An Adaptive Traffic Lights System using Machine Learning

Mohammad Ashraf Ottom
Information Systems Department,
Yarmouk University,
Jordan
ottom.ma@yu.edu.jo

Ahmad Al-Omari
Information Systems Department,
Yarmouk University,
Jordan
ahmadalomari3000@gmail.com

Abstract: *Traffic congestion is a major problem in many cities of the Hashemite Kingdom of Jordan as in most countries. The rapidly increase of vehicles and dealing with the fixed infrastructure have caused traffic congestion. One of the main problems is that the current infrastructure cannot be expanded further. Therefore, there is a need to make the system work differently with more sophistication to manage the traffic better, rather than creating a new infrastructure. In this research, a new adaptive traffic lights system is proposed to determine vehicles type, calculate the number of vehicles in a traffic junction using patterns detection methods, and suggest the necessary time for each side of the traffic junction using machine learning tools. In this context, the contributions of this paper are: (a) creating a new image-based dataset for vehicles, (b) proposing a new time management formula for traffic lights, and (c) providing literature of many studies that contributed to the development of the traffic lights system in the past decade. For training the vehicle detector, we have created an image-based dataset related to our work and contains images for traffic. We utilized Region-Based Convolutional Neural Networks (R-CNN), Fast Region-Based Convolutional Neural Networks (Fast R-CNN), Faster Region-Based Convolutional Neural Networks (Faster R-CNN), Single Shot Detector (SSD), and You Only Look Once v4 (YOLO v4) deep learning algorithms to train the model and obtain the suggested mathematical formula to the required process and give the appropriate timeslot for every junction. For evaluation, we used the mean Average Precision (mAP) metric. The obtained results were as follows: 78.2%, 71%, 75.2%, 79.8%, and 86.4% for SSD, R-CNN, Fast R-CNN, Faster R-CNN, and YOLO v4, respectively. Based on our experimental results, it is found that YOLO v4 achieved the highest mAP of the identification of vehicles with (86.4%) mAP. For time division (the junctions timeslot), we proposed a formula that reduces about 10% of the waiting time for vehicles.*

Keywords: *Traffic time management, image processing and objects detection, vehicles dataset.*

Received April 23, 2022; accepted February 5, 2023
https://doi.org/10.34028/iajit/20/3/13

1. Introduction

Traffic accidents in recent years have become a remarkable phenomenon in the world and the Hashemite Kingdom of Jordan as well, because of the speedy increase in the number of vehicles, population inflation, and the current political conditions surrounding Jordan, which forced millions of people to immigrate forcedly into Jordan (Forced Immigration). In addition, people are financially capable of owning vehicles, which put additional pressure on transport infrastructures and creates traffic jams in many locations, particularly in major cities. The public security directorate in Jordan revealed that the percentage of increase in registered vehicles throughout the last ten years was 96%, the growth within the variety of registered drivers was 82%, while for foreign vehicles reached an increase of 64%, and a recent report by Traffic management authority in Jordan stated that between 2014 and 2018, the growth of registered vehicles of 23%, which shows a significant increase in the number of vehicles on the road, along with a lack in infrastructure which led to traffic jams (Traffic congestion) [1]. Traffic congestion is a critical problem worldwide, and it is a significant problem for many countries, which affects the transportation system in many cities.

In many cities, public transport has become convenient and reliable, and at the same time cheaper than small cars. However, using public transportation compared to small cars is still challenging because small cars are still faster and give the passenger more privacy and comfort. On the other hand, waiting and time loss is due to inefficient traffic lights timeslot. Therefore, we have to mitigate the issue, by developing traffic signal timing systems that can handle large numbers of vehicles and reduces the waiting time. Most transport systems are based on a pre-established timing system using models that do not respond well to variable demand. Currently, Computer science and engineering concepts such as Artificial Intelligence (AI), Machine Learning (ML), communications, the internet, and many other emerging technologies are coming together with the fields of civil and mechanical engineering research and development; to build what is called Intelligent Transportation System (ITS) [20]. ITS combines different information and communication technologies to create networks and systems that manage traffic and protect roads, vehicles, and pedestrians. It also can provide drivers and passengers with instant information and forecasts on traffic and weather conditions. In addition to that, it offers higher resource efficiency and better management of physical flows [12]. According to (traffic congestion ranking, 2020), The Top-Ten Cities in Traffic Congestion Rates 2019/2020 is shown in Figure 1.

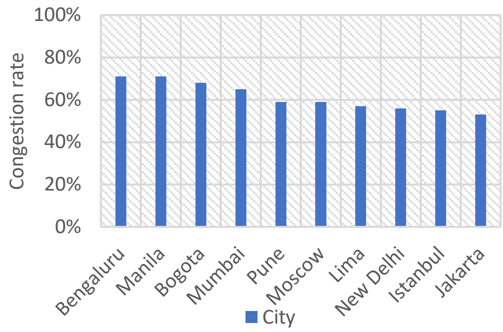

Figure 1. Top-ten cities in traffic congestion rates 2019/2020.

Through this research, a six-classes vehicle image-based dataset was created, then a model was built that is able to detect all types of vehicles used in the Hashemite Kingdom of Jordan by training it on the created database. Finally, a formula is proposed to divide the time fairly, based on the volume of the vehicle density on all sides of the junction of the traffic light.

The paper is organized as follows: the rest of section 1 lists an overview about the techniques used in managing traffic light and the techniques used in each of them. Section 2 mentions some studies that contributed to managing the traffic lights and reducing waiting time by using image processing technology. Section 3 provides an overview machine learning algorithm, YOLO. Section 4 provides an adequate explanation of all stages of the methodology used in the implementation of the experiment. Section 5 explains the proposed equations for time management of traffic lights. Section 6 presents the results extracted from experiments examining the algorithms used in the learning process and the proposed equations for time management. Section 7 discusses research results, conclusions, and future work.

According to Pashupatimath and Madhavanavar [13], there are four techniques applied to traffic lights to decrease traffic congestion: Automatic Traffic Management Technique (usually used), Intelligent Traffic Management Technique supported by image processing, Traffic Management System exploiting Wireless Technologies, and finally Intelligent Roadway Information System (IRIS). From the beginning of the vehicles industry, various techniques have been used in traffic systems to facilitate traffic flow. Over time, these traffic systems have evolved to this day. This section discusses all of these traffic light systems and shows how they work.

1.1. Automatic Traffic Management Technique

The first and most straightforward style of traffic management involves humans within the technique. In this traditional technique, police officers were the primary traffic control system, a traffic police officer stands on every road junction and tries to manage traffic flow using traffic signs. Over time, with the increasing number of cars, the dependence on humans has become insufficient to regulate traffic. Therefore, simple traffic signals replaced police officers to eliminate most weaknesses of the human-based traffic control system. The automatic traffic management technique includes three traffic light colors: red, green, and yellow, usually for every side 30 seconds of allowing passage. However, this could vary in some city areas and also depends on traffic volume [13]. There are weaknesses in this automatic traffic management technique such as:

1. Work statically for example, we need to wait for at least 30 seconds even if there is no traffic on other sides of the junction, which leads to wasting time.
2. It does not consider the traffic density on each side of the traffic junction or the importance of the roadside.

1.2. Intelligent Traffic Management Technique Based on Image Processing

This technique uses cameras, which capture the image of the traffic density on the road. These cameras are placed on a high pole so that they will envelop long distances. A processor chip to detect vehicles on the road analyzes the image captured by the camera, the processor then calculates the times for red and green signals to control and manage the traffic flow [10]. This technique will allow a dynamic time slot for each side of junction, however sometimes, the camera cannot cover long distances in heavy traffic, or the captured images are unclear due to weather conditions. On top of that, vehicle length is not considered an input that makes small and big vehicles equal.

1.3. Traffic Management System using Internet of Things (IoT) Technologies

This technology is designed only to facilitate the movement of emergency vehicles at traffic lights and to give them the estimated time to pass. For example, if an ambulance faced a traffic signal, it sends wireless signals to a receiver installed on the signal pole, then the green light is lit according to the time set by the control unit, and the rest of the signals are closed. Once the estimated time to pass has ended, the normal system will be returned [10, 13].

1.4. Intelligent Roadway Information System

Minnesota Department of Transportation developed a system called an Advanced Traffic Management system

(ATMs). The idea of this system is to observe and handle the freeway traffic flow. The systems did not provide traffic information in the previous techniques and was only used to monitor traffic congestion. On the other hand, Intelligent Roadway Information System (IRIS) delivers real-time information on road conditions to detect traffic accidents, manage traffic flow, and broadcast passenger information. The drawbacks of this method are costly and difficult to use in mixed traffic [11].

2. Related Work

There are several techniques used to reduce traffic density or congestion. Generally, traffic management system techniques based on time division are classified into two groups of solutions: either Wireless Sensor Networks (WSN) or image processing. We will discuss previously published literature on improving traffic management systems based on time division and discuss the previous works of literature on deep learning method Convolutional Neural Networks (CNNs) and how they developed.

Salama *et al.* [16] provided a model of an intelligent system for managing and monitoring traffic lights based on disseminated sensor sited before and after the traffic lights. Their model consists of four main components:

1. Control system unit: which is installed in the traffic control unit placed in the traffic location and its switches the lights, treating the emergency cases and getting the data from the sensor and storing it in a central database.
2. Sensors: there are two types of sensor, the first one is used to test whether roadway extensions are clear or not and while the second type consists of four sensors distributed on the sides of the traffic light.
3. Emergency cases alarm: used to identify police, ambulance and fire vehicles to open road for them.
4. Traffic lights: which is connected to the central control unit and controlled by the central control system.

According to the data received from sensors, the total weight for each side will be calculated, and high priority is given to the most significant total weight traffic that will be open in the next.
They provided the following solutions:

1. Control the traffic jam by facilitating the flow of traffic and less waiting time.
2. Provide a safe way for pedestrians.
3. Facilitate the movement of emergency vehicles.
4. Identification the traffic lights that do not have any cars to exclude them from distribution.

Collotta *et al.* [4] proposed dynamic methods for green periods in traffic light system of a junction based on a WSN. The main objective of this study is to reduce the waiting time for the vehicles and reduce the Red-Light Running (RLR) phenomenon happening. "RLR is a behavioral phenomenon that occurs when the driver must choose to cross (or not) the road when the traffic light changes from green to yellow" [4]. Their system consists of two parts. The first one is to gather information about the traffic using a WSN and dynamic traffic light manager to determine the traffic light cycles based on vehicle number in the queue. Reduced Function Devices (RFD) monitor the road junction placed every 12 meters, and magnetic sensors provide it to detect the vehicles at the junction. After RFD collects the data, it transmits to their eligible Full Function Device (FFD) placed every 60 meters, which sends them to the First Pan Controller (FPC) placed at the end of the road. For evaluation, they used simulation cases for fixed and dynamic traffic light cycles. In their experiment, the length of the road was 1.5 km, and the duration of the fixed traffic light cycle was the 30s, 50s75s, and 90s-120s. They found out that the percentage of passing vehicles from the queue was 80% in the dynamic traffic light, while 43.35% in the fixed traffic light.

Ghazal *et al.* [6] proposed a system that evaluates the traffic density using infrared (IR) sensors and achieves dynamic timing periods with different levels based on a Programmable Intelligent Controller (PIC) microcontroller. They also took into consideration solving the problem of emergency vehicles pause in traffic congestion through a mobile controller. The proposed system is implemented based on numerous electrical modules that comprise:

1. PIC 16F877A microcontroller consisting of a simple Central Processing Unit (CPU), RAM and ROM to make the required processes.
2. Liquid Crystal Display (LCD) display device to alert the manager if the emergency case is working.
3. XBee transceivers to supports a secure and simple full-duplex communication between microcontrollers over serial port data transmission.
4. A pair of IR (InfraRed) sensors for to obstacles detection and differentiate between objects depending on its feature.
5. Push buttons.
6. Many-colored Light Emitting Diode (LEDs) that demonstrate the colors (red, green, and yellow) of the traffic lights.

The traffic master controller determines the duration and the timetable of the two arrangements and their stated levels for different modes of traffic. In addition, the status of the different lights is determined using the traffic master controller by commanding the triggered switches connected to the PIC ports. The microcontroller is also associated with infrared sensors whose production powers are accountable for shifting the counter of the cars incoming at the junction. Finally, the XBee module takes the commands to form the mobile controller and calls the matching emergency sub-functions.

A live video technique is proposed by Kanungo *et al.* [9] by putting cameras at traffic junctions for capturing videos with 30 frames per second and transmit them to the servers where analyzing video and image to calculate the traffic density in real-time on every side of the junction. To improve traffic, they focused on the technique for swapping the traffic lights based on the relative weight of each side by calculating the height of the traffic density in the frame no matter how many vehicles on the side. They used an empty street image as a reference image to help them calculate the change between the reference image and the current image by calculating the height difference between them. After measuring the density of the sides having a red light, their switching system prioritizes a road with a greater density by giving it the green light. Finally, they evaluated their system by comparing it and the traditional approach by testing 100 different scenarios. The experiment showed traffic is reduced by 23% using the proposed method.

Badura and Lieskovsky [2] presented a new technique for the adaptive traffic lights system, based on image processing and data broadcasting over Mobile Ad-hoc NETworks (MANET). The proposed approach is divided into two components: the first one for acquisition data and the second for delivering data. They used 22 videos; each video contains (300 to 500) frames. Their methodology depends on object recognition, based on the distance of feature vector to detect moving objects on the street and classify them into three types of objects: walkers, individual cars, and other vehicles categories, they presented a new connection structure, applied for data transmission in a traffic system using a MANET protocol called Common Alerting Protocol (CAP). They evaluated their model, and it was around 98% of correct identification. Shan and Zhu in [18] worked with high-quality videos to improve vehicle detection and counting using enhanced Singleshot Multibox Detector (SSD) and residual neural network (ResNet). The videos are recorded at five traffic junctions with (3840*2178) resolution; they annotated the vehicle images dataset for training-the results of the proposed system for vehicle detection and counting were 93% in regards to accuracy.

Islam *et al*. [7] proposed a real-time video processing technology to design an intelligent traffic control system based on traffic density. In their system, four High Definition (HD) cameras were placed in one junction, a camera for each side. These cameras were connected with CPUs for video processing. In addition to Radio Frequency Identification (RFID) was placed on the street to detect vehicles and a microprocessor for the traffic light controlling process. All four cameras will send their data to the central server processor to analyze the video and transfer it to the microcontroller. The video processing is done by detecting the vehicles and counting them on each side to compare results and use them for the switching model. They used the OpenCV Python library, and the approach decreases the traffic density and the waiting time significantly.

Rodríguez and García [15] designed a simultaneous traffic observing system. The system is dynamic with conditions and is able to work for a long time. It works in all-weather situations and repeatedly chooses a suitable method for a day, night, or changing times. The system is successful with fast and slow brightness alterations and is able to manage with long broken shadows. They placed a camera used to identify the vehicles passing on the road. They also provide a successful management method for heavy vehicles, to collect the results. Their system segments the video by extracting the moving objects of the location and classifies moving and static objects. When the image is created, it will be segmented by extracting the moving objects using background subtraction methods. The system detects and tracks the vehicles to make with higher performance than the existing system.

Swathi *et al.* [19] proposed an Intelligent Transportation System (ITS) that takes the summarized route to achieve minimum traffic overcrowding. They used sensors to obtain statistics about traffic flow and density. The sensors preserved transferring InfraRed (IR) light and when an object passes nearby, the sensors for vehicle detection and calculating the traffic density by surveilling the mirrored light from the vehicles. They indicate that there is an alteration in data obtained from IR sensors when the temperature and humidity are changed.

Another study by George *et al.* [5] aims to enhance the throughput of traffic and reduces the delay of the road. They presented a new technique for improving the traffic management system using Internet of Things (IoT) and Adaptive Neuro-Fuzzy Inference System (ANFIS). Adaptive Neuro-Fuzzy traffic light controller developed based on traffic density and time of vehicle inputs. They improved the flow of random traffic volumes using Image processing for the identification of vehicles in the traffic and intelligent controls on traffic statistics. The image sensor is used to identify vehicles in the traffic. They used the ThingSpeak Platform and Arduino UNO to transfer images to the cloud. ThingSpeak platform to analyze and display collected data in real-time. They also used Daytime and Nighttime models to detect vehicles for data analysis. The daytime module gathers chunks of pixels in motion using optical flow and background subtraction methods. The optical flow algorithm calculates the motion volume and direction for each image point. Using image processing techniques, the centralized point of the vehicle is identified. The distance traveled by the vehicle is the movement of the centralized point across the frames and velocity is estimated. They used background subtraction to detect constant vehicles. Speed thresholding and formalism 35 processes were done on the binary image. In the nighttime method, they used Ostu's method of image thresholding for vehicle detection images are then segmented to magnify the contrast between foreground and background category. Using blob analysis, the vehicle illuminations are discovered as blobs and traffic density for the junctions is estimated by dividing the total amount by 2.

Javaid *et al.* [8] proposed a hybrid system using both centralized and decentralized approaches to improve traffic movement on highways, and an algorithm is developed to control different traffic cases. The system consists of a camera to record traffic flow, an algorithm to calculate the traffic density, Radio-frequency identification to give the green light to the ambulances, and fire truck vehicles throughout traffic congestion. They concluded that the system changes the traffic light timing smartly based on the density of the traffic, and manages the traffic flow by interacting with a server more effectively than ever. The decentralized approach makes it optimized and efficient as the system works even if the local server or central server goes down. The centralized server connects to the nearby help centers if there an emergency vehicle. They used the algorithm to assign a red-light interval for a particular lane of the junction and transmitted it to the microcontroller and the server.

Sankaran [17] presented proposed a method to enhance the traffic monitoring system performance using vehicle estimation pattern matching performance. The ROI from the input frame is extracted for analyzing the features and templates matching in the process of density estimation. The vehicles were detected at first from the ROI based on coordinate axis and physical features on using reference and current frames. These methods are specified to detect the vehicle amount using the true positives and decreasing the false negatives for reaching better density estimation accuracy and decreasing the rate of error. The proposed technique does not take into account the time spent in detecting vehicles.

Ramakrishnan and Radhakrishnan [14] discussed deep learning topics, Cloud- based Connected and Automated Vehicles, challenges of their applications, and studied Machine learning technologies for autonomous cloud vehicles. He used the proposed Convolutional Neural Networks (CNN) algorithm to simulate the traffic model. After the experimental results, they concluded that CNN is superior to other algorithms, achieving a detection rate of 71% for compounds. The results also showed that cloud vehicles are able to increase their knowledge to avoid accidents.

3. YOLO V4

You Only Look Once (YOLO) v4 is the latest version of the YOLO algorithm and the state-of-the-art object detection algorithm that improves upon the previous versions of YOLO in terms of accuracy, speed, and efficiency. The new version consists of the following [3]:

- Backbone: the Backbone is related to the feature-extraction architecture. It is one of the ways to increase accuracy and implement a deeper network to extend the receptive field and increase model complexity.
- Neck (Feature Pyramid Network (FPN), Spatial Pyramid Pooling (SPP)): the purpose of the neck block is to add additional layers between the backbone and the head to improve the prediction.
- Head (Dense Prediction): Here is the detection of the bounding box coordinates (x, y, w, h) in addition to the confidence score for a class. Also, dividing the image into a grid of multiple cells and then for every cell to predict the probability of having an object using anchor boxes.
- Sparse Prediction: related to the output is a vector with bounding box coordinates and probability classes. Figure 2 shows the structure of YOLO v4.

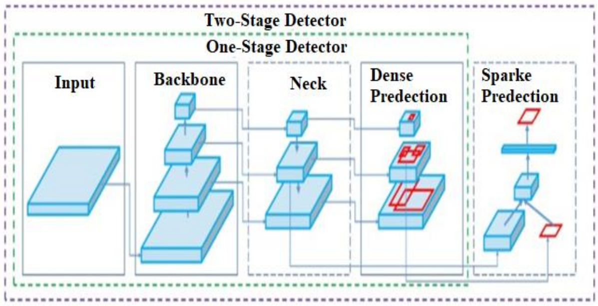

Figure 2. YOLO v4 structure.

4. The Proposed Methodology

4.1. Overall Research Design

This section describes how to investigate the research problem. At first, we recorded videos for traffic to collect vehicle views. Then extracted the frames from videos and generated the annotation files for each frame to construct the dataset. We used YOLO v4 deep learning algorithm for training on our dataset for vehicle detection. We tested our vehicle detector model and compared it with other algorithms. We designed a formula for time management into the junction sides. Figure 3 simplifies the overall research design of the proposed method.

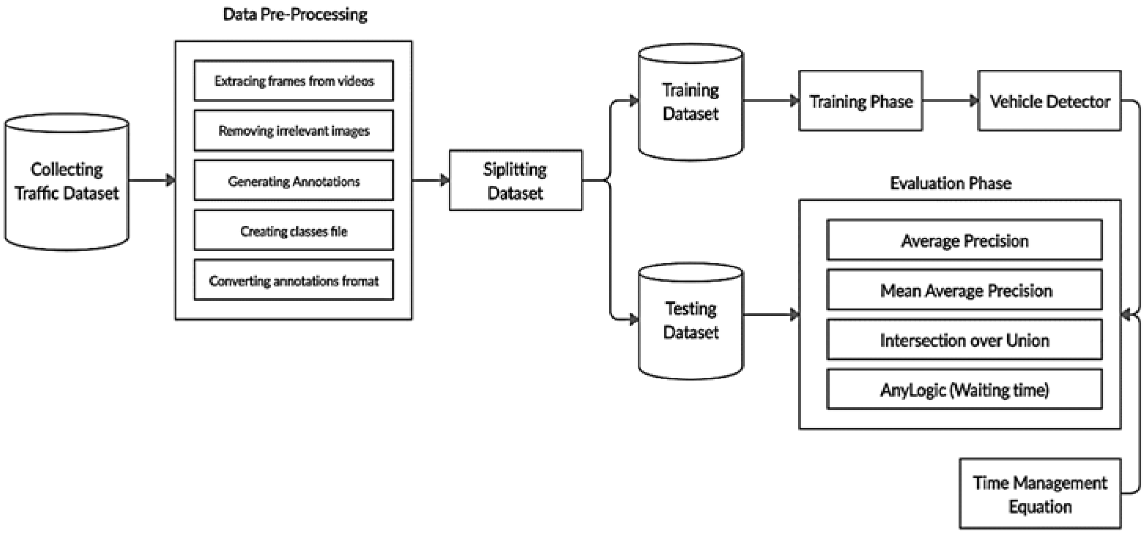

Figure 3. Overall research design.

4.2. Research Phases

4.2.1. Dataset Collection

In this research, we built a new dataset for traffic in Irbid city in the Hashemite Kingdom of Jordan. Our dataset consists of (61,384) frames of real-world traffic views acquired by a Canon EOS 7D camera with a resolution of 1280x720. The camera is placed at the height of 3 meters. The frames are divided into (55,371) images for training and (6,013) images for testing. They are manually annotated using Visual Object Tagging Tool (VoTT) with a total of (234,092) labelled bounding boxes for training and (11,132) labelled bounding boxes for testing. We took 90% of the data for training to get more accurate results. Figure 4 shows a sample of the dataset.

Figure 4. Sample dataset images.

4.2.2. Data Pre-Processing

The first phase in our approach is data pre-processing to ensure data quality and improve the model results. Data pre-processing phase is shown in Figure 5.

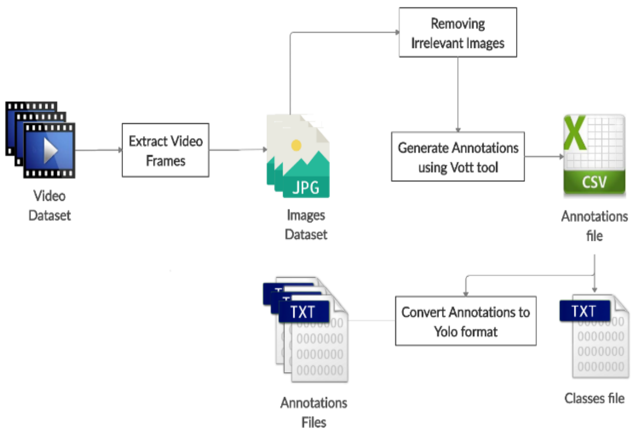

Figure 5. Data pre-processing.

Data pre-processing phase involves four steps, the first step is to extract images frames from the video, the videos were divided into frames (images), precisely five frames per second. The second step is removing irrelevant images; this step is important to enhance the privacy in our dataset, we have removed all images that do not contain vehicles, put masks on people's faces if they appear in the picture, and a red label around plate numbers to ensure people's privacy. The third step is Object Tagging. The tagging process is a pre-processing phase to complete the dataset, it is used to define the objects in the image for the learning process (classification task), object detection models rely on accurate bounding box annotations that define the object's location and size in the image. If the annotations are imprecise or incomplete, the model's performance can be adversely affected. In this step, we used VoTT to assign tags to the objects in each image in the dataset. We created six tags named Car, Bus, Truck, Motorbike, Van, and Minitruck. In each image, the coordinates (x1, y1, x2, y2) were determined for each vehicle, as shown in Figure 6. The output is a Comma-Separated Values (CSV) file, which contains the following data:

- Image_id: the id of the image file.
- x1: the left coordinate of the box.
- y1: the top coordinate of the box.
- x2: the right coordinate of the box.
- y2: the bottom coordinate of the box.
- class_name: the vehicle type.

A sample of tagging process is shown in Table 1. The fourth step in data pre-processing phase is to create classes file for all possible vehicles in the dataset as shown in Table 2. Lastly, converting annotations to YOLO format; where the CSV file format (Image_id, x1, y1, x2, y2, class_name) will be converted into YOLO format. At the end of this step, each image will have annotation file with the same id and contains box and class in each line, a sample of YOLO annotations file is shown in Table 3.

Figure 6. Assigning tags to vehicles using VoTT tool.

Table 1. Sample of the output of the tagging process (CSV file).

Image_id	X1	Y1	X2	Y2	class_name
1	765	319	793	349	motorbike
1	624	281	656	337	bus
1	521	138	679	275	bus
1	440	117	505	188	car
5	164	400	340	451	car
6	174	366	343	434	bus
7	559	384	952	698	car
8	518	443	774	627	car
9	319	347	387	387	car
10	387	363	577	484	car
11	6	393	150	561	car
12	718	313	876	349	bus
13	463	324	593	425	bus
14	1102	374	1244	424	bus
15	3	298	205	391	bus

Table 2. Data Classes and number of tagged objects for each class.

ID	Class	Number of tagged objects
0	Car	91,233
1	Motorbike	6,244
2	bus	32,762
3	truck	11,736
4	van	54,652
5	mini-truck	48,597

We converted the csv file format (Image_name, x1, y1, x2, y2, class_name) to Yolo format. Each image will have annotation file with the same name contains box and class in each line. The following Equations (1), (2), (3), and (4) represent each column of the Table 3 as follow: class_id: the ID of the class in Table 2.

$$\frac{x_{\_center}}{w} = \frac{x_{center\ of\ the\ tagged\ box}}{image\ width} \quad (1)$$

$$\frac{y_{\_center}}{h} = \frac{y_{center\ of\ the\ tagged\ box}}{image\ height} \quad (2)$$

$$\frac{width}{w} = \frac{width\ of\ the\ tagged\ box}{image\ width} \quad (3)$$

$$\frac{height}{h} = \frac{height\ of\ the\ tagged\ box}{image\ height} \quad (4)$$

All values after conversion are between 0 and 1 because the process is done by dividing a part by the dimensions of the whole image. Sample of result are shown in Table 3.

Table 3. YOLO annotation file for one sample image.

Class_id	x_center/w	y_center/h	width/w	height/h
0	0.203906	0.231944	0.06875	0.088888
0	0.298437	0.225	0.067968	0.091666
0	0.271093	0.315277	0.086718	0.126388
0	0.561718	0.429166	0.105468	0.155555
0	0.467187	0.301388	0.105468	0.1625
1	0.3375	0.408333	0.047656	0.179166

4.2.3. Learning Phase

Figure 7 explains the learning phase and its inputs and output. We used YOLO v4 algorithm to train our dataset. The learning process was done with the following specifications (mini-batch gradient descent with 32 batch-size, Iterative process with 50 epochs, and split training dataset for validation with 0.2 ratio) and computer specification was NVIDIA RTX 2070 Graphical Processing Unit (GPU), Intel CORE I7-8th Gen, and 32GB RAM. The learning process took five days and three hours.

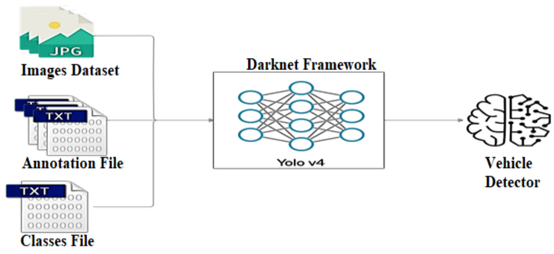

Figure 7. Learning phase.

5. The Proposed Time Traffic Management Formula and System Design

5.1. The Proposed Time Traffic Management Formula

After taking measurements of Irbid city traffic lights using the timer, we concluded that the typical traffic light cycle in Irbid is 120 seconds in the static traffic lights systems. There is time wasted during yellow light periods or during "all red" periods to avoid accidents and ensure the junction is clear. The researchers also concluded that the average of lost time is three seconds (for all red+yellow light periods). Therefore, the actual period of green light for the four-leg junction is 108 seconds. In the static system, the average green light time is 27 seconds for each side. To achieve the primary goal of this research, we proposed the following Equations (5), (6), and (7) to reduce the traffic delay.

$$n_i = 4 * T + 2 * B + C + V + MT + 0.5 * M \tag{5}$$

Equation (5) is related to the outputs of the proposed model, which is the types of vehicles on the street and the number of every vehicle type. T is the number of trucks in a certain junction, B is the number of buses, C is the number of cars, V is the number of vans, MT is the number of mini-trucks, and M is the number of motorbikes.

We measured the average length of each of the six vehicles types, and the approximate lengths were as follows [car ≈ 4.5m], [truck ≈ 18.4m] [bus ≈ 10m] [motorbike ≈ 2.5m]. We also assumed the length of Van and Mini-truck equal the length of Car as shown in Figure 8. The factors in Equation (5) represent the average vehicles' lengths divided by the car length to standardize the equation along with car length because it is the most used on the street. To build the equation, we suggest putting a factor for each vehicle type based on the average length, such as the factor 4 for Truck and 2 for Bus… etc., The maximum factor given was 4 for trucks because it is longest vehicle and can occupy more space than other vehicles, then the factor 2 for buses. We also assumed the length of Van and Mini-truck equal the length of Car with factor 1.

$$Avg = \frac{n_1+n_2+n_3\ldots+n_k}{K} \tag{6}$$

Equation (6) uses the output of Equation (5) to all sides of the junction and then taking the average for them. Where n_i is presented in Equation (5) and k is the number of sides of the junction.

$$t_i = \frac{n_i * 27}{Avg} \tag{7}$$

The time allocated to the green light for a specific side is presented as t_i in Equation (7). Where n_i is presented in Equation (5) and Avg is presented in Equation (6).

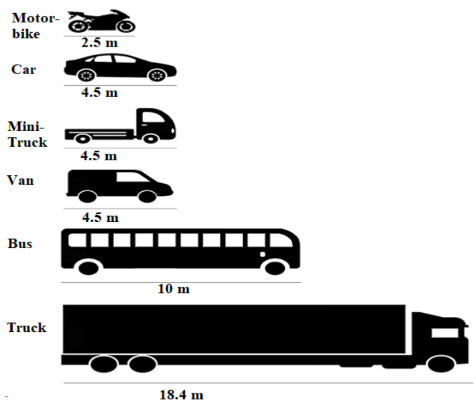

Figure 8. Average lengths f vehicles.

To evaluate the proposed time management equation, we used Anylogic simulation tool. We take barada traffic light as a test field. Barada traffic light is located in a commercial area and was chosen for the following reasons:

1. It's located in the city center of Irbid.
2. It is located on a street linking the regional areas with the city center.
3. It handles a lot of daily traffic compared to other traffic lights.
4. Difference in traffic between its lanes.
5. It crosses through it all kinds of vehicles that were used in the experiment.

There are some criteria that were taken in this research to choose the traffic light in the experiment: A traffic light that deals with daily traffic pressure, the importance of its location in the city, the volume of the traffic, the possibility of applying the equation to all types of vehicles, the possibility of applying the proposed methodology on it (for example: it suffers from the problem of fixed time for all its lanes, although some lanes need less than others).

Barada traffic light placed at for-leg junction, and each side has three lanes. We measured manually the green light intervals several times a day and the green light time set for each side was constant throughout the day. This is a negative point for the current system. Figure 9 shows an aerial photograph of the Barada traffic light from google maps.

Figure 9. Aerial photograph of the "Barada" traffic light from google maps.

5.2. System Design

This section explains how the proposed system works. Cameras will be placed on the traffic light so that each camera covers one side. These cameras detect and count vehicles waiting for a traffic signal using the built vehicle detector. Then the system takes the number of vehicles for each side to the proposed equation to calculate the time required for each side of the traffic junction. The overall system design structure is explained in Figure 10.

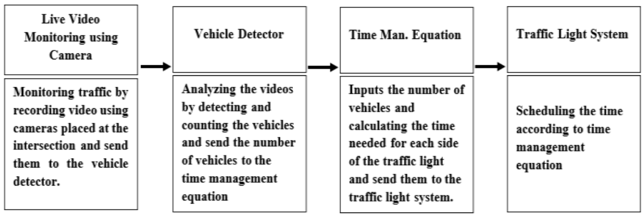

Figure 10. System design.

6. Experimental Results and Analysis

This section demonstrates the experimental and evaluation results with relevant and appropriate inferential statistical analyses and interpretation reports.

6.1. The Performance of Vehicle Detection

Mean Average Precision (mAP) is a metric used to evaluate the performance of object detection models. Average Precision is calculated as the weighted mean of precisions at each threshold. mAP is calculated by summing the Average Precision (AP) of all classes and dividing by their number.

We tested our vehicle detector on the test dataset to calculate the Average Precision for each class and the mAP for all classes. We assumed the prediction is correct if (IoU $\geq$ 0.5) where Intersection over Union (IoU) denotes to Junction over Union. Figure 11 shows the sample of testing results.

Figure 11. Sample of tested images showing the detected vehicles.

Figure 12 summarizes the results of the mAP for YOLO v4 and shows that the (Car) class has the highest mAP of 97% and the (Motorbike) class has the lowest mAP of 73%. While the other classes have their mAP values ranging between the highest and the lowest values, in descending order (Mini-truck), (Van), (Bus), and (Truck), respectively.

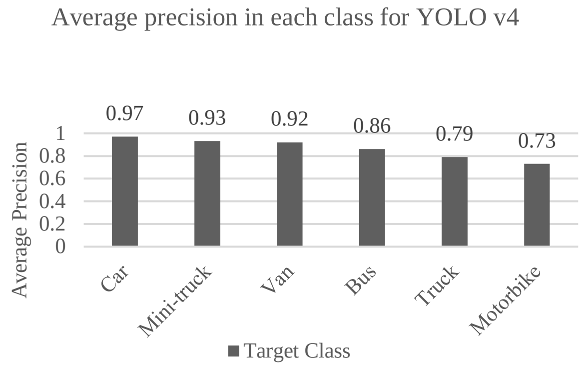

Figure 12. Average precision in each class for YOLO v4.

We repeated the learning process using R-CNN, Fast R-CNN, Faster R-CNN, and Single Shot Detector (SSD) algorithms. We calculated mAP for each of them compared to the YOLO v4 algorithm to demonstrate the efficiency of the YOLO v4 algorithm. Figures 13, 14, and 15 shows model evaluation using R-CNN, Fast R-CNN, and Faster R-CNN respectively.

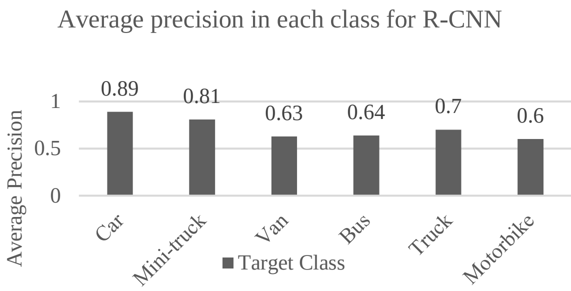

Figure 13. Average precision in each class for R-CNN.

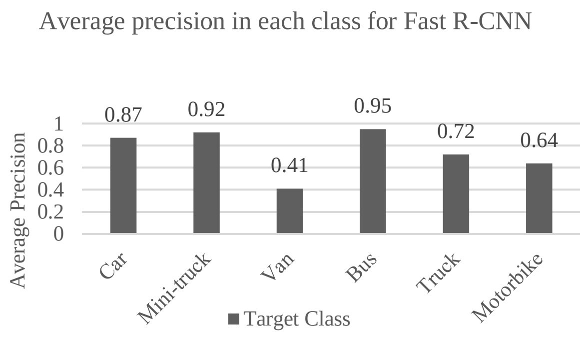

Figure 14. Average precision in each class for Fast R-CNN.

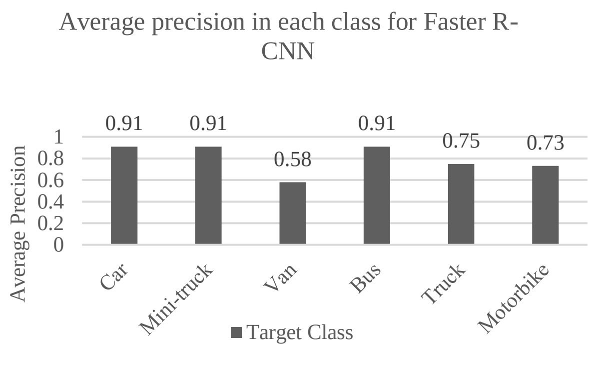

Figure 15. Average precision in each class for Faster R-CNN.

Figure 13 recaps the results of the AP for R-CNN and shows that (Car) and (Mini-truck) classes have the highest AP and (Motorbike) class has the lowest AP. While the other classes have their AP values ranging between the highest and the lowest, in descending order respectively (Truck), (Bus), and (Van). Figure 14 shows the evaluation results for the Fast R-CNN model, AP for detecting Minitrucks and Buses above 92%, and the lowest was Van with an AP of 41%. Detection AP for Cars, Trucks and Motorbikes were between 87-64%. Figure 15 shows the evaluation results for the Faster R-CNN model, AP for detecting Cars, buses, and Minitrucks were 91%, and the lowest was Van with an AP of 58%. Detection AP for Tucks and Motorbikes were between 73-75%.

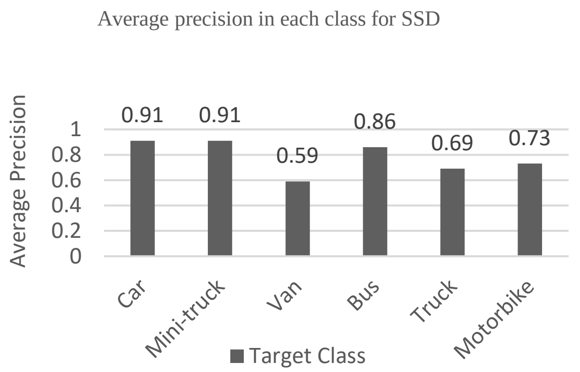

Figure 16. Average precision in each class for SSD.

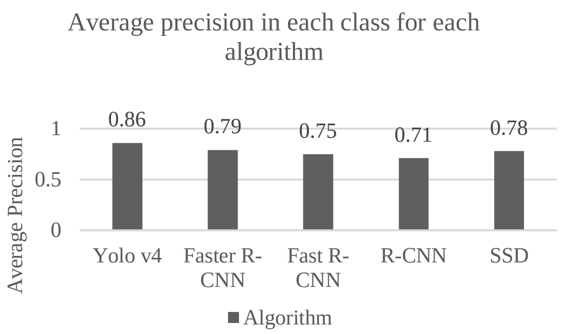

Figure 17. Mean average precision for each algorithm.

Figure 16 shows the evaluation results for SSD model, AP for detecting Cars and Minitrucks were 91%, Buses were 86%, and the lowest was Van with an AP of 59%. Detection AP for Tucks and Motorbikes were between 69%-73%.

Figure 17 reviews all mAP results for all algorithms, YOLO v4, SSD, R-CNN, Fast R-CNN, and Faster R-CNN. It shows that YOLO v4 has the highest mAP overall, with mAP about 86%. The outcomes for all algorithms are very encouraging due to the diversity of data and because we captured images from different angles. We also noticed a variation in mAP for some classes. For example, all algorithms are better at detecting Cars than Motorbikes.

6.2. The Efficiency of the Proposed Formula for Traffic Management

We created simulations of traffic using the Anylogic simulation tool for the traditional system and the proposed approach. We suggested nine scenarios with a different number of vehicles per hour and recorded the waiting time for each vehicle, as shown in Table 4. The experimental results showed that the average waiting time in the traditional system was 38.4 seconds and the average waiting time after applying the proposed equation was 34.7 seconds. The proposed method reduced the vehicle waiting time by 4 seconds, with about 10% saving in the waiting time.

These results indicate that our proposed equation for time management is more efficient than the fixed time used in the current systems. Because it divides the time based on the number of vehicles with different weights for six types of vehicles. The proposed time management formula achieved less waiting time than the current system. Our findings in this research can be concluded in the following points:

Table 4. Average waiting time comparison between static system and the proposed approach.

Scenario Number	Number of vehicles per hour	Average waiting time in the static system (seconds)	Average waiting time in the proposed system (seconds)
1	500	39.7	35.2
2	600	38.8	34.8
3	650	39.6	35.3
4	700	37.4	34.5
5	750	40.1	36.6
6	800	37.5	32.1
7	850	37.1	34.2
8	900	37.7	34.2
9	950	39.5	35.2

7. Conclusions and Future Work

This research proposed a new approach for traffic management using deep learning algorithms based on created vehicles dataset. In addition, a traffic time management equation was proposed. For detection purposes, we used the YOLO v4 deep learning algorithm to train a vehicle detector. The results proved that the proposed equation reduces the waiting time for vehicles by 4 seconds and saves about 10% of the previous waiting time, and the model can detect the vehicles with mAP of 86.4%. Real-time traffic analysis gives the needed time to vehicles fairly and reduces waiting time, in our case, about 10%. The vehicle length or size can affect the mAP metric because the detection of small vehicle can be more challenging than the detection of large objects. This is because small objects may have fewer distinctive features and can be more easily confused with the background or other objects in the image. As a result, a model that performs well on detecting large objects may not perform as well on small objects. In addition, each technique used in this study may produce a different output from other techniques, depends on technique architecture and assumptions. Finally, Van was difficult to detect in all algorithms except YOLO v4 because the algorithms identified it as a car and not as a van, but YOLO v4 was the most accurate in distinguishing and detecting all objects. The best detection of the motorbikes was not only in the SSD, but also in YOLO v4 and Faster R-CNN, and this is due to the reason for their ability to detect relatively faster and smaller objects (vehicles). For future work, the plan is to expand the dataset and capture traffic in other cities in Jordan to ensure better results in the learning process. Also, we aim to use augmentation and balancing techniques in machine learning to solve the issue of imbalance vehicle types. In addition, we believe adding new factors can reduce vehicle waiting time, such as considering the number of lanes in each direction.

Acknowledgments

Necessary approvals were taken from the officials in Irbid and Yarmouk University to collect the dataset.

References

[1] Alharahsheh Y., "Traffic Reports in Jordan 2016," [Online].Available:https://www.psd.gov.jo/images/traffic/docs/0v10.pdf, Last Visited, 2020.

[2] Badura S. and Lieskovsky A., "Intelligent Traffic System: Cooperation of MANET and Image Processing," *in Proceedings of the 1st International Conference on Integrated Intelligent Computing*, Bangalore, pp. 119-123, 2010.

[3] Bochkovskiy A., Wang C., and Liao H., "Yolov4: Optimal Speed and Accuracy of Object Detection," *arXiv:2004.10934*, 2020.

[4] Collotta M., Pau G., Scatà G., and Campisi T., "A Dynamic Traffic Light Management System Based on Wireless Sensor Networks for the Reduction of the Red-light Running Phenomenon," *Journal of Transport and Telecommunicaions*, vol. 15, no. 1, pp. 1-11, 2014.

[5] George A., George V., and George M., "Iot Based Smart Traffic Light Control System," *in Proceedings of the International Conference on Control, Power, Communication and Computing Technologies*, Kannur, pp. 148-151, 2018.

[6] Ghazal B., ElKhatib K., Chahine K., and Kherfan M., "Smart Traffic Light Control System," *in Proceedings of the 3rd International Conference on Electrical, Electronics, Computer Engineering and their Applications*, Beirut, pp. 140-145, 2016

[7] Islam M., Shahid N., Ul Karim D., Al Mamun A., and Rhaman M., "An Efficient Algorithm for Detecting Traffic Congestion and a Framework for Smart Traffic Control System," *in Proceedings of the 18th International Conference on Advanced Communication Technology*, PyeongChang, pp. 802-807, 2016.

[8] Javaid S., Sufian A., Pervaiz S., and Tanveer M., "Smart Traffic Management System Using Internet of Things," *in Proceedings of the 20th International Conference on Advanced Communication Technology*, Chuncheon, pp. 393-398, 2018.

[9] Kanungo A., Sharma A., and Singla C., "Smart Traffic Lights Switching and Traffic Density Calculation Using Video Processing," *in Proceedings of the Recent Advances in Engineering and Computational Sciences*, Chadigarh, pp. 1-6, 2014.

[10] Keerthi R. and HariharaGopalan S., "A Survey on Various Traffic Management Schemes for Traffic Clearance, Stolen Vehicle and Emergency Vehicle," *International Journal of Emerging Technology in Computer Science and Electronics*, vol. 19, no. 2, pp. 46-50, 2016.

[11] Lach D. and Macioszek E., "Analysis of Selected ITS Services Used in Public Collective Transport," *WUT Journal of Transportation Engineering*, vol. 121, no. 1, pp. 211-222, 2018.

[12] Mangiaracina R., Perego A., Salvadori G., and Tumino A., "A Comprehensive View of Intelligent Transport Systems for Urban Smart Mobility," *International Journal of Logistics Research and Applications*, vol. 20, no. 1, pp. 39-52, 2017.

[13] Pashupatimath S. and Madhavanavar S., "Different Techniques Used in Traffic Control System: An Introduction," *International Journal of Engineering Research and Technology*, vol. 6, no. 3, pp. 1-4, 2018.

[14] Ramakrishnan D. and Radhakrishnan K., "Applying Deep Convolutional Neural Network (DCNN) Algorithm in the Cloud Autonomous Vehicles Traffic Model," *The International Arab Journal of Information Technology*, vol. 19, no. 2, pp. 186-194, 2022.

[15] Rodríguez T. and García N., "An Adaptive, Real-Time, Traffic Monitoring System," *Machine Vision and Applications*, vol. 21, pp. 555-576, 2010.

[16] Salama A., Saleh B., and Eassa M., "Intelligent Cross Road Traffic Management System (ICRTMS)," *in Proceedings of the ICCTD 2nd International Conference on Computer Technology and Development*, Cairo, pp. 27-31, 2010.

[17] Sankaran S., "Pattern Matching Based Vehicle Density Estimation Technique for Traffic Monitoring Systems," *The International Arab Journal of Information Technology*, vol. 19, no. 4, pp. 575-581, 2022.

[18] Shan Z. and Zhu Q., "Camera Location for Real-Time Traffic State Estimation in Urban Road Network Using Big GPS Data,"*Neurocomputing*, vol. 169, pp. 134-143, 2015.

[19] Swathi K., Sivanagaraju V., Manikanta A., and Kumar S., "Traffic Density Control and Accident Indicator Using WSN," *International Journal for Modern Trends in Science and Technology*, vol. 2, no. 04, pp. 18-21, 2016.

[20] Zear A., Singh P., and Singh Y., "Intelligent Transport System: A Progressive Review," *Indian Journal of Science and Technology*, vol. 9, no. 32, pp. 1-8, 2016.

Mohammad Ashraf Ottom is an Information Systems associate professor and the director of entrepreneurship and innovation center at Yarmouk University in Jordan. He obtained his Ph.D. from the University of Canberra, M.S. degree from the University of New South Wales in Australia, and Bs. In computer science from JUST. Mohammad is an expert in data science, computational modelling, and machine learning. He was involved in several projects and research. He served as the vice dean in the deanship of student's affairs. Dr. Ottom received the Fulbright Visiting Scholar award, supported by the ECA and FFSB, and joined SOCR at the University of Michigan Ann Arbor for the year 2021/2022.

Ahmad Al-Omari is a skilled and dedicated professional in the field of computer information systems. He earned his bachelor's degree in computer information systems from Yarmouk University and went on to pursue his master's degree in information systems from the same institution. Ahmads research of interests include but are not limited, deep learning, machine learning, data science, and computer vision.