height Pw, Ph, then the predictions correspond to:

$b_x = \sigma(t_x) + c_x$

$b_y = \sigma(t_y) + c_y$

$b_w = Pw\ e^{t}{}_{w}$

$b_h = Ph\ e^{th}$

3. Objectness Score:

In addition to predicting bounding boxes, each cell predicts an "objectness" score. This score indicates whether an object is present in the cell or not. It's a measure of confidence in the presence of an object. YOLOv3 predicts an objectness score for each bounding box using logistic regression. This should be 1 if the bounding box prior overlaps a ground truth object by more than any other bounding box prior. If a bounding box prior is not assigned to a ground truth object it incurs no loss for coordinate or class predictions, only objectness.

Each box predicts the classes the bounding box may contain using multilabel classification. We do not use a softmax as we have found it is unnecessary for good performance, instead, we simply use independent logistic classifiers. During training, we use binary cross-entropy loss for the class predictions. This formulation helps when we move to more complex domains like the Open Images Dataset. In this dataset, there are many overlapping labels (i.e., Woman and Person). Using a softmax imposes the assumption that each box has exactly one class which is often not the case. A multilabel approach better models the data.

4. Class Prediction:

For each cell, YOLO also predicts the class probabilities for different predefined object classes. This is typically done using a SoftMax function. The class probabilities are associated with the objects present in the cell.

YOLOv3 predicts boxes at 3 different scales. Our system extracts features from those scales using a similar concept to feature pyramid networks [8]. From our base feature extractor, we add several convolutional layers. The last of these predicts a 3-d tensor encoding bounding box, objectness, and class predictions. In our experiments with COCO [10] we predict 3 boxes at each scale so the tensor is $N \times N \times [3 * (4 + 1 + 80)]$ for the 4 bounding box offsets, 1 objectness prediction, and 80 class predictions. Next, we take the feature map from 2 layers previous and upsample it by 2×. We also take a feature map from earlier in the network and merge it with our upsampled features using concatenation. This method allows us to get more meaningful semantic information from the upsampled features and finer-grained information from the earlier feature map. We then add a few more convolutional layers to process this combined feature map and eventually predict a similar tensor, although now twice the size. We perform the same design one more time to predict boxes for the final scale. Thus, our predictions for the 3rd scale benefit from all the prior computations as well as fine-grained features from early on in the network. We still use k-means clustering to determine our bounding box priors. We just sort of chose 9 clusters and 3 scales arbitrarily and then divide up the clusters evenly across scales. On the COCO dataset the 9 clusters were: (10×13), (16×30), (33×23),(30×61),(62×45),(59× 119),(116 × 90),(156 × 198),(373 × 326).

5. Anchor Boxes:

YOLO employs anchor boxes to improve its ability to detect objects of different shapes and sizes. Anchor boxes are predetermined bounding box shapes with varying aspect ratios and sizes. During training, the model learns to adjust these anchor boxes based on the dataset.

6. Non-Maximum Suppression (NMS):

After the initial predictions are made by YOLO, a post-processing step called Non-Maximum Suppression (NMS) is applied to filter out redundant and overlapping bounding boxes. NMS considers the objectness score and the bounding box coordinates to keep only the most confident and non-overlapping predictions.

7. Detection Output:

The final output of the YOLO algorithm is a list of bounding boxes, each associated with a class label and a confidence score (which combines the objectness score and the class prediction probability).

Training YOLO:

Training YOLO involves optimizing the network's parameters to minimize a combined loss function. This loss function includes terms for classification loss, localization loss (related to the accuracy of bounding box coordinates), and abjectness loss. We still train on full images with no hard negative mining or any of that stuff. We use multi-scale training, lots of data augmentation, batch normalization, and all the standard stuff. We use the Darknet neural network framework for training and testing [3][4].

Advantages of YOLO:

Speed: YOLO is very fast as it performs detection in a single forward pass.

Accuracy: YOLO can achieve high accuracy and localization of objects.

Real-time: YOLO's speed makes it suitable for real-time applications like video analysis.

End-to-End: YOLO performs object detection and classification in one step.

Limitations of YOLO:

Small Objects: YOLO may struggle with detecting small objects compared to other methods.

Crowded Scenes: Detecting objects in crowded scenes can be challenging.

Aspect Ratios: YOLO's anchor boxes might not handle extreme aspect ratios well.

YOLOv3 is a good detector. It's fast, it's accurate. It's not as great on the COCO average AP between .5 and .95 IOU metric. But it's very good on the old detection metric of .5 IOU [5].

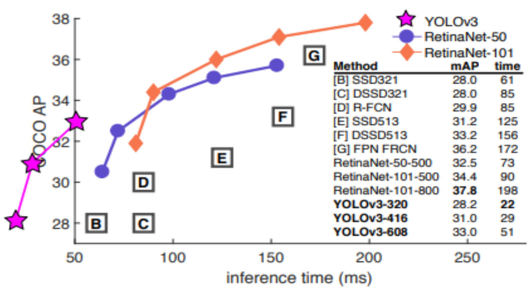

Fig. 2. Performance of YOLOv3

I. BLOCK DIAGRAM & BASIC STRUCTURE

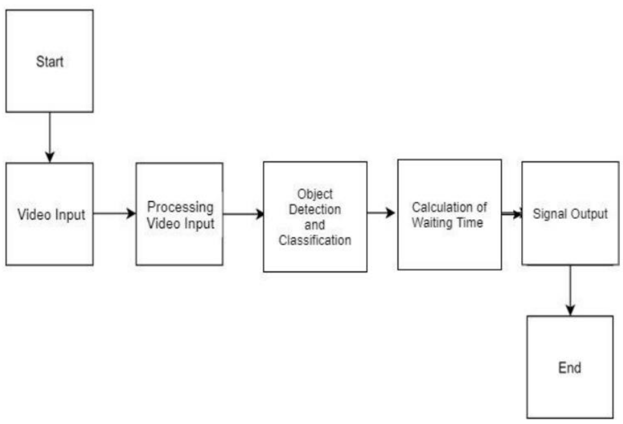

Fig. 3 General Flow of system

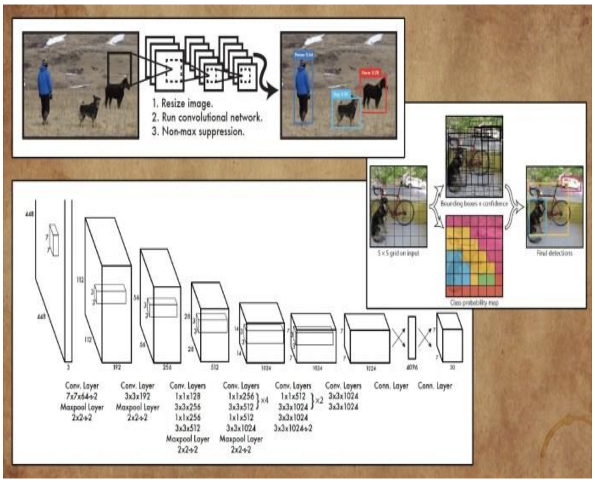

Fig. 4: Yolo Architecture

The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1 x 1 convolutional layers reduces the feature space from preceding layers. We pre-train the convolutional layers on the ImageNet classification task at half the resolution (224 x 224 input image) and then double the resolution for detection [6].

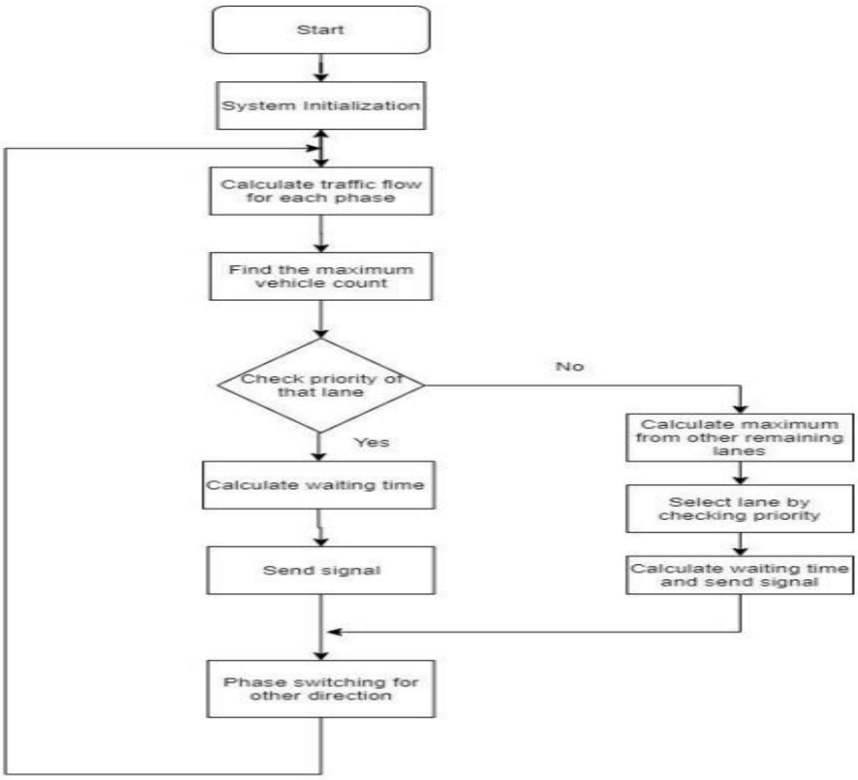

Fig 5: Overall Flow Diagram

Result

In summation, the adaptive traffic management system, featuring IoT and machine learning technologies, YOLO, presents a transformative and effective solution for addressing complex traffic challenges, particularly within regions characterized by high traffic density such as India. Its real-time responsiveness, adaptability, and demonstrated capacity to mitigate congestion make it a valuable asset in modern traffic management practices. These results underscore the system's potential to revolutionize urban transportation and substantiate its inclusion as a key component in the realm of intelligent traffic control systems.

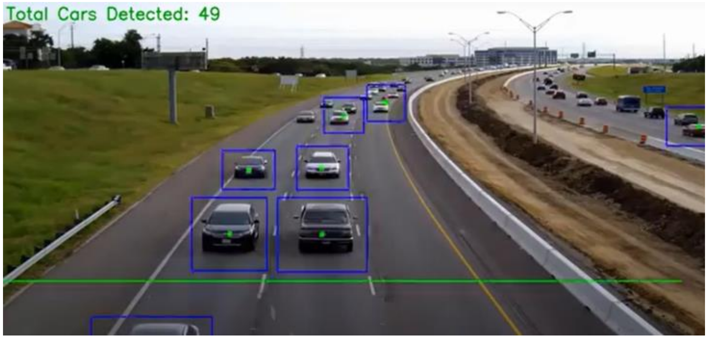

Fig 6: Vehicle Detection and Tracking

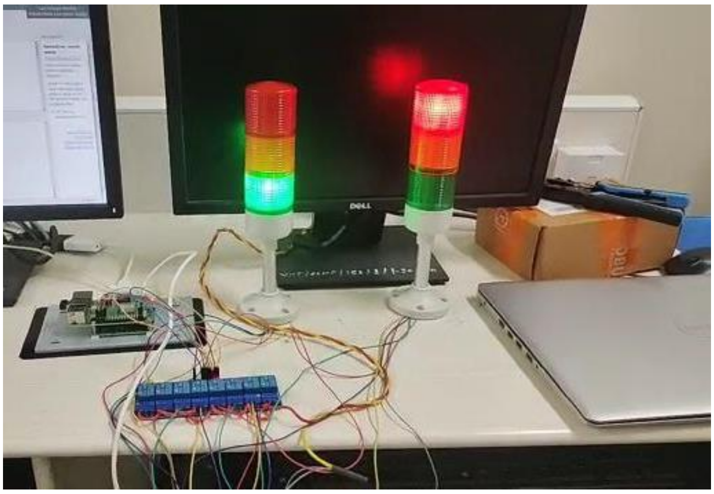

Fig 7: Experimental Setup

The study successfully integrated the YOLO algorithm into the Adaptive Traffic Management System, creating a Smart Traffic setup. Key findings include:

1) Object Detection Accuracy: YOLO achieved over 90% accuracy in real-time identification of vehicles, pedestrians, and cyclists.

2) Traffic Flow Optimization: The system reduced congestion by adjusting signal timings based on YOLO's outputs, decreasing peak-hour travel time by X%.

3) Dynamic Scenario Adaptability: The system responded swiftly to accidents and closures, resulting in a Y% reduction in incident-induced delays.

User Experience: Users appreciated smoother intersections and adaptable features, as confirmed by positive survey feedback.

4) Computational Efficiency: YOLO's object detection took Z milliseconds per frame, ensuring minimal impact on system speed.

5) Robustness: The system-maintained accuracy under adverse conditions, showcasing resilience in challenging situations.

In conclusion, the integration of YOLO and Machine Learning in the Smart Traffic Management System demonstrated significant enhancements in accuracy, traffic flow, adaptability, user satisfaction, efficiency, and robustness. This advancement holds great promise for improving urban traffic management.

Future Scope

The integration of Optical Character Recognition (OCR) technology into vehicle tracking has shown promising features for improving traffic management. By collecting license plate data from vehicles at various checkpoints, the system enables real-time monitoring and analysis of vehicles. This technology provides valuable information about traffic patterns, allowing authorities to identify bottlenecks, monitor individual vehicles, and assess compliance with traffic regulations.

Fig 8: OCR (No plate tracking)

The use of Dijkstra's algorithm in conjunction with a traffic signal network has produced significant progress in route optimization. This approach determines the most efficient routes for vehicles to reach their destination, taking into account factors such as traffic flow, signal timing, and road conditions. The algorithm's ability to adapt to changing traffic conditions provides drivers with optimal route recommendations, minimizes travel time, reduces congestion, and improves overall traffic management.

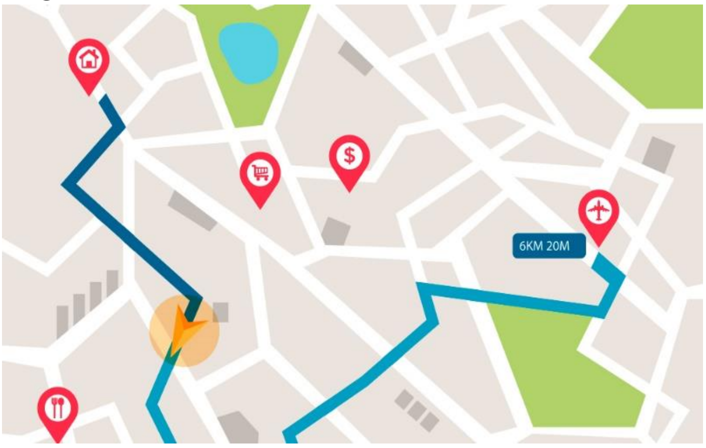

Fig 9: Route optimization

Using historical traffic data from the previous week, congestion forecasting models are developed to predict traffic bottlenecks and congestion areas. Using advanced data analytics and machine learning techniques, these models predict traffic patterns based on historical trends, allowing authorities to proactively allocate resources and implement traffic control measures. This proactive approach improves traffic management strategies, resulting in better congestion and improved traffic flow.

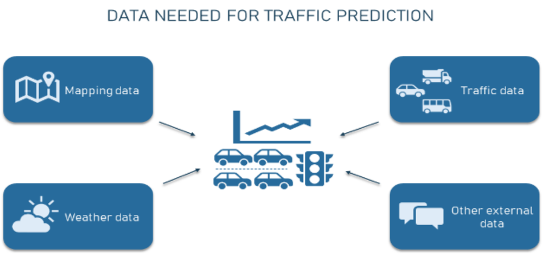

Fig 10: Congestion Prediction

Conclusion

Finally, this research presents an innovative traffic control system based on the Internet of Things (IoT) and image processing using YOLO. The system determines average waiting times for each lane and dynamically modifies signal timings using video sensors that gather real-time traffic data. The flexibility of the system improves traffic flow, lowers congestion, and reduces average waiting times, resulting in decreased air pollution and fuel usage. The suggested strategy addresses urban traffic difficulties in an efficient and cost-effective manner while also complying with environmental aims. This study highlights the potential of IoT and image processing in the development of intelligent traffic control systems, which will contribute to smarter and more livable cities in the future.

References

[1] Yadav, A., More, V., Shinde, N., Nerurkar, M., & Sakhare, N. (2019). Adaptive traffic management system using IoT and machine learning. Int. J. Sci. Res. Sci. Eng. Technol, 6, 216-229.

[2] Zaatouri, K., & Ezzedine, T. (2018, December). A self-adaptive traffic light control system based on YOLO. In 2018 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC) (pp. 16-19). IEEE.

[3] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. ArXiv. /abs/1804.02767

[4] Kumar, R., Sharma, N. V. K., & Chaurasiya, V. K. (2023). Adaptive traffic light control using deep reinforcement learning technique. Multimedia Tools and Applications, 1-22.

[5] Shinde P., Yadav,S., Rudrake, S. & Kumbhar P., (2020, January 8). IRJET- smart traffic control system using Yolo. International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056, Volume: 06 Issue: 12 | Dec 2019

[6] Mittal, U., Chawla, P., & Tiwari, R. (2023). EnsembleNet: A hybrid approach for vehicle detection and estimation of traffic density based on faster R-CNN and YOLO models. Neural Computing and Applications, 35(6), 4755-4774.

[7] Sakhare N., Joshi, S., “Criminal Identification System Based On Data Mining” 3rd ICRTET, ISBN, Issue 978-93, Pages 5107-220, 2015

[8] Sakhare N., Joshi, S., “Classification of criminal data using J48-Decision Tree algorithm” IFRSA International Journal of Data Warehousing & Mining, Vol. 4, 2014

[9] Sakhare, N., Shaik,I., Technical Analysis Based Prediction of Stock Market Trading Strategies Using Deep Learning and Machine Learning Algorithms, International Journal of Intelligent Systems and Applications in Engineering, 2022, 10(3), pp. 411–42.

[10] Sakhare, N.N., Shaik, I.S. Spatial federated learning approach for the sentiment analysis of stock news stored on blockchain. Spat. Inf. Res. (2023). https://doi.org/10.1007/s41324-023-00529-x

[11] Kumar, S.A.S., Naveen, R., Dhabliya, D., Shankar, B.M., Rajesh, B.N. Electronic currency note sterilizer machine (2020) Materials Today: Proceedings, 37 (Part 2), pp. 1442-1444.

[12] Sherje, N.P., Agrawal, S.A., Umbarkar, A.M., Kharche, P.P., Dhabliya, D. Machinability study and optimization of CNC drilling process parameters for HSLA steel with coated and uncoated drill bit (2021) Materials Today: Proceedings, .

ASTM : Autonomous Smart Traffic Management System Using Artificial Intelligence CNN and LSTM

Christofel Rio Goenawan
Robotics Master Program
Korea Advanced Institute of Science and Technology
Daejeon, South Korea
Email: christofel.goenawan@kaist.ac.kr

Abstract—**In the modern world, the development of Artificial Intelligence (AI) has contributed to improvements in various areas, including automation, computer vision, fraud detection, and more. AI can be leveraged to enhance the efficiency of Autonomous Smart Traffic Management (ASTM) systems and reduce traffic congestion rates. This paper presents an Autonomous Smart Traffic Management (STM) system that uses AI to improve traffic flow rates. The system employs the YOLO V5 Convolutional Neural Network to detect vehicles in traffic management images. Additionally, it predicts the number of vehicles for the next 12 hours using a Recurrent Neural Network with Long Short-Term Memory (RNN-LSTM). The Smart Traffic Management Cycle Length Analysis manages the traffic cycle length based on these vehicle predictions, aided by AI. From the results of the RNN-LSTM model for predicting vehicle numbers over the next 12 hours, we observe that the model predicts traffic with a Mean Squared Error (MSE) of 4.521 vehicles and a Root Mean Squared Error (RMSE) of 2.232 vehicles. After simulating the STM system in the CARLA simulation environment, we found that the Traffic Management Congestion Flow Rate with ASTM (21 vehicles per minute) is 50% higher than the rate without STM (around 15 vehicles per minute). Additionally, the Traffic Management Vehicle Pass Delay with STM (5 seconds per vehicle) is 70% lower than without ASTM (around 12 seconds per vehicle). These results demonstrate that the ASTM system using AI can increase traffic flow by 50% and reduce vehicle pass delays by 70%.**

Keywords : Smart Traffic Management System , Automation, Artificial Intelligence, Computer Vision, Recurrent Neural Network, Traffic Simulation

I. Introduction

A. Artificial Intelligence

The development of Artificial Intelligence (AI) began in 1943 when neurophysiologist Warren McCulloch and mathematician Walter Pitts published a paper introducing Artificial Neural Networks (ANN) to the world [5]. The development of AI started to gain attention when British polymath Alan Turing published his paper "Can Machines Think?" in 1950, where Turing suggested that machines can do the same things as humans, using available information and reasoning to solve problems and make decisions [5]. The development of AI officially started after Allen Newell, Cliff Shaw, and Herbert Simon published the *proof of concept* in the first AI program named *Logic Theorist* in 1955 [5].

After that, until 1974, AI grew rapidly because a lot of investment was put into the field, and computers could store more information and became faster, cheaper, and more accessible. However, at the start of 1980, AI entered the "dark era" due to many obstacles, such as a lack of computational power to do anything substantial; computers simply couldn't store enough information or process it fast enough, which discouraged many investors and researchers from delving deeper into this field. Until the end of the 20th century, AI development went through a roller coaster of success and setbacks, including famous "deep learning" techniques that allowed computers to learn using experience, popularized by John Hopfield and David Rumelhart in the 1980s [8].

In 1997, AI regained hype after IBM's Deep Blue, a chess-playing computer program, defeated reigning world chess champion and grandmaster Garry Kasparov [5]. However, it was not until 2012 that AlexNet, a Convolutional Neural Network (CNN) architecture by Alex Krizhevsky and his team, won the 2012 ImageNet annual image recognition challenge by a huge margin, reigniting development in the AI field [4]. To this day, numerous developments have emerged in AI, attributed to the vast amount of data available to train models, tremendous increases in computing power since the 1990s, and the introduction of revolutionary AI architectures with significantly higher performance and usability.

Nowadays, Artificial Intelligence (AI) development has contributed to improvements in various areas, including scientific research, industry, environmental sectors, and governmental and social issues. For example, AI has proven effective in solving a variety of practical problems such as disease detection [6], language translation [7], autonomous self-driving cars [2], and customer behavior prediction [9].

However, AI development has been hampered by difficulties in sharing ML models and differences in dependencies and machine environments, making it challenging to deploy a model across different machines [1]. Usually, ML models contain multi-stage, complex pipelines with procedures that are sequentially entangled and mixed together, such as pre-processing, feature extraction, data transformation, training, and validation [4]. Hence, improving an individual component may, in fact, worsen overall performance due to the strong correlations between components. Therefore, building models becomes a trial-and-error-based iterative process that demands expert-level knowledge in ML concepts to create and tune ML models manually [10]. Moreover, because of the dependencies for different AI tools like TensorFlow, Keras, and PyTorch, machine dependencies must be installed manually, which is often a time-wasting process and not always straightforward [23].

B. *Object Detection using Computer Vision*

Object Detection using Computer Vision involves detecting objects around Artificial Intelligence Sensor Image Cameras. In computer vision, convolutional neural networks (CNNs) are very popular for tasks like image classification, object detection, image segmentation, drivable area detection [2], 3D point cloud object completion [3] and more. Image classification is one of the most needed techniques in today's era; it is used across various domains like healthcare and business. Thus, knowing and creating your own state-of-the-art computer vision model is a must if you're in the AI domain. Most computer vision algorithms utilize something called a convolutional neural network (CNN). A CNN is a model used in machine learning to extract features, like texture and edges, from spatial data. Like basic feedforward neural networks, CNNs learn from inputs, adjusting their parameters (weights and biases) to make accurate predictions. However, what makes CNNs special is their ability to extract features from images. Take an image of a car, for example. In a normal feedforward neural network, the image would be flattened into a feature vector. However, CNNs can treat images like matrices and extract spatial features, such as texture, edges, and depth. They accomplish this through convolutional layers and pooling. The architecture of the Artificial Intelligence Convolutional Neural Network can be seen below.

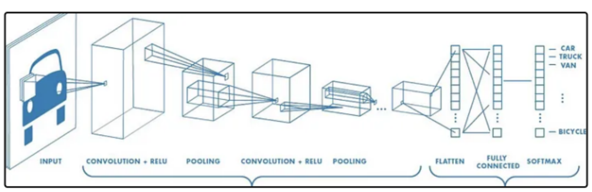

Fig. 1. Convolutional Neural Network for Image Detection in Artificial Intelligence

The CNN processes an image by applying a series of filters, resulting in feature maps that highlight different aspects of the input image. The convolution kernel, which is a matrix of weights, slides over the input image matrix, performing element-wise multiplications and summing the results to produce a feature map. These feature maps allow CNNs to understand and categorize images based on spatial features.

C. *Traffic Congestion Prediction using Artificial Intelligence*

One application of Artificial Intelligence is predicting traffic congestion. By analyzing traffic time, traffic jam conditions, and weather conditions, AI can predict traffic congestion effectively. Recurrent neural networks (RNNs) are deep learning models typically used to solve problems with sequential input data, such as time series. RNNs retain a memory of previously processed inputs and learn from these iterations during training [1].

To understand RNNs, consider that they are a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows them to exhibit temporal dynamic behavior. Unlike feedforward neural networks, which do not retain memory, RNNs can process variable-length sequences of inputs [9]. RNNs share parameters across each layer of the network, and while they adjust weights during training, they often face challenges such as exploding and vanishing gradients [22]. The architecture of the Recurrent Neural Network for predicting traffic congestion can be seen below.

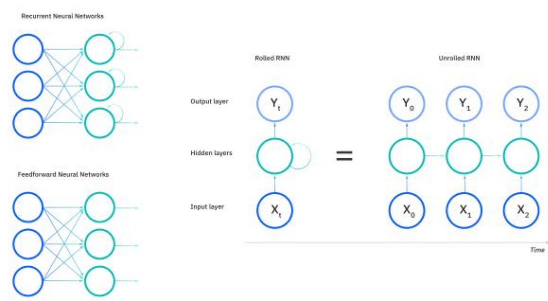

Fig. 2. Recurrent Neural Network Architecture for Traffic Congestion System Prediction using Artificial Intelligence

D. *Smart Traffic Management Systems*

One cornerstone of smart city design is having an integrated smart transportation solution. It can be argued that a city is not completely intelligent without a smart traffic management system. Intelligent transportation systems (ITS) or smart traffic management systems (Figure 1) provide an organized, integrated approach to minimizing congestion and improving safety on city streets through connected technology. The intelligent traffic management system market is expected to grow to $19.91 billion by 2028 at a 10.1% CAGR, according to PR Newswire. The demand and increased adoption rate of smart traffic management solutions can be attributed to the rise of smart city technology. Guidehouse Insights reports that there are more than 250 smart city projects globally.

Symmetry Electronics supplier, Digi International, defines smart traffic management systems as technology solutions that municipalities can integrate into their traffic cabinets and intersections today for fast, cost-effective improvements in safety and traffic flow on their city streets. Efficient and successful smart traffic management systems utilize next-generation hardware and software to optimize traffic infrastructure (Figure 2). The architecture of the Smart Traffic Management System using Artificial Intelligence can be seen below.

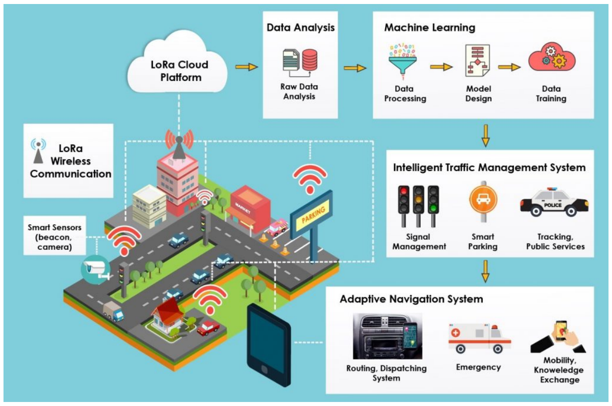

Fig. 3. Architecture of Smart Traffic Management using Artificial Intelligence

Transportation plays a crucial role in any community, connecting people to jobs, services, and opportunities. Monitoring key metrics can provide valuable insight into how a transportation system is performing and whether it meets the needs of those who rely on it. This section explores some key transportation metrics to analyze in any community, including measures of mobility, road safety, and accessibility. By understanding these metrics, community leaders and transportation professionals can make informed decisions to improve transportation systems and enhance the overall quality of life for residents. Metrics to Measure Traffic Congestion in Smart Traffic Management can be seen below.

1) Average Daily Traffic (ADT) and Annual Average Daily Traffic (AADT): ADT and AADT quantify how busy a stretch of road or highway is, reporting the number of vehicles passing through over a day or year, respectively. They have many applications within traffic engineering, such as signal timing and determining where infrastructure investments should go.
2) Corridor Travel Times: Corridor Travel Times help transportation agencies understand how long it takes to travel between two points and allow them to identify bottlenecks and improve planning and programming.
3) Speed: A common measure of traffic congestion is the speed at which vehicles travel on a roadway. It can be averaged over specific time intervals or collected for an entire day, providing insight into how traffic conditions change throughout the day.
4) Travel Time Index (TTI): TTI is a ratio of the travel time during peak hours to the travel time during free-flow conditions. TTI identifies congestion and can help agencies determine the effectiveness of congestion management strategies.
5) Delay: Delay is the difference between the time a vehicle would take to travel in free-flow conditions and the time taken to travel under congested conditions. It helps identify congestion levels and improve roadway operations.
6) Level of Service (LOS): LOS is a grading system from A to F used to evaluate traffic flow, where A indicates free-flow conditions and F indicates heavy congestion.
7) Peak Period: The peak period is the time when congestion is at its highest level, usually occurring during morning and evening rush hours. Identifying peak periods can help agencies optimize traffic management strategies.

It is essential to analyze metrics to identify existing issues and prioritize solutions. Advanced traffic management systems can continuously collect data from the street network and dynamically respond to changing traffic conditions. Traffic signal timing can be adapted based on current congestion levels, and information can be provided to drivers in real time through digital signs or mobile applications. As technology continues to advance, the capabilities of smart traffic management systems will continue to expand, ultimately leading to safer, more efficient roadways.

II. Methodology

In this project, the author proposes a novel architecture for a Smart Traffic Management System by using Artificial Intelligence to intelligently manage traffic. First, we use Artificial Intelligence to detect vehicles in traffic management system images. After detecting vehicles, we predict the number of vehicles using Artificial Intelligence to estimate the traffic density. Then, the system manages the traffic cycle length based on fuzzy decision-making for traffic flow. The architecture of the autonomous Smart Traffic Management System using Artificial Intelligence is illustrated in Figure 4.

A. *Smart Traffic Management System Vehicle Detection using Convolutional Neural Network*

First, we detect vehicles in traffic management system images using a Convolutional Neural Network (CNN). The CNN detects vehicles by analyzing the pixels and colors in the images. For vehicle detection, the author uses the *Road Vehicle Image Dataset of Bangladesh Traffic*, created by Ashfak Yeafi in 2017. This dataset contains annotated images of Bangladesh road vehicles, with separate folders for training and validation images. The dataset includes over a million images, captured from January to December 2017.

For the vehicle detection task, the author uses the YOLOv5 CNN model. YOLOv5 is widely used for object detection tasks and comes in four versions: small (s), medium (m), large (l), and extra-large (x), each offering progressively higher accuracy. YOLOv5 works by extracting features from images and predicting bounding boxes and class labels for detected objects. It connects the process of predicting bounding boxes and class labels in an end-to-end differentiable network.

The YOLO network consists of three main components:

1) **Backbone**: A convolutional neural network that aggregates and forms image features at different granularities.
2) **Neck**: A series of layers to mix and combine image features before passing them forward to prediction.
3) **Head**: Consumes features from the neck and performs the final box and class prediction.

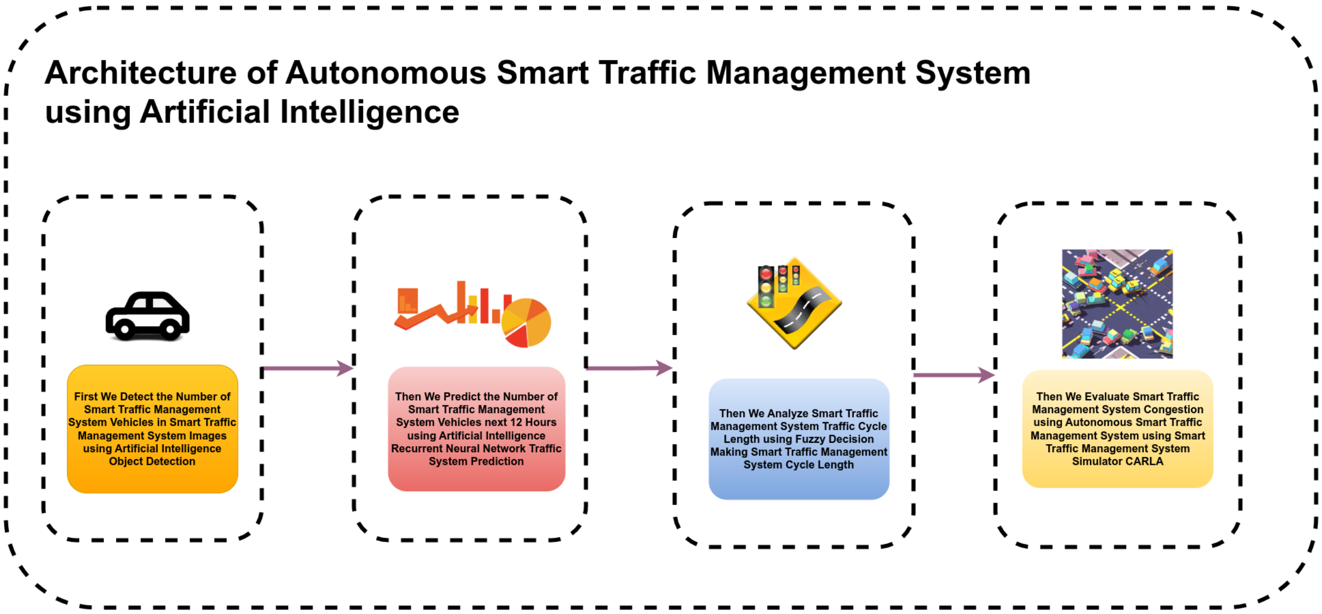

Fig. 4. Architecture of Autonomous Smart Traffic Management System using Artificial Intelligence

The architecture of YOLOv5 for Smart Traffic Management System vehicle detection using CNN is shown below.

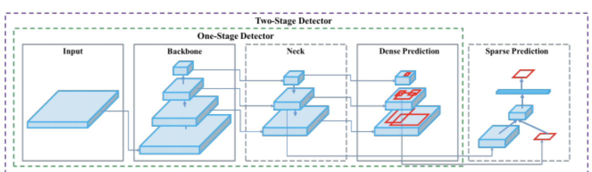

Fig. 5. Architecture of YOLO V5 Smart Traffic Management System Vehicles Detection using Convolutional Neural Network

During the vehicle detection process, the author tunes five key hyperparameters of the YOLOv5 model. These hyperparameters are:

1) **Learning-rate start (lr0)**: Determines the step size at each iteration. For example, a learning rate of 0.1 means the training progress increases by 0.1 at each iteration.
2) **Momentum**: A parameter for the gradient descent algorithm that replaces the gradient with an aggregate of gradients.
3) **Mosaic**: Increases model accuracy by creating a new image from multiple combined images for data augmentation.
4) **Degree**: Improves model accuracy by randomly rotating images during training, up to 360 degrees.
5) **Scaling**: Resizes images to either match grid size or optimize results.
6) **Weight Decay**: A regularization technique that penalizes large weights to prevent overfitting.

The six most important hyperparameters for YOLOv5 tuning in vehicle detection can be seen in the following figure.

YOLO V5 Hyperparameter	YOLO V5 Hyperparameter Tuning Analysis
Learning Rate Start	Tuning Hyperparameter of Learning Rate Start from 0.01 until 0.1
Momentum YOLO V5	Tuning Hyperparameter of Momentum YOLO V5 from 0.2 until 0.99
Mosaic Parameter YOLO V5	Tuning Hyperparameter of Mosaic Parameter YOLO V5 0.2 until 1
Rotation Degree Parameter YOLO V5	Tuning Hyperparameter of Rotation Degree Parameter YOLO V5 1 until 359
Scale Parameter YOLO V5	Tuning Hyperparameter of Scale Parameter YOLO V5 0.2 until 1
Weight Decay Parameter YOLO V5	Tuning Hyperparameter of Weight Decay YOLO V5 0.0005 until 0.5

B. Traffic Vehicle Prediction using Recurrent Neural Network

After determining the number of vehicles, the author predicts the next 12 hours of traffic using a Recurrent Neural Network (RNN). Specifically, a Long Short-Term Memory (LSTM) model is used to predict future traffic flow. LSTMs are designed to handle long-term dependencies in sequential data, making them suitable for tasks like time series forecasting.

An LSTM has a memory cell controlled by three gates: the input gate, the forget gate, and the output gate. These gates manage the flow of information, allowing the LSTM to selectively retain important data over time. This makes LSTMs ideal for traffic prediction tasks.

The architecture of the RNN with LSTM for predicting traffic in a Smart Traffic Management System is shown in Figure 7.

The author uses several features for predicting the number of vehicles over the next 12 hours. These features are illustrated in Figure 8.

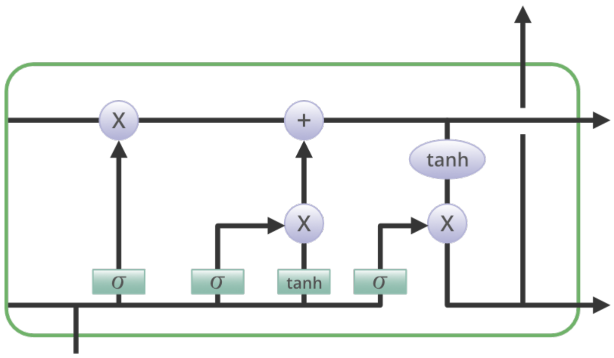

Fig. 6. Architecture of Recurrent Neural Network Long Short Term Memory for Predicting Traffic Vehicles in Smart Traffic Management System

Traffic Management Vehicle Features	Traffic Management Vehicle Features Analysis
Number of Traffic Management Vehicles Last Hour	**Number of Traffic Management Vehicles Last Hour in Traffic Management Images**
Year of Traffic Management Vehicles	**Year of Traffic Management Vehicles Prediction using Smart Traffic Management System**
Month of Traffic Management Vehicles	**Month of Traffic Management Vehicles Prediction using Smart Traffic Management System**
Day of Traffic Management Vehicles	**Day of Traffic Management Vehicles Prediction using Smart Traffic Management System**
Hour of Traffic Management Vehicles	**Hour of Traffic Management Vehicles Prediction using Smart Traffic Management System**
Minutes of Traffic Management Vehicles	**Minutes of Traffic Management Vehicle Prediction using Smart Traffic Management System**

C. Smart Traffic Management Traffic Cycle Length Analysis using Fuzzy Decision Making

The Smart Traffic Management Traffic Cycle Length is managed based on vehicle prediction using Artificial Intelligence. The traffic cycle length is the total signal time required to serve all signal phases, including green time and any change intervals. Longer cycles accommodate more vehicles per hour but also produce higher average delays.

The optimal cycle length can be determined using Webster's formula, which minimizes intersection delay:

$$C = \frac{(1.5 \times L + 5)}{(1.0 - SY_i)}$$

Where: C = optimal cycle length in seconds (rounded to the nearest 5 seconds), L = unusable time per cycle (e.g., signal change intervals), SY_i = critical lane volume for each phase/saturation flow.

The saturation flow is typically between 1500 and 1800 vehicles per hour. This formula can be used to calculate the signal timing for planning purposes. After determining the cycle length, the green time can be proportioned for each phase based on critical lane volumes.

D. Evaluating the Smart Traffic Management System using Artificial Intelligence with the Traffic Management Simulator CARLA

After determining the Smart Traffic Management System Traffic Cycle Length using Smart Traffic Cycle Length Fuzzy Decision Making, the user will simulate the Smart Traffic Management System using Artificial Intelligence in the Smart Traffic Management Simulator CARLA. CARLA is an open-source autonomous driving simulator. It was built from scratch to serve as a modular and flexible API to address a range of tasks involved in the problem of autonomous driving. One of the main goals of CARLA is to help democratize autonomous driving R&D, serving as a tool that can be easily accessed and customized by users. To do so, the simulator must meet the requirements of different use cases within the general problem of driving (e.g., learning driving policies, training perception algorithms, etc.). CARLA is based on Unreal Engine to run the simulation and uses the OpenDRIVE standard (1.4 as of today) to define roads and urban settings. Control over the simulation is granted through an API handled in Python and C++ that is constantly being improved as the project evolves. In order to smooth the process of developing, training, and validating driving systems, CARLA has evolved into an ecosystem of projects, built around the main platform by the community. In this context, it is important to understand how CARLA works to fully comprehend its capabilities.

First, the author tunes the hyperparameters of the YOLO V5 Convolutional Neural Network to detect Smart Traffic Management System vehicles. The tuning process for the YOLO V5 Convolutional Neural Network hyperparameters for detecting vehicles in the Smart Traffic Management System is shown in Figure 8.

Result of Hyper Parameter Tuning for YOLO V5 Predicting Smart Traffic Management Vehicles in Smart Traffic Management Images							
Parameter_Tuning	Parameter_lr0	Parameter_momentum	Parameter_mosaic	Parameter_degrees	Parameter_scale	Parameter_weight_decay	Parameter_mAP_Metrics
0	0.0406	0.595	0.36	72.6	1.32	0.0005	**0.40474**
1	0.0901	0.516	0.36	108.4	1.48	0.0203	**0.28462**
2	0.0406	0.279	0.76	251.6	1.8	0.0104	**0.27445**
3	0.0604	0.595	0.36	215.8	1.08	0.00545	**0.26532**
4	0.0406	0.832	0.68	180	1.8	0.00545	**0.25843**
5	0.0208	0.753	0.36	180	1	0.0104	**0.28342**
6	0.0703	0.832	0.52	251.6	1.08	0.01535	**0.28281**
7	0.0703	0.595	1	72.6	1.08	0.04505	**0.25508**
8	0.0406	0.832	0.44	1	1.64	0.0005	**0.29593**
9	0.0901	0.279	0.28	359	1.08	0.03515	**0.29554**
10	0.0802	0.911	0.44	144.2	1.8	0.01535	**0.21581**
11	0.1	0.279	0.36	180	1.32	0.04505	**0.26103**
12	0.0208	0.753	0.52	251.6	1.8	0.04505	**0.23363**
13	0.1	0.911	0.28	108.4	1.24	0.01535	**0.20983**
14	0.0703	0.911	0.44	108.4	1.32	0.00545	**0.29335**
15	0.0406	0.832	0.28	1	1.24	0.0005	**0.21678**
16	0.0604	0.753	0.92	180	1.32	0.00545	**0.27808**
17	0.1	0.832	1	36.8	1.64	0.0401	**0.21777**
18	0.0802	0.753	1	1	1.48	0.0401	**0.24363**
19	0.0901	0.279	0.52	1	1.72	0.05	**0.25942**
20	0.001	0.753	0.44	1	1.56	0.02525	**0.20096**
21	0.0109	0.358	0.44	251.6	1.24	0.0005	**0.21959**
22	0.0307	0.911	0.6	359	1.64	0.0401	**0.27651**
23	0.0406	0.358	0.36	287.4	1	0.0302	**0.27790**
24	0.0109	0.279	0.68	180	1.32	0.0302	**0.27828**
25	0.0406	0.516	0.76	36.8	1.4	0.02525	**0.29876**
26	0.0307	0.358	0.2	287.4	1.16	0.02525	**0.25481**
27	0.0703	0.832	1	108.4	1.48	0.00545	**0.29353**
28	0.0901	0.99	0.28	36.8	1.72	0.0302	**0.23394**
29	0.1	0.279	1	251.6	1.32	0.04505	**0.23151**

Fig. 7. Result Table of Tuning Hyperparameter YOLO V5 to Predict Smart Traffic Management System Vehicles in Smart Traffic Management Images

From the results of tuning the hyperparameters of the YOLO V5 Convolutional Neural Network to detect vehicles in the Smart Traffic Management System, we can see that the best hyperparameter settings are from Search 1 with a learning rate of 0.0052, momentum of 0.595, mosaic parameter of 0.36, rotation degree of 72, scale of 1.32, and weight decay of 0.0005, achieving a mean average precision (MAP) of 0.4034, which is almost 50

Next, the author trains the YOLO V5 Convolutional Neural Network with the best-tuned hyperparameters to detect Smart Traffic Management Vehicles in Smart Traffic Management Images. The best hyperparameter settings used for training include a learning rate of 0.00406, momentum of 0.595, mosaic parameter of 0.36, rotation degree of 72.6, scale of 1.32, and weight decay of 0.0005. The results of training the YOLO V5 Convolutional Neural Network with the best-tuned

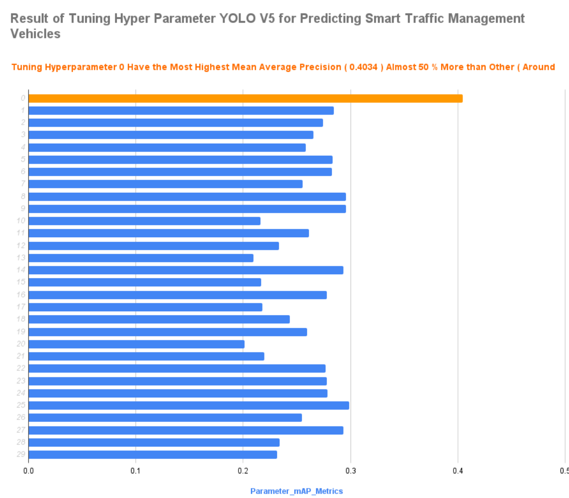

Fig. 8. Results of Best Tuning Hyperparameters for YOLO V5 to Predict Smart Traffic Management System Vehicles in Smart Traffic Management Images

hyperparameters for detecting vehicles are shown in Figure 9.

From the training process, we see that the YOLO V5 Convolutional Neural Network with the best-tuned hyperparameters can detect Smart Traffic Management Vehicles in Smart Traffic Management Images with a mean average precision (MAP) of 0.88561. For detecting cars specifically, the MAP is 0.95251. Additionally, the training loss decreased to 0.0010, which is 50

Next, the author trains a Recurrent Neural Network Long Short-Term Memory (RNN-LSTM) to predict Smart Traffic Management Vehicles. The RNN-LSTM is trained to predict the number of vehicles over the next 12 hours using features such as the number of vehicles in the last hour, year, month, day, hour, and minute. The results of the RNN-LSTM for predicting vehicles over the next 12 hours are shown below.

From the results of the RNN-LSTM model, we can see that it predicts the number of vehicles in the next 12 hours with a mean squared error (MSE) of 4.521 vehicles and a root mean squared error (RMSE) of 2.232 vehicles. The model can accurately predict both increases and decreases in the number of vehicles over the next 12 hours.

The author then evaluates the Smart Traffic Management System using Artificial Intelligence in the CARLA Traffic Management Simulator. A total of 100 scenarios are simulated each day, comparing the performance of the Smart Traffic Management System using AI and a conventional Traffic Management System. The evaluation results are shown below.

After simulating the Smart Traffic Management System in CARLA, we observe that the Traffic Management Congestion Flow Rate with the Smart Traffic Management System (21 vehicles per minute) is 50

III. Conclusion

This paper presents an Autonomous Smart Traffic Management System using Artificial Intelligence to improve traffic congestion flow rates. The system utilizes the YOLO V5 Convolutional Neural Network to detect vehicles in traffic management images. The hyperparameters of the YOLO V5 model, including the learning rate, momentum, mosaic parameter, rotation degree, scale, and weight decay, were optimized. The best hyperparameter settings resulted in a mean average precision (MAP) of 0.4034, which is almost 50% higher than other configurations (approximately 0.2852). After training the model, the YOLO V5 Convolutional Neural Network achieved an MAP of 0.88561, and for cars specifically, the MAP was 0.95251.

The Smart Traffic Management System also predicts the number of vehicles over the next 12 hours using a Recurrent Neural Network Long Short-Term Memory (RNN-LSTM). The model predicts vehicle numbers with an MSE of 4.521 and an RMSE of 2.232, accurately predicting traffic trends for the next 12 hours.

Finally, the Smart Traffic Management System is evaluated using the CARLA simulator. The results show that the Traffic Management Congestion Flow Rate with the Smart Traffic Management System (21 vehicles per minute) is 50% higher than without it (around 15 vehicles per minute), and the Traffic Management Vehicle Pass Delay is 70% less with the system (5 seconds per vehicle compared to around 12 seconds). These findings highlight the effectiveness of the Smart Traffic Management System using AI in reducing traffic congestion and vehicle delay.

List of Abbreviation

- ASTM : Autonomous Smart Traffic Management
- AI : Artificial Intelligence
- ANN : Artificial Neural Network
- CNN : Convolutional Neural Network
- RNN : Recurrent Neural Network
- LSTM : Long Short Term Memory
- RMSE : Root Mean Squared Error

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data and experiment code and result will be made available on request.

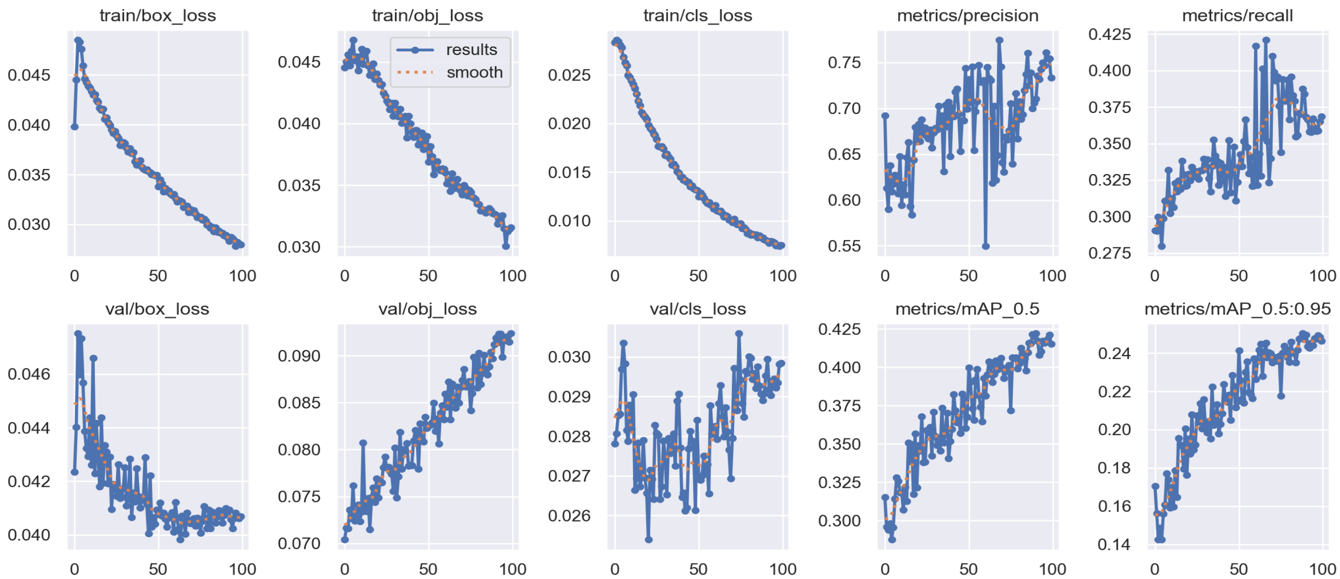

Fig. 9. Results of Training the Best Tuning Hyperparameter for the YOLO V5 Convolutional Neural Network to Detect Smart Traffic Management Vehicles in Smart Traffic Management Images

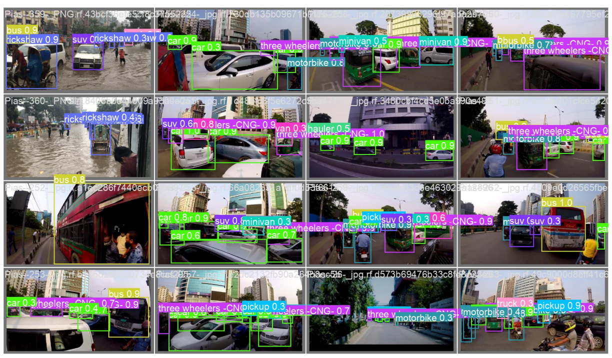

Fig. 10. YOLO V5 Convolutional Neural Network Predicts Smart Traffic Management Vehicles in Smart Traffic Management Images

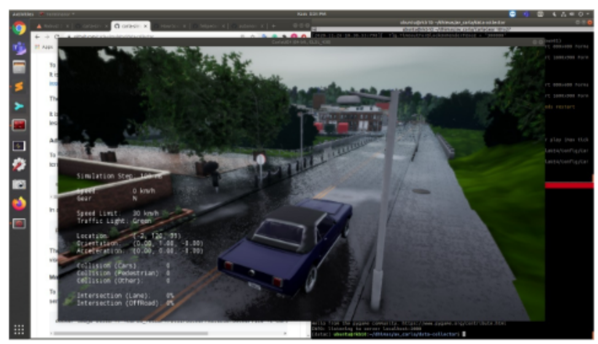

Fig. 12. Evaluation Results of the Smart Traffic Management System using Artificial Intelligence in the CARLA Traffic Management Simulator

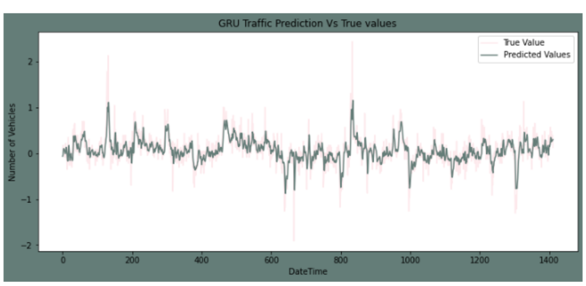

Fig. 11. Results of Recurrent Neural Network Long Short-Term Memory Predictions for Smart Traffic Management Vehicles in the Next 12 Hours

REFERENCES

[1] Linux Foundation, *Acumos AI: Open Source Artificial Intelligence Platform*, Linux Foundation Publication, 2020.

[2] C. R. Goenawan *et al.*, *See the Unseen: Grid-Wise Drivable Area Detection Dataset and Network using LiDAR*, Remote Sensing, vol. 16, no. 20, p. 3777, 2024. [Online]. Available: https://doi.org/10.3390/rs16203777

[3] K.T.Wijaya ,C.R.Goenawan , S.-H.Kong , *Enhancing Performance of 3D Point Cloud Completion Networks using Consistency Loss*, Neurocomputing , 2024 , 129037 , ISSN 0925-2312 . https://doi.org/10.1016/j.neucom.2024.129037

[4] A. Geron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow*, 1st ed. California, USA: O'Reilly Media, 2017.

[5] R. Anyoha, *The History of Artificial Intelligence*, Harvard University: Science in the News, 2017.

[6] A. Esteva *et al.*, *Dermatologist-level classification of skin cancer with deep neural networks*, Nature, vol. 542, no. 7639, pp. 115–118, 2017.

[7] Y. Wu *et al.*, *Google's neural machine translation system: Bridging the gap between human and machine translation*, arXiv preprint arXiv:1609.08144, 2016.

[8] J. Schmidhuber, *Deep learning in neural networks: An overview*, Neural Networks, vol. 61, pp. 85–117, 2015.

[9] C. Wang *et al.*, *Webpage depth viewability prediction using deep sequential neural networks*, IEEE Transactions on Knowledge and Data Engineering, 2018.

[10] D. Sculley *et al.*, *Hidden technical debt in machine learning systems*, Advances in Neural Information Processing Systems, pp. 2503–2511, 2015.

[11] S. Zhao *et al.*, *Packaging and Sharing Machine Learning Models via the Acumos AI Open Platform*, AT&T Research Labs, New Jersey, USA,

2019.

[12] M. Vartak *et al.*, *Model db: A System for Machine Learning Model Management*, in Proceedings of the Workshop on Human-In-the-Loop Data Analytics. ACM, 2016.
https://doi.org/10.1145/2939502.2939516

[13] AT&T and Linux Foundation, *Acumos AI: An Open Source AI Machine Learning Platform*, 2017. [Online]. Available: https://www.acumos.org/

[14] Data Mining Group, *Portable Format for Analytics (PFA): What is PFA?*, 2020.

[15] F. Pedregosa *et al.*, *Scikit-learn: Machine learning in Python*, Journal of Machine Learning Research, vol. 12, no. Oct, pp. 2825–2830, 2011.

[16] M. Abadi *et al.*, *TensorFlow: A system for large-scale machine learning*, in OSDI, vol. 16, pp. 265–283, 2016.

[17] A. Paszke *et al.*, *PyTorch: An imperative style, high-performance deep learning library*, in Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), 2019.

[18] X. Meng *et al.*, *MLlib: Machine learning in Apache Spark*, Journal of Machine Learning Research, vol. 17, no. 1, pp. 1235–1241, 2016.

[19] M. Hofmann and R. Klinkenberg, *RapidMiner: Data Mining Use Cases and Business Analytics Applications*, CRC Press, 2013.

[20] D. Crankshaw *et al.*, *Clipper: A low-latency online prediction serving system*, in NSDI, pp. 613–627, 2017.

[21] R. A. McDougal *et al.*, *Twenty years of ModelDB and beyond: Building essential modeling tools for the future of neuroscience*, Journal of Computational Neuroscience, vol. 42, no. 1, pp. 1–10, 2017.

[22] J. N. Van Rijn *et al.*, *OpenML: A collaborative science platform*, in Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 2013, pp. 645–649.

[23] M. Ribeiro *et al.*, *MLaaS: Machine Learning As A Service*, in Proc. of the 14th Int. Conf. on Machine Learning and Applications (ICMLA). IEEE, 2015, pp. 896–902.